

Introducing Predictive Analytics for Decision Support in the Cultural Domain

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Michael Heil

Matrikelnummer 0826358

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

Wien, 27.01.2014

(Unterschrift Michael Heil)

(Unterschrift Betreuung)

Introducing Predictive Analytics for Decision Support in the Cultural Domain

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Michael Heil

Registration Number 0826358

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber

Vienna, 27.01.2014

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Michael Heil

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Michael Heil)

Acknowledgements

First of all I want to express my profound thanks to my advisor Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Rauber. He has always lent an ear to my issues, despite his tight schedule. His feedback and ideas were essential for this work. However, I especially want to thank him for giving me the opportunity to work on this project at all, because without his support I would never have had the chance to cooperate with the originator of the idea.

With regards to this originator I want to thank Gerald Stockinger who has also invested a lot of time in handling my concerns and requests. His effort and input has helped a lot in understanding the domain and the data.

I also want to mention my family here, which really helped me making my life easier in the course of this work in many different ways. They have always supported me wherever possible and I want to thank them very much for that!

The same goes for my significant other - my girlfriend Teresa which in addition to that was greatly patient with me especially on free weekends when I devoted more time to this thesis than to her.

In the end I also want to mention and express my gratitude to my loyal calculation servant to whom this study meant a great deal of work, possibly even more than for me. There were periods when it had to do nonstop-calculations at full duty which made me sweat even more during the record summer and lessened my heating costs during the cold time. Furthermore, for the first time the investment in a big main memory really paid off!

Abstract

Data mining is an effective means to extract useful meaning from the myriads of data that has accumulated over the years. As the base of the Big Data-hype, it is increasingly common in companies, but often fails due to its high complexity or excessive expectations. To counter this situation by reducing the necessity of expertise and intuition required for conducting such projects, a methodology shall be developed by the means of a feasibility study. This study is intended to answer different forecasting questions using sales data of a cultural establishment in the German speaking area. These questions include predicting the occupancy rate of single events, determining “typical” sales trend patterns and predicting the general success as well as daily sales figures of productions.

Firstly and according to the CRISP-DM process model, the understanding of the data as well as their business context are addressed. Additionally, data needs to be prepared in general before it can be used in our data mining context. The domain related questions are answered sequentially by reusing insights gained and data structures prepared. Further, various evaluations are conducted by comparing different solution approaches and configurations. However, a comparison with a conventional forecast method can be drawn for the occupancy rate prediction of events only, as there have been no such efforts concerning the other questions.

From a methodological point of view different possibilities to solve problems are unveiled and aspects that need to be taken into consideration are pointed out. In the end, the work is intended to facilitate getting the feel of the working principles and eventually being able to reproduce the process in a different environment, even different from the cultural context. The tool used for almost all tasks is Weka, which is open source, offering a great flexibility and an appropriate range of functions.

When it comes to the results of this feasibility study, it is demonstrated that on the one hand, data mining is suitable and on the other hand, the data available is sufficient to yield useful results for the bigger part of the domain questions. The conventional approach to forecast events is surpassed by providing a solution that is up to 11% more accurate on average depending on the horizon. Further, simple and classic approaches to forecast time series are outclassed by the ML-NARX approach proposed especially for projecting sales figures of productions. Taken as a whole, many new insights are gained, but also several deficiencies are encountered. Most of them are due to deliberate interferences caused by marketing measures, sales to key accounts and partner companies, which lack granularity and the general scarcity of data. In addition to this, manifestations of a phenomenon called “concept drift” are experienced.

In the end, the conclusion can be made that despite facilitating the repetition of such approach by this formalization of the basic procedure, the principles of applying data mining remain what they are – art as much as science.

Kurzfassung

Data Mining ist ein effektives Instrument um aus gesammelten Datenfluten nutzbares Wissen zu extrahieren. Es findet als Basis des Big Data-Hypes verstärkt Einsatz in Unternehmen, scheitert aber oftmals an der Komplexität und falschen Erwartungen. Um dem entgegenzuwirken und den Anteil an Erfahrung und Intuition, den ein solches Projekt erfordert, zu verringern, soll hier anhand einer Machbarkeitsstudie, in der es darum geht, mithilfe von Verkaufsdaten einer kulturellen Einrichtung im deutschsprachigen Raum verschiedene Vorhersage-Fragestellungen zu beantworten, eine Methodologie erarbeitet werden. Diese Fragestellungen umfassen unter anderem die Prognose von Auslastungen einzelner Vorstellungen, dem ermitteln "typischer" Auslastungsverlaufsmuster, die Vorhersage von Verkaufszahlen ganzer Produktionen sowie Erfolgsvorhersagen derselben.

Dem Vorgehensmodell CRISP-DM folgend wird zunächst ein Verständnis der zugrundeliegenden Daten sowie dem betriebswirtschaftlichen Kontext erarbeitet. Dies ist essentiell für die danach folgende generelle Aufbereitung und Vorbereitung der Daten für den Einsatz im Data Mining. Die fachlichen Fragestellungen werden nacheinander beantwortet, wobei bereits gewonnene Erkenntnisse sowie aufgebaute Datenstrukturen wiederverwendet werden. Soweit möglich, werden im Zuge dessen auch verschiedene Evaluierungen durchgeführt, wobei ein erfolgsbeurteilender Vergleich mit der herkömmlichen Methode nur im Fall der Auslastungsprognose einzelner Vorstellungen gezogen werden kann, da es in den anderen Fällen keine solchen Bemühungen gegeben hat.

Aus methodologischer Sicht werden die verschiedenen Möglichkeiten zur Problemlösung sowie die Aspekte, auf die dabei zu achten ist, aufgezeigt. Im Endeffekt soll ein Gefühl für die Funktionsweisen vermittelt werden, um den Prozess auch unter anderen Rahmenbedingungen effektiv nachzeichnen zu können. In allen Bereichen wird das Open Source-Tool Weka eingesetzt, da es eine große Flexibilität und einen angemessenen Funktionsumfang bietet.

Was die Ergebnisse der Machbarkeitsstudie angeht, so wird gezeigt, dass Data Mining im Wesentlichen geeignet und die Datenlage ausreichend ist, um brauchbare Ergebnisse für den Großteil der Fragestellungen zu liefern. Der bisherige Ansatz zur Auslastungsprognose liefert je nach Horizont der Prognose eine durchschnittlich um bis zu 11 % akkuratere Aussage. Auch klassische und simple Ansätze zur Prognose von Zeitreihen, werden durch den ML-NARX-Ansatz, welcher insbesondere auf die Prognose von Produktionen anwendbar ist, in den Schatten gestellt. Insgesamt können viele neue Erkenntnisse gewonnen werden, es treten jedoch auch einige Unzulänglichkeiten zu Tage. Diese werden durch in den Daten inhärent enthaltene störende

Eingriffe durch Marketing-Aktionen, Großkundenverkäufe, welche nicht feingranular genug abgebildet sind, sowie der generellen Knappheit an Daten verursacht. Zudem wird das Phänomen der "Concept Drift" angetroffen.

Zuletzt kann mit der Erkenntnis verblieben werden, dass durch die Formalisierung des grundsätzlichen Vorgehens das Ziel der Erleichterung der Wiederholung des Prozesses zwar erreicht wird, die Anwendung von Data Mining im Wesentlichen jedoch gleichermaßen Kunst wie Wissenschaft bleibt.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement	2
1.3	Research Questions and Goals	3
1.4	Methodology and Structure	4
2	Related Work	7
2.1	Data Mining	7
2.2	Time Series Analysis	25
2.3	Comparable Studies	29
2.4	Summary	33
3	Data Analysis and Preparation	35
3.1	Business Background	35
3.2	Data Model	36
3.3	Data Analysis	37
3.4	General Data Preparation	37
3.5	Summary	44
4	Methodology	45
4.1	Domain Question 1 – Sales Trend Patterns	45
4.2	Domain Question 2 – Event Capacity Utilization Forecast	56
4.3	Domain Question 3 – Influencing Factors	68
4.4	Domain Question 4 – Breaking down into Price Categories	70
4.5	Domain Question 5 – Production Sales Figures Forecast	74
4.6	Domain Question 6 – Extension of a Running Production	94
4.7	Domain Question 7 – Success of a Production	97
4.8	Summary	99
5	Discussion and Limitations	101
5.1	Analogism and Quantitative Limitations – Concept Drift	101
5.2	Control Measure Interference	102
5.3	Expert Discussion and Feedback	103
5.4	Generalizability	104

6 Conclusion	107
6.1 Summary	107
6.2 Future Work	110
A Data Description	113
B Additional Figures	123
B.1 Attribute correlations of DQ2	123
B.2 Capacity Utilization Rate Prediction at Different Cutoff Days	125
B.3 Scatterplots of Influential Attributes (DQ2)	126
B.4 Capacity Utilization Rate Prediction at Different Cutoff Days (Blau)	126
B.5 Sales Prediction with Sliding Window Integrated	127
B.6 Sales Prediction with Sliding Window Difference	127
B.7 Sliding Window Evaluation Examples	128
B.8 Sales Prediction with the Micro-founded Approach	129
C Attribute Groups	135
Bibliography	137

List of Figures

1.1	The CRISP-DM process model	5
2.1	Comparison of linear and nonlinear classification	14
2.2	A simple decision tree example	15
2.3	A simple multilayer perceptron	16
2.4	The support vectors and maximum margin hyperplane of a simple model	17
2.5	A three-dimensional regression problem	18
2.6	Comparison two k-nearest-neighbor models with different k values	19
2.7	Naive Bayes and Bayes optimal classifier at a one-dimensional classification problem	19
2.8	A self-organizing map based on several attributes of DQ1 of which two are shown .	24
2.9	A simple time series of air passenger numbers	26
3.1	Data model of the problem domain	37
3.2	Comparison of sales figures of production 4068 with and without batch filtering . .	42
4.1	Comparison of different clustering algorithms with different quantities of clusters .	46
4.2	Comparison of different clustering algorithms with no accumulation of input factors	47
4.3	10 clusters of sales trends of all events identified by k-means clustering	48
4.4	Cluster number 0 with all event sales trends that constitute it	48
4.5	Cluster allocations for all productions that are involved	49
4.6	Cluster allocations at structural criteria with a comparably high distinguishability .	49
4.7	Accuracy of a naive Bayes classifier with various attribute subsets	53
4.8	Accuracy of several classifiers on different attribute sets	54
4.9	Accuracy of well-performing classifier configurations at different cutoff days . . .	55
4.10	Summary of the high-level steps needed to answer DQ1	56
4.11	Accuracy of several regression algorithms to forecast the occupancy rate	57
4.12	Forecasting the occupancy rate when using events of the respective production only	59
4.13	Accuracy of a support vector regression at different cutoff days	60
4.14	Histogram of the final event capacity utilization rates	60
4.15	Comparison of regression result with mean value and conditional density classification	61
4.16	Comparison of regression result with the result of classifying the trend cluster . . .	62
4.17	Forecast of capacity utilization and trend cluster of event 5182 (10)	63
4.18	Forecast of capacity utilization and trend cluster of event 5182 (30)	63
4.19	Comparison of the normal and vague forecasting approaches	64

4.20	The deviations to the final occupancy of the baseline and data mining approach . . .	66
4.21	The improvement of the data mining over the baseline forecast	67
4.22	A heavily pruned C4.5 decision tree for a DQ1 classification problem	69
4.23	A heavily pruned regression tree for a DQ2 regression problem	70
4.24	The regression coefficients of DQ2 with their corresponding p-values	71
4.25	The absolute amount of tickets of each main price category	72
4.26	Clustering of all tickets of the price category Rot	72
4.27	Clustering of all tickets of the price category Stehplatz	73
4.28	Comparison of the DQ1 and DQ2 performance for each price category	73
4.29	The idea behind the sliding window	74
4.30	Accuracy of several regression algorithms for the sliding window	78
4.31	Comparison between the Google Trend and the sales of production 5830	79
4.32	Sliding window forecasting within each production	80
4.33	Seven-step-ahead forecasts for each day in different phases of production 4068 . . .	81
4.34	Comparison of regression performance with and without past data	81
4.35	Sales forecast of production 4068 using the sliding window approach	83
4.36	Sales forecast scenarios when approaching the final prediction day	84
4.37	Comparison of regression performance with a holistic model	84
4.38	The idea of the sliding window smoothed variant	85
4.39	Comparison of the integrated and smoothed approach	86
4.40	Comparison of different algorithms to project daily fraction of weekly sales	87
4.41	Sales forecast of production 4068 using the sliding window difference approach . .	88
4.42	Comparison of sliding window and classical forecasting methods (1 – 21)	89
4.43	Comparison of sliding window and classical forecasting methods (1 – 365)	90
4.44	Comparison of sliding window and classical forecasting methods on a weekly basis	91
4.45	The idea of the micro-founded prognosis	92
4.46	Sales forecast of production 4068 using the micro-founded approach	93
4.47	Sales figures forecast to decide upon an extension (sliding window integrated) . . .	96
4.48	Sales figures forecast to decide upon an extension (sliding window difference) . . .	97
4.49	Sales figures forecast to decide upon an extension (micro-founded)	98
B.1	Correlogram of selected qualitative attributes and the target variable of DQ2	123
B.2	Correlogram of the sales figures attributes and the target variable of DQ2	124
B.3	Correlogram of price category sales data and the target variable of DQ2	124
B.4	Prospective sales trends of event 53412 using DQ1 and DQ2 models	125
B.5	Scatterplots of the most important attributes against the target of DQ2	126
B.6	Prospective sales trends of event 53412 using DQ1 and DQ2 models (Blau)	127
B.7	Sales prediction of production 4652 using the sliding window integrated approach .	128
B.8	Sales prediction of production 4652 using the sliding window difference approach .	129
B.9	Comparison of different algorithms using the sliding window integrated approach .	130
B.10	Comparison of different attribute configurations	130
B.11	Sales prediction of production 4068 using the micro-founded approach (300–400) .	131
B.12	Sales prediction of production 4068 using the micro-founded approach (400–500) .	131

B.13	Sales prediction of production 4068 using the micro-founded approach (end)	132
B.14	Sales prediction of production 5830 using the micro-founded approach	133

List of Tables

4.1	Results of a simple cluster classification by using past sales figures attributes	50
4.2	Results of a simple cluster classification using past sales figures and date related properties	51
4.3	Results of a cluster classification using past sales figures, date related properties and characteristics of tickets already sold	52
4.4	Results of a comprehensive cluster classification model	52
4.5	Results of the optimal regression to predict the final capacity utilization rate	58
4.6	Statistical evaluation of the baseline and data mining approach	67
4.7	The most influential attributes for DQ1 identified by wrapping	68
4.8	The most influential attributes for DQ2 identified by wrapping	69
4.9	Results of a regression based on the sales of the preceding week	75
4.10	Results of a simple regression without filtering of batch sales	75
4.11	Results of a simple regression with sales of previous weeks	76
4.12	Results of a simple regression with different structural features	77
4.13	Results of the total regression model	77
A.1	VORST_ID attribute summary	113
A.2	KUNDE_Postleitzahl attribute summary	113
A.3	KUNDE_Land attribute summary	114
A.4	KUNDE_Geschlecht attribute summary	114
A.5	KUNDE_Flag_Firma attribute summary	115
A.6	POS_Hierarchie0 attribute summary	115
A.7	POS_Hierarchie1 attribute summary	115
A.8	POS_Hierarchie2 attribute summary	115
A.9	TICKET_Ermäßigungsnummer summary	116
A.10	TICKET_Ermäßigungsbezeichnung attribute summary	116
A.11	TICKET_Ermäßigungsgruppe attribute summary	116
A.12	TICKET_Preiskategorie_ID attribute summary	116
A.13	TICKET_Preiskategorie attribute summary	117
A.14	PREIS_Platzkapazität attribute summary	117
A.15	PREIS_Verkaufte_Tickets attribute summary	117
A.16	TICKET_Vollpreis_It_Liste attribute summary	117

A.17	TICKET_Tatsaechlicher_Preis attribute summary	117
A.18	TICKET_Reihe attribute summary	118
A.19	TICKET_Sitz attribute summary	118
A.20	TICKET_Verkaufstyp attribute summary	118
A.21	TICKET_Rechnungsnummer attribute summary	118
A.22	TICKET_Buchungsdatum attribute summary	118
A.23	TICKET_Zahlart attribute summary	119
A.24	TICKET_Versandart attribute summary	119
A.25	TICKET_Kauf_Tage_vor_Vorstellung attribute summary	119
A.26	TICKET_Kauf_Tage_nach_Premiere attribute summary	120
A.27	KONTINGENT_ID attribute summary	120
A.28	KONTINGENT_Max_Tickets attribute summary	120
A.29	KONTINGENT_Verkaufte_Tickets attribute summary	120
A.30	VERANST_ID attribute summary	121
A.31	VORST_Veranstaltungsreihe attribute summary	121
A.32	VORST_Datum attribute summary	121
A.33	VORST_Wochentag attribute summary	121
A.34	VORST_Startzeit attribute summary	121
A.35	VORST_Spielort attribute summary	122
A.36	VORST_Platzkapazitaet attribute summary	122

Introduction

1.1 Motivation

In the modern world of competitive and globalizing market economy it is crucial for a market player to be ahead of the competition. In compliance with that, a profound mental picture of prospective developments of key indicators as a solid basis for decision making has become more important than ever. Modern computers in connection with the vast amount of data that is available today open up a new world of such future predictions based on pattern recognition.

Recently, the buzz-words big data and predictive analytics have become increasingly common, indicating the next big hype. Many companies and institutions enthuse about it and more and more are about to jump on the bandwagon. Further, reports on success stories are becoming increasingly frequent. They point out the improvement of understanding of one's business, the partners, the competition, and, of course, the customers, caused by these modern means when used adequately and correctly. There is a beautiful paraphrase made by Dan Ariely which perfectly summarizes the current situation: "Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it...". So whereas these means have become greatly ubiquitous, tracking and assessing the many marks that all of us leave behind, only a few people are actually aware of the technology and its capabilities at all. And even less of them are able to envision the societal and economic effects that are about to arise.

One major technology behind is data mining. Its main purpose is to extract exploitable meaning from data in situations where relationships are unclear. What has been known as data fishing in the 1960s as a derided offshoot of the prevalent approach of manual formation and testing of hypotheses has emerged to a prospering and promising tool in many different areas as from the 1990s [Selvin and Stuart, 1966; Fayyad et al., 1996]. Its ultimate goal can be seen as transforming human gut feeling to the computer to become less dependent on subjective intuition. This generic idea to teach the computer to get an intuition by itself is applicable to almost every domain imaginable. The only requirements are sufficient data and the assumption of coherences among them.

Nevertheless, many data mining efforts were facing severe problems and quite a few of them were doomed to fail. This circumstance kind of reminds of the situation the software industry was fighting with in the 1960s, which became known as the software crisis [Marbán et al., 2009]. Furthermore, the availability of scientific material on conducting and guiding such endeavor is very limited. This causes the proportion of craftsmanship and intuition that is required to be successful to still make up a significant proportion, despite the fact that the variety on scientifically grounded methods is tremendous [Pyle, 1999, chap. 3].

Hence, the process and success rate shall be improved by creating methodologies, just as it was done to overcome the software crisis. The core purpose of this thesis is to support this endeavor by providing a methodology on the basis of a data mining project conducted for a real world problem.

1.2 Problem Statement

Many companies have been collecting vast quantities of data over the last years either with or without a specific purpose. In many cases these data have not been used in any way at all. They have just been stored up to hope for better times. As early as 1982, futurist John Naisbitt stated his famous line “we are drowning in information, but starved for knowledge” [Naisbitt, 1982]. The amount of data that accumulates grows exponentially and doubles about every 20 months [Hilbert and López, 2011; Witten et al., 2011]. For them, better times have already begun with the emergence of data mining. For the first time these data can be used to exploit highly valuable economic knowledge.

The starting position of this thesis is represented by data of a cultural establishment in the German speaking area. This data consist of transactional sales information of theatre plays during a period of approximately 8.5 years. In the past it has been a challenge for this enterprise to make decisions as predictions were mainly based on intuition and gut feeling. Although this intuition came from highly experienced people, decisions have always had a subjective flavor. Over time, a simple model was created which takes three factors into account. Of course, the accuracy of that approach is limited, but subjectivity is reduced already. To overcome these shortcomings and remain competitive, a state-of-the-art and scientifically sound solution should be brought on board to be able to draw forecasts and support decision making of miscellaneous questions.

This study is intended as a feasibility study to point out the possibilities that can be achieved by a data mining approach and the information that is available. It is also meant to pave the way for potential subsequent follow-up projects, depending on whether the results are able to keep up with the expectations of the management.

The possibilities and chances that are promised by making use of big data and data mining, respectively, may convey the impression that there is some kind of magic involved. The only thing that would be necessary is to pour in the data and spectacular results would be spit out automatically [Fayyad et al., 1996; Pyle, 1999, p. 17]. Data mining itself indeed works exceptionally well as it is a mature field of research. Nevertheless, in order to solve a business case there is a lot more to do than it seems. The most prominent and underestimated working step is the preprocessing. It represents the critical step when raw economic data is put into a shape

data mining can make use of effectively and efficiently. This step actually makes up the major portion of the total effort of such project in most cases [Zhang et al., 2003; Myatt and Johnson, 2009, p. 2].

Preprocessing and other critical steps are encountered very rarely or in an insufficient breadth in studies that are available to date. Instead, it is rather common to present the data mining inputs readily converted and to concentrate on core data mining issues such as algorithms and optimization issues. In other cases, when preprocessing is brought into focus explicitly, the problem setting is only of a theoretical and hypothetical nature. But a combination of these two is exactly what is needed to work out a guideline with the goal to unify and simplify the process. The scientific aim of this thesis is to do exactly that.

1.3 Research Questions and Goals

One of the first steps that is necessary to employ data mining in an effective way is to define a clear-cut task description, including questions that are to be answered eventually [Pyle, 1999, p. 15]. Of course such questions may turn out to be inadequate or unanswerable by the means of data mining in further consequence. This could be due to quantitative or qualitative deficiencies that become explicit during the implementation. Nevertheless, several questions were defined as a preliminary point to assess the adequacy of this approach. The object of interest for all of them is the behavior of the end customer. These domain questions are as follows:

- DQ1** Is it possible to reduce the sales trends of single events to distinct patterns and how accurately can such pattern be predicted for an event prior to its performance?
- DQ2** How accurate are predictions of the final capacity utilization rate of an event in general and how does the confidence change over the time?
- DQ3** What are the influencing factors that play an important role for these problems?
- DQ4** How do predictions of sales trends and final capacity utilization rates look like when breaking them down to the individual price categories?
- DQ5** How can sales trends of entire productions be predicted and can simple prediction approaches be outperformed?
- DQ6** Is it possible to provide decision support for questions concerning the extension horizon of an ongoing production?
- DQ7** How can the success of a production be predicted very early in its life cycle, possibly even before its premiere?

Besides providing answers to these questions the task of this study from a functional point of view is to provide a technical base of operations for refining the solutions to these questions or to even elaborate new ones in successive research projects. For it must be clear that these questions may cover some interesting situations and may even provide reasonable decision support, but in

order to make use of them in a professional environment a lot more work is necessary than is demonstrated here.

On the other hand, while working out a methodology for this study the focus shall be explicitly drawn on all steps that are necessary to answer these questions. In this way it is possible to point out the tasks that are important in general and provide a real world example for carrying them out in one breath. Some prototypical and exemplary technical questions that will mostly reoccur at several questions are summarized in the following list:

- How can strong factors of influence be distinguished from weak factors and how does the performance change when just subsets thereof are used?
- Which techniques does data mining provide to answer the questions and how do their performances compare with each other?
- How can transactional data be converted to an appropriate time series representation and how can this in turn be processed by data mining?
- How can a problem be converted from a classification to a regression problem and which approach is more promising?
- How does the normalization of numeric attributes affect the final predictive power of a model?

Altogether, a thorough, holistic and sound overview should be provided by digging into the details of the process. The reader should have a feeling about the functional principles that he often does not get from the studies available. In order to support this purpose we do not just point out and describe the one and only solution that was selected in the end. Instead, alternatives are discussed as well, sometimes even those that did not prove expedient in the end but help to understand the problem. However, one needs to be aware that such methodology can never be complete. A data mining project can be continued and improved in lots of, seemingly unlimited ways.

The ultimate goal therefore can be regarded as the creation of a “sales prediction approach for cultural events by the means of data mining”. Hence, the endeavor to reduce the necessity of intuition and experience that is required to carry out such projects shall be facilitated. Last but not least this work will augment the range of data mining studies to comprise the context of cultural events that so far has not been dealt with at all.

1.4 Methodology and Structure

At the beginning of a data mining based solution solid and sufficient theoretical background knowledge must be acquired. We take a look at the basics of data mining, including the methods that are available and general solutions to problems that occur frequently. Additionally, we try to extract good practice from the variety of case studies available which could prove useful and inspirational for this one. This literary focal point will be presented in the adjoining chapter 2.

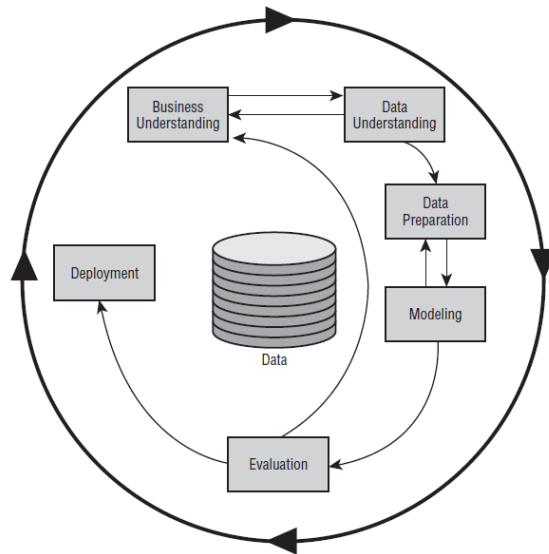


Figure 1.1: An overview of the CRISP-DM process model showing its six phases (taken from Chapman et al. [2000]).

After this overview we start with the core case study. The general procedure follows the widely accepted CRISP-DM process model that is depicted in figure 1.1 and which will briefly be explained in chapter 2.1. According to this model we conduct an analysis of the data as well as the the originating business domain in chapter 3 at first. We take a look at the attributes and assess their value range, distribution, whether and why data are missing and explain their technical context. Working out these first two blocks of CRISP-DM will be severely limited due to space constraints. However, we show some coherences and provide background information in the following chapters wherever relevant. We start the methodology with the general preparation of the data in the same chapter already. That implies cleaning, enriching and filtering data on a very low level to have them at hand in an appropriate form. With that, we address the domain questions sequentially in chapter 4, partially building up on each other by reusing insights already gained and data structures already prepared.

Basically, there is a variety of tools available to perform data mining tasks. The most prominent ones are comprehensive commercial suites that provide a considerable range of functionality and facilitate many steps. The biggest vendor in that field is SAS with its Enterprise Miner suite – closely followed by IBM with its recently acquired competencies of SPSS. SAP enters this top market as well slowly but steadily by providing a solid solution that follows a well-conceived strategy. Besides those heavyweights there are several smaller vendors that provide all kinds of solutions. [Gualtieri, 2013]

In this study we deliberately refrain from applying such solution but instead make use of the tools that are available freely as open source. By doing that we can depict our solutions in a detailed way and describe many tasks by carrying them out manually. Moreover, we can enjoy the pleasure of maximum flexibility despite the fact that this means more work in some cases.

The primarily used tool is Weka. Its way of data representation and analytical capabilities have been used for all research questions. We selected it because of its excellent integration in the Java environment – the platform that also represents the foundation of the small framework that we mentioned before. Another reason why we chose Weka is the fact that it is used frequently in the scientific context and offers a wide variety of algorithms. Important open source alternatives to Weka are R, RapidMiner and KNIME. In this study, R was used for some statistical evaluation and visualizations only.

On a micro-level, our approach exhibits an iterative character. That means, that we approximate each question step by step, and always try to improve the results by adjusting inputs and methods. We especially place value on comparing different configurations – i.e. data selection and preparation issues and algorithm selection issues, for our goal is to point out what happens when the screw is turned at different positions. In addition to that we repeatedly float hypotheses and answer them inline.

We want to stress that the true purpose of this feasibility study is not to create or make use of fancy or novel data mining approaches. Moreover, we mostly do not elaborate on the parameter optimization problem as the possibilities there would be sufficient for a study on its own. Instead, we focus on employing fundamental and established techniques to avoid drifting off the aim to work out a methodology that is as universal as possible. Interested readers are directed to literature that focuses on such innovative and interesting approaches and optimization issues of any kind. Nevertheless, certain approaches are required to solve the problems, especially DQ5.

Each domain question will, explicitly or/and implicitly, be accompanied by an evaluation to assess the presented solution. The only real comparison that can be drawn is between domain question two and the formula based on experience that is in use now, as mentioned before. To maintain readability and clearness we conclude the most important methodological issues at the end of each question as well. Due to the fact that most evaluation issues are treated along the corresponding question, the following chapter 5 is only to contain universally valid and general issues concerning the methodology as for instance the apportionment to similar domains. In the end, chapter 6 will present a conclusion of the whole case study and highlight some of the matters that could not be implemented due to several reasons but are desirable in subsequent research projects.

Related Work

As a precondition to carrying out this data mining project first of all knowledge about the basics of several related areas must be acquired. We will primarily concentrate on data mining as this represents the core of our approach. Nevertheless, due to the fact that the data at hand is transactional and especially not customer oriented, many of the questions concern time series indeed. We therefore will deal with time series analysis subsequently and illustrate the possibilities to elegantly conjoin these two different disciplines.

Following up to this theoretical background overview we will take a look at comparable scientific studies. We will point out the most important circumstances there and show why they can or can not be adapted for our problem.

2.1 Data Mining

Data mining is a manifold field of science. There are a lot of definitions, which in essence amount to the same thing in the end – namely, that its core purpose is to extract useful information from a big and rather orderless bulk of data. This information is manifested in the discovery of reoccurring patterns that may or rather may not be graspable for humans instantaneously. This information can then be used for different purposes such as forecasting and decision support as the most important ones.

Already in the 1990s, when data mining was in its initial years and the world seemed to drown in information already [Fayyad et al., 1996], the innovative approach could gain acceptance as groundbreaking tool in many different fields. Nowadays, the most important fields of application are marketing (for many different tasks such as customer segmentation, relationship marketing, association rule learning to detect regularities in buying behavior or prophylactic churn detection), fraud detection (for instance to curb tax evasion, but mainly at insurance companies [Gupta, 2006]) and credit ranking prediction at bank establishments [Witten et al., 2011]. However, data mining may also be used in other areas as in premature detection of crime as many cities in the US do in a Minority Report manner, or in general system monitoring [Witten et al.,

2011] and, last but not least, in health care to test the effectiveness of medication or to establish a rule based system to be applied in critical situations [Mayer-Schönberger and Cukier, 2013]. Of course this is just a subset of all fields, as data mining can be employed almost everywhere, as long as the availability of data is sufficient [Myatt and Johnson, 2009].

In fact, data mining is just the one single analysis step in the superordinate process of knowledge discovery in databases (KDD). However, it is often used synonymously, just as it is done here. KDD itself is an iterative process wherefore many different procedure models have been developed in the past years. The most common and de-facto approach is CRSIP-DM, which stands for Cross Industry Standard Process for Data Mining [Marbán et al., 2009; Gupta, 2006]. We already took a brief look at it in figure 1.1 in the introduction.

This model was conceived in 1996 by a consortium of five companies – SPSS, Teradata, Daimler AG, NCR Corporation and OHRA. It divides the KDD process into six broadly defined phases, which in turn constitute a hierarchical arrangement of the fine-grained working steps. These six phases are as follows:

Business Understanding In this phase the problem domain is analyzed in order to be understood. At the same time, the goals of the process are elaborated and brought into a form so that they are solvable by the means of data mining. The project plan is set up.

Data Understanding This step comprises data acquisition as well as getting to know and familiarizing with their peculiarities. This involves the necessity for a distribution, frequency and outlier analysis. Consequently, this and the previous step are traversed iteratively and interactively.

Data Preparation The preparation phase is passed through for each problem (and once for a general preparation in this case) in order to process data in respect to data selection, aggregation, cleaning, enrichment, attribute selection, transformation and derivation in order to make it suitable for the respective data mining approach in the end.

Modeling Modeling describes the application of data mining algorithms to extract knowledge from the previously prepared data – the core data mining step. Thereby, parameters are optimized and data may need to be refactored, resulting in several iterations of the preparation phase.

Evaluation The results of knowledge discovery are evaluated in this step, which is decisive for whether the solution is ready to be deployed, i.e. it is sufficient to fulfil the initially defined business objective in a qualitative respect.

Deployment Depending on the approach that was used, the results gathered so far are cast into a system that allows the customer of the project to derive a benefit. This could mean to simply apply the previously built model to generate statistical reports for instance – or it encompasses a full process, i.e. to build several models with live-data and to integrate them into an already existing management system. Nevertheless, the project is concluded in this step.

To get a more precise picture of the model we refer to the official manual [Chapman et al., 2000]. Basically, we adhere to this model in our methodology. The most serious deviation is that the last step of deployment is left out, as it generally very much depends on the system that is used afterwards. Nevertheless, we want to emphasize the importance of this step and, despite the fact that this work is a feasibility study, we integrated some parts of the models into the management system of the organization for demonstration purposes as well.

In essence, data mining makes use of statistics to achieve its goal. Many of the methods used actually originate from the notorious artificial intelligence, or more precisely, the machine learning field. Although these two scientific areas have very different origins, they are astonishingly similar in some respects. For instance, the widely used decision tree idea has been formulated by statisticians and machine learning researchers in the 1980s almost simultaneously [Witten et al., 2011].

By its very nature, data mining is restricted to finding empirical nexus. It is not possible to observe theoretical connections that do not appear in the data or states and processes that lay behind them. For that reason any identified relationship may not constitute a connection based on causality, but rather on correlation. We want to illustrate this important distinction with the example of stock price prediction. In the complex world of stock markets, prices are the result of the interaction of millions of mostly irrational decisions. There might be distinctive behavioral patterns that reoccur from time to time, but they usually can not be ascribed to specific causes (we deliberately neglect the undeniable impact of news reports in that example). Data mining may be used to evaluate hypotheses that were floated beforehand, but to extract reasonable and usable knowledge from price patterns only is doomed to fail. Despite that, any results gathered cannot be made use of in a sustained matter, for every measure represents a regulation in the stock market context. The extracted information, provided that it really increases success, would influence the implicit systematics in a way that they get invalidated in almost the same breath as more market players follow. This problem is a realization of the concept drift issue and will be encountered at a later point in this study as well.

Another problem that is related to some extent is the so called data dredging. Data mining works by floating and testing hypotheses automatically. In the light of the vast amounts of hypotheses involved the chance is big that the relationships found are in fact not statistically significant. No matter what significance level is chosen, the probability for that misjudgment is equivalent to the type I error. Apart from that it is important to be aware of the fact that data mining can only solve problems that were contained in the data it has been provided, for the only thing that data mining can do is to learn from them and extract statistical knowledge [Verleysen and François, 2005]. Novel problem settings and other situations which apparently mean different characteristics in the data can thus only be solved to a limited extent.

Despite these and other shortcomings, some scientists consider the data based approach the fourth paradigm of science, succeeding the empirical, theoretical and computational science. In this context, data mining is called data-intensive science, data driven science or simply eScience. [Hey et al., 2009]

For our case study, the two main areas classification and regression, respectively, and cluster analysis are relevant. Hence, we will concentrate on these two concepts and related relevant aspects of the technology. For the sake of brevity we wish to concentrate on the basics and would

like to draw attention of interested readers to widely available advanced literature. Beyond that, data mining encompasses anomaly detection techniques, association rule learning and summarization for the purpose of visualizing compressed characteristics of data for humans [Fayyad et al., 1996].

PREPROCESSING

As mentioned already, even the best methods are useless if the data brought into play exhibits an inappropriate form or quality. Hence, a lot of attention has to be paid to a proper preparation. Normally, the major portion, sometimes even up to 80% [Zhang et al., 2003; Piramuthu, 2004] of the total effort are put into this step alone [Pyle, 1999; Witten et al., 2011].

There are many different things to be considered. As most of them are encountered in data mining projects frequently, well accepted and tested solutions exist. We briefly want to describe those of them that are also met in this study.

Missing values are a problem especially encountered in business related data [Witten et al., 2011; Fayyad et al., 1996]. They occur when certain characteristics of an observation, or also called instance, are unknown. The reasons for that can be manifold. A common source for them is when they are optional in a form that is filled out manually. In other cases it may be concerned with measurement difficulties in general or in situations when the scheme has changed over the time resulting in a structural break. The interesting question is how to handle such cases. A straightforward approach is to simply ignore their existence and make use of a method that is capable of dealing with them. Usually this means that the missing value manifestation is treated as a value on its own. Another strategy to avoid problems is to estimate their likeliest value by looking at instances that are similar with respect to the other, nonmissing characteristics. In other cases, when data is not scarce, a simple way of avoiding missing values is to remove the observation that they belong to. Especially in this case, but also in all the others, missing values restrict the prospects of an analysis. In our study, we make use of algorithms that are capable of treating missing values as we experienced the best results by doing so.

More severe than missing values are measurement inaccuracies or faulty data in general as it is often difficult, or even impossible, to detect them and prevent them from becoming part of the data base. Moreover, even if they are detected it is not always clear how to treat them. However, one could follow the ideas demonstrated for missing values and remove the value or guess them from other instances, or even remove the whole instance. We detected several misentries in our data that are mainly a result of faulty user input. A special case of misentries are outliers which are usually easier to detect. However, due to the fact that many algorithms (for instance those who make use of least squares estimates) are very sensitive to outliers, resulting in distortions and biased coherences, it is even more important to detect and remove them [Witten et al., 2011].

In relation to this, it is crucial for all algorithms that use distances to normalize values, for a distance should not dependent on the unit it is expressed in. This so called feature scaling is relevant at numeric attributes only, i.e. those that only contain numerical values (in contrast to nominal attributes). The ultimate aim is to harmonize them and make them comparable and equally weighted in the end [Wu et al., 2008]. Consider for instance an example where each observation holds the height of a person in meters and its weight in kilograms. If two people are compared, they are equally similar when having one meter difference in height and one kilogram

difference in weight – an obvious fallacy. There exist different ways of normalization, including min/max normalization, zero-mean-unit-variance standardization or unit length normalization. If we for instance standardize the two attributes of our example, a comparison would take the sample distribution of each of them into consideration and eventually allow for a more sound comparison. Many algorithms automatically normalize data to a form that is reasonable for them to work with.

Some algorithms may require numeric attributes to be converted to nominal ones in order to increase their performance or make them applicable at all. This is especially relevant for classification algorithms that shall be used on numerical target variables. The solution to that problem is called binning and means that values are pooled into bins that represent a certain value range. These bins can have an equal range size or equal amount of instances that fall into them. The other way round, when nominal attributes have to be converted to numerical ones, is called binarization or 1-to-N coding. The result there is a new attribute for each nominal value which contains a binary value indicating its presence or absence. Other attribute transformations are important for several mechanisms such as the support vector machine – an algorithm that we will pay attention to later.

With that said, we want to finish this overview of generic preprocessing issues. Of course, preprocessing is not limited to that, but rather includes more specific, problem related actions that need to be taken into account. They will be demonstrated in the course of our methodology. For a comprehensive overview of the data preparation topic we want to refer to [Pyle, 1999].

Attribute Selection

In many cases, attributes are irrelevant or redundant for a specific problem. Irrelevancy means that an attribute adds no additional knowledge and redundancy that its information surplus is negligible compared to the others and just arbitrary noise is added [Piramuthu, 2004]. Many algorithms are distracted by such attributes, causing their run time to increase or the resulting models to be needlessly complex or even wrong [Witten et al., 2011; Piramuthu, 2004]. This statement may seem paradox at first as one assumes that the more information an algorithm gets the better it is able to extract knowledge. As we will also see in later chapters this indeed is not always the case. In the end, this problem is attributed to deficiencies of the algorithms. A closely related problem is called “curse of dimensionality” and attributed to [Bellman, 1961]. It describes the demand of an algorithm for data of sufficient quantity to the effect that there are enough examples available for each dimension. However, if this is not the case and the density is low in the value space, an algorithm is not capable of deriving meaningful insights as combinations of certain values are rare or even not present. The demand for observations is generally exponentially larger than the amount of attributes [Verleysen and François, 2005; Lutu and Engelbrecht, 2010].

Consequently, in order to avoid these problems it is a crucial issue to make a selection of attributes that are used for learning eventually [Piramuthu, 2004]. Due to the high relevancy, many scientific works exist and there are many ways for solving this attribute selection problem.

In a simple yet efficient first step, attributes are screened for relevancy manually. An important group of attributes that must be omitted a priori contains those that allow a single instance to be identified. Accordingly, especially IDs are concerned, but also date fields and any other

attribute that has many different values and each value belongs to a very small amount of instances only. These attributes are number one candidates for leading algorithms astray as the result would kind of represent a functional dependency. After assessing the usefulness of attributes manually one can make use of automated procedures.

The problem of attribute selection is generally NP-hard as many different combinations have to be tested exhaustively [John et al., 1994; Meiri and Zahavi, 2006]. Moreover, the results do not just depend on the subset that has been included in the learning process but also the algorithm used. Consequently, heuristics such as genetic algorithms, simulated annealing or greedy hillclimbing are usually used for solving that problem [Meiri and Zahavi, 2006]. A distinction is made between backward elimination (starting with all attributes and keeping removing them until the performance decline is significant), forward selection (starting with an empty set and keeping adding attributes until the performance improves no more or an inserted random attribute is selected) and bidirectional search, which combines these two.

Further, one distinguishes between wrapper and filter approaches. The latter try to select the optimal subset by the means of a universal relevancy criterion. This approach is very fast, but an effective universal relevancy criterion could not be found yet [Witten et al., 2011]. In most cases, correlation is used as the so called correlation-based feature selection [Ooi et al., 2007]. Especially in this domain, many different methods have been developed, for instance one that is based on decision rules and includes domain knowledge into the process (see Lutu and Engelbrecht [2010]). The filter approach also allows for single attribute selection. That is, every attribute is screened for its information content independently, resulting in a relevancy ranking. From this ranking the top x could be selected for instance. This method is even faster, but intercorrelations, i.e. dependencies among attributes, may be lost in doing so. Hence, subset selectors that assess subsets instead should be preferred.

This is also the idea that is used for the wrapper approach. There, the performance of a subset is assessed by the means of an algorithm that is to be used subsequently. Consequently, using this method, the optimal subset for a specific algorithm can be identified [Witten et al., 2011]. In our case we mainly make use of this approach, but we will also take a look at the filter approach to compare them.

As an alternative to attribute selection there is also the concept of feature extraction. When applied, new attributes are extracted from the old ones, replacing them eventually. The most prominent example here is the principal component analysis. Due to the fact that this approach yielded disappointing results, we will not go any further here.

In the end, we want to point out that rigorously reducing dimensionality to only keeping attributes that exhibit a certain influence complies with the important principle of parsimony, i.e. the “keep it as simple as possible but no simpler”-idea. It is therefore worth to be pursued with a high priority. For that reason, and also because of the relatively high scarcity that we are going to bear, attribute selection is a big issue for us as well.

REGRESSION AND CLASSIFICATION

Classification and regression, respectively, represent a form of supervised learning. There is a goal that is known a priori and the system is able to use for learning. In the course of learning

internal structures are adapted in a way that when receiving a specific input (independent variables), the corresponding output (dependent variable(s)) is generated automatically, provided that it has learned when to do so beforehand. This could for instance be the sex, age or probability of churn of a customer when providing his or her buying behavior as input.

In a general formulation, this problem can be expressed as the result of a function which has input parameters x_1, \dots, x_n . This function now may be an arbitrary method, or to be precise, algorithm, that has been calibrated on the basis of some training data.

$$\hat{y} = f(x_1, x_2, \dots, x_n) \quad (2.1)$$

Nowadays, there exists a variety of algorithms that differ from each other in several characteristics. It is emphasized that there exists no general purpose state-of-the-art algorithm that works better than most others – it is still quite an art to find the one that is optimal to a specific problem [Fayyad et al., 1996]. In the following list we take a look at three prominent distinguishing features:

Laziness When an algorithm is not lazy, a model is built in advance – the step that is known as learning. This model is then applied to new observations and tasks to obtain a result. Most algorithms work that way. But lazy (or instance-based) algorithms, such as the nearest neighbor classification refrain from that step and directly classify on the basis of training data by comparing observations with the new one. Consequently, the main memory requirements are very high and computational costs are shifted towards classification as it may take longer than normal, but the cost-expensive step of model creation can be dropped [Witten et al., 2011]. Therefore lazy algorithms are especially useful in situations where the learning base changes frequently and classification is done rarely.

Linearity In many problems, relationships between input and output variables are nonlinear. In the best case, a linear algorithm such as the linear regression is capable of approximating such behavior as it just considers direct input to output dependencies. Nonlinear algorithms are additionally able to consider dependencies among input variables, leading to more complex behaviors. In the end, when looking at a two dimensional input space, a linear problem can be solved using a linear separation line (see figure 2.1a) while nonlinear problems require a curve of a more complex shape (see figure 2.1b).

Classification/Regression So far we have used classification and regression as one concept – but what distinguishes them? The difference is simple yet crucial. In a classification problem the output, or target variable, is nominal, i.e. there is a finite set of possible discrete solutions that the result can take, the so called class. In a regression problem the target variable is numeric. As we will see, several algorithms have manifestations of both concepts.

After this overview we will take a brief look at some popular algorithms that have also been applied later in this feasibility study.

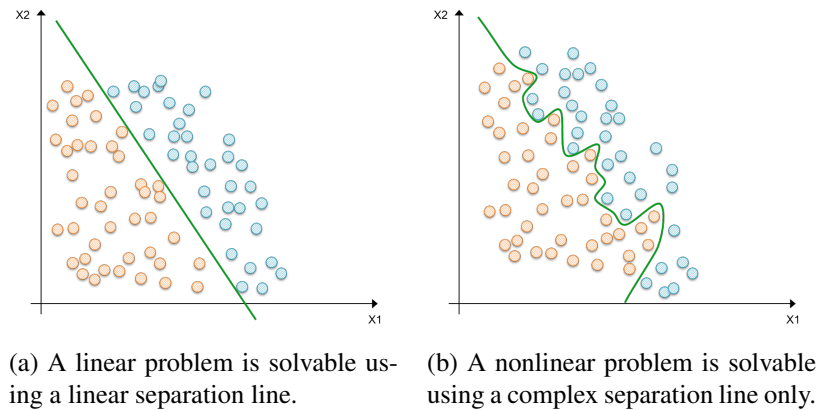


Figure 2.1: Two two-dimensional classification problems to demonstrate the difference between linear and nonlinear classification.

Decision Trees

The decision tree is a classical non-lazy, nonlinear classification and regression algorithm. As the name implies, its structure follows the idea of a tree. Each node represents a decision point where an attribute is tested for certain conditions. After passing through all relevant decisions, the leaf node that was reached eventually represents the result. This result can be a class for classification problems or a single numeric value in the so called regression tree or a linear regression formula in the so called model tree. These formulas can further be smoothed to ensure that the break between different linear models is not too big. There are several different ways to construct a tree from training data, but we will not consider them here.

The principle of decision trees is relatively simple and easily understandable. Nevertheless, they suffer from being instable, meaning that the removal or adaption of a very small fraction of training data can effect huge adaptations to the tree [Wu et al., 2008]. Further, trees are prone to overlearning – a problem that we will take a look at later in chapter 2.1 – which is usually met by pruning, i.e. the removal of seemingly arbitrary and insignificant paths.

There exist many different implementations of decision trees, such as the ID3 based C4.5 and its commercial successor C5.0, which represent the most commonly used variations [Witten et al., 2011], or the CART algorithm. Due to their practicable interpretability, decision trees can also be used for data description and help to better understand a problem, as we will also see at a later point. Apart from that, decision trees can be converted to decision rules (and vice versa) by deriving rules from the paths that are contained.

Generally speaking, decision trees are well suited for high-dimensional problems, missing values and can handle numeric values equally well as nominal ones [Fayyad et al., 1996, p. 12]. Finally, a simple two-dimensional classification decision tree is depicted in figure 2.2. On the left hand side there is the tree to the corresponding inputs that are displayed on the right hand side.

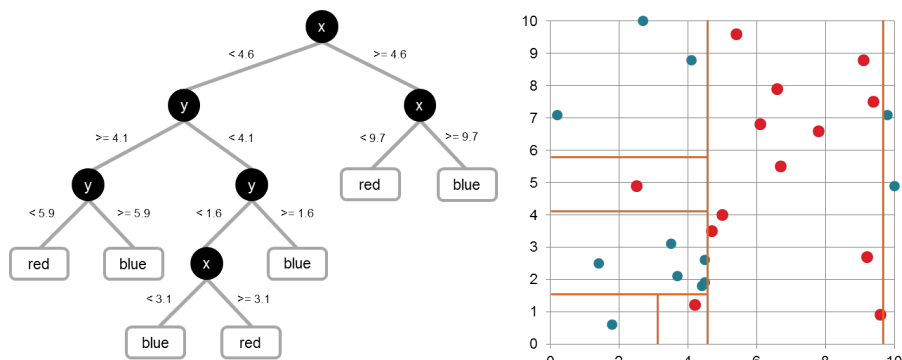


Figure 2.2: An unpruned decision tree to classify observations in a two-dimensional vector space.

Artificial Neural Networks

Another classic machine learn-algorithm is the artificial neural network. It is non-lazy and non-linear as well and can be used for classification and regression. The core idea follows the mechanics of a brain, i.e. there are many independent, but interconnected neurons. Interestingly, because of its groundbreaking novelty, such systems were built and tested intensively, long before the underlying theory was actually explored and developed [Kecman, 2005].

A very simple artificial neural network that is used in this context is the perceptron, which is organized in layers and in its simplest form contains just one of it – the input layer. Due to its restriction to linear problems, it has been extended to contain multiple, so called hidden layers, which allows for nonlinear behavior as well. One such multilayer perceptron is illustrated in figure 2.3. The connections between the neurons of different layers are provided with weights, which indicate their importance. By adjusting these weights and using an activation function, that is a stimulus threshold for making a neuron “fire” and pass a value to its output, neural networks can be trained and eventually used for classification.

They have been developed in the 1960s already and consequently are savagely well explored, developed and also applied. The dominant model architecture nowadays is the so called feed forward network (such as the perceptron) using the backpropagation method for training [Witten et al., 2011; Ahmed et al., 2010]. They are very flexible and adjustable, but on the other hand the resulting configurations are hardly comprehensible as it is hard to get a meaning out of connection weights. Additionally, artificial neural networks have many different input parameters, such as the amount of layers, the amount of neurons in each layer, the activation function, learn rate, learning cycles, etc. making it nontrivial to find the optimal configuration for a specific problem [Kim, 2003].

Support Vector Machines

The idea of support vector machines has evolved over several years and is as simple as effective. In their core, they are just capable of constructing linear separation hyperplanes in a vector

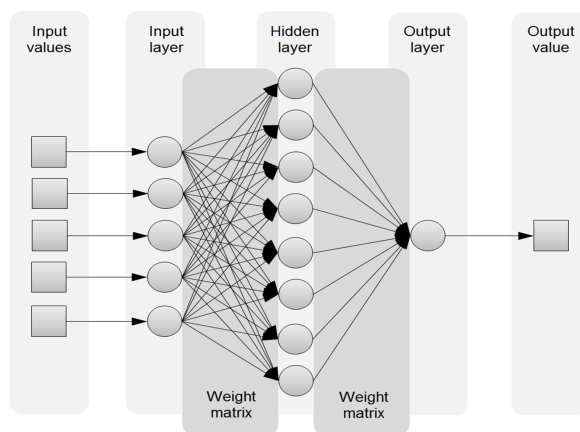


Figure 2.3: A feedforward artificial neural network – the multilayer perceptron with one hidden layer, five input neurons and one output neuron.

space. But by applying the so called kernel function, the original nonlinear problem can be projected to a higher dimensional problem space. Depending on the kernel, a linear separation is now possible in this new space. When calculating this separation back to the original space the result is a nonlinear separation of arbitrary shapes. This procedure is called kernel trick. There are different kernel functions available, but most of the time a linear (no kernel actually), polynomial or RBF kernel is used [Üstün et al., 2006].

There is also a variant of the support vector machine that is capable of predicting numeric values, the support vector regression. It has been developed by Vapnik in 1995 (see Cortes and Vapnik [1995]), who was also involved in the formulation the original idea of this algorithm. When applying a regression, a function is approximated in the high-dimensional space. This approximation is blind to small deviations and as flat as possible to avoid overfitting [Smola and Schölkopf, 2004].

In the end, a support vector machine can be interpreted as an extension of the concept of an artificial neural network [Witten et al., 2011]. However, due to the fact that not all observations are taken into account when constructing the maximum margin hyperplane, but just the so called support vectors (giving the algorithm its name), they are much more robust [Kim, 2003]. In figure 2.4 an exemplary demonstration of this characteristic can be seen. In addition to this, instead of just localizing a local optimum as artificial neural networks do, global optima can be identified more accurately [Kim, 2003]. Moreover, they have proven effective at many different problem domains, especially in those where data is sparse [Wu et al., 2008]. A drawback, however, is their considerable computational effort [Wu et al., 2008].

In the end we want to point out that there exists an alternative to the difficult quest of finding the optimal kernel and its configuration. This alternative is achieved by a flexible kernel, the PUK (Pearson VII universal kernel). The kernel is capable of taking any form between a Gaussian and Lorentzian shape and by that, substitute many classical kernels. This flexibility leads to an even increased performance and robustness. We also experienced that in our study. [Üstün et al., 2006]

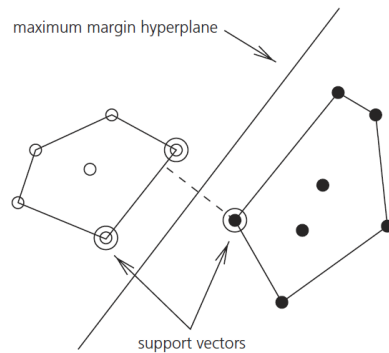


Figure 2.4: The support vectors of a simple support vector model (taken from Witten et al. [2011]). Their corresponding maximum margin hyperplane is the separation line that is as far away from them as possible.

Linear Regression

Linear regression is an almost ancient statistical technique and specialization of the regression analysis. Nevertheless, it is perfectly embeddable to the data mining context. As the name implies it is only capable of regarding linear dependencies for numeric target variables. In its basic form, the output Y is expressed by the input factors (the covariates) x_1, \dots, x_p and their respective coefficients β_1, \dots, β_p . The term ε represents disturbance or in other words the part that is not explainable by the model.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon \quad (2.2)$$

An important prerequisite to performing a linear regression is the independence of the covariates, usually assessed by a correlation matrix. The unknown parameters β_1, \dots, β_p are “trained” by least squares fitting in most cases. That is, the squared distances of the estimation are minimized. Figure 2.5 shows an optimal hyperplane for a three-dimensional problem and the respective distances to the training observations. To incorporate nominal input factors, we simply apply the already introduced concept of binarization.

In order to perform a linear regression in a classification context, the so called logistic regression has been developed. This concept uses a model for each class where the target variable is 1 if the respective observation corresponds to this class or 0 if not. By that, a probability can be calculated for each class. In order to prevent invalid probabilities (less than 0 or more than 1), a logit transformation is done, providing the name of this approach. In the end, every numeric predictor can be transformed using this idea [Witten et al., 2011]. Another difference to the classical linear regression is the parameter estimation technique. As a direct estimation of the parameters is not possible, they need to be estimated iteratively using a maximum-likelihood approach [Agresti, 2002].

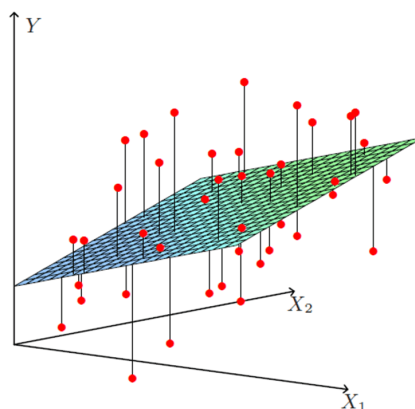


Figure 2.5: A three-dimensional regression problem and the regression hyperplane estimated via least squares (taken from Trevor et al. [2001]).

K-Nearest-Neighbor

The k-nearest-neighbor algorithm is applicable to classification and regression tasks and is inherently nonlinear. As a lazy learner it compares the instance to be classified with the k most similar instances in the data base. To assess the similarity of two observations a distance metric is used. There are several such measures as the Euclidian or the Manhattan distance. The prediction is formulated as the most common target value (or average in the regression case). Consequently, the choice of an appropriate k is important, yet difficult. If chosen too large, the outcome will be too general and not of great value. Instead, if chosen too small, the model tends to be overtrained [Wu et al., 2008]. To visualize this, we compare the results of a k-nearest-neighbor classification with k set to 15 and 1, respectively on the same problem in figure 2.6.

Naive Bayes

This classifier stems from statistics as well, as it works with hypotheses about the class membership of an instance to be classified. These hypotheses are then tested using conditional probabilities, i.e. the probability that a class results from the input parameters, assessed by the training data that is available. The class that has the highest probability is then selected. Consequently, this algorithm is only able to do classification, but can provide probabilities for class-membership. The idea is depicted in figure 2.7 as a one-dimensional problem, that is the body height of a person as input and the gender as the class. The dashed lines represent the true population distributions, while the other two represent the distributions of the training set. A naive Bayes classifier will set the separation line when the probability changes from one to the other, that is the solid red line which touches the x-axis at approximately 172.5 centimeters. The green dashed line on the other hand represents the true separation line, the so called Bayes optimal classifier, which of course is only theoretically known.

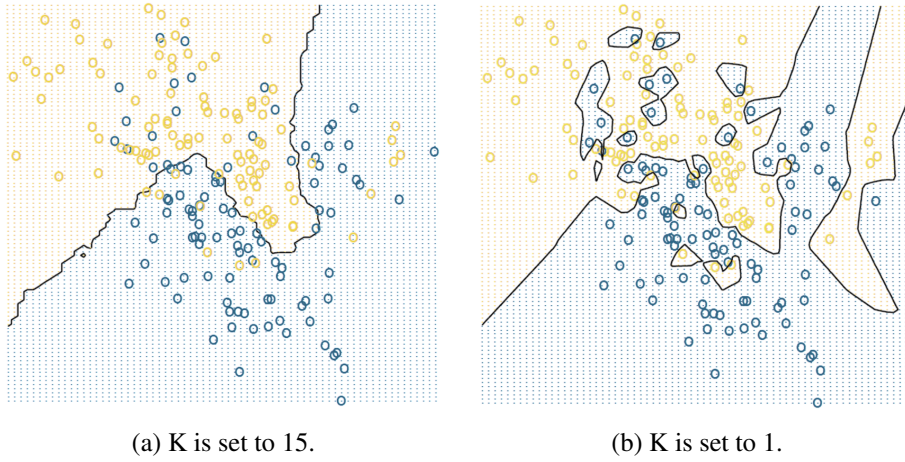


Figure 2.6: Comparison of two different k-values for the k-nearest-neighbor algorithm with the same example (taken from Trevor et al. [2001]).

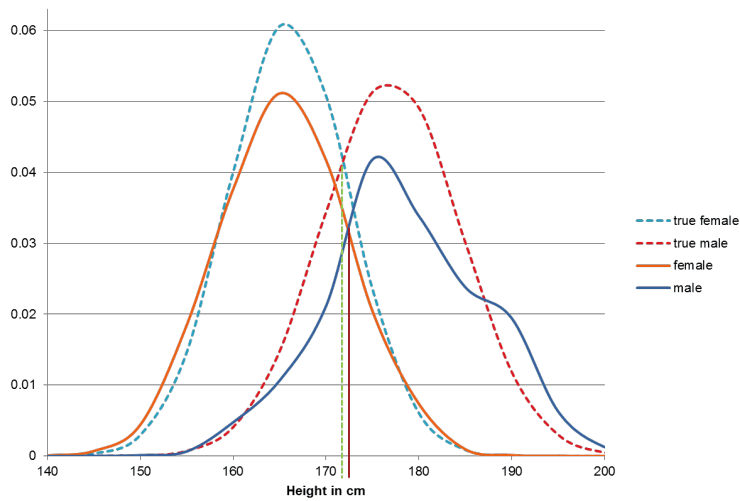


Figure 2.7: A naive Bayes and Bayes optimal classifier at a one-dimensional classification problem. The two lines represent the separation borders.

Due to its mechanics, a naive Bayes classifier can easily be updated when new training data becomes available. As with the linear regression, the major drawback of this method is the strict assumption of attribute independence, as the attribute values are combined as compound probabilities. This is not the case in most real world problems. On the other hand, this algorithm is very fast and statistically exact [Wu et al., 2008].

Meta-Learners

A very different concept than the algorithms that have been shown so far is the idea of meta or ensemble learning – ensemble, because multiple “weak learners” are combined, resulting in a meta-model. There are many different manifestations available, specific ones that are developed for an individual problem and generic ones that can be regarded as normal algorithms. We will concentrate on the latter here.

The so called **bagging** (from bootstrap aggregating) combines multiple models that have been built using the same algorithm. The idea is to draw multiple samples of the data base (with replacement) and build a classifier (or regression model) on each of them. The results are then combined, each weighing equally. This concept is relatively new as it has been developed 1994 by Breiman (see Breiman [1996]). In the end, a stabilized version of the base algorithm can be obtained. This stabilization is reached by artificially increasing the training sample size. Theoretically, by increasing this size to infinity, bagging can improve the quality of the base learner to its theoretical limit [Witten et al., 2011, p. 350]. Of course, as the complexity is increased, interpretability of the resulting models becomes impractical as well. In order to receive a performance surplus, the base algorithm needs to be instable basically. Consequently, trees or k-nearest-neighbor-algorithms with a small k are perfectly suited for that. On the other side, a linear regression or support vector machine can hardly be improved as they are relatively stable already. Bagging was used very much as it is effective, suitable for classification and regression and easy to be parallelized.

A similar concept to bagging is **boosting**. The major difference is that the models are not weighted equally but depending on their quality. Boosting has turned out the most powerful methods in recent years [Trevor et al., 2001; Witten et al., 2011]. It is even able to turn a weak learner such as ZeroR (a rule based classifier with just one rule) into a strong learner [Wu et al., 2008; Trevor et al., 2001]. We, however, could not make use of it in most cases as it is not possible to boost a numeric predictor.

Another fascinating concept is the so called stacking, a concept that especially fits to the term meta-learner, as it uses a machine learning algorithm to learn which algorithm fits best to a specific observation. Due to space limitations and low relevancy for our problem we will not go any further here.

Parameters

In the end of this chapter we want to put the light onto something that was observable partially already, that is the variety of parameters that most of the algorithms provide. We paid attention to the different configuration possibilities an artificial neural network or support vector machine has, but also trees for instance are to be configured concerning the pruning or the construction,

for instance. In some cases, selecting the right configuration can decide whether an algorithm is suitable for a problem at all, but it definitely has an impact on the model performance [Kim, 2003].

As already pointed out in the introduction we do not want to concentrate on this part of data mining too much as the possibilities there are vast. Nevertheless, we will see different base-configurations in comparison to each other later in the methodology.

EVALUATION

To be able to compare different models, i.e. the result of a parameterized algorithm and underlying training data, we have to make use of proper evaluation techniques. Some of them will be treated here.

Performance Measures for Classification

To assess the performance of a classification result, the simplest way is to look at the error rate, i.e. the fraction of instances that have been classified incorrectly and its complementary metrics – the accuracy.

$$\text{error rate} = \frac{\# \text{ misclassified instances}}{\# \text{ total instances}} \quad (2.3)$$

$$\text{accuracy} = \frac{\# \text{ correctly classified instances}}{\# \text{ total instances}} = 1 - \text{error rate} \quad (2.4)$$

However, when the class sizes are not homogenous, these measures might be misleading. Imagine for instance of one big class that contains almost all instances and one small one. If a classifier classifies all instances as the big class, the error rate is relatively small, but the classifier itself is miserable. To overcome this, two additional measures are available that are calculated for each class separately. The **recall** (or true positive) takes into account the amount of instances of a respective class that have been classified correctly in comparison to all instances of this class. Hence, the higher this value, the better the correct identification of examples of this class.

$$\text{recall}(x) = \frac{\# \text{ correctly classified instances of class } x}{\# \text{ instances of class } x} \quad (2.5)$$

The **precision**, in contrast, compares the amount of correctly classified instances of a class with all instances that have been identified as such. The bigger this value, the smaller the fraction of black sheep.

$$\text{precision}(x) = \frac{\# \text{ correctly classified instances of class } x}{\# \text{ all instances classified as class } x} \quad (2.6)$$

The **f-measure** (F) combines these two measures as the harmonic mean.

$$F(x) = 2 \cdot \frac{\text{precision}(x) \cdot \text{recall}(x)}{\text{precision}(x) + \text{recall}(x)} \quad (2.7)$$

To get a better insight of the classification result, the so called **confusion matrix** confronts true class-memberships with the estimated counterparts. Hence, in the ideal case, the only values that have values above zero appear on the diagonal.

Performance Measures for Regression

The picture of performance assessment looks quite different for numeric predictions. Here, the differences between all n predictions and the corresponding real values, the errors $(y_i - \hat{y}_i)$, represent the basis of most measures.

The **mean absolute error** stands for the average absolute deviation of all predictions.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.8)$$

The **root mean squared error** does not use absolute, but squared deviations. By doing so, small deviations are paid less attention and big ones more. In the end, the root is taken of the result to normalize the value.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (2.9)$$

Both these values are sufficient to assess different algorithms or configurations with the same data base, but not different ones. The reason is, that the mean value may be different, making it inappropriate to conduct a comparison based on these absolute values. Instead, relative measures can be used – relative to a simple predictor, as for instance the average value of the target variable in the training set.

In analogy to above, there is the **relative absolute error**

$$\text{RAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - \bar{y}|} \quad (2.10)$$

and the **root relative squared error**

$$\text{RRSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2.11)$$

In the end, an obvious quality measure is to simply take the **correlation coefficient** of the predictions and actual values.

Cross-Validation

As pointed out multiple times already, models are built upon training data. Consequently, a model contains information of exactly this data and accordingly, can value them quite accurately. To have a meaningful evaluation however, the performance of the classification of unknown observations needs to be assessed. Therefore, some instances need to be withheld from being used for training to have them at hand for testing afterwards. So a distinction is made between training and test (or holdout) set.

Of course, it is desirable to have as much training data as possible in order to obtain a model of maximum validity. On the other hand, the evaluation result should be meaningful as well, leading to an increased need for testing data. As a solution for this dilemma, the so called k -fold cross-validation was developed and has become standard practice [Trevor et al., 2001]. The

idea is to perform k steps where each time a fraction $1/k$ is used for testing and the remaining fraction for training. Hence, every instance is tested exactly once. In practice, 10 was found to be a reasonable value for k [Witten et al., 2011]. All in all, $k + 1$ models are needed – k evaluation models and the overall model which uses all instances for training. One could also go a step further and leave out just one instance each time, resulting in a huge amount of models. In return, the evaluation result has a great accuracy in most cases. This concept is called leave-one-out cross-validation. K -fold cross-validation was also widely applied in our context, but not all the time as we will see.

CLUSTERING

In contrast to regression and classification tasks, a correct solution is not known a priori in clustering tasks, hence representing a form of unsupervised learning. Instead, the task of clustering is it to explore structural similarities among the observations and generate new, “natural” classes [Witten et al., 2011]. As clustering is only relevant for one question, DQ1, we will just take a brief look of the most important algorithms here.

K-Means

K-means is one of the simplest and most popular clustering algorithms [Gupta, 2006]. It starts with k arbitrarily defined clustering centroids that are placed in the vector space. The distance of each instance is then calculated to those centroids by the means of a distance metric, which is the Euclidian distance in most cases [Gupta, 2006]. With that, each instance is assigned to the closest centroid. This centroid is then updated in the way that it centers its newly assigned instances. These steps are repeated several times until a stopping criterion is met. Hence, this algorithm works iteratively.

As most other greedy algorithms, K-means one is limited to find local optima. It is also very sensible to centroid initialization, making it practical to repeat the procedure several times. Additionally, the choice of a reasonable amount, k , is not trivial and subject to trial and error. Another drawback of this method is the fact that it cannot cover non-convex cluster shapes. On the other hand, and that is what makes it so successful, it is easy to understand and interpret, and a lot of extensions and improvements have been proposed in the past [Witten et al., 2011].

Expectation-Maximization

Compared to k-means, the expectation-maximization algorithm assumes a distinct probability distribution of each attribute. Consequently, as clusters can have different sizes and density distributions, the result is more accurate in many cases [Wu et al., 2008; Gupta, 2006].

As k-means, it starts with random centroids. In the first step, the expectation step, the affinity of each instance to each centroid is assessed. Based on that, the distribution parameters of the clusters are adjusted to fit those affiliations (maximization step). Therefore, the maximum likelihood parameters are used, i.e. the parameters that are most likely under the prevailing conditions. It, however, assumes independence among all attributes.

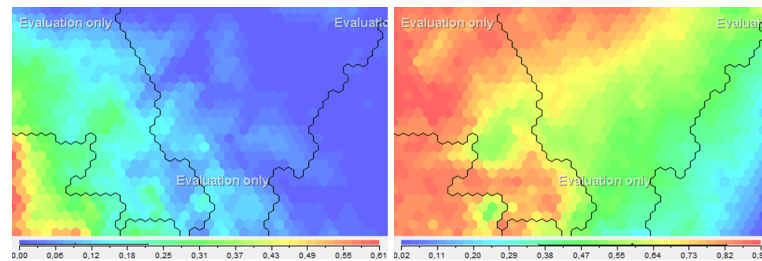


Figure 2.8: A self-organizing map based on several attributes of DQ1 of which two are demonstrated. The graph was created using an evaluation version of Viscovery SOMine 6.

Self-Organizing Maps

The self-organizing map is based on an artificial neural network – the Kohonen network. It consists of one input layer and one Kohonen layer, which interconnects all neurons with each other. As an artificial neural network, it emulates function principles of the brain itself. More precisely, it is based on a nonlinear projection of a high-dimensional space to a low-dimensional, most of the time two- or one-dimensional as this is easier to interpret and visualize [Kohonen, 2001]. This projection results in a constrained topological map that preserves original structures. This projection can then be used for clustering. Figure 2.8 depicts such a mapping and the corresponding clustering for a subset of two attributes based on DQ1.

Other Approaches

Apart from these algorithms, there are two methods that shall be introduced, although they have not been used in this study. The first one is the so called hierarchical clustering. According to this algorithm, clusters are joined step by step in the agglomerative, or bottom up, approach. The opposite, the divisive or top down mode starts with one cluster and breaks it down until each instance represents its own cluster. The major drawback is its computational complexity which is $O(n^3)$ [Gupta, 2006].

The second method is the group of density based approaches such as DBSCAN (density-based spatial clustering of applications with noise) that solely use the proximity of observations in order to form clusters. As a result, all instances are very close to each other within a cluster, making it possible to form clusters of complex and non-convex shapes [Gupta, 2006].

Evaluation

The evaluation of a clustering result is not as straight-forward as for classification or regression. The reason is obvious – there is no “true” result that can be compared. Instead, we have to use other measures. A simple one is the “within-cluster variation vs. between-cluster variation” which works good in many cases, but is not capable of regarding complex or non-convex shapes. As a result, it favors the results of k-means over those of DBSCAN nine times out of ten [Gupta, 2006]. In many cases, the only thing that remains is a visual analysis, which is quite hard when looking at high-dimensional data.

For our study we only look at the within-cluster variation to assess performance. Due to the characteristics of the problem, this approach can be considered valid.

PRIVACY

Privacy is a very important issue that is often neglected in data mining projects, especially when personal data are involved [Witten et al., 2011; Gupta, 2006]. Nowadays, in the age of transparent mankind, where tracking is becoming more and more common due to smartphones, and big governmental institutions observe citizens and people around the world, this topic has become more topical than ever.

In many cases, the data that is used for data mining has not been collected for this purpose [Witten et al., 2011]. As a consequence, entities, such as people or companies, do not know what their information is used for and far less are able to raise an objection to that. However, even when disregarding personal identities, data mining can lead to discrimination for instance when assessing creditworthiness of customers or classification of presumptive terrorists [Gupta, 2006].

Nevertheless, great attention should always be paid to reasonable anonymization of the applied data. This problem is tough in general as the combination of several attributes allows for an identification in many cases even if identification attributes have been removed already. The most established solution to that problem is the so called k-anonymization. It proposes the idea to manipulate the data base in a way that the subset of remaining identifying attributes fits to at least k different instances. By setting k sufficiently high a reidentification is not effectively possible [Gupta, 2006].

As concerns this case study, true identification features have been omitted a priori, when extracting the data from the source system. The remaining individual-related attributes are gender, country and postal code, so performing a k-anonymization is not necessary. Instead, an anonymization was conducted in order to mask the data provider himself. Accordingly, for this study every occurrence which could make an identification possible, as for instance a performance's or theatre's name was replaced by a random value. Furthermore, all sales trend graphs in this study that might reveal financially sensitive information have been deliberately distorted. We introduced shifts in order to always maintain the relative trends and characteristics revealed by the analysis while preventing the deduction of underlying absolute values.

2.2 Time Series Analysis

A time series is a set of observations, that can be put in order along a timeline. In this timeline, points are discrete and their intervals have a fixed length. Figure 2.9 shows a simple example of a monthly time series. Time series analysis is concerned with mathematical-statistical investigations of time series to reduce them to certain characteristics and describe the underlying data generating process, respectively. Time series forecasting in turn uses these resulting models to predict future values. [Palit and Popovic, 2005]

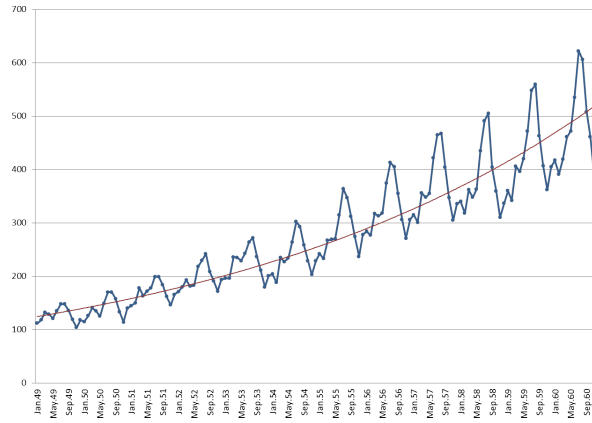


Figure 2.9: A simple time series of the air passenger dataset from Box and Jenkins [1976] that exhibits seasonality and a global trend.

The data that to be used offers two possibilities to create time series. Tickets can be arranged based on the time distance to the performance date of the respective event they belong to, resulting in as many time series as events. The other possibility is to focus on a whole production and create a time series for each of them, taking into account the distance that a ticket has in respect to the premiere date.

CLASSICAL APPROACHES

Just as the linear regression, time series analysis is a statistical special field of regression analysis that has been studied for decades. Over time, one dominant methodology has evolved – the Box-Jenkins methodology proposed in 1970. This work has caused a paradigm shift from the prevailing trend model, which assumes a deterministic process, towards a stochastic model that is capable of modeling almost any time series. The Box-Jenkins methodology popularized the autoregressive moving average model (ARMA), which was developed by Whittle in 1951. This model is composed of an autoregressive and a moving average polynomial. The first one, AR(p), formalizes the fact that each realization y is based on p preceding values, each weighted by a coefficient φ , resulting in a smoothed aggregation. ε represents the error term.

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t \quad (2.12)$$

The moving average part – MA(q) – defines that the white noise, or deviations ε of the past q values, again weighted by a coefficient θ , have an influence on the current value as well.

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (2.13)$$

The combination, ARMA(p, q) simply summarizes these polynomes.

$$y_t = \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (2.14)$$

An important precondition for the application of such model is the property of stationary, i.e. the constancy of the structural parameters mean and variance over time. In order to model a non-stationary process, a first order differentiation is necessary, resulting in the ARIMA extension. In a similar fashion, by calculating differences of adequate length, even seasonal factors can be considered, leading to the definition of the SARIMA model in further consequence. [Brockwell and Davis, 2002] Apart from that, many other extensions and generalizations of this model have been developed, as for instance the ARARMA, VARIMA, CARMAC, ARCH or GARCH models. However, space limitations prevent us from going into detail.

Basically, both our time series are non-stationary as they represent different phases of a finite life cycle. Additionally, they exhibit seasonal fluctuations as there is a summer break, over-averaged winter etc.

NONLINEAR APPROACHES AND OUR STRATEGY

There are a lot of different methods available to estimate the parameters of ARMA and similar models. Most of these methods assume a linear dependency of the independent variables as this restriction allows for simpler models and procedures [Lu and Chon, 2003; Bontempi et al., 2013]. However, as it turned out, this assumption is not appropriate for many real-world problems [De Gooijer and Hyndman, 2006]. Consequently, a manageable amount of nonlinear models have been developed as for instance regime switching, or the ARCH and GARCH models (for more information on that consult for instance Palit and Popovic [2005]). This development is still in its infancy compared to classical approaches, however [De Gooijer and Hyndman, 2006].

At that point we also want to mention the perception of time series problems that results from the dynamic systems theory – the state space analysis. It assumes that a value results from the respective state of a system. To obtain new forecasts this state needs to be reconstructed using past values. This so called space state reconstruction is conducted by the means of a state space model. This generic approach allows to describe ARMA and any other linear model. It is possible, however, to also model nonlinear models in a chaotic deterministic system perspective using extended concepts. [Durbin and Koopman, 2012]

In the end, also artificial neural networks have been used for time series analysis as they became popular very early and achieved good results in modeling nonlinear problems. For a time series forecast problem can be represented as supervised learning task straight away as the target value results from an arbitrary composition of preceding values. One could label this approach ML-AR. We, however, can easily extend this approach to include weekday, day of the month, quarter of the year and other date related information. This model would go beyond the scope of the basic ARMA model already and could identify seasonalities and, if including the year or another acyclic influence as well, even trends.

A lot of studies have been conducted to compare this and similar approaches that are based on artificial neural networks with classical models. In most cases they have proven superior [Ahmed et al., 2010; Bodyanskiy and Popov, 2006; Zhang et al., 1998]. Nevertheless, it must not be forgotten that this algorithm needs to be configured appropriately. Many solution attempts have been proposed for that, but none of them has become standard yet (see Mayer and Schwaiger [1999]; Peralta et al. [2010]; Montañés et al. [2002]; Bodyanskiy and Popov [2006];

Doganis et al. [2006] for examples). We will seize this idea in our study and use machine learning based models instead of classical statistical methods to forecast values. However, we will not confine ourselves to artificial neural networks to avoid the problem of parameter selection. In general, machine learning has been used for time series forecasting for two decades now, usually taking the lead in competitions [Bontempi et al., 2013; Ahmed et al., 2010]. Apart from that, it can be assumed that our time series exhibit nonlinear behavior as exogenous factors like press reviews, marketing actions etc. could effect temporary breaks in structural systematics. Nevertheless, even if it turns out that this is not a big issue, the flexibility of many algorithms allows them to fall back to a linear behavior in one way or the other.

However, the most important advantage of this idea is the fact that not just the autoregressive and date-related information can be passed to a machine learning algorithm, but also virtually any other influencing factor, even those that do not describe the past, but future characteristics that are known beforehand. And there are a lot of them available in the data base, for instance customer or ticket related qualitative information. However, a niche of time series analysis is to be mentioned, the idea of ARX or ARMAX models. These models are capable to incorporating one exogenous factor into a time series, and in more sophisticated approaches even more of them [Shumway and Stoffer, 2011]. Below, an ARMAX model is listed with one such exogenous factor as represented by the last polynomial. There, η_x represents the value of an exogenous time series at time x , weighted by a coefficient d .

$$y_t = \varepsilon_t + \sum_{i=1}^p \varphi_i y_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \sum_{i=0}^b \eta_i d_{t-i} \quad (2.15)$$

In the end, even if making use of such an approach, there would still be the problem that time series are relatively short in our problem domain. The first, event-oriented group would have a maximum length of about 300 days in certain cases, while the second, production-oriented one has only eight distinct series with a length ranging from 170 to 1,063 days. This would not be enough to estimate reasonable parameters using classical approaches. Moreover, as the series are self-contained, there are structural peculiarities that occur across all time series of the respective group, such as the already mentioned summer break or push in sales one month before a specific event, that could be valuable when forecasting a new production.

Taking into account all of these considerations, we chose this approach and call it **ML-NARX** – N for nonlinear, AR for autoregression and X for (multiple) exogenous inputs for our methodology.

MULTISTEP FORECASTS

As we will see, answering our research questions is not possible by using single-step forecasts only as it is normal business in time series analysis. There, just the successive point in time is predicted. Instead, multi-step forecasts need to be used which are capable of drawing predictions of an arbitrary horizon. Basically, there are two strategies to accomplish that.

Following the **recursive-strategy**, concatenated single-step forecasts are made, i.e. the output of one model is used as input for the next one and so on. Consequently, just one single model is needed which is applied as often as there are days between the cutoff day and the target

day. The drawback of this method is that errors can accumulate possibly leading to a distorted trend [Ben Taieb et al., 2012]. The following equation shows the idea. Again, y_x stands for the time series value and ε_x for the error term at time x .

$$y_{t+1} = f(y_t, \dots, y_{t-n}) + \varepsilon_{t+1} \quad (2.16)$$

Alternatively, there is the **direct-strategy**, which directly predicts the target value without considering any predictions in between. As a result, one model is needed for each horizon h , but errors cannot accumulate. Other drawbacks are the increased complexity due to uncertainty that results from long term horizons and the statistical independence of the predictions, which may lead to interesting situations when looking at several forecasts simultaneously. [Ben Taieb et al., 2012]

$$y_{t+h} = f_h(y_t, \dots, y_{t-n}) + \varepsilon_{t+h} \quad (2.17)$$

These two approaches can also be combined in an attempt called **DirRec**. Here, all previous forecasts of different horizons are used as input for the current horizon prognosis [Sorjamaa and Lendasse, 2006]. All these models have advantages and disadvantages. However, just one approach is suitable for our purpose – the direct-strategy. The reason is simple, as we do not use just the preceding values as input parameters, but also other qualitative factors that would not be available in successive models in the recursive strategy.

2.3 Comparable Studies

As we have seen in this overview, the theoretical background of the concepts involved in our study is well advanced. In this section we now want to complete our theoretical introduction by taking a look at scientific works showing their application in specific situations.

As a first starting point the small examples and case studies that are widely available in many fundamental books introducing for instance time series analysis or data mining could be considered. These include Gupta [2006]; Trevor et al. [2001]; Witten et al. [2011]; Myatt and Johnson [2009]; Brockwell and Davis [2002]; Shumway and Stoffer [2011]. However, in almost all cases they are very narrow in scope and illustrate a very specific subarea only, usually the one that was treated in the respective chapter. For that purpose they are just right – but in order to obtain an overview about the solution process in general and to be able to reproduce it, it will need more than that.

Thus, we will rather look at conceptually similar studies of other domains presented in papers and making up the large proportion of material available. A subset of them will be presented here. Each of them is relevant for our case study in one point or another by dealing with data mining, time series analysis or both. We will not consider papers that introduce a novel theoretical concept and show it by an example or compare it with different approaches. Instead, the focus is put on the application of established and well-founded concepts. In the end it is the methodology and not the method that counts for us, taking into account that a separation is not always easy. Apart from that, we want to point out that the choice of the former is considerably larger. This, as already mentioned in the introduction, is one of the main reasons why we create this case study.

Many of the papers deal with consumer goods, as the need for accurate forecasts is rather huge there. This domain is rather different from ours, but as concerns the technical means and proceedings a compliance is given to a certain degree. There is, for instance, one paper by Meulstee and Pechenizkiy [2008] that predicts 17 food and beverage products using an approach that is somewhat comparable to ours. Sales figures are aggregated on a weekly basis there. We deliberately refrained from doing so to obtain a better insight and have more data points available. The time series were further transformed using SAX – a symbolic representation of time series for data mining tasks – as sales figures were almost the only inputs available. Another way to overcome this shortcoming was to extract different key figures such as moving averages of different lengths, slopes, past values, etc. In contrast to our situation, where many exogenous factors are available and we do not have to rely on such derivatives only, the ones in this study are holidays, weather and promotion data of the retailers. Another considerable difference is that time series data of different products are incorporated into the forecast model. For us, this is not appropriate as there are usually just two productions in parallel, exposing heavily non-stationary characteristics, which results in a low expressiveness in the end. The authors followed a meta learning approach using k-nearest-neighbor to select the best base-learner for a series of given properties. The performance did not turn out to be that successful however, as the previously used moving average forecaster was beaten in some cases only. This was also important for us to decide against a sophisticated approach like this, as bagging and other simpler concepts are more flexible and possibly even better. But on top of all that, as we will see, there are only eight time series at hand, making it hard to work out reasonable distinctions. To overcome the performance issue, later studies Žliobaitė et al. [2009] and Žliobaitė et al. [2012] introduced an extended ensemble learner that distinguishes between predictable and seemingly random time series. Based on that, either the classical moving average or the new model is used. However, as an effective distinction is impossible due to lacking qualitative discriminatory factors, the result still underachieves.

In Chen et al. [2010] food, or fresh food to be more precise, is to be forecasted as well. It is concluded that artificial neural networks perform better than a moving average predictor, which is not surprising. However, it also beats a logistic regression which contradicts to our findings.

Clothes are subject of forecast efforts in the study Thomassey [2010] and elder papers of the same author. There is a distinction made between short-term predictions (a few weeks) for stock management and long-term predictions (one year) for a general sales estimation. This distinction makes sense in our case as well, as short term forecasts could be used for promotion management and long term ones to deal with problems like DQ6 and DQ7. As we will see, the gap between these two is less significant because of the short lifespan most productions have. The author makes use of a fuzzy inference system for the long term forecast, which represents a very interesting approach basically. It carries out a decomposition of sales based on their root influences such as promotion, weather, price, etc. to a structural baseline. Eventually, these factors are then reapplied on the baseline forecast. This approach is, however, inherently limited to considering linear relationships. Additionally, it has turned out to succumb a classical ARMAX approach. For this reason, we will stick with conventional data mining approaches. On the other hand, the method proposed to conduct short term predictions is rather similar to ours in DQ1. It provides that the time series forecast problem is substituted by a clustering of sales

figures trends and consequently estimated using a classifier. The inputs for that are qualitative factors which are available plentifully.

Moving away from consumer products, but similar to this clustering substitution approach, Yao et al. [2010] describes how customers are clustered on the basis of their buying behavior using a SOM based approach. To predict this behavior, the support vector machine, an artificial neural network and a boosted decision tree are used consequently. 10 demographic and behavioral independent variables serve as inputs. These approaches performed very well, with the SVM and the boosted tree being the best option. They are eventually augmented by several ensemble-methods, increasing the classification performance even more. We will refrain from doing so for reasons mentioned already.

From a domain perspective, we move even further away by looking at room temperature forecasts described in Mateo et al. [2012]. The paper formulates a time series problem with exogenous inputs by using the humidity, outside temperature, an indication whether temperature should be reduced or increased and the thermal power on an hourly basis. It briefly describes a preprocessing by transforming the time axis to have a cyclic character which is not necessary in our case as appropriate date related attributes can be derived. The authors compare a classical ARX approach with several others, including one that is called MLP-NARX, sharing many properties of our approach. It turned out to be the most effective one, encouraging us to stick with our endeavor. In the end, a single attribute based feature selection method is used (just as in Žliobaitė et al. [2012]), which has severe drawbacks as already pointed out. Consequently, the performance could be increased in one case only.

A technological similarity was also perceived in Ragg et al. [2002] by forecasting newspaper sales figures. In this context, a wide variety of 47 input factors, including holidays, important news of sport events and the like is used. Such factors would also be a good assistance for our models, but unfortunately they are not available as we will see. The task is to predict the day one week ahead. An artificial neural network is used to correct a moving average simple predictor, which in turn had to be improved eventually using feature selection. By applying a filter attribute set selector with mutual information as criterion, they realized that the model got significantly better when using data of the weekday that is to be forecasted only. We will also encounter this idea when dealing with DQ5 as we are facing a similar situation of weekday variations.

There are studies in related domains, like cinema admissions in Hand and Judge [2012]. This paper describes the use of Google Trend as input for a forecast of sales time series of cinema movies and clearly was an inspiration especially for DQ5. We will see that their findings can be confirmed as the result can be improved indeed.

Cinema movies, or their success to be exact, are also the focus of Stimpert et al. [2011]. However, the authors do not follow a time series approach but confine themselves to the prediction of the total box office revenues. They do this using a linear regression model with reviews, the genre, a rating of the leading actor, movie ratings, marketing expenses, an indication whether it is a sequel or not and the amount of screens it has been opened at together with the premier date as input factors. This configuration is interesting for us because it would allow for predicting the success of a production as is the goal of DQ7. In the end, reviews, sequel and the amount of screens turned out to be the most influencing parameters. Despite the fact that non-linear relationships cannot be considered, the linear regression provides an insight into simple

relationships and allows to derive simple rules of thumb. We will also use them for that purpose later especially in DQ3. A similar approach to predict music sales figures was proposed in Dhar and Chang [2009]. It additionally uses the amount of references in blogs and the amount of friends at Myspace as input variables. A linear regression model as well as a model tree was used to analyze the magnitude of influence of these two metrics. They eventually turned out to be rather valuable.

Another paper dealing with cinema movies is Marshall et al. [2013]. The authors, however, follow an approach that differs very much by using a mathematical model – the Bass diffusion model – which is ranked among the most established methods in the marketing context. It distinguishes between innovation-driven sales and imitation-driven sales, i.e. word-of-mouth-effects. We mention this model because it represents an important alternative to our data mining based approach and might overcome some of the deficiencies encountered. Other alternatives are behavioral models that are suggested in Ateca-Amestoy and Prieto-Rodriguez [2013] to forecast attendances of museums and jazz concerts.

Eventually, we also want to highlight another approach that is a great contrast to data mining. The paper Garber et al. [2004] proposes the spatial divergence as leading indicator for the success of supermarket products that were launched recently. The thesis is that word-of-mouth-marketing plays an important role therefore. This, according to the authors, can be measured using the spatial distribution of sales i.e. when they are irregular and form clusters. In the end, seven out of eight cases could be predicted correctly. Basically, we could also follow such idea as the country and zip code are available. However, as we will see afterwards, the quality of the values there would have to be improved and a solid mechanism to translate them to a spatial map is needed. Additionally, there are some rather severe shortcomings pointed out in the study, but it would be an interesting idea nevertheless.

SUMMARY OF EXISTING APPROACHES

As we have seen, there are no studies available that try to predict sales figures in the cultural context of a theatre establishment, especially using data mining or a comparable time series approach. Instead, one needs to fall back to different domains which are similar from a technological perspective although, as data mining is universally applicable. The domain that comes closest is cinema movie as well as music sales. Consequently, this study represents an important contribution by throwing the light on a novel domain.

In most cases, despite neglecting the data preparation phase, the last two phases of the CRISP-DM model, the evaluation in terms of a comparison with existing approaches, and the deployment into the operative system are treated. For us, this is not possible however, as just one domain question is answered up to now on the one hand and the application is not the focus of this feasibility study on the other. Moreover, data acquisition usually turns out to be much more sophisticated, which is why we are just paying very little attention to that step.

Going into methodological detail, existing time series studies mainly perform single-step predictions on a weekly basis as this is sufficient in their context. As our domain questions require different horizons, multi-step predictions are involved as well, performed on a daily basis to obtain a result that is as finely grained as possible. Beyond that, many studies, also those that were not mentioned, employ and compare a small selection of algorithms only. In

some cases, the artificial neural networks is the only real machine learning algorithm involved at all. This signifies the dominance of this approach that still holds today, despite the fact that some others might prove superior. Anyway, we could exploit some ideas by some of the papers for our study.

In the end, we want to cast the light on one of the success stories that were mentioned in the introduction that exhibits some astonishing similarities to our endeavor, namely Ostendorf-Rupp [2013]. This blog post superficially describes an attempt to forecast sales figures of a professional orchestra using similar parameters. As it is no scientific contribution, however, no insight into the methodology or detailed results is granted whatsoever.

2.4 Summary

Data mining is a technology that has been well researched in theory and extensively used in practice. Accordingly, there are many different algorithms available, exhibiting various strengths and weaknesses, depending on the context. Further, problems such as missing values, irrelevant or redundant attributes, but also privacy issues or obtaining reasonable evaluation figures have been experienced regularly. Eventually, generic solutions of different kind were brought forth to tackle them. Time series analysis represents a classical discipline as well, but only some aspects thereof were shown to be relevant in our solution context.

Using these technological means and the insights that have been elaborated in similar studies already, it is now up to us to appropriately employ and combine the tools in order to achieve beneficial results. A first step was already made by proposing the idea of ML-NARX, but more is to follow.

Data Analysis and Preparation

Just as a solid background knowledge, engaging with data is a necessity and advisable at the very outset of each data mining project. Hence, we are going to address step one and two of CRISP-DM – the understanding of the business and the data. As we are very limited in space, many issues of the former are delayed to the methodology when they become relevant. After this overview, we conduct a low level data preprocessing to prepare the data for its application in data mining. However, we only address those issues that are relevant for all domain questions. As a consequence, parts of step three – the data preparation – are covered here as well already.

3.1 Business Background

The data at hand originates from a cultural establishment, which arranges theatre performances in an en-suite manner. This establishment operates a small number of physical theatres and some other minor facilities. The plays, which are called productions in this context, can be distinguished into short-lived ones, which consist of very few single performances (also called events or shows) only, in some cases even just one, and large productions, which stretch over several years consisting of hundreds of events. In general, plays are performed simultaneously in the different locations, but each production is usually performed only once a day, five to seven times a week. Altogether, there are 55 productions and 3,240 events in our data, amounting to an average of 59 events per production. As the data is clipped out more or less arbitrarily during normal operation, some productions are not covered during their whole life cycle, i.e. from the first ticket sold until the last event. Hence, we need to take care of that when analyzing and processing them. Nevertheless, at least we can be sure that each event is contained completely.

The focus of this study are sales data, that is all information that concerns ticket sales. This data is held within an enterprise resource planning (ERP) system by which all sales are handled and recorded. Therefore, several distribution channels are available, comprising own box office and online sales, as well as external sales which are conducted via a partner company. Such tickets can be recognized by looking at this distribution channel information in general, but to

get more detailed information, contingent data and discount group fields need to be consulted. At this point, we want to remind that we focus on the behavior of distinct end customers. However, in some cases, external sales are represented in a coarse-granular way only, missing end customer and exact purchase date information. This may lead to a distorted picture when looking at sales figures trends in some situations. Hence, it is vital to filter such “misrepresentations” to meet our goal and be able to provide a clear picture about end customer sales.

There is a lot of customer-related information available, provided that it has been recorded during the purchase process. As a consequence, this is especially the case for tickets that were bought online. This circumstance would basically also allow for an orientation towards the customer him- or herself, resulting in customer behavior predictions, recommendations and other marketing related use cases. Most of this data, above all the identification field, is not available for our purpose although, conserving this considerable playground for further studies.

In total, 3,188,911 bookings are available. In the entire width of the relational structure this results in approximately 630 megabytes of raw data. While this may not be considered “big data”, the technical means concerning data mining are the same in the end. These bookings spread over about eight years and a half, but the events they belong to are contained within a time span of eight years exactly. It should be noted here that there is a summer break reaching from July to August, at which almost no events are scheduled apart from some exceptions and smaller productions, and as a consequence, sales are much lower than usual, especially than in the winter months.

After this brief overview about the origin and the business behind the data we want to cast the light on its structure and constitution.

3.2 Data Model

The data scheme of the ERP system is relational. However, it very much reminds of a typical snowflake-scheme already, which is common for data warehouses. Accordingly, in the center there is the fact table which holds a ticket and around that there are the dimensional tables event, contingent, price, discount, customer and point of sale. This data model is depicted in figure 3.1. Due to space limitations, we are not going into detail of all fields here. Instead, we refer to appendix A for a brief overview and translation of each attribute.

This scheme is ideal to conduct a denormalization, i.e. to abstain from relational dependencies and all advantages that result from it and instead condense each sale to contain all related information in one data set. This is especially important for data mining as most algorithms cannot deal with relational data. The biggest advantage is that data sets need not be merged on each access resulting in great performance losses, but once for all at the very outset. Since normalization is a very useful concept though, data is harder to maintain when denormalized. Due to the fact that there is no need to perform any changes in this context however, anomalies are a non-issue. Of course, denormalization also results in a higher memory requirement, which can easily be coped with in our dimensions.

In the final configuration, a record consists of 36 attributes that will prove more or less valuable. A closer look at them (as done in appendix A) reveals that the quality is sufficient to allow for further processing in general. Some attributes, however, need to be used carefully,

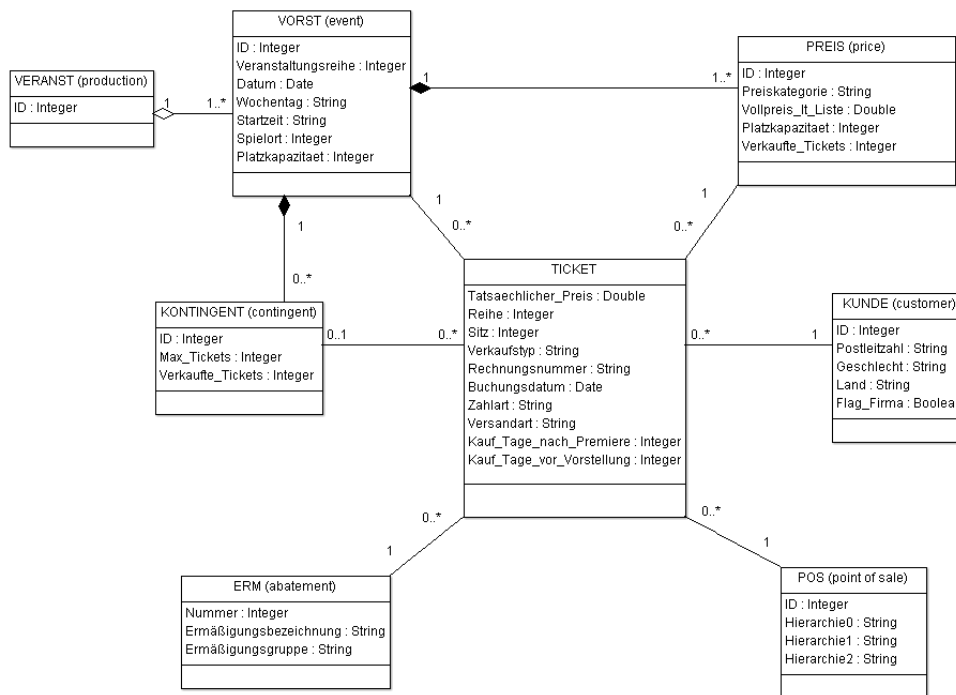


Figure 3.1: The data model of the problem domain, representing a typical snowflake-scheme. It only contains attributes that are available for our study.

raising the necessity of preprocessing. This is particularly relevant for the zip code and home country fields of the customer.

3.3 Data Analysis

At this point, following our process model, we would take a look at the data from different perspectives, drill down on different dimensions, aggregate data and look at distribution for different filters, as for instance each weekday or production. As this would go beyond the scope and space of this work, we cannot show the results though. Nevertheless, the so called data scientist or any person responsible for this project is committed to doing so in order to get a feeling and deep understanding about the data and its circumstances.

However, we are going to see some of the coherences in the following as already announced.

3.4 General Data Preparation

After this technical overview, but before addressing any of the domain questions, the data needs to be prepared in a very general way. This preparation comprises the extraction of the source system and the transformation to a basic data format that allows for being used in further analysis.

STEP 1 – DATA EXTRACTION

Even if data is collected explicitly for the purpose of being used by data mining, it might be still unstructured, heterogeneous and distributed among several sources. Hence, data needs to be collected, harmonized and brought into a sound and consequent shape. This problem, which is known as extract, transform and load (ETL) in the data warehousing domain, also represents a huge obstacle to data mining tasks [Witten et al., 2011].

In our case, data could be extracted from a centralized data storage with a uniform schema. It has indeed not been collected for data mining purposes, but as all applications make use of an ERP-system that relies on this base, consistency is very strong already and many steps that are usually required can be skipped. A simple CSV-export was performed in order to extract the data.

One of the advantages of the Weka internal format ARFF might be its performance, but building such file requires some extra effort if nominal attributes are involved. The reason for that is the need to create a dictionary of all values that might occur for each attribute beforehand so that an instance only needs to reference the according entry. Consequently, each export file must be processed twice, once for constructing the dictionary and once to load the observations. In that step all tables were joined and data was concatenated with the result that each ticket sale is denormalized as far as possible.

STEP 2 – DATA CLEANSING

When looking at the data in detail, we explored that some attributes exhibit obvious errors and impurities. They can now be removed right away up to a certain degree. Despite attribute specific rules that need to be elaborated for that purpose, we also need to look at cross-attribute values, such as '?', '-', 'no', or technical expressions such as 'unknown', 'moved away', which basically indicate missing values. Hence, they should be removed in order to have a consistent representation. Some attribute specific cleaning rules are:

KUNDE_Postleitzahl If the city has been appended to the postal code, this appendix is removed. The same goes for country prefixes that might come with a dash or not as well as customer salutations. In cases where this field has been used for telephone numbers (which can easily be detected by its different length) the value is set missing as well. Beyond that, there is a huge variety of misentries (as we have seen, there are more than 14 thousand different values) that complicate any further processing. They include any kind of misspellings such as arbitrarily entered special characters or country and city details. Many cases can be treated, but still there is a huge proportion of misentries, for instance zip codes that might seem, but indeed are not valid at all. It would be possible to correct them, but the benefit would not justify the efforts required therefore. By applying these rules, 9,871 instances could be repaired.

KUNDE_Geschlecht und KUNDE_Flag_Firma Both attributes have one occurrence of multilabeling that is corrected: at the first 'f' is replaced with 'w' (64 instances) and in the second '0' with 'y' (12,016 instances).

KUNDE_Land Any email-address that has been entered in this field is basically set missing. However, if the top level domain allows the country to be identified, this information is set as the new value. Multilabeled countries, such as fully written-out names, were reduced to the respective country code. Altogether, 72,748 instances were concerned.

TICKET_Preiskategorie For this attribute, multilabeling was a big issue as well. All categories were “stemmed”, i.e. reduced to their common root. Exactly 1,137,700 instances were cleaned here.

In the end, we can say that a 100% clean result can hardly be obtained, but, according to the Pareto principle the result can significantly be improved by a reasonable effort already. Nevertheless, especially for this step, a sufficient understanding of the data is essential.

STEP 3 – DATA FUSION

To reduce memory consumption and computational costs, that are both tremendous in many cases, we apply a simple compression to the data. As indicated by the invoice number attribute, a sale in most cases not only concerns one, but several tickets, i.e. when a family visits the theatre or a company arranges an outing. These tickets can be condensed to a single instance, as in most cases the only information that is different is contained in **TICKET_Sitz** (seat) and **TICKET_Reihe** (row). (We remember that the gender and other individual-related information is taken from the purchaser only.) Of course, we lose this information by doing so, but as these two attributes represent candidates for creating functional dependencies, this does not matter at all. Further, the information provided by the seat row attribute can substituted by the price category information quite well. To prevent other information from getting lost, we create a new instance each time a relevant attribute changes. In the end, the data can be compressed from 3,188,911 data records to 878,159, resulting in a compression rate of 27,54% without any effective loss of information. This approach represents a simple variant of the concept that is known as data fusion or data merging [van der Putten et al., 2002].

Of course the amount of observations that are represented by one of those new instances needs to be retained, as it would result in tremendous distortions. Therefore, we make use of the ARFF-internal instance weight concept as well as create a new numeric attribute.

In the course of this compression step some further attributes that are that are not relevant for further analysis can be removed. This early manual attribute selection considers all attributes that allow for an identification, i.e. all identifiers such as **TICKET_Rechnungsnummer** or **TICKET_Ermäßigungsnummer**. Further, we already know that the attributes **VORST_Spielort**, **VORST_Veranstaltungsreihe**, **KONTINGENT_Max_Tickets** and **KONTINGENT_Verkaufte_Tickets** will be irrelevant, so we remove them as well.

STEP 4 – DATA FILTERING

We now proceed with a domain related preparation step, as we conduct a low level data selection.

Basically, tickets for premieres are not sold in an ordinary way, but assigned to important persons such as press representatives, persons responsible and other “notabilities”. Consequently,

the sales data of such special events are useless for our analysis. They can be identified quite easily by comparing the values of the time before the show (TICKET_Kauf_Tage_vor_Vorstellung) and days after the premiere (TICKET_Kauf_Tage_nach_Premiere) attributes. Using this rule, about 16 thousand tickets (not instances) were identified and removed.

For administrative reasons, not all productions that are contained in the data hold their respective premiere sales. Instead, they are often recorded in separate “productions” that only consist of this one event. In a similar manner, administrative productions exist that hold ticket information of warm up events and rehearses. They are to be excluded as well, as these tickets are distributed extraordinarily too. At this point, it is also advisable to take a closer look at the productions in general. It becomes apparent that there are some further administrative entries as well as very short productions and special occasions with a few seats only (less than 100). As our goal is to predict big productions and their single events, we eliminate them as well. In this manual assessment, 32 of the 55 productions were identified as irrelevant, resulting in the deletion of approximately 100 thousand tickets.

We now continue with a filtering on the booking-level. In further consequence, a distinction has to be made between filters that affect the effective seating capacity and the consequent capacity utilization rate calculation and those that do not. Free tickets, for instance, are given out for productions that underachieve or are not fully occupied in most cases, representing a severe control influence. But, as the costs would incur anyway, these measures are legitimate for the establishment. In our context, they can be viewed as empty seats. On the other side there are sales that, because of their properties, cannot be used for sales forecast in general. The reasons for that are different as we will see in the following list of rules:

1. As mentioned already, free tickets are to be excluded. They can easily be detected by looking at the discount group attribute. The capacity is not affected by this step.
2. After that, we are going to remove special seat tickets. This group of tickets exists for almost every event and usually comprises 8–14 seats that are freely distributed to firefighters for instance. Consequently, these removals must be considered for the effective capacity, as they are not sold ordinarily, but if they were normal seats, they could.
3. We continue with virtually free tickets. They can be identified by comparing the list price and actual price, as well as looking at the actual price only. If the discount exceeds 99% or the ticket costs less than a certain threshold, they can be considered as gifts and consequently do not count as sales.
4. As we have seen already, there are tickets that are booked into the system belatedly. These tickets have a negative value at the attribute that indicates the days before the performance takes place. In most cases these tickets are correction transactions, so they need to be considered for the capacity.
5. It is a general rule that, the finer-grained data is, the better. However, it is a common problem in data mining that data is only available in an aggregated form [Zhang et al., 2003]. This is also the case in our study, as pointed out in chapter 3.1. In the data, there are larger re-sale partners, responsible for a bigger fraction of all ticket sales. These bookings

are bulk-loaded, usually over the period of three to one week before the respective show is performed, resulting in the loss of information about the exact purchase date as well as about the purchaser. This would not be much of a problem if the share of tickets sold via such partner would be approximately the same for all events. In fact, this is by far not the case as in some situations a partner is not even involved and in some cases sales make up to 20%. Hence, to preserve consistency and avoid distorting effects, we have to exclude this significant amount of tickets. Of course, they need to be considered for the capacity.

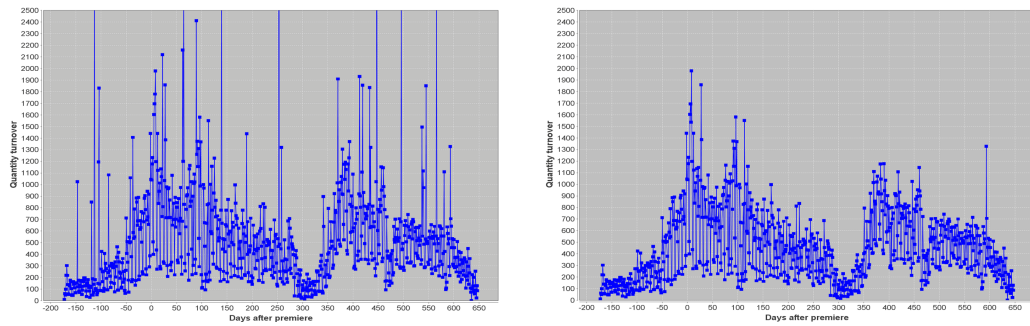
These tickets can basically be identified by looking at the contingent identifier. As we miss the field that describes the company behind, we collected them in the source system and used an exported list of identifiers for filtering. This list of IDs must of course be extended when further data is added.

6. The last step is even more sophisticated, but also a result of contingent sales of partners. Some of these sales, which can be identified using the discount group, are distributed and recorded in batches. In many cases these are season related promotions or direct promotions of the respective partner. They are usually distributed very early in the life cycle of a specific event – about 300 to 200 days before the performance. In figure 3.2a such batch sales can be seen as peaks in production 4068.

A forecasting algorithm will never be good enough to foresee such peaks as there are simply no conceivable systematics behind them that could be exploited. Instead, these data must also be removed and regarded in capacity determination. To identify discount groups that are predominantly distributed in batches, we developed a heuristic, as a clear separation is not possible. This algorithm works as follows: we start by grouping all tickets according to their discount group and booking date. We then remove all those groups from the list that have more than 30 distinct dates for at least one production, which indicates that they occur on a regular basis. Likewise, all groups that have more than 100 distinct dates when ignoring the production are removed. As a result, groups that have high numbers per day by nature, such as the full price group, are excluded. The remaining groups are assessed for their average sales per date – again by ignoring the production. Eventually, we remove those that have more than 50 tickets sold each day on average. What is left are all groups whose tickets are “bursty”. But before removing them, we might manually assess the list of groups for those that should be included nonetheless. After doing all that, approximately 150 discount groups could be identified. The parameters mentioned were chosen by trial and error.

The comparison at figure 3.2 shows the effectiveness of this approach. Of course, it is far from being perfect. But again, following the Pareto principle, we could achieve a reasonable compromise for cleaning our data base to contain data items that are valuable for our analysis. Moreover, some of the remaining peaks are a result of marketing campaigns that were active on single days only, which legitimates their existence.

After working through all those steps the effective seating capacity can be determined by subtracting the accumulated amount of tickets from the real capacity. This needs to be done for the event as a whole, as well as for the respective price categories that are affected. In some



(a) Sales figures without filtering, i.e. all discount groups.

(b) Sales figures without discount groups that are sold in batches.

Figure 3.2: Comparison of sales figures of production 4068 with and without batch filtering. Most of the peaks can be removed while “ordinary” days are not affected at all.

cases a curious situation appears as the resulting effective capacity is negative. This, however, is only true for price category capacities and due to the fact that their assignments are incorrect sometimes. To overcome this problem, we simply adjust the effective capacity to be at least as large as the amount of tickets sold. As we do not know which categories have been mixed up, we cannot reduce their capacities in turn. By using these numbers, we can calculate relative utilization rates by normalizing absolute sales figures. All this information is added to our data base, resulting in four new attributes. One could eventually consider removing all events that have a remaining effective capacity of less than 100 as the increased granularity could result in distortions. We did not do this, however, as performance was not influenced by that at all.

In the end, approximately 20% of all tickets were removed by these booking-level-rules.

STEP 5 – ATTRIBUTE ENRICHMENT

After applying these steps we have obtained a result that is clean enough for further processing. But before that we extend our data base by some attributes that could prove useful in further consequence. We start with simple derived attributes such as the weekday, day of the month, month, day of the year and year for the booking date as well as the performance date. These attributes are necessary as we will not use the respective date fields as they could result in functional dependencies. We have seen earlier that this is inappropriate.

Additionally, we append information about holidays and vacation. The former can be obtained by using a third party library called Jollyday¹ which provides the list of holidays for many countries of the world. To incorporate vacation information we manually created a list of all periods of no school such as the winter or summer holidays for the respective country. This information is then added for the booking and performance date as well. In cases the respective country of the customer is known, it can be used for holiday identification in Jollyday¹. If not, we fall back to the country of the cultural establishment as this is the most likely one.

¹<http://sourceforge.net/projects/jollyday/>

Other simple derived attributes are absolute discount (list price minus actual price) and relative discount (absolute discount divided by list price) which are added as numeric attributes. Using the original seating capacity we can infer the theatre that was involved. Fortunately, this seating capacities are distinct and allow for a disjoint separation. As already mentioned above we cannot use the attribute VORST_Spielort for theatre identification as its values represent an administrative allocation only. Hence, we create a new nominal attribute that has individual manifestations for the main locations and one for all other small venues.

After these rather simple attributes, we are going to introduce some that are more complex and harder to acquire. The first is a distance metric between the purchaser's home location that can be obtained when looking at the country and zip code and the theatre's location that is known now. For that purpose we took a rather pragmatic way by using a website, i.e. <http://www.luftlinie.org> as no webservice could be found that was capable of doing that in a simple way without encountering limits on the number of requests imposed by most routing services for instance. We wrote a simple web-client that went through all possible combinations of home and theatre location (altogether there are about 17,000), sent a request to the website for each of them and parsed the result to obtain the distance in kilometers. In the course of doing that, we actually realized how many of the codes were actually invalid as about 1,500 combinations did not yield a result. In order to increase hit ratio, as the website is unable to resolve all zip codes, even if they are valid, we retracted the city labels if originally appended to be able to provide a more general description of the source location. We could even increase the amount of city supplements by consulting another website that represents a zip code dictionary for all German speaking countries, i.e. <http://www.plz-suche.org>. The result could be improved, but not that much as many misinterpretations and invalidities remain. Additionally, we realized that some distances are invalid or simple nonsense as a specific number was returned for those that are unknown, resulting in a peak in the histogram. We tried to rule out such entries as good as possible, but again the Pareto principle applies as for most cases the result is just good enough. In the end, this new numeric attribute has about 35% missing values and a mean distance of 139.68 kilometers.

In a similar way weather data was collected – again for the date of purchase as well as the performance date. We used the website <http://www.wunderground.com> that allows for retrieving different weather metrics at different locations up to many years in the past. For our case, we will just use the precipitation and the mean temperature of the respective day. By collecting this data using another web-client, our newly created attributes have mean values of 8.77 and 8.31 for the temperature fields (booking date and performance date), and 0.84 and 0.82 for the precipitation fields.

As we have seen in Hand and Judge [2012], Google Trend data can have a positive impact on forecasting performance. Hence, we are going to add this data at this point already as it might prove useful for different purposes. A Google Trend generally stands for the relative interest of a specific search term. This interest is measured on the basis of frequency this term is “googled”. To incorporate this measure, we identified a representative term for each production with the help of related search terms which indicated when a term was too general or too specific. In most cases it was the name of the production in connection with the city it was performed in. With this term, we restricted the search period to the actual playing period and obtained a list of

weekly Google Trend values. These values are normalized to be between zero and 100, reaching from no to the maximum interest. Of course, it is a great pity that the granularity is coarse as we do not get daily values. But we can help ourselves by interpolating them. This outcome is eventually added for the booking date, resulting in about 16% missing values and a mean of 34.82.

Similarly, we could also think about querying Google about the trends of all productions at the same time, which would result in a relative comparison between them. The values would, however, be even more coarse as the measurement interval is changed to monthly values automatically if the period that is observed becomes too large. Further, the procedure that was shown three times now can be used to incorporate any other external input such as Facebook, Twitter or website click data, promotion data, press information, economic factors and many more. We will look at that later in the discussion, chapter 5.

3.5 Summary

In this chapter we could catch a brief glimpse of the background of the data and the data itself. Although circumstances could be covered in a superficial way only, many issues were found to be highly relevant as they could be used to improve the data base in order to make it ready for action. Hence, we want to emphasize the importance of a sufficient understanding of the data and the problem domain to be aware of the special treatments and special conditions that lie within the data and to have an idea how to react to them.

By applying a basic cleaning of the data, the necessity to consider some issues became obvious, as for instances “irregular” sales could bear great distractions in the light of our objectives. Another technical peculiarity was found when dealing with contingent sales or discount groups that are distributed in batches to partner companies. A considerable amount of time needs to be spent with such essential preparation issues already. Some of the measures taken will probably not even lead to a quantitatively improved result at classification or regression tasks that are conducted later, but rather improve the representation in terms of reflecting end customer behavior.

On the whole, the Pareto principle was encountered several times, i.e. the fact that a simple attempt can effect a huge improvement of data quality already. In most cases, we left it at that since we are doing a feasibility study here.

Methodology

After dealing with steps one, two and also parts of step three of the CRISP-DM model, resulting in the data base and an overview about it, we can now proceed in our methodology. Hence, we next address the essential data mining phases – data preparation, modeling and evaluation. Each of them is basically required for each domain question. We are going to treat them sequentially to be able to reuse knowledge and configurations already prepared whenever possible.

At this point we again want to emphasize that the goal of this feasibility study, from the functional point of view, is not to provide a full scale decision support system that can be employed within a reporting- or dashboard-system right away. Instead, we want to assess the theoretical possibilities and provide an approach for them.

4.1 Domain Question 1 – Sales Trend Patterns

CLUSTERING OF TRENDS

The task of the first domain question is to identify “common” sales trends. We start with creating time series by ordering sales using the time distance to the actual performance date. This results in one sales trend for each event. To obtain our common trends, we follow a classical clustering approach. The input parameters for the clustering algorithm could be the sales amount for each day before the event takes place. However, as this would result in more than 300 parameters, the algorithm would struggle to identify coarse distinctive characteristics. Furthermore, there are many days, especially those that are very early in the life cycle, that would not contain any sales for most events at all. If we would follow this approach, these “insignificant” days would gain as much attention as the performance date itself, which usually is much more busy and more valuable as a consequence. Hence, we instead perform an aggregation into intervals that are orientated towards the frequencies of all events. The result is an almost equal frequency binning into the following 20 groups: infinity–300, 300–250, 250–200, 200–175, 175–150, 150–125, 125–100, 100–90, 90–80, 80–70, 70–60, 60–50, 50–40, 40–30, 30–20, 20–15, 15–10, 10–5, 5–1 and 1–0.

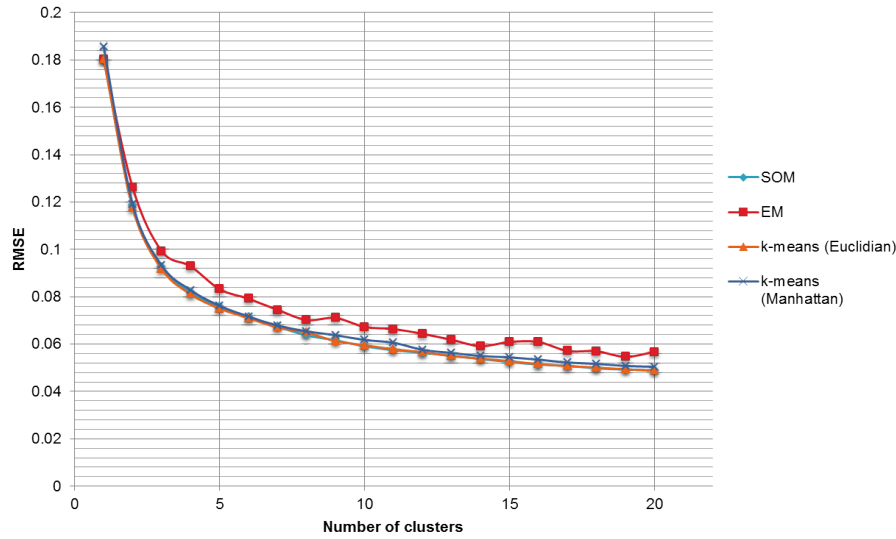


Figure 4.1: Comparison of different clustering algorithms with different quantities of clusters.

With that, we are going to carry out our first normalization already. In fact, the result would be heavily distorted if we used absolute sales figures after having filtered out so many tickets before with the effective capacities ranging from 59 to 1,227. Instead, we are going to use relative sales only, i.e. the capacity utilization rate at these respective points in time. We are also going to accumulate those values so that the last attribute holds the final rate. In the end of that step there is an instance for each event, holding 20 numeric attributes with no missing values.

As already mentioned, we are going to evaluate the effectiveness of our clustering approach by using the within-cluster variation – that is the mean deviation of a specific trend to its cluster, or the deviation of each time interval to be more exact. Accordingly, the MAE as well as the RMSE can be calculated for each interval as absolute numbers, representing the occupancy rate deviations.

Figure 4.1 compares the results obtained by applying a SOM (using a one dimensional lattice), EM, k-means with a Manhattan-distance and one with Euclidian distance. The x-axis stands for the amount of clusters that was created and the y-axis for the RMSE.

It becomes obvious that, using one cluster only, all algorithms except k-means with the Manhattan measure identify the same “representative” sales trend, that is the average of all events. The Manhattan distance measure is responsible for finding a different combination that results in a slightly higher RMSE. In general, EM underperforms in all cases, possibly attributable to its requirement of attribute independence, which is obviously not given in this case. The SOM and k-means with Euclidian distance perform almost equally well. Due to performance reasons (building a SOM takes a considerable amount of time) we chose k-means with Euclidian distance to be used in further consequence. We set the amount of clusters to 10, which is rather arbitrary, but seems to represent a good compromise between average deviation and parsimony.

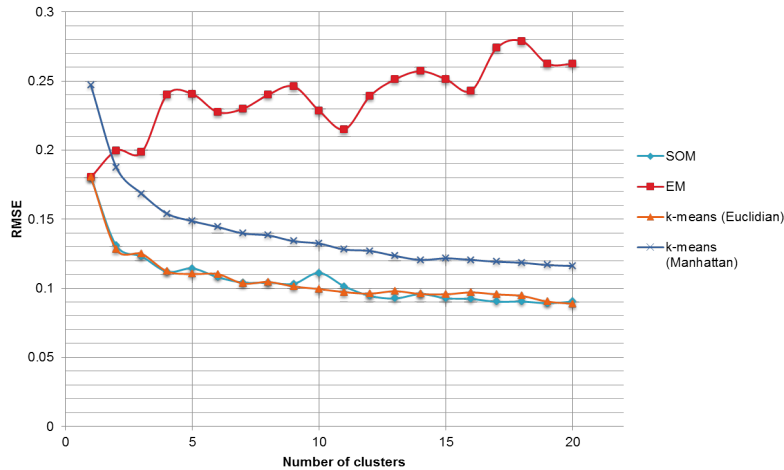


Figure 4.2: Comparison of different clustering algorithms with no accumulation of input factors.

▷ **Hypothesis:** We float our first hypothesis by claiming that performance is affected by the accumulation of the occupancy rate values.

As we can see in figure 4.2 a clustering that has been conducted without accumulated values (they are accumulated afterwards for evaluation) results in a RMSE that is way above the accumulated version. Apart from that, EM stands out especially, exhibiting even raising RMSE values when increasing the amount of clusters. We could not identify a clear reason for that, but obviously the algorithm is unsuitable for this problem. With these results, we can conclude that algorithms do care about accumulations as the lower performance can be attributed to the fact that deviations build up themselves here while they do not when using accumulations. ⇒ **YES** □

After defining the configuration we can take a look at the results of the clustering, i.e. the ten different prototypic sales paths depicted in figure 4.3. As we can see, the event distribution is rather homogenous, which allows for the accuracy to be used as comparative measurement in further consequence. For each cluster we can also take a look at the distinct trends that make up its shape. Figure 4.4 displays all events for cluster 0.

▷ **Hypothesis:** The clusters allow to infer the production of most events that constitute it.

When plotting the cluster distribution for each production as done in figure 4.5, we can see that they are not really equivalent, despite the fact that in some cases clusters are more frequent than others. Instead, clusters are a result of structural differences that are reflected in those productions, largely responsible for the inhomogeneity. Figure 4.6 illustrates this with some structural examples. Accordingly, there are clusters which rather stand for events that are set around the summer break, some that are more common for Saturday shows and some that are especially significant when the average price paid per ticket is low. ⇒ **NO** □

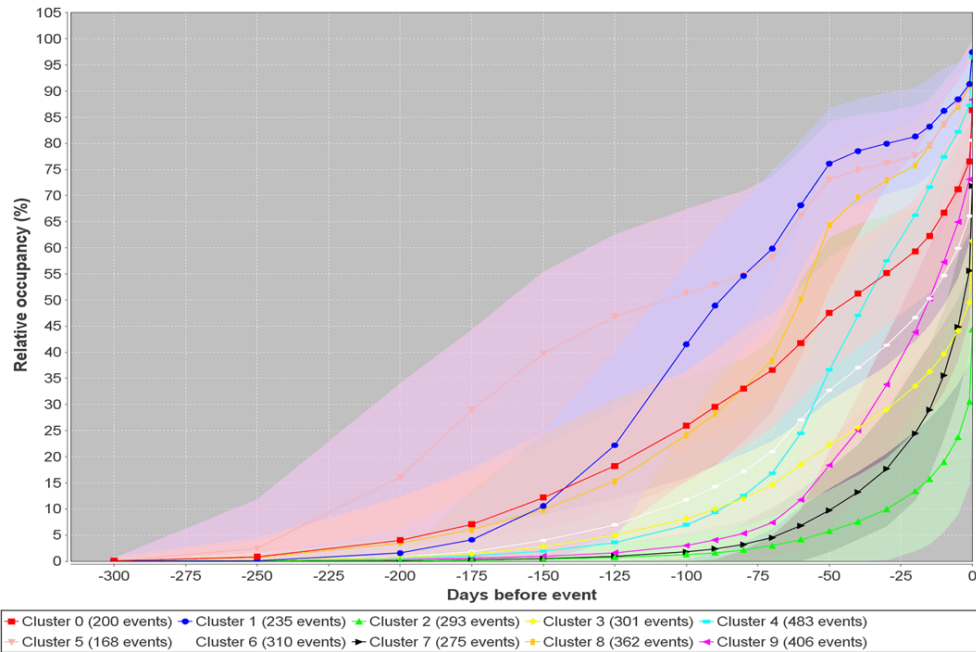


Figure 4.3: 10 clusters of sales trends of all events identified by k-means clustering. For each path, the variation is displayed as shade around the line with the width equal to its standard deviation. The legend informs about the amount of events that belong to each of them.

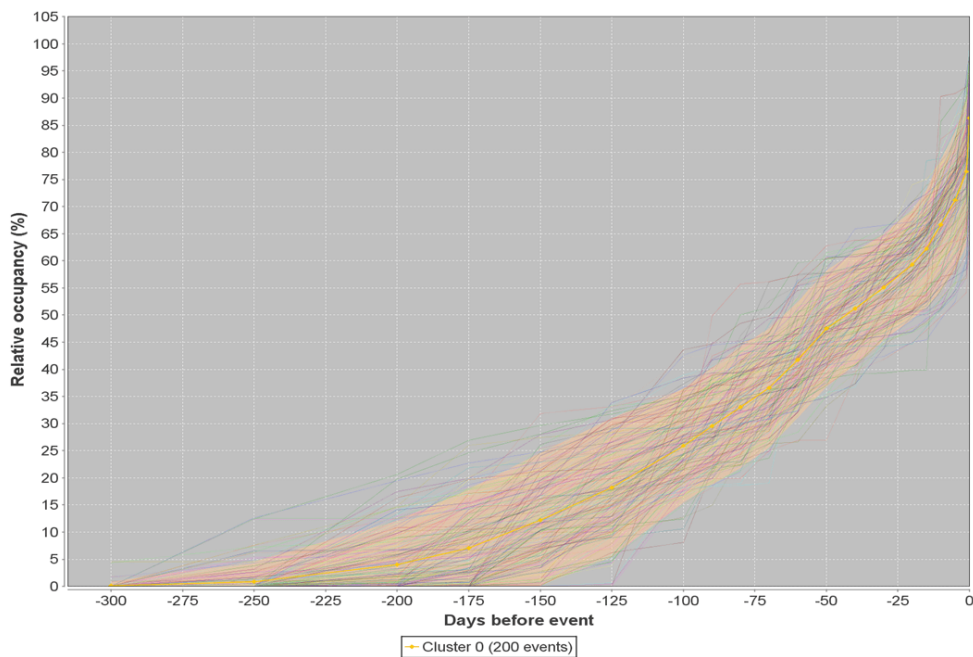


Figure 4.4: Cluster number 0 with all event sales trends that constitute it.

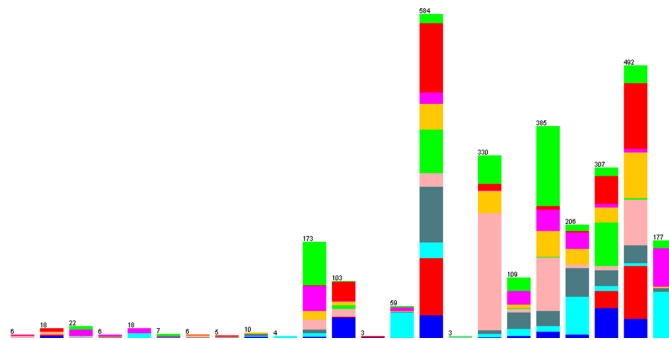
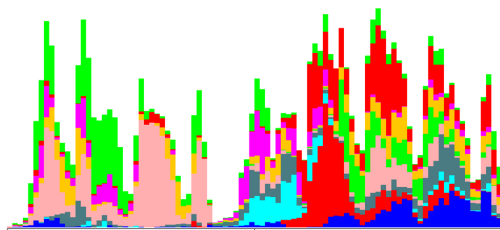
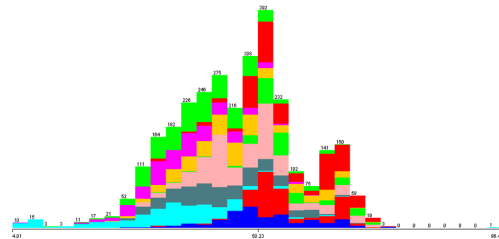


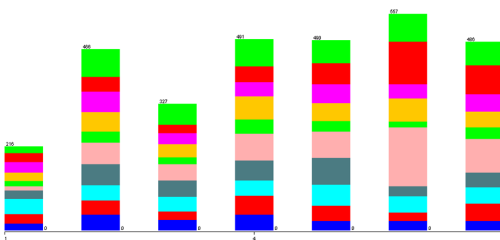
Figure 4.5: Cluster allocations for all productions that are involved. Each color represents one cluster and each bar one production.



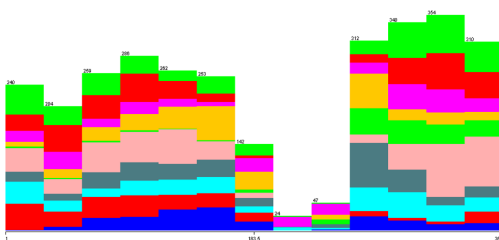
(a) Time of purchase.



(b) Average actual price.



(c) Weekday of the performance.



(d) Day of the year of the performance.

Figure 4.6: Cluster allocations at some structural criteria that expose a comparably high distinguishability. Again, each color represents one cluster.

FORECASTING FUTURE TRENDS

Answering the first part of the domain question was rather easy, but now it is about to build a model that is able to predict the cluster of an event that has not been performed yet to be able to have an idea about its future sales development. In a pragmatic attempt we could simply take the preceding sales trend and compare it with those events that have already be performed to obtain a result. In the following, we will refer to these parameters as SF, or sales figure attributes. The data base for such model, or the training data to be more precise, would then consist of these attributes under the condition that only those are taken into account that are also available for the event to be classified. The reason for that can be justified as follows: assume for instance that there is one attribute among those that are not available which exhibits a very high information content (discriminability), e.g. the one that indicates the sales of the performance day. The instance to be classified does not have that information available, forcing the model to fall back to those available, possibly resulting in a bias as an important portion of the combined (nonlinear) solution breaks away.

Hence, we create a data corpus of all events except the one to be classified and all sales figures attributes up to the point in time that the prediction is to be made, e.g. 10 days before the show. Unlike before, we are not going to use accumulated values as multicollinearity is a confounder for many algorithms.

Table 4.1 shows a first classification result by using a naive Bayes classifier. We selected this approach as it is simple, fast and sufficiently accurate for our purpose, but most of all because it does not make use of any automatic attribute selection or any other optimization as we want to demonstrate this later. We chose 10 days before the show as cutoff date. On the left hand side several evaluation criteria as introduced in chapter 2.1 are displayed whereas on the right hand side the confusion matrix can be seen. Each row stands for the actual class and each column for the estimated one. The attribute groups used in this and all further examples are itemized in the appendix C. All in all, about 59.77% of all instances are classified correctly, which is not that overwhelming.

Table 4.1: Results of a simple cluster classification 10 days before the show by using past sales figures attributes only.

Naive Bayes		94	16	1	4	9	16	35	1	23	1
		9	195	0	0	0	12	0	0	19	0
Attributes	SF (17)	0	0	199	35	0	0	0	48	0	0
Accuracy	59.77%	6	0	16	176	4	6	58	10	2	23
∅ precision	60.9%	5	0	2	2	352	0	36	0	18	68
∅ recall	59.8%	13	4	1	1	0	146	0	0	3	0
∅ f-measure	59.2%	27	6	4	39	32	3	166	6	12	15
		0	0	48	29	0	0	7	129	0	62
		47	52	2	0	24	24	12	0	200	1
		0	0	8	11	47	0	16	45	1	278

As we have seen, there is a lot more qualitative information available for each event, so this rather simple solution can be extended. In the first place, we could take date related attributes that were extracted in step five of the general tasks. Obviously, information about the weather may not be used as it is usually not known precisely enough 10 days in advance. With these

four new attributes we obtain the result shown in table 4.2. A very slightly improved accuracy of about 60.23% is achieved.

Table 4.2: Results of a simple cluster classification using past sales figures and date related properties of the events.

Naive Bayes		50	47	0	2	1	16	65	0	19	0
		8	206	0	0	0	7	0	0	14	0
Attributes	SF+EQ (21)	0	0	199	21	0	0	0	62	0	0
Accuracy	60.23%	5	5	0	186	3	8	22	43	1	28
\emptyset precision	61.2%	4	2	1	1	305	0	43	0	13	114
\emptyset recall	60.2%	7	1	0	1	0	151	0	0	8	0
\emptyset f-measure	59.6%	22	4	0	83	57	4	118	0	4	18
		0	0	45	13	2	0	0	192	0	23
		20	100	0	0	19	23	29	0	171	0
		0	0	1	14	36	0	10	103	0	242

▷ **Hypothesis:** The production gives an indication of the success of an event.

This indeed is the case. By including the production attribute into our data base, the result can be improved to 62.48% accuracy and an average f-measure of 61.9%. However, we must admit that we neglect any temporal ordering in this context. Further, when classifying events of a new production, an improved accuracy is not to be expected as the model got no chance to learn about it. Hence, we refrain from using it in further consequence but again keep looking for further structural relationships that were already partially visible in the clustering context.
 ⇒ **YES** □

Depending on the cutoff day, there might be some tickets in the data base already, whose characteristics could be valuable as well. To attain representative values, we set the minimum amount to 20. The characteristics that are going to be used are the average purchase price, the average amount of days the tickets were bought before the show, and others. The complete list can again be seen in the appendix C as the group “TQ”. In addition to this, one could add the variance, minimum and maximum values for each of them as well, but we will refrain from that. Concerning the nominal attributes the binarization-concept will be used. That is, we create a new attribute for each distinct value and count the amount of tickets that have this value. The result is a good many new attributes of which we made a selection that seemed and proved useful. These include the gender, the amount of customers from the same country of the cultural establishment, the amount of contingent sales and the sales for each main price category that are additionally normalized by their respective effective capacity. The classification result of this model is listed in table 4.3. We can see that the predictive performance is greatly increased.

Intuitively, an event has poor sales when figures stagnate over long periods of time. To incorporate this information, we take measurements of the time that is needed in order to reach the next decile of occupancy rate. This results in nine new numeric attributes – one to indicate the point in time the first decile has been reached (as taking the interval from the first ticket sold would result in a tremendous variance) and eight that express the amount of days that have passed before reaching the next one. As a consequence, again depending on the horizon, many events lack values especially for higher deciles, as some of them might not even reach them in

Table 4.3: Results of a cluster classification using past sales figures, date related properties of the events and several characteristics of tickets already sold.

Naive Bayes		150	2	0	0	7	2	25	0	14	0
Attributes SF+EQ+TQ+x1 (45)		10	204	0	0	0	6	0	0	15	0
Accuracy 77.73%		0	0	210	20	0	0	0	52	0	0
\emptyset precision 78.2%		2	0	0	231	0	0	24	28	0	16
\emptyset recall 77.7%		6	0	0	0	418	0	14	0	18	27
\emptyset f-measure 77.8%		10	1	0	0	0	151	0	0	6	0
		18	0	0	33	8	0	227	0	0	24
		0	0	38	19	0	0	1	198	0	19
		19	63	0	0	18	23	0	0	239	0
		2	0	0	11	12	0	31	29	0	321

the very end. One could now either accept this as a matter of fact or throw away all attributes that miss more values than a certain threshold. Experiments have shown that the benefit of such sparse attributes is negligible, leading us to select the latter approach with a level of 50%. In our 10 day scenario, this is enough for the first six decile attributes. The classification accuracy can be improved to 78.06% with an average f-measure of 78.1%.

▷ **Hypothesis:** These interval attributes are highly correlated with the sales figures attributes and can be substituted by them.

To verify this hypothesis, we are going to express them in both directions by using a state-of-the-art algorithm, the support vector regression, for we want to know whether this is possible at all. When taking the decile attributes as target values, we attain an average prediction correlation of 74.81% whereas in the reverse case it amounts to 62.76%. Consequently, their substitutability is rather limited, as we are also going to see when applying attribute selection techniques. Thus, we will retain both for analysis. ⇒ **NO** □

Up to now we have only incorporated aggregate values of the lead time as a whole. But it might also make sense to use information from the cutoff day itself, i.e. as close as possible to the period unknown. Therefore, attributes are selected that are available in any case, that is the weekday, month, Google Trend, average temperature and precipitation. The result of this configuration is summarized in table 4.4.

Table 4.4: Results of a comprehensive cluster classification model. It consists of several different attribute groups.

Naive Bayes		157	2	0	0	6	2	23	0	10	0
Attributes SF+EQ+TQ+x1+DC+CD (56)		12	206	0	0	0	7	0	0	10	0
Accuracy 77.86%		0	0	207	23	0	0	0	52	0	0
\emptyset precision 78.6%		3	0	0	224	0	0	26	26	0	22
\emptyset recall 77.9%		5	0	0	0	416	0	19	0	17	26
\emptyset f-measure 77.9%		13	1	0	0	0	149	0	0	5	0
		23	0	0	31	11	0	227	0	0	18
		0	0	34	19	0	0	1	200	0	21
		18	74	0	0	21	10	0	0	239	0
		0	0	0	13	13	0	26	26	0	328

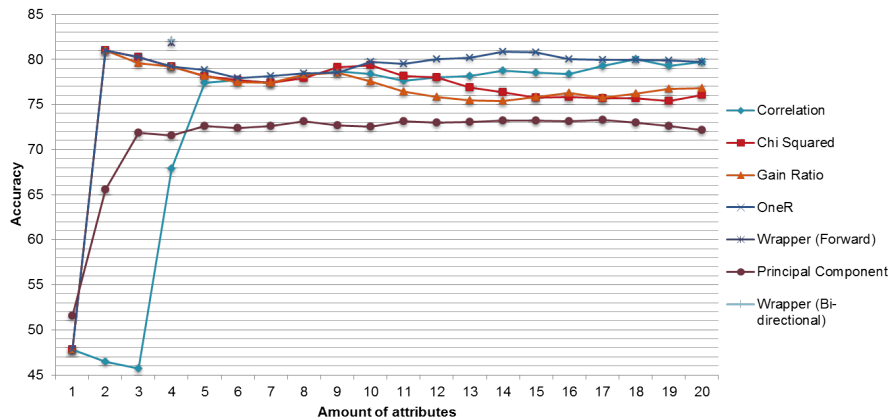


Figure 4.7: The accuracy of a naive Bayes classifier with attribute subsets selected by various selectors.

In the end we may not only consider past but also future-related information. The holidays are known, for instance, no matter how far they are in the future. Counting the holidays that are between the cutoff day and the performance day may reveal important information, of course also depending on the kind of holidays. A nonlinear algorithm could use date-related information, i.e. whether they are summer, or Easter holidays, etc. The same can be done for all past days when tickets have been sold. Another future-related value is the distance of the event date to the premiere date, which allows for a slight incorporation of production-life-cycle characteristics. By extending the model with those three attributes (NX1+x2), an accuracy of 78.23% and an average f-measure of 78.3% can be achieved.

FINE-TUNING THE MODEL

We now have a model that consists of many and diverse attributes, able to predict the correct cluster in about four out of five cases (let alone that a misclassification could concern a cluster that is rather similar). However, as we have seen, classification performance did not improve at a certain point. What is more, its performance even decreased in one case. We have suffered from the curse of dimensionality, as discussed in chapter 2.1, despite the fact that attribute sets were filtered ex ante. So we are going to apply an automated feature selection. In figure 4.7 we compare different filter-methods, or more precisely filters with different relevancy criteria, the principal component analysis and two wrapper-approaches, one forward and one bi-directional.

What we see here is remarkable – a subset consisting of just two attributes identified in three cases, that is the average amount of days all tickets were bought before the show, somehow acting as a substitute for the sales figures parameters, and the total occupancy rate so far providing a good indication for the previous sales performance as well, already gained a result of almost 81% accuracy! For the algorithm, these two fields provide more information than the full set of 59 that was created painstakingly. The even better result – as expected – was reached by the two wrapper approaches, i.e. 81.8% with the forward and 82.14% with the bi-directional approach,

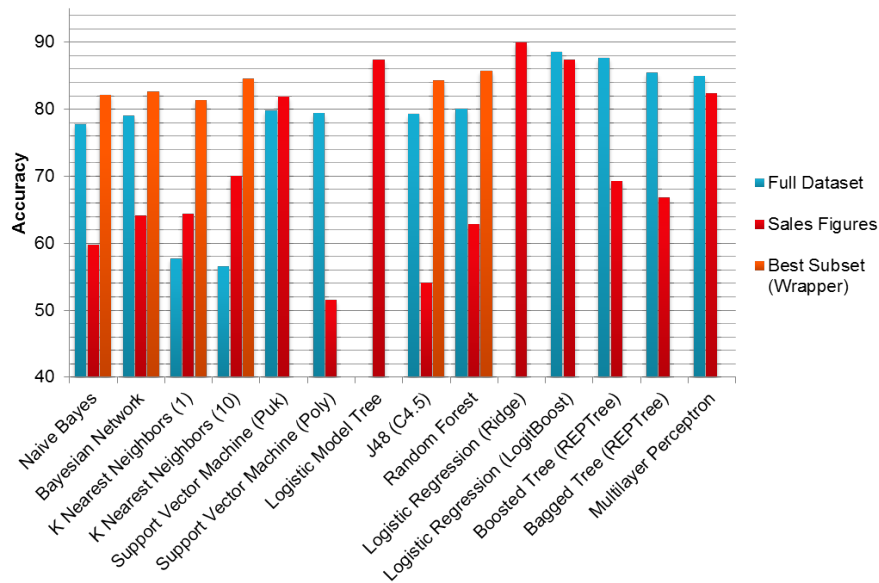


Figure 4.8: The accuracy of several classifiers on different attribute sets.

both selecting four attributes. They are, beyond the two that were already mentioned, the day of the year the event is to take place and the decile number six for the first one and the interval attributes 125–100 and 20–15 for the second. We used a greedy-hillclimbing approach with backtracking facility. When using the backward approach 29 attributes were selected, leading to a classification accuracy of even 82.67%. As more combinations must be tested, runtime increases significantly. Due to that and in the light of the limited benefit that can be reached, we will not use this method. Instead, we are going to use the bi-directional approach as a compromise in further consequence.

All filter selectors were evaluating each attribute separately and selected the topmost ones. In many cases, starting with about five attributes, adding further ones lead to poorer results, again indicating the curse of dimensionality problem. This is aggravated by the fact that naive Bayes classifiers despise multicollinearity, which is certainly introduced in some cases. As already mentioned in chapter 2.1, the principal component decomposition performs worst.

Before summarizing, we want to assess the performance of different algorithms with different attribute sets, as we have only seen naive Bayes in action so far. We did not optimize any parameters, as this is not our intention, but still made use of means incorporated by some algorithms to normalize and improve data, for instance at the support vector machine. The result can be seen in figure 4.8. The optimal subset could not be determined for some algorithms as the runtime simply was too high – for even if it takes just one minute, it needs to be run through hundreds or even thousands of times. The logistic model tree and logistic regression using the ridge attribute-selector were even unable to process the full data set in a reasonable amount of time.

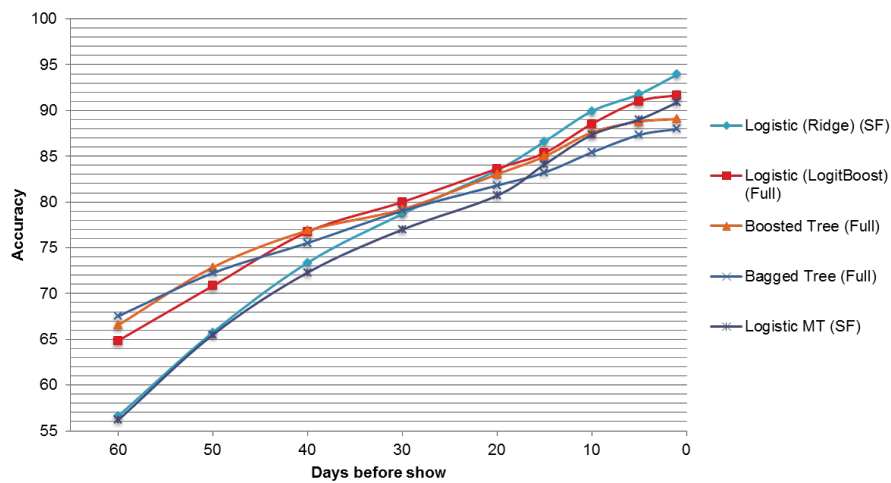


Figure 4.9: The accuracy of well-performing classifier configurations at different cutoff days.

The curse of dimensionality is clearly recognizable at all algorithms where an optimal subset could be identified, indicated by its superior performance. This is especially the case for the two k-nearest-neighbor variants, as this algorithm is particularly prone to that phenomenon. In all optimal subsets, the average amount of days all tickets were bought before the show and the total capacity utilization rate are present again. Beyond those two, sales figures or decile attributes were selected mostly, providing a more detailed picture of the past trend. However, the total amount is very low with less or equal than eight selected attributes in five cases out of six to avoid multicollinearity issues. Logistic regression and ensemble learners based on trees proved the most effective algorithms, outperforming simple and complex ones such as the support vector machine, even in their optimal feature configuration.

Just as we compared different algorithms, we now want to compare different horizons. So instead of taking 10 as cutoff day, we are going to take all values that are represented by a sales figures attribute up to 60. Therefore, the algorithms that proved most effective are used. Figure 4.9 illustrates this comparison.

The logistic regression with a ridge-parameter-selection performs by far best in the beginning, but drops behind other algorithms with an increasing horizon. The same goes for the logistic model tree. The reason for that is that both configurations make use of the sales figures attribute set only. As a consequence, the increased need for information that results from greater horizons cannot be met here. The boosted tree is preferable to the bagged one, as expected – except for the last interval. All in all, it can be concluded that choosing the optimal algorithm itself is a nontrivial task as it depends on many different factors. Beyond that, it is remarkable that two months before the performance, the classification is correct in two out of three cases already.

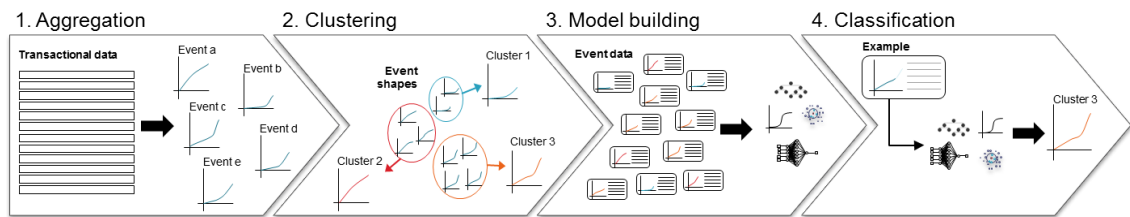


Figure 4.10: Summary of the high-level steps needed to answer DQ1.

SUMMARY

Figure 4.10 summarizes the high-level steps that were conducted to answer DQ1. We started by aggregating our transactional base sales data according to their events and time distance to the performance date. The resulting sales trend time series were fed to several clustering algorithms that identified an arbitrary amount of distinct patterns. We have seen that the performance pretty much depends on the algorithm that is used. When trying to forecast the cluster of a show that has not been performed yet it is important to keep in mind that the data base that is used for training may only contain information that is available at the cutoff distance at each event.

So we built a classification model using various attributes – past sales figures, qualitative attributes of the event, some characteristics of the tickets already sold, information about the cutoff day, the amount of upcoming holidays, and some others. With that rather wide attribute space we experienced the curse of dimensionality problem as the result could be considerably improved by rigorously filtering attributes. In order to allow an algorithm to detect more complex relationships it would need several orders of magnitude more than those 3,000 instances that are available, depending on the horizon of the forecast. Nevertheless, the models that were built suffice to allow for a rather confident prediction.

4.2 Domain Question 2 – Event Capacity Utilization Forecast

BASIC ANALYSIS

Instead of predicting the upcoming sales figures shape as a whole, the final capacity utilization rate can be used as a goal for a forecast model as well. Consequently, instead of having a target class, the target value is now numeric, resulting in a regression problem. All other input parameters can be retained from the preceding question. We will not take the final capacity utilization rate as the target however, but only the increase in percentage terms as the final rate can easily be ascertained by summarizing it with the rate to date. Interestingly, the performance even worsened significantly when using the final rate.

Hence, this approach suddenly represents kind of a time series forecast problem as a future series value is tried to be expressed now. The autoregressive part is depicted by the sales figures attributes and partially also by the decile intervals. All other inputs represent exogenous factors. For the consequent analyses, we again chose 10 days as cutoff day. Further, the RMSE is chosen as evaluation criterion as big deviations are disprized. In figure 4.11 we show the performance

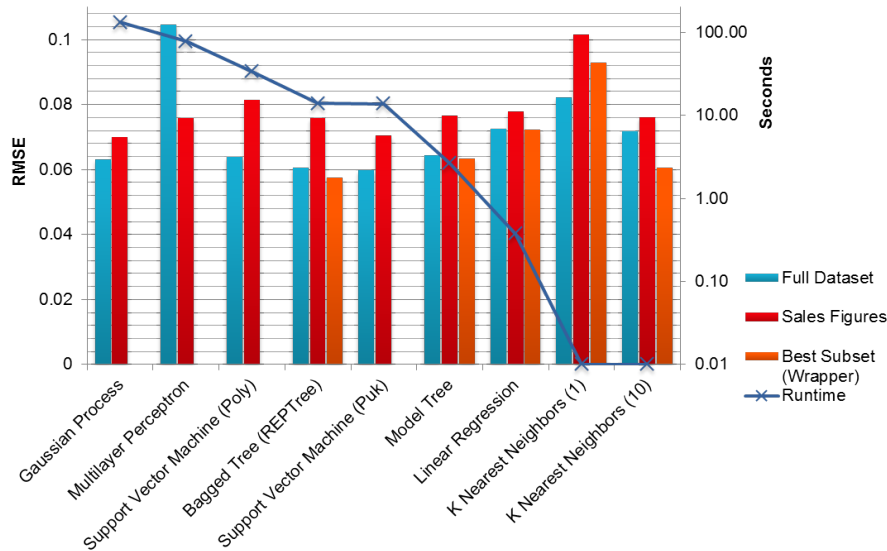


Figure 4.11: Accuracy of several regression algorithms to forecast the occupancy rate. The runtime follows the logarithmic scale on the right hand side.

results of several algorithms using different attribute sets, ordered by their single-threaded runtime measured with a 3 GHz AMD CPU. Unlike before, smaller values are better.

All sales figures attribute set configurations are now worse than the respective full attribute set, which is due to the fact that the target value now shares less information with these values. Instead, structural and qualitative factors play a bigger role. Now, the support vector machine and again the tree based ensemble learners proved most effective. The best result was an RMSE of 0.0575 by the bagged tree using the optimal attribute subset. Neural networks were found to be instable as the worst result was obtained when providing the full attribute set. It could be improved however, by reducing the amount of attributes. The set that was identified optimal for the linear regression was considerably large in the light of the multicollinearity-sensitivity of this algorithm as it consists of 28 attributes. Apart from that, it is worth pointing out that the k-nearest-neighbor algorithm with k set to one was the only case where the subset was performing worse than the full set. When taking a closer look, the reason becomes obvious, for there is only one attribute selected after 354 combinations – the first decile attribute.

To provide a better insight about these transformed variables, we refer to three correlograms depicted in appendix B.1.

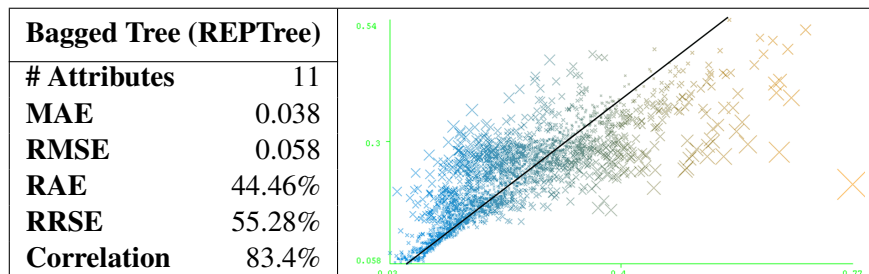
We also want to mention here that using the absolute sales figures as target values, i.e. abandoning the normalization step, the performance gets significantly worse with an RMSE of 0.0624 if the numbers are normalized afterwards again.

▷ **Hypothesis:** When looking at the attributes selected by the wrapper approach, there are differences to DQ1.

Again, except for the k-nearest-neighbor with k set to one, all algorithms make use of the average amount of days all tickets were bought before the show and the total occupancy rate so far as they simply prove most valuable. But additionally, in order to overcome insufficiency, sales figures and decile attributes are incorporated in slightly increased quantities. Also, as already indicated in the last diagram, qualitative attributes play a more important role, as for instance the holiday information about the event, the average discount and the average degrees for the dates the tickets were bought are included in some cases now. Moreover, also the Google Trend of the cutoff day is used. Another important attribute that was selected in all but one cases is the average days after the premiere the tickets were bought. The model tree is the only one which also uses the amount of prospective holidays. So in the end, there is a difference as the target value is more unacquainted than in the previous question. ⇒ **YES** □

In table 4.5 we take a look at the optimal model configuration identified before. The root relative squared error of 55% indicates that the configuration provides a real surplus compared to a simple predictor and also the high correlation signalizes a good result. On the right hand side, real values (x-axis) are compared with predicted values (y-axis). The size and color of the crosses provide information about the deviation. One cross immediately stands out, namely the one with a real value of 77% and a prediction of 21.4%. Generally, there are some rather big deltas that are underestimated by the model and probably due to unforeseen marketing actions, as 77% of the tickets sold as from 10 days before the show is rather unlikely. In turn, low sales are underestimated, indicated by the large proportion that lies above the optimal line in the left half.

Table 4.5: Results of the optimal regression to predict the final capacity utilization rate.



▷ **Hypothesis:** The coherence within a production is higher, improving the accuracy when exclusively using events of the respective production for model building.

We again ignore the temporal aspect in answering this question. We evaluate the performance of the bagged tree with the optimal subset for each production separately. In the end, the weighted mean RMSE is calculated. The results can be seen in figure 4.12.

Obviously, events of bigger productions can be forecasted better than those of smaller ones. This is due to the presence of a higher homogeneity within them and the fact that the diversity and amount of examples is sufficient. The total sum of events is smaller because only productions were evaluated here whose amount of events was sufficient to create a reasonable model.

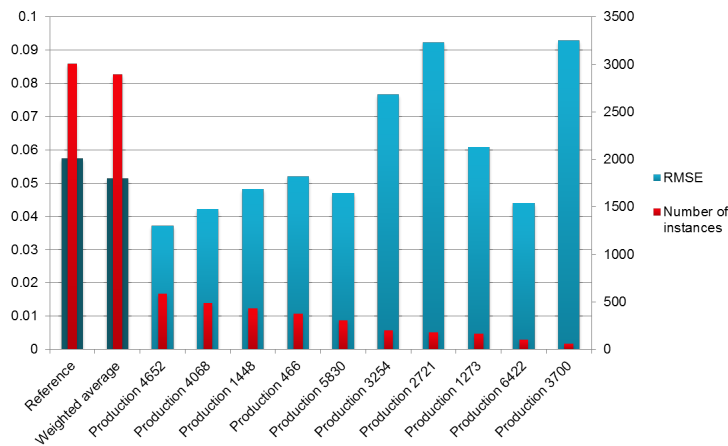


Figure 4.12: Forecasting the occupancy rate when using events of the respective production only. The weighted RMSE is compared to the RMSE of the holistic model on the very left.

As a consequence, some uncertainty remains concerning the others. The subsets selected for each production differ quite considerably and are smaller than before, which emphasizes their instability. In the end, a clear answer cannot be provided as it very much depends on the production, but at least in this ten day horizon-scenario the weighted average RMSE significantly surpasses the holistic one by 0.006. ⇒ **YES/NO** □

We now want to assess how confidence changes when varying the horizon again. In figure 4.13 we used the support vector regression with PUK-kernel as this algorithm proved effective without time-consuming attribute selection. Two months before the show, the average deviation is almost 7%. It sags down in a more or less logarithmic way to eventually reach 2.33% at the day before the show.

REGRESSION VS. CLASSIFICATION

As we have seen in chapter 2.1, each regression problem can be converted to a classification problem rather easily by binning, i.e. discretizing the target variable. We shall prefer equal frequency binning over equal size binning as most events have a final capacity utilization rate of more than 50% and its distribution is rather uneven as can be seen in figure 4.14. As a consequence, also the delta target variable is unevenly distributed. The reason for the mode to be 99 instead of a hundred percent is that in many cases a few free tickets are given out.

▷ **Hypothesis:** The obvious assumption is that a classification via this binned result works better than a regression.

We use the mean value of the interval predicted as target to be able to calculate a difference and consequently compare the result with the bagged tree used previously. These values are presented on the left hand side of figure 4.15. It becomes obvious that the information loss that

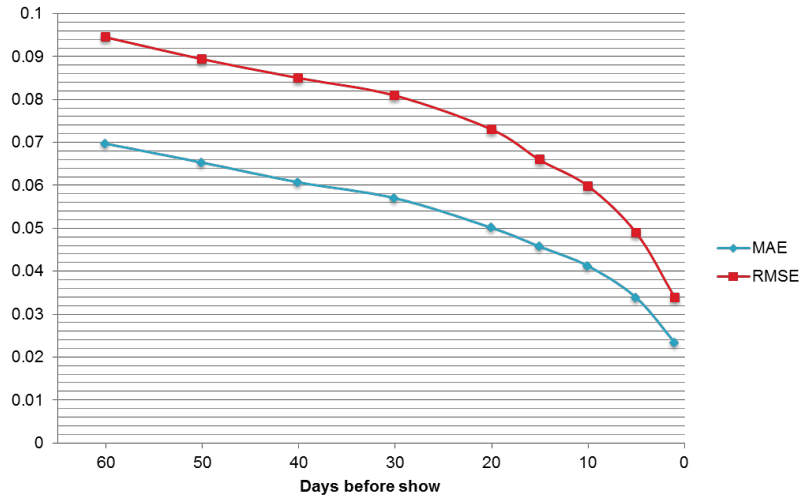


Figure 4.13: The accuracy of a support vector regression at different cutoff days. We added the mean average error to allow for an easier comparison with real world examples.

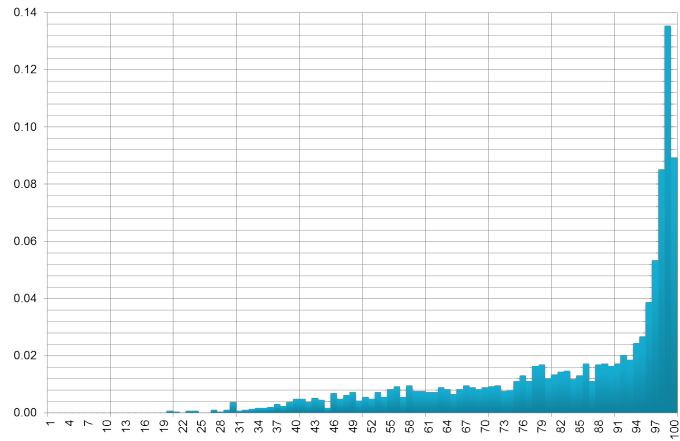


Figure 4.14: Histogram of the final event capacity utilization rates.

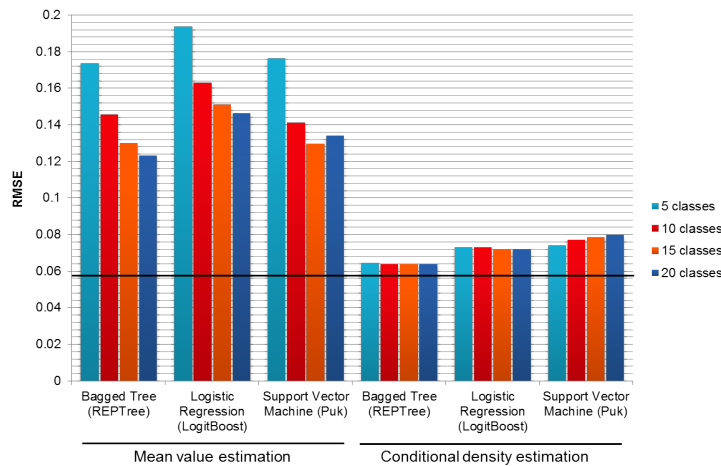


Figure 4.15: Comparison of the best regression result of the capacity utilization rate with several classifiers using the mean value prediction and a conditional density estimation. The black line indicates the regression result.

is effected by the discretization is irreversible and even increasing the amount of classes cannot help to overcome this considerably.

But also more sophisticated classification-via-regression approaches have been developed, such as the conditional density estimation. It uses a classifier that is capable of predicting probabilities (such as naive Bayes) to estimate the distribution of the target value and eventually determine the value that is most likely. Hence, the result is much more accurate than just taking the mean value as before. The fact that a confidence interval can be expressed additionally is left out deliberately (see Frank and Bouckaert [2009] for more information thereon). When looking at the right hand side of the graph, the superiority of this concept can be seen, although it is still overtopped by a regression. Interestingly, increasing the amount of classes has no clear effect on the performance as the probabilities suffice to obtain a finely-grained result already. ⇒ **NO** □

▷ **Hypothesis:** These classification approaches proved ineffective, but taking the final value of the predicted sales trend cluster offers a way out.

Again, the results with different quantities of classes are compared, which in this case is the amount of clusters. The algorithm used is the same as in DQ1. The results are depicted in figure 4.16 on the right hand side. The previous mean value predictions are added to allow for a comparison on the left hand side.

This approach cannot beat the regression performance either, but represents a compromise between the previous approaches. The fact that the performance surpasses mean value prediction is based on a binning that is more appropriate. The algorithms perform almost equally when using the same amount of clusters in this scenario. In the end it is to be noted that the result does not improve when using more than 15 clusters. ⇒ **NO** □

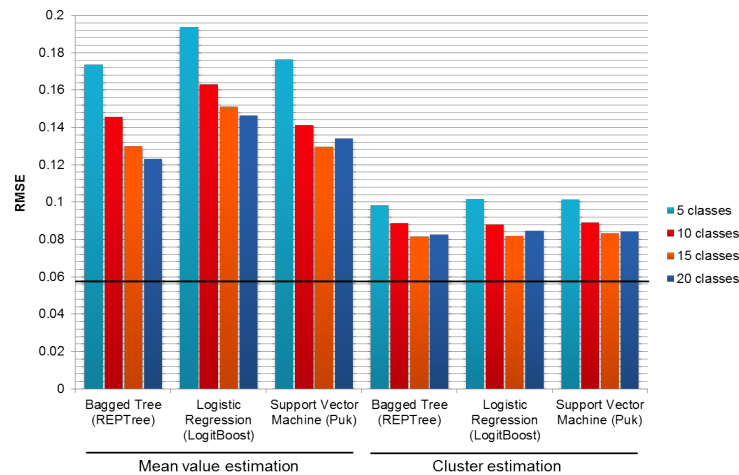


Figure 4.16: Comparison of the best regression result of the capacity utilization rate with several classifiers using the last value of the sales trend cluster.

SALES TREND PREDICTION

Up to now we used the sales between the cutoff day and the performance day as target. But we can also define any other time interval and as a consequence are able to predict any prospective day, resulting in a more fine-granular sales trend prediction than in DQ1. By comparing predictions and actual values of other events, it is even possible to express the confidence of the target value of a certain horizon as the range of variation.

In plotting several individual forecasts simultaneously, as needed when predicting each future day, we experience an interesting phenomenon indicated in chapter 2.2. As each prediction is independent, there might be the situation that sales are negative between two days. In the end, this reflects the uncertainty of the forecast in the respective area.

We take event 5182 of the production 466 as an example to forecast the trend. For this, and also all further examples, we use bagging without attribute selection as this algorithm is parallelizable. Figure 4.17 and 4.18 assume 10 and 30 as the cutoff day, respectively. In addition to that, the predicted cluster of question one is displayed.

In both cases, the trend can be approximated quite well. Also the cluster is estimated correctly. In the second graph we can already experience negative sales especially one week before the show as fluctuations are quite high then usually. As these graphs need a lot of space, we moved some further examples to the appendix B.2.

A Reduced Approach

The further we are away from the day of performance, the less tickets are already sold in general. We already mentioned that in order to preserve representativeness of the ticket related derivatives, each event is required to have at least 20 tickets to become a training instance. As

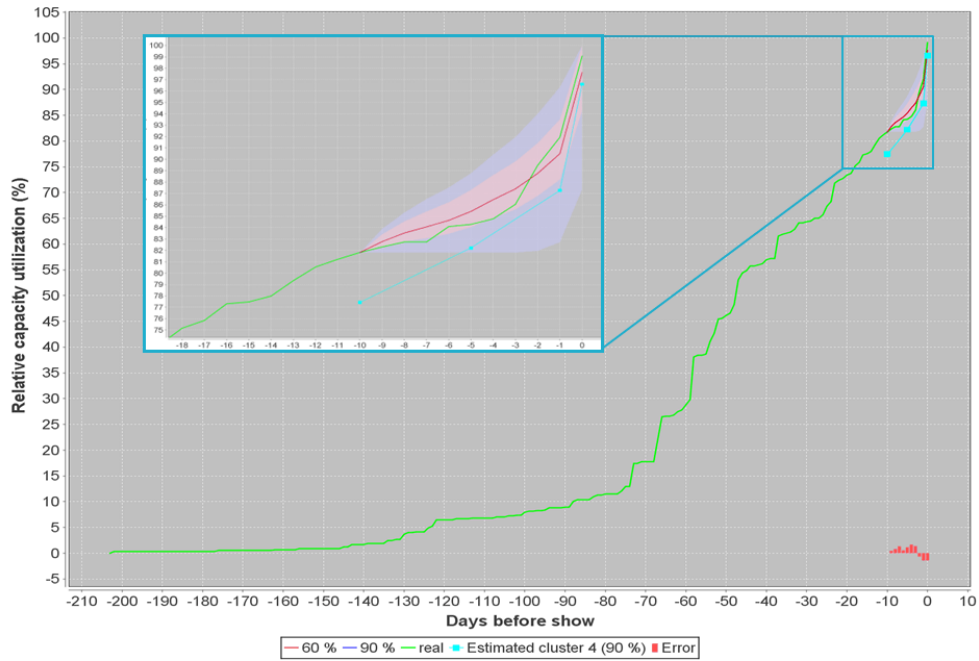


Figure 4.17: Forecast of capacity utilization and trend cluster of event 5182, 10 days before show.

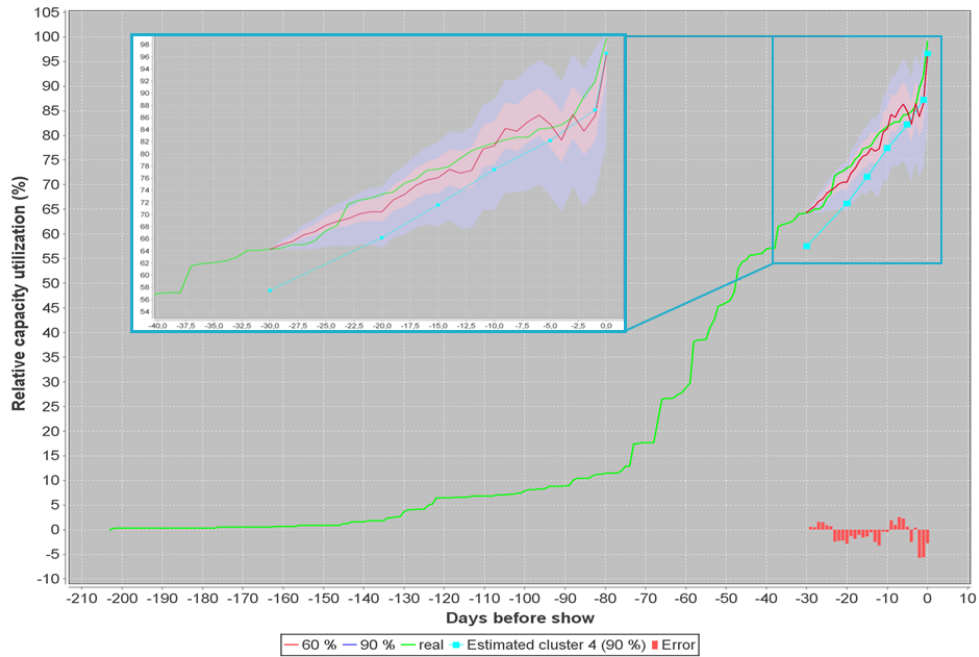


Figure 4.18: Forecast of capacity utilization and trend cluster of event 5182, 30 days before show.

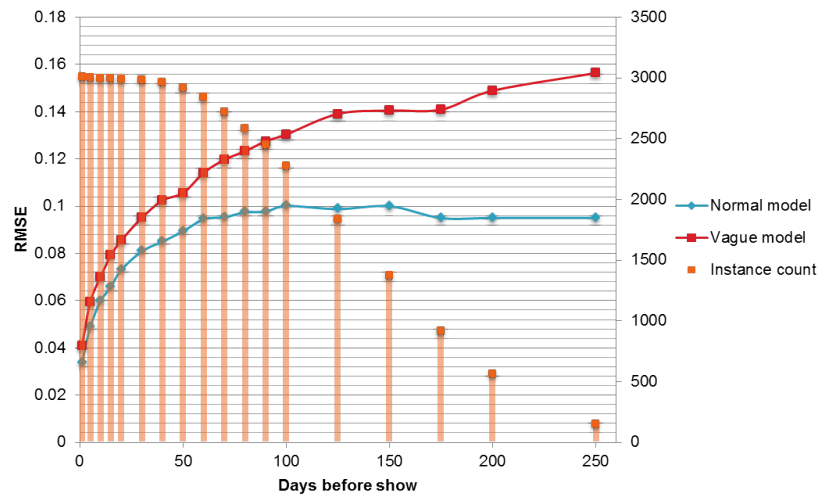


Figure 4.19: A comparison of the normal and vague forecasting approaches in short and long term problem settings. The orange bars indicate the amount of instances the normal model has for the respective horizon.

a consequence, the amount of instances becomes smaller and smaller. In turn, to preserve representativeness of the models by avoiding them to get too few examples, we create a reduced approach and call it “vague”. It uses attributes that are assessable in any situation, that is the date related information of the event (EQ), the cutoff day information (CD), the amount of tickets sold so far (x_1) to have a simple equivalent of the past sales and the target day related attributes (NX1). Eventually, each model obtains all events as the learning base. Hence, we trade variance for bias. This approach is intended for long-term forecasts only, as its accuracy is of course lower than the one of the normal model.

In figure 4.19 both approaches are compared for horizons until 250 days before the show. It is visible that up to 50 days the amount of instances does not decrease significantly, but it does as from 100 days. Representativeness drops so rapidly, that the error even decreases, for the only events that remain having tickets sold so early already are likely to reach a high final occupancy rate. By taking a look at the clusters identified before, we can see that just 0, 1, 5 and 8 are active that far ahead.

The question now is when to draw the line and fall back to this approach. We selected 150 days as a good compromise between representativeness and the exploitation of the implicit success indication. Of course, this reduced approach may also be applied to forecast the cluster of DQ1.

EVALUATION

After elaborating this approach, its effectiveness should be tested in comparison with the method that is currently used to predict the final sales. In this context we shall call it “baseline”. It enables to calculate the final amount of seats sold at any point in time during pre-sale. As

mentioned initially, it is an inflexible formula only, that is a regression with three parameters. Those parameters are all tickets that have not been sold or reserved so far, those that are reserved and all contingent reservations. The coefficients x_1 to x_3 are variable, but estimated manually and based on intuition, each time a forecast shall be made, usually once a month per production. This intuition takes into account the performance of the production itself and its life cycle phase. It reaches from 50-90-90 in optimistic cases to 25-50-50 in pessimistic ones.

$$\begin{aligned} \text{estimated total tickets sold} = & \text{tickets sold so far} + \text{unsold and unreserved tickets} \cdot x_1\% + \\ & \text{reservations} \cdot x_2\% + \text{contingent reservations} \cdot x_3\% \end{aligned} \quad (4.1)$$

As one would expect when considering the data structure, we cannot use this formula with our data as we are unable to determine the reservations that have been made. Instead, we just have those which were converted into real tickets eventually. However, contingent reservations could be recovered by looking at the respective attribute which indicates the maximum amount of tickets that can be sold. For this reason, we have to rely on some individual forecasts only, which were obtained ex post.

These cases come from one production that is not contained in our data base but used exclusively for this evaluation purpose. Of course, all preparation steps described need to be applied as well. Altogether, we have 445 forecasts made for 246 events, reaching up to 100 days before the respective performance. The events are scheduled over a ten month timespan, thus encompassing almost all seasonal phases. Due to the fact that our data differs from the figures obtained with regard to free tickets and the like, we may not compare absolute but relative numbers of seats sold. The plot in figure 4.20 depicts the deviations of the true final occupancy rate with the baseline approach and our data mining based approach in the course of time. For the latter, all 246 events that are concerned are forecasted at every point in time between 100 and one day prior to the event.

The most striking discovery is that rates are underestimated systematically by the baseline approach and this underestimation increases when the horizon becomes bigger. Beyond 50 days, only two out of 101 forecasts are higher than the actual value at all. This is due to the inflexibility of the linear formula. Instead, the data mining approach works pretty well for small horizons but eventually overestimates sales for this production. In a quick visual assessment, it looks like absolute deviations are bigger at the previous approach.

▷ **Hypothesis:** The novel data mining based approach outperforms the baseline forecasting approach.

In order to provide an answer we are going to statistically evaluate each interval of 10 days, as shown in table 4.6. For matching variances the normal two sample t-test is used, for non-matching ones the Welch's t-test. The significance level was set to 5%.

As the horizon increases, so does the average deviation of the data mining method. But because it increases less than the baseline forecast, the average improvement becomes better in absolute numbers. It raises from a little more than one to almost 11 percent between a horizon of 71 and 80 days. It is to be noted however, that as from day 61 the amount of observations

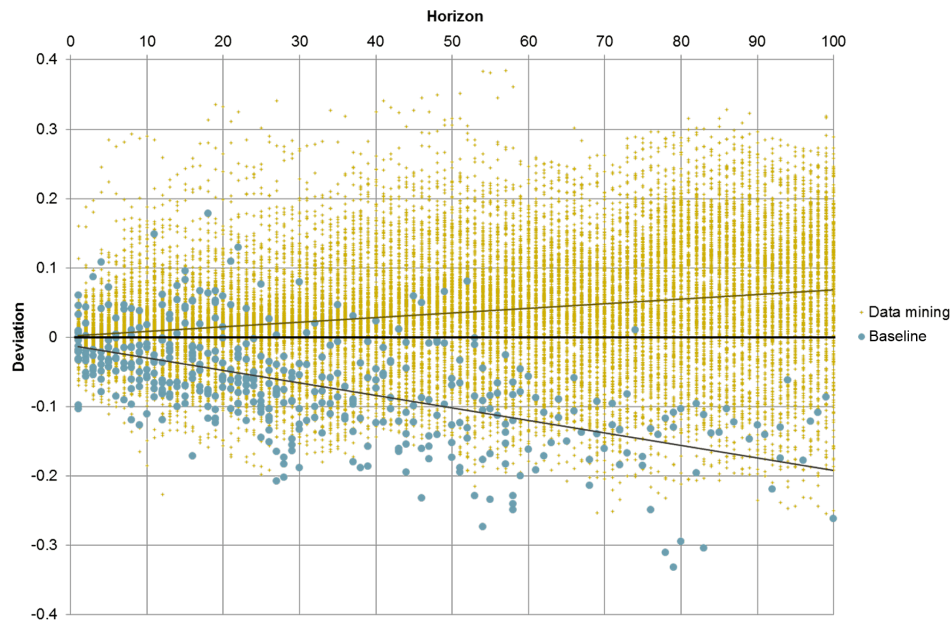


Figure 4.20: The deviations to the final occupancy of the previously used baseline and the data mining approach. The colored lines represent the linear regression lines of the respective method.

drops painfully, reducing its expressiveness. Nevertheless, the differences remain statistically significant. The hypothesis can thus be considered confirmed.

After this fundamental comparison, we compare the forecasts one on one. Therefore, the difference is calculated in each case using the formula: $\text{improvement} = |y - \hat{y}_1| - |y - \hat{y}_2|$. Here, \hat{y}_1 stands for the estimation of the baseline approach and \hat{y}_2 for the data mining one. The results are depicted in figure 4.21.

In the majority of 312 out of 445 cases our new approach surpasses the baseline approach. Again, due to the systematically increasing underestimation of the latter, the improvement increases, as becomes clear when looking at the regression line. On average, when ignoring the horizon, the forecast provided by our data mining approach is better by 2.096% of capacity utilization rate. To statistically verify the result a Wilcoxon signed-rank test was conducted to compare the mean values pairwise. The resulting p-value is highly significant amounting to $3.2196e-44$ ($z = -13.9484$).

In the end, we can conclude that the data mining approach exceeds the baseline approach and, by implication, also human intuition, which is reflected in the coefficients up to a certain degree, especially for long term horizons. \Rightarrow **YES** \square

Table 4.6: Statistical evaluation of the baseline and data mining approach.

Horizon	Meth.	n	Mean	T-test stat.	P-value T-test	Improvement	Variance	F-test stat.	P-value F-test
1-10	BL DM	94 2460	-0.0222 0.0102	-7.071	2e-12	1.204%	0.0019 0.0019	1	0.963
11-20	BL DM	88 2460	-0.0267 0.0125	-6.053	1.6e-9	1.416%	0.0042 0.0035	1.18	0.25
21-30	BL DM	81 2460	-0.0732 0.0145	-11.684	9.3e-31	5.874%	0.0042 0.0044	1.062	0.75
31-40	BL DM	43 2460	-0.0766 0.0215	-10.999	3.9e-28	5.51%	0.0033 0.0056	1.673	0.04
41-50	BL DM	43 2460	-0.0886 0.0394	-10.545	1.8e-25	4.92%	0.0053 0.0062	1.194	0.478
51-60	BL DM	40 2460	-0.1207 0.0372	-11.664	1.2e-30	8.345%	0.0061 0.0072	1.192	0.502
61-70	BL DM	16 2460	-0.1346 0.0343	-15.629	4.6e-55	10.03%	0.0018 0.0075	4.125	0.003
71-80	BL DM	20 2460	-0.1594 0.0506	-11.23	2.9e-29	10.87%	0.0069 0.0091	1.314	0.489
81-90	BL DM	10 2460	-0.1483 0.0674	-6.769	1.6e-11	8.082%	0.0038 0.0101	2.67	0.105
91-100	BL DM	10 2460	-0.148 0.0656	-6.798	1.3e-11	8.238%	0.0037 0.0099	2.658	0.106

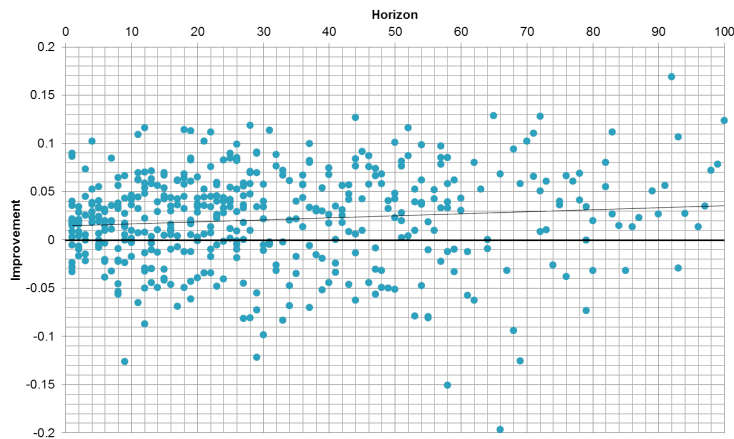


Figure 4.21: The direct improvement of the data mining forecast over the baseline forecast.

SUMMARY

Using the transformed variables of the previous problem, the final occupancy rate can be predicted by simply exchanging the target variable. As we are now confronted with a regression problem, other algorithms come into play. Both approaches are rather similar, nevertheless. By trying to predict this final rate using classification surrogates of different kind, also using the final value of the sales trend prediction, we could show that the regression delivers a superior performance.

By varying the target interval, it is also possible to predict individual days in the course of sales, resulting in a more finely grained presentation of the prospective sales trend than in DQ1. In the course of this, we showed that for long term forecasts, representativeness is reduced as training instances break away. To overcome this, we introduced a reduced approach which only uses input factors that are known in any case.

In the end, we could demonstrate the supremacy of our approach compared to the one which is currently used in the establishment.

4.3 Domain Question 3 – Influencing Factors

As we have seen in chapter 2.1 many algorithms and their resulting models are incomprehensible for humans due to their complexity and the unbounded possibilities to form inner structures and mechanics. In order to provide an answer by pointing out the most influential input factors, different solution approaches are presented subsequently.

ATTRIBUTE SELECTION

A simple way out could be to look at the features that have been selected by various feature selectors. Therefore, filters as well as wrappers are suitable. Concerning the former we could for instance look at the correlation of each attribute with the target variable. We mentioned this already when pointing to the chapter B.1 in the appendix. Depending on the metric used, the influence direction and strength can be assessed. The subset selected by a wrapper selector in turn depends on the algorithm used. Thus, it is advisable to look at several algorithms simultaneously and select the most common attributes. This has been done in table 4.7 for DQ1 and in table 4.8 for DQ2 with each being selected in at least one third or half of all cases, respectively.

Table 4.7: The most influential attributes for DQ1 identified by a wrapping based attribute selection for several algorithms. They are listed in the order of their frequency from left to right.

occupancy rate until cutoff day	average days before the show	sales figures 150–175
sales figures 100–125	sales figures 20–30	decile 1
decile 6	sales figures 150–175	decile 5

Unlike any attribute set selected by a filter approach, which assesses each attribute individually, nonlinear relationships can be considered here as well, thus yielding a result that is more accurate. On the other hand, the direction of influence cannot be provided, as it inherently

Table 4.8: The most influential attributes for DQ2 identified by a wrapping based attribute selection for several algorithms. They are listed in the order of their frequency from left to right.

occupancy rate until cutoff day	average days before the show	average days after the premiere
month of event	average relative discount	sales figures 10–15
is the event date a holiday	month of cutoff day	Google Trend of cutoff day

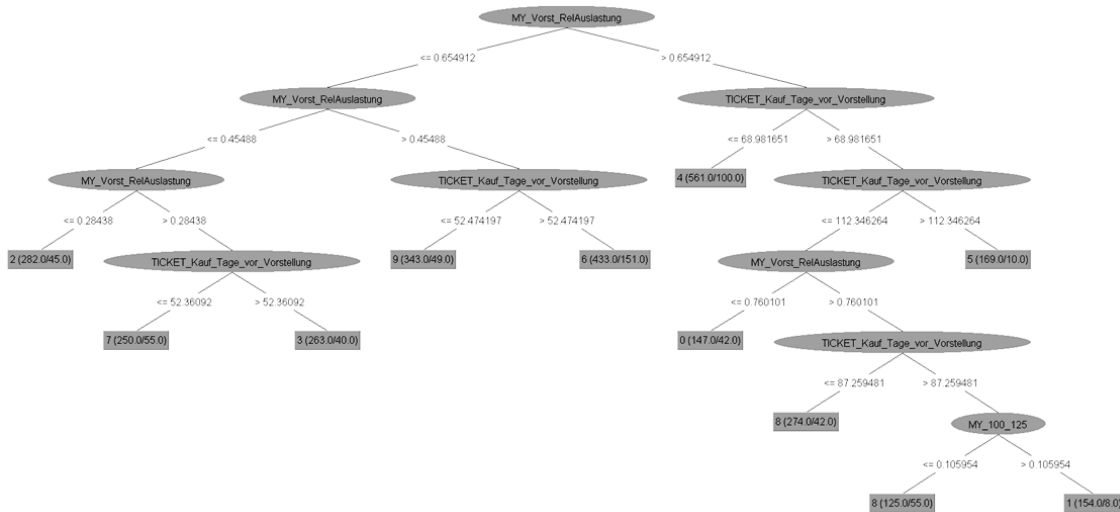


Figure 4.22: A heavily pruned C4.5 decision tree for a DQ1 classification problem. The leaves declare the resulting cluster, whereby the first number stands for the events concerned and the second one for how many have been misclassified.

changes in nonlinear models depending on the values of other attributes. Anyway, one also needs to consider the instability of optimal subsets.

STRUCTURES OF SIMPLE ALGORITHMS

Another possibility to obtain an overview about high level coherences is to make use of an algorithm that is simple and comprehensible enough to allow looking at its insides. Therefore, for instance a simple tree with a very low number of nodes and leaves can be used. The most influencing attributes can be identified on the basis of their position and even their directions can be assessed despite the nonlinearity of the approach. We created a classification tree in figure 4.22 for DQ1 and a regression tree in figure 4.23 for DQ2, each time using the full attribute set.

On the top of the first tree, the occupancy rate so far represents the most discriminatory attribute. When going to the left, i.e. falling below the level of 65.5%, a further distinction is made by the means of the same attribute as well as the average amount of days before the show. These two attributes also play a role on the right hand side of the top node, accompanied by the

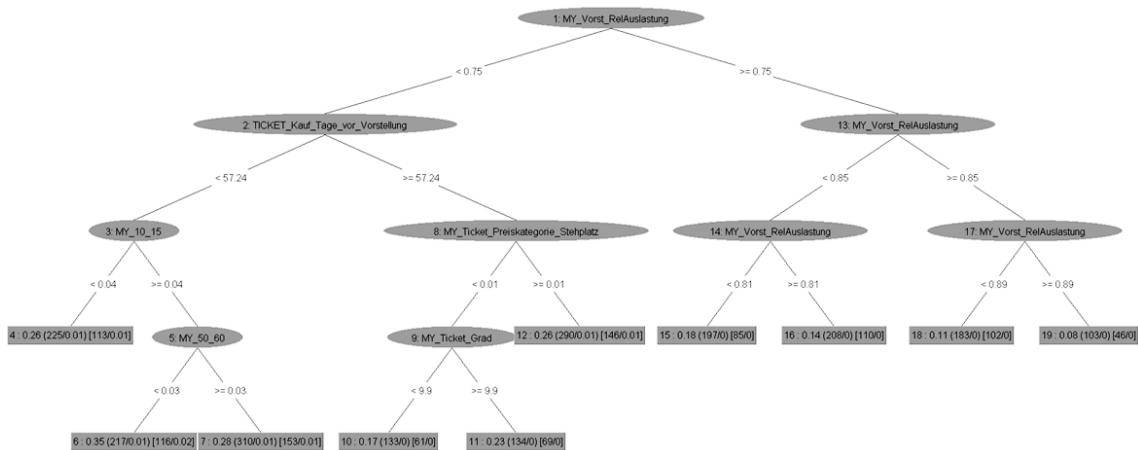


Figure 4.23: A heavily pruned regression tree for a DQ2 regression problem. The leaf nodes stand for the average value of the estimation, the first number for the amount of events yet again and the second one for the average deviation.

sales figures attribute 100–125. The accuracy of this very simple tree is remarkable – 78.97%. The regression tree looks somewhat similar as our two top candidates play an important role as well. However, in this case four other attributes are involved which corresponds to our previous findings.

Another simple algorithm is the linear regression model. Here, the direction and strength of an attributes influence can be assessed by the corresponding coefficients very easily, but again only linear relationships can be covered. Figure 4.24 shows the most significant ones. The top four attributes are further assessed in the appendix B.3 as scatterplots against the target variable.

One might go further and use other algorithms, for instance a logistic regression or naive Bayes classifier, as they are still simple enough to yield useful insights. We will refrain from doing so due to space limitations.

SUMMARY

Getting to know the most important influence factors is nontrivial in general. One could use the attribute sets selected by feature selectors, i.e. filter or wrapper approaches. The former may lose nonlinear relationships while the latter are instable, based on the algorithm used, and blind to influence directions and strengths. A different way is to use simple algorithms and look into their inner structures, allowing to deduce knowledge of different levels of sophistication, depending on the algorithm and complexity level.

4.4 Domain Question 4 – Breaking down into Price Categories

Instead of predicting the sales trend cluster or target utilization rate for events as a whole, we now want to focus on specific price categories as this information is important for targeted marketing

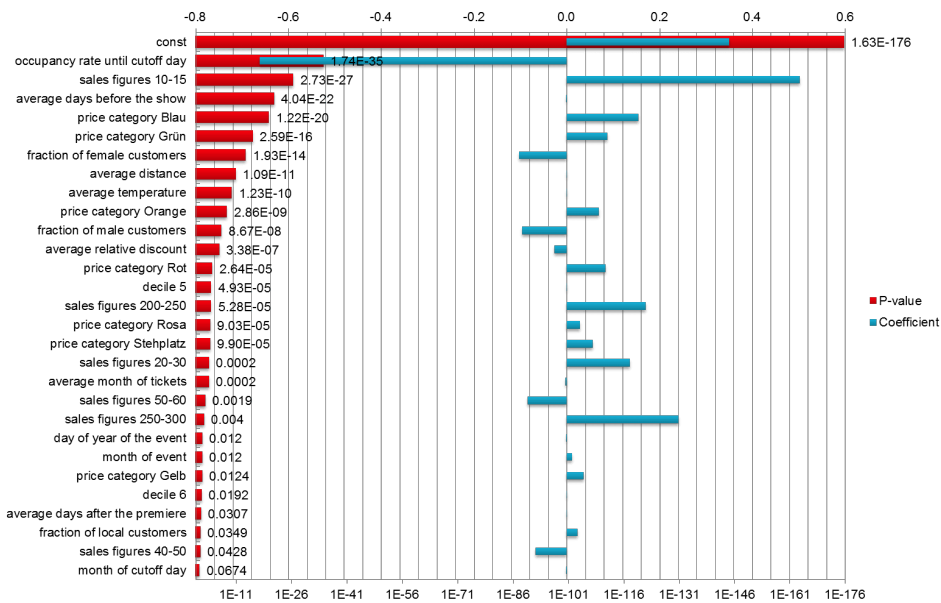


Figure 4.24: The regression coefficients of DQ2 with their corresponding p-values.

campaigns. We have seen in the description of the attribute TICKET_Preiskategorie that there are many distinct values (105) as they include not only main categories but also their price level derivatives. But in order to look at those categories and make sure that each of them is the same for all events, they need to be aggregated. Since information about the price is contained as a separate attribute anyway, it is advisable to perform this step in the very beginning already as deriving frequency attributes of the previous two questions is made easier thereby as well. In this process, we need to be aware that capacities of the respective categories need to be accumulated likewise. We need to consider that due to administrative reasons, a price category may not be unique for each event.

The resulting main categories are the colors Blau, Gelb, Grün, Rot, Rosa, Orange und Stehplatz (standing). Figure 4.25 depicts the frequency distribution in absolute numbers. Despite those categories there are some smaller ones occurring in some events or small productions only. They will be ignored henceforth.

The only thing that remains now is to select the price category to be forecasted by applying a filter on the data base which considers the transformed price category attribute. With that, all previous analyses can now be repeated on the basis of the respective tickets only. As we lose tickets of other categories in doing so, some attributes that were used previously cannot be formed, that is their respective occupancy rate so far. To overcome this, they need to be counted separately to regain them. We are further going to change the normalization basis from event capacity to the respective category capacity.

We may now for instance perform a clustering of all tickets of price category Rot. The resulting picture, as shown in figure 4.26 looks astonishingly similar to all tickets which is

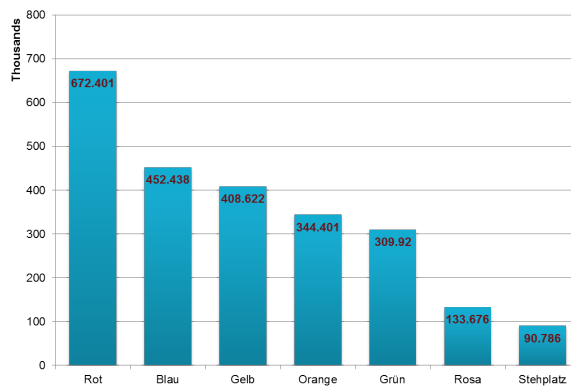


Figure 4.25: The absolute amount of tickets of each main price category.

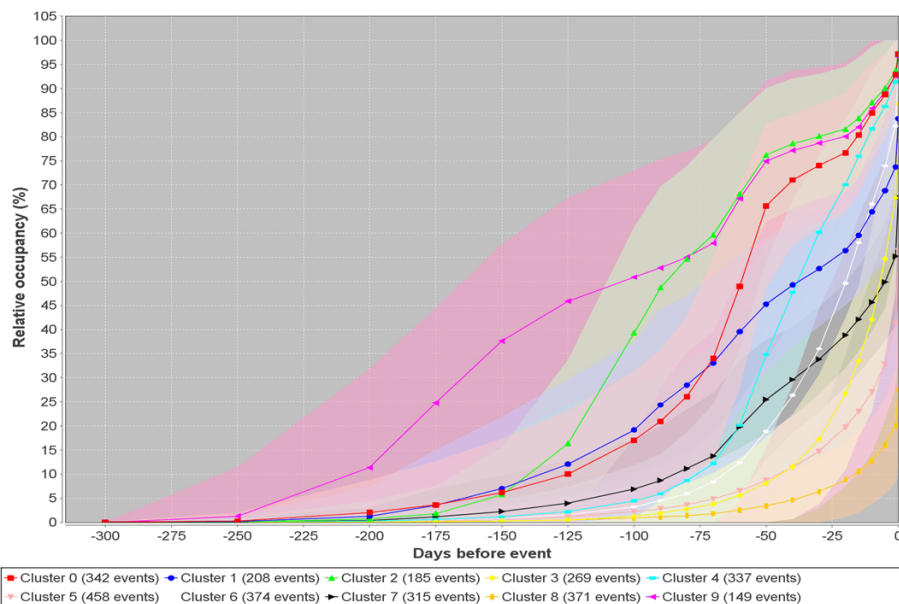


Figure 4.26: Clustering of all tickets of the price category Rot.

probably due to the size of the category acting as a reasonable representative. But it does not when looking at standing tickets, as they are sold on the day of performance only in most cases (figure 4.27).

In appendix B.4 we attached a collage of the event used previously to demonstrate future trend prediction, but now for price category Blau.

▷ **Hypothesis:** All price categories are predictable equally well.

For this purpose, we compare the RMSE of the target rate prediction and the error rate of the cluster prediction (and not the accuracy as before to avoid confusing the reader's eye) for each

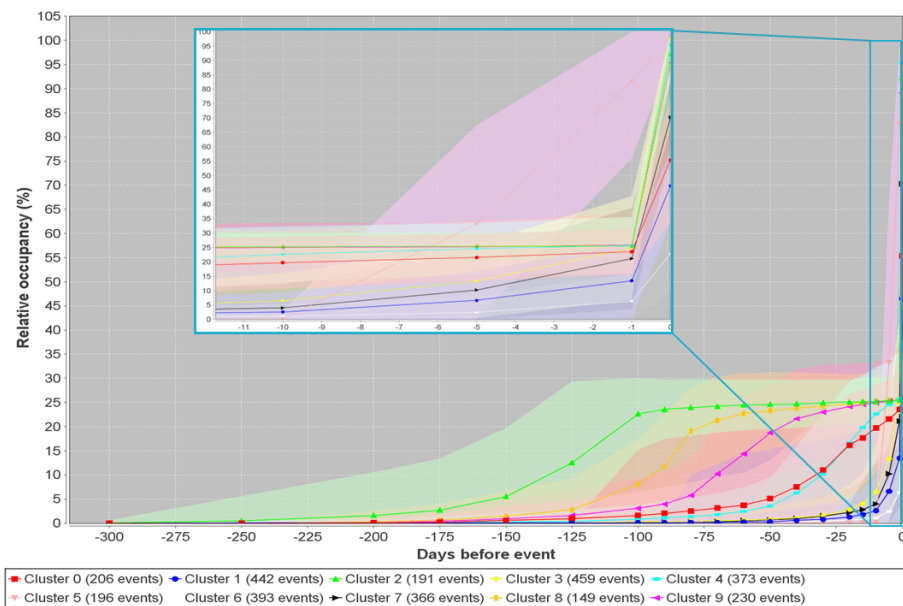


Figure 4.27: Clustering of all tickets of the price category Stehplatz.

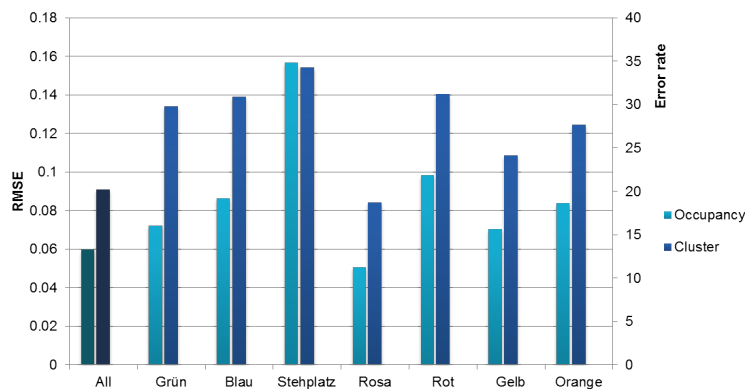


Figure 4.28: Comparison of the DQ1 and DQ2 performance for each price category.

price category. We used the support vector machine with PUK-kernel again on the full data set with 10 as cutoff day. In figure 4.28 the result is depicted.

Apart from price category Rosa, the coherency within each price category is lower than in the total set. Hence, the model is able to extract information that is present across all categories. This especially goes for the category Stehplatz as it acts rather different as we have seen. For this category, we lowered the minimum amount of tickets to 10 in order to obtain a reasonable amount of events and changed the cutoff day to the day before the show. ⇒ **NO** □

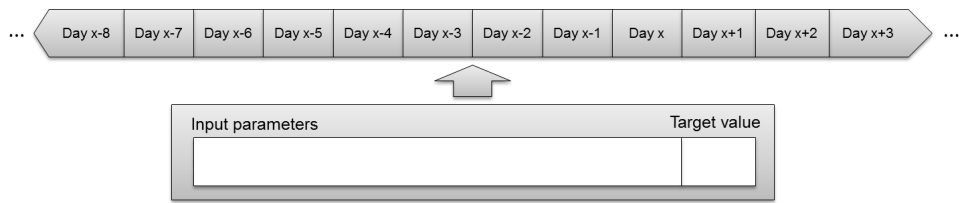


Figure 4.29: The idea behind the sliding window. This template may be applied anywhere on the data. It comprises seven input values (AR) and the target value.

SUMMARY

By aggregating the price category attribute to maintain the main categories only, and selecting the respective tickets by filtering, it is very easy to reapply the previous (and also subsequent) models. One needs to consider though, that normalization scale needs to be switched to the price category capacity, and sales figures of other categories need to be assessed separately, if needed. We have seen that there exist cross-category characteristics as the performance of price category prediction is lower than the prediction of events as a whole.

4.5 Domain Question 5 – Production Sales Figures Forecast

We now leave the event level and move towards the superordinate concept of productions. The goal of this question is to predict subsequent days at a certain point in time of an ongoing production, just as done before with events. Hence, we again face a time series problem. We are going to present three different solution attempts, each of them handled in its own section.

SLIDING WINDOW

Building the Model

The idea behind the sliding window approach is equal to an autoregressive time series model. As depicted in figure 4.29 there is a cutout of fixed length that can be applied anywhere within the time series. Unlike before, we now have several instances for each time series, i.e. one for each target day. Hence, due to the nature of the question, the time series problem is now transformed to a data mining problem in a direct and appropriate way. The difference that remains to a classical time series analysis is that we do not just use observations of the series at hand for prediction (i.e. parameter calibration), but also of different series in order to extract patterns and characteristics that are universally valid, and to overcome the self-containedness, as already mentioned in chapter 2.2. We have to make sure, however, that we confine ourselves to productions that are available during their whole lifespan as representativeness would suffer otherwise. If, for instance, the phase before the premiere would be overrepresented in the data base, so would the characteristics that are typical for that phase. Hence, we end up with a list of eight productions that are appropriate for this model.

We create the model by selecting the first seven lags to provide the sales of the previous week in a first step. The horizon is set to seven as well, as we want to know the sales of the day one week ahead. A bagged tree is used because the runtime of a support vector machine would be unnecessarily long due to its nonlinear time complexity, for now there are 5,221 instances. The model statistics are summarized in table 4.9.

Table 4.9: Results of a regression based on the sales of the preceding week.

Bagged tree (REPTree)	
Attributes	LG (7)
MAE	125.22
RMSE	200.11
RAE	45.5%
RRSE	57.03%
Correlation	82.1%

On average, the prediction is wrong by 125 tickets. In this context, we need to be careful however, as the result might be too optimistic as the temporal order is ignored when applying cross-validation. Additionally, each instance shares a lot of information with its neighbors by nature, which amplifies this effect. Later, when making use of the model, we are going to introduce another evaluation approach, but so far, it suffices as we confine ourselves to comparing the results. When looking at the plot on the right hand side, it becomes apparent that, like before, there are outliers on the very right, representing the remaining peaks that are not replicable by the model. If we go further and apply this model to the data base with no batch filtering as described in chapter 3.4 at all (as depicted in table 4.10), the result is even worse. The optimal axis turns to the left as the amount and amplitude of outliers increases.

Table 4.10: Results of a simple regression without filtering of batch sales.

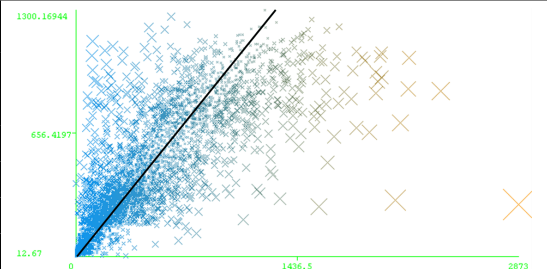
Bagged tree (REPTree)	
Attributes	LG (7)
MAE	190.7
RMSE	374.5
RAE	57.35%
RRSE	77.57%
Correlation	63.1%

Like before, this simple model is now expanded step by step. So instead of using the sales of the preceding week only, we might also take tickets into account that were sold further in the past. To limit the amount of attributes, again aggregations are used, i.e. one that holds the amount of tickets sold in all four weeks prior to the starting week, normalized by the amount of days involved. To keep the information content that comes from sales of the same weekday (as in Ragg et al. [2002], discussed in chapter 2.3), we subsume their normalized figures in a separate

attribute. The same two attributes may also be created for all past weeks. We eventually require that at least two weeks prior to the week of observation must be available to obtain meaningful values. This of course leads to the fact that models need at least three weeks of lead time. The resulting model is summarized in table 4.11.

Table 4.11: Results of a simple regression with sales of previous weeks.

Bagged tree (REPTree)	
Attributes	LG+PW (11)
MAE	118.96
RMSE	192.96
RAE	43.23%
RRSE	54.99%
Correlation	83.52%



The result slightly improves. When both weekday attributes are omitted, we end up with an RMSE of 198.84, which justifies their application. It has become evident that the longer the horizon, the bigger the influence of these past weeks. When predicting just one day however, the performance even decreases. In comparison, when creating a model with each past day as a single attribute (limited to 28), the RMSE is just 194.52 as a result of the curse of dimensionality.

▷ **Hypothesis:** There is an advantage in normalizing the aggregate sales figures by the amount of days.

As the RMSE reduces to 192.44 when omitting normalization, the presence of an effect described in Ahmed et al. [2010, chap. 7] could be assumed. That is, the model is able to extract just very little, but useful information in terms of the amount of weeks that is involved at all. As the difference is vanishingly low, we are going to conduct a statistical test this time. Therefore, the corrected resampled paired t-test as described in Witten et al. [2011, chap. 5.5] is used. The resulting p-value of 0.291 indicates that the difference is far from being significant. ⇒ **NO** □

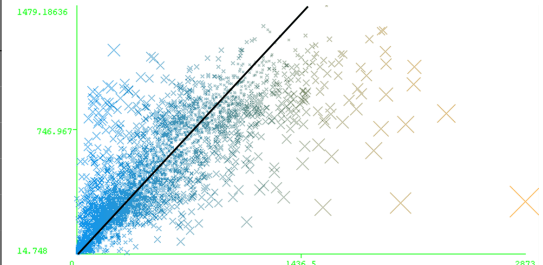
We move on by adding qualitative properties extracted from tickets sold in the week of observation, bringing the exogenous part of the model into play. Again, at least 20 tickets are required, which shall be distributed over three distinct days at least. These values could be optimized in further consequence, but so far we set them by intuition. This time we are going to forego all binarized nominal attributes as their contribution proved negligible. Instead, the average or modal values of the attributes of group EX (appendix C) are used. The model that results from using all 26 attributes yields a RMSE of 184.08 (RRSE 52.46%). Consequently, the algorithm is able to use these structural values to improve its accuracy.

We could also extract these properties from the tickets sold in the weeks prior, but the benefit cannot compensate the increasing complexity. Instead, we are going to derive other measures to describe the sales situation. Therefore, the amount of events that are concerned by the sales are determined. This value can then be divided into events which were performed in this week already and those which were not. The average distance to the performance date can still be

derived when looking at the averaged attribute created before. This separation is also possible for sales figures themselves, i.e. one for the amount of tickets that were consumed already and one for those who will be used prospectively. These numbers do not have to be normalized as the divisor would be the same for all instances. The model is summarized in table 4.12. The performance was greatly improved.

Table 4.12: Results of a simple regression with different structural features.

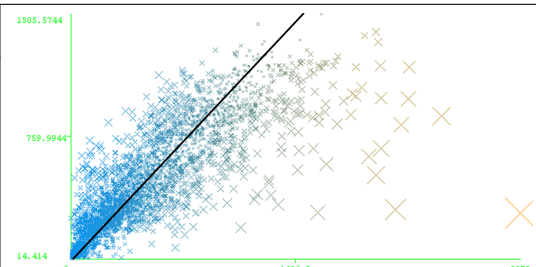
Bagged tree (REPTree)	
Attributes	LG+PW+EX+SS (31)
MAE	111.95
RMSE	180.03
RAE	40.32%
RRSE	51.30%
Correlation	85.85%



The approach described so far corresponds exactly to the proposed **ML-NARX** model introduced in chapter 2.2. It reflects trends and seasonalities already, for it is aware of temporal aspects such as the season of the year. The information added at last further enables it to approximate complex development patterns such as a steep growth after the premiere or gradual decrease at the end of the season as well. To further enhance these capabilities we will supply information about the target day available in any case. That is the amount of performances that is scheduled for the respective production which can be assessed by examining the schedule for instance. The target date itself can also be described in terms of the day of the week, the day of the month and whether it is a holiday. We could of course go further by adding more and more derivatives, but for our purpose we will leave it at that. The final configuration result is shown in table 4.13. Altogether, an RRSE of 48.1% and a correlation of 87.7% could be achieved which certainly qualifies this model to be used in further analysis.

Table 4.13: Results of the total regression model. It consists of several different attribute groups.

Bagged tree (REPTree)	
Attributes	LG+PW+EX+SS+NX5 (35)
MAE	104.98
RMSE	168.77
RAE	38.15%
RRSE	48.1%
Correlation	87.69%



Model Evaluations

The performance of different algorithms on this attribute configuration can be assessed in figure 4.30. We again conducted a feature selection via wrapper to meet sparseness. This time,

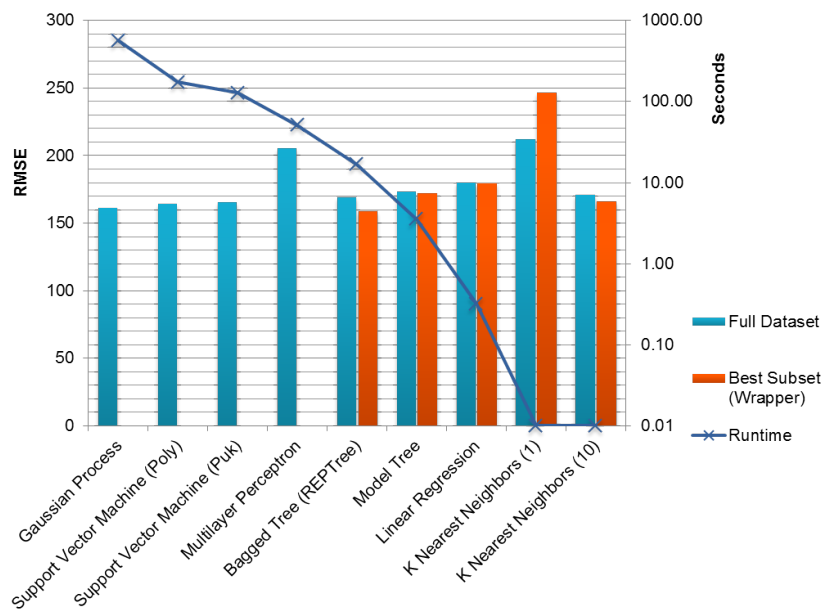


Figure 4.30: Accuracy of several regression algorithms for the sliding window. The runtime follows the logarithmic scale on the right hand side.

however, as only valuable attributes found their way into the model, the performance cannot be improved considerably. The Gaussian process performed best, but very much at the expense of runtime. The multilayer perceptron should be substituted by support vector approaches again. In the end, the bagged tree was identified as a good compromise between runtime and performance.

We also tested whether standardizing all numeric attributes would increase the performance, which is not the case for any of the algorithms shown. In many cases, the results are not even different, but the runtime increases. If the target variable is standardized as well, the result gets worse in every case, for harmonizing the value range cancels out an important distinguishing feature.

▷ **Hypothesis:** Google Trend values improve the regression performance.

We are going to verify the findings of Hand and Judge [2012] by looking at the influence of the respective attribute. We could derive this information from the fact that it has been selected by the wrapper attribute selector in two out of five cases – the linear regression and the model tree. Hence, there seems to be an influence worth mentioning. A similar picture can be obtained by means of a correlation analysis of the target value. The resulting coefficients are between 47.8 and 48.8% ($p = 2.2e - 16$) depending on the horizon (up to two weeks). Figure 4.31 further shows the sales and Google Trend for production 5830, visualizing the correlation.

In the end, we draw a direct comparison between the model with and without the attribute, which however should be used with caution. By making use of the full attribute set, the average

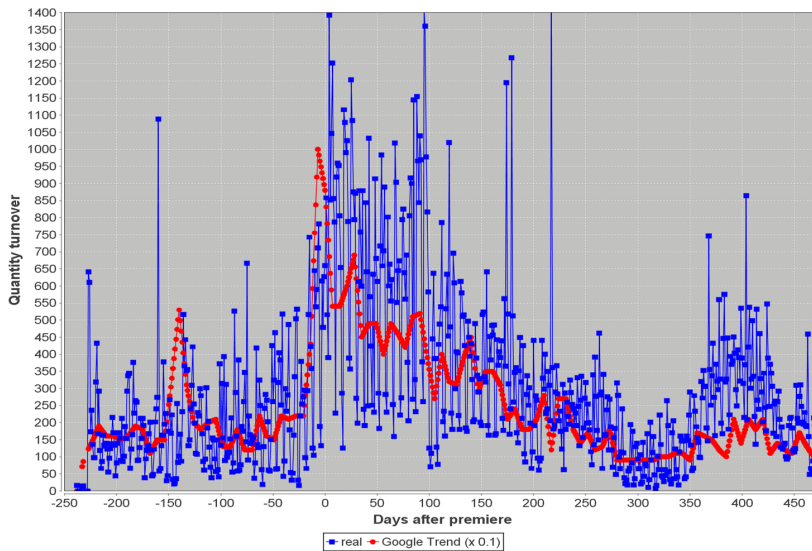


Figure 4.31: Comparison between the Google Trend and the sales of production 5830.

performance deterioration is 1.76 RMSE, or almost one percent, when looking at all algorithms. However, the result got better for the multilayer perceptron and the model tree.

Roughly, we can confirm the positive effect of this attribute, which is possibly bigger in reality when not having to interpolate the weekly values. The procedures shown here can be applied to assess the influence of any attribute in general for any domain question. ⇒ **YES** □

▷ **Hypothesis:** It is possible to observe differences in sales characteristics based on the life cycle phase of a production.

We are going test this hypothesis by means of the relevancy of the attribute that indicates the distance to the premiere using the steps shown just before. First of all, the attribute has been selected by the wrappers two times. A correlation analysis is inappropriate however, for we assume the influence of this attribute to be inherently nonlinear due to the presence of different phases. Eventually, a direct comparison yields an average difference of -0.2921 RMSE which is not statistically significant. Hence, the picture obtained is inconclusive, possibly due to algorithm deficiencies. ⇒ **YES/NO** □

▷ **Hypothesis:** Sliding window predictions work better within one production than across all.

To simplify matters and stick with our previous procedure we ignore the temporal aspect in applying cross-validation. This time, the support vector machine with PUK-kernel is used. The results are shown in figure 4.32. Like in DQ2, the coherence is higher on the whole as the weighted average RMSE is 161.33 compared to 165.05 of the holistic model, but a consistent picture cannot be provided. ⇒ **YES/NO** □

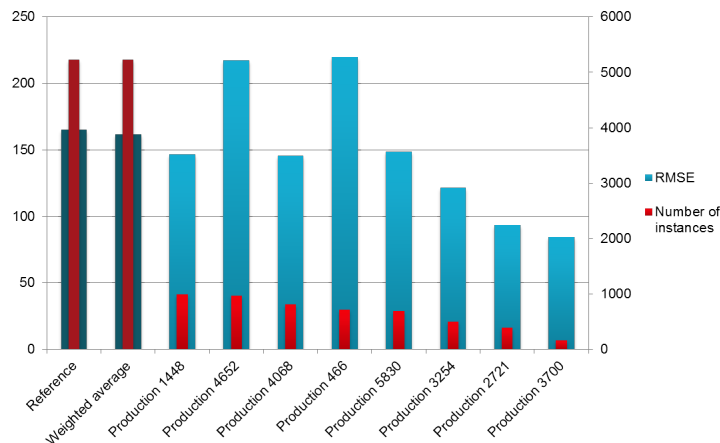


Figure 4.32: Sliding window forecasting within each production. The weighted RMSE is compared to the RMSE of the holistic model on the very left.

Predicting Sales Figures

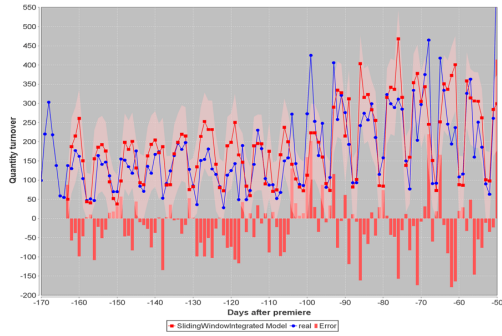
With this model configuration we can now create predictions of arbitrary horizons. For each of them, the confidence can be expressed by considering the errors of other instances in analogy to DQ2. As we have pointed out, evaluation results are inappropriate in absolute terms as they are too optimistic. Hence, an alternative leave-one-out cross-validation approach is proposed by not taking an arbitrary fraction for evaluation but exactly one production that was left out for training. This is repeated until every production has been evaluated. The resulting representative RMSE is now 172.88, which is about 2.4% more than using k-fold cross-validation.

Figure 4.33 depicts a seven-step-ahead forecast for each day of production 4068. The training data for the model makes use of all other productions. For the sake of clarity, we visualized four phases only. The algorithm used for this and all subsequent examples is the bagged tree.

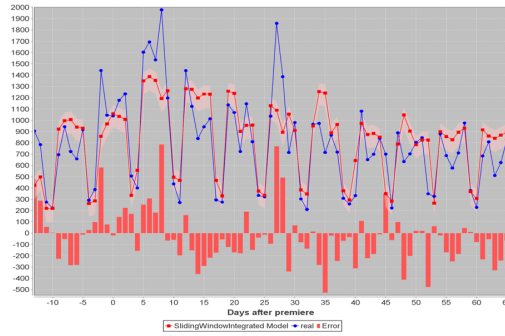
▷ **Hypothesis:** The performance can be improved when using data of the current production as well.

We try to answer this question by comparing the RMSE of the production-level cross-validation with the one that results from evaluating the last half of each production when the first half was used for training as well. The remaining first half is evaluated as usual with information from other productions only. The resulting picture is shown in figure 4.34. We can see that the error decreases by about 3.5% altogether, which can be considered significant. It does so for all productions except 2721. Hence, it makes sense to use previous data as well as it provides more information to the model. ⇒ **YES** □

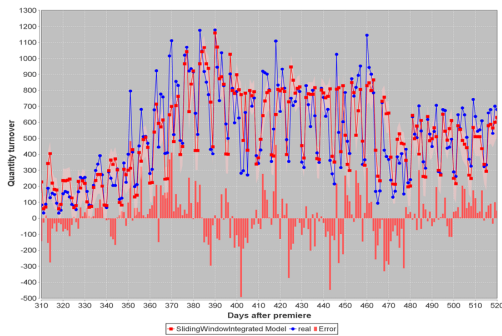
We now forecast the prospective sales trend of production 4068 by applying several independent models of different horizons, limited to 100 days. Figure 4.35 depicts that for two different cutoff days – one at the premiere and one 300 days after, during summer break. It becomes clear



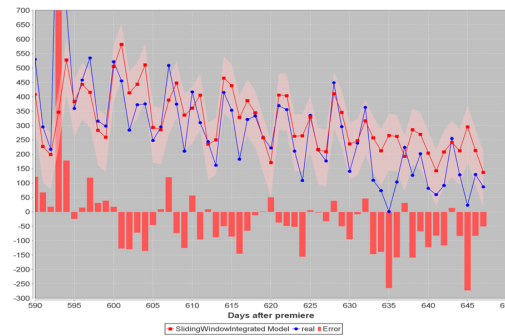
(a) In the beginning of pre-sales.



(b) Around the premiere.



(c) One year after the premiere.



(d) At the end of the production.

Figure 4.33: Seven-step-ahead forecasts for each day in different phases of production 4068. The blue line represents the true sales figures whereas the red one the prediction. The red shaded area marks the 60% confidence interval.

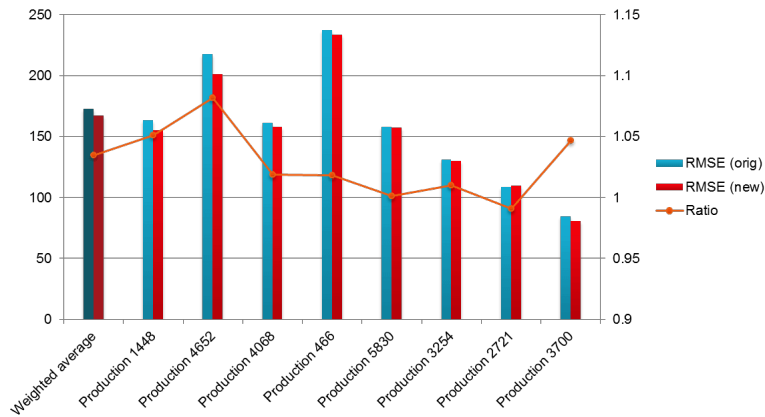


Figure 4.34: Comparison of regression performance with and without previous data of the same production for each of them. On the very left the weighted averages are illustrated. The orange line orientated on the right hand scale describes the ration between the two values.

that weekly fluctuations, seasonalities and trends can be covered quite well, just as structural breaks for instance when the summer break ends in the second graph. Salient and atypical patterns which occur especially in the first graph cannot, however. The RMSE of the first example is 224.21, representing an RRSE of 54.87%, and for the second it is 137.33 and 41.3%, respectively. The final deviations are just 632.3 and -624.4 tickets. It should be noted that the range of variation does not increase significantly with bigger horizons, but instead reaches a high level very soon after a few days only.

In appendix B.5 we added some further examples of production 4652.

We now assess whether certainty increases when time passes by means of the first cutoff day example. In figure 4.36 we make four steps of 20 days towards the last prediction day. As can be seen, the accuracy does not improve substantially. The RMSE evolves from the original 224.21 to 225.92, 201.33, 206.24 and 218.87 ultimately while the absolute deviation changes from 632 to -5,095, -919, -476 to 1,054 in the end. This finding corresponds to the fact that confidence intervals do not expand over time, for the information the model receives allows a high accuracy to be achieved for a very short period of time only. Uncertainty then remains more or less equal for all subsequent days.

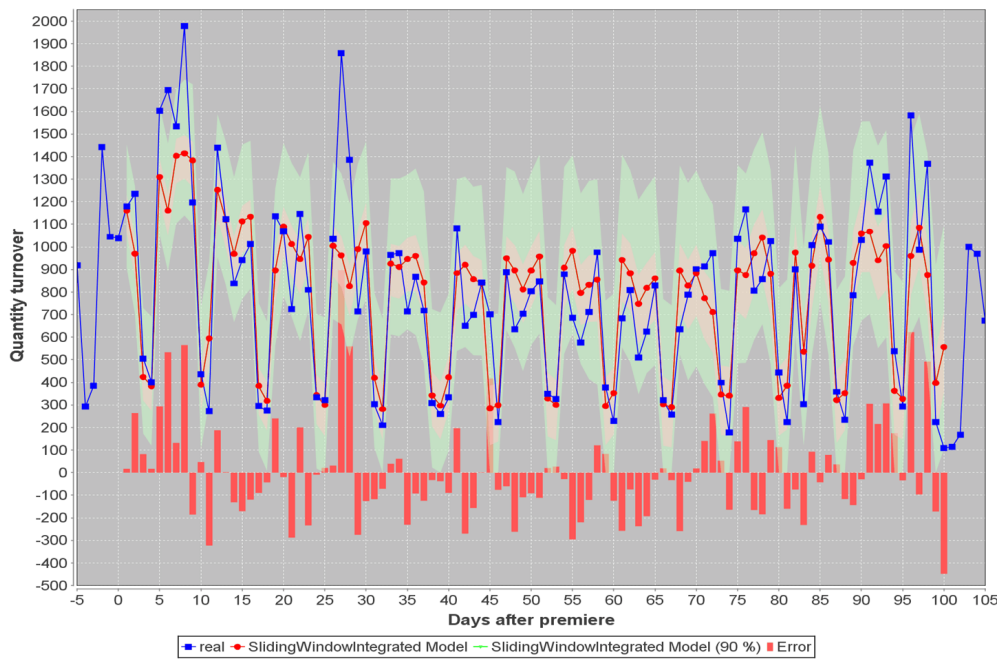
In a completely different approach than before we could add the horizon as attribute to the model and add instances of different horizons, resulting in a holistic model that is applicable for all horizons contained. Theoretically, such model could be as good as the individual models, but in practice this is not the case as shown in figure 4.37. There, we compare the performance previously reached (without taking previous observations of the same production into account) with the performance of a model that contains horizons one to 14 evaluated using the same instances, that is those of horizon seven.

The performance is better in one case only, that is again production 2721 which obviously holds some interesting characteristics. In total, the performance decreases by 3.5%. Apparently, the huge complexity prevents the algorithm from detecting structural relationships and differentiate between different horizons appropriately. Hence, we are going to refrain from this approach. This finding is also applicable to DQ1 and DQ2.

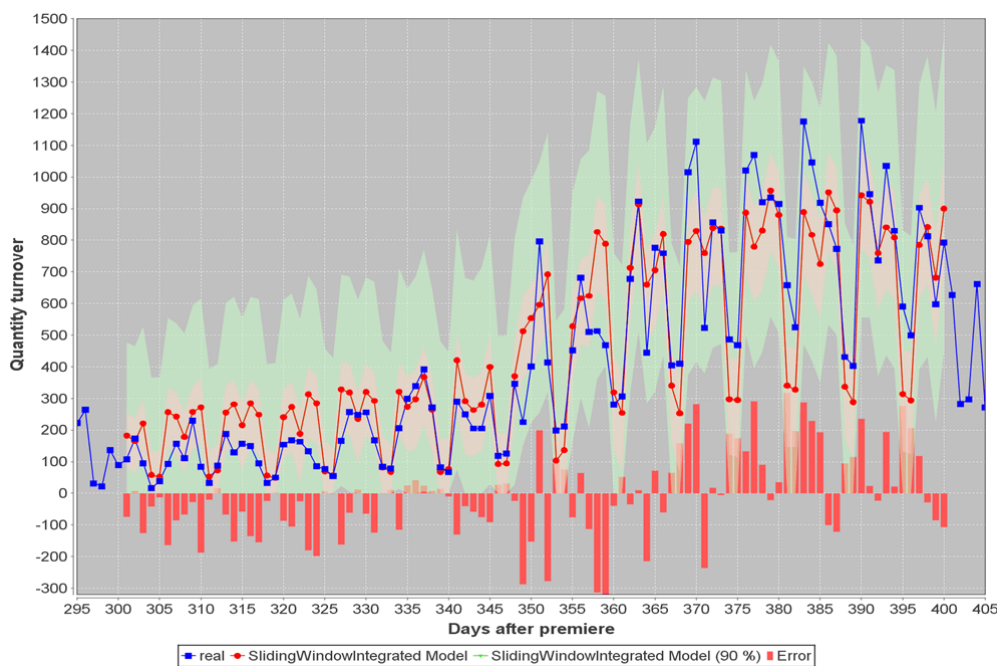
Before moving on with the next approach we want to mention that there exists a time series extension for Weka, the *Time Series Environment*, which enables to create NARX models. The AR-part can be created automatically, but any further exogenous factors need to be created manually again, except for date related information fields. Beyond that, it provides some very interesting analysis facilities, but as a third-party prototype, it is immature and buggy. We refer to the homepage¹ for further information.

The sliding window approach described here shall be called sliding window integrated to allow for a differentiation between the following one. Integrated stands for the fact that forecasts on a daily basis can be created inherently. We are going to see further evaluations of this approach in chapter 4.5.

¹<http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka>



(a) Cutoff day is set to the day of premiere.



(b) Cutoff day is set to 300 days after premiere.

Figure 4.35: Sales forecast of production 4068 using the sliding window approach at two different cutoff days. The line colors have the same meaning as before, but the green shaded area now additionally describes the 90% confidence interval.

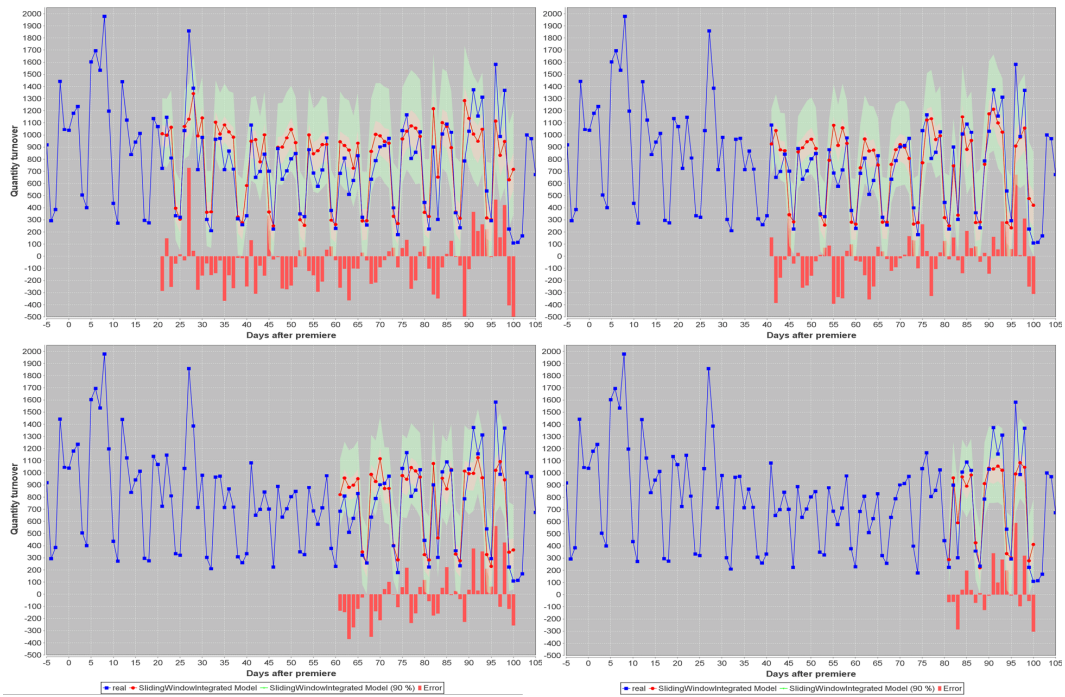


Figure 4.36: Sales forecast scenarios when approaching the final prediction day.

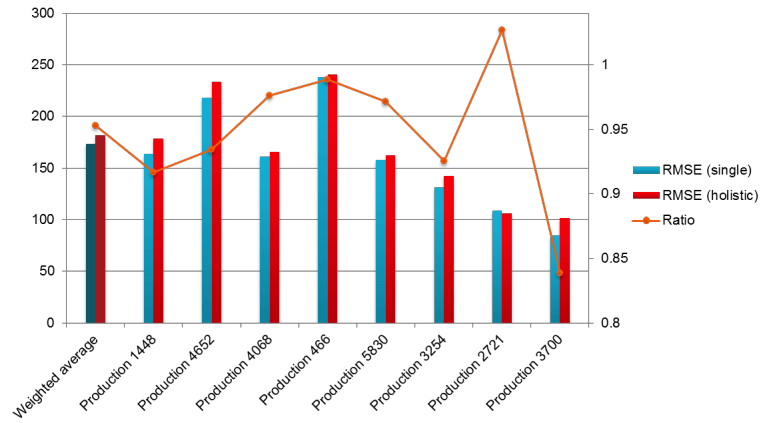


Figure 4.37: Comparison of previous regression performance with a holistic model that contains data of several horizons.

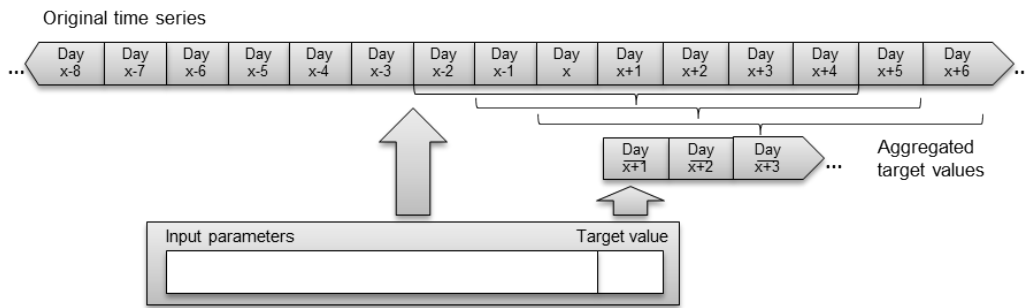


Figure 4.38: The idea of the sliding window smoothed variant. The target value represents the average of the three days before and after the respective day.

SLIDING WINDOW DIFFERENCE

The approach we want to present in this chapter is divided into two phases. Firstly, a forecast is made on a weekly basis using a similar approach as before, called sliding window smoothed. Then, to incorporate fluctuations of a weekly seasonality, an intra-week difference model is applied – hence the name.

We start with the first phase by reusing the model created previously. To predict weekly values, first of all the target variable needs to be replaced. As depicted in figure 4.38 we want to predict the average of the week that encompasses the respective day, i.e. three days before and after. This creates a smoothing by simultaneously losing granularity. Of course this leads to the fact that predictions of a horizon smaller than four days are too optimistic as a part of the solution is available as input. But since predictions are usually made for bigger time spans this is not much of a problem.

Another difference is that the two parameters describing the sales of the respective weekdays of prior weeks cannot be used, as they turn out irrelevant in this context. In the end, the attributes describing the target day need to be replaced as multiple days are involved now. Hence, we are going to take the average month, count the holidays and remove the weekday attribute. In figure 4.39 we compare the resulting configuration is compared with the integrated approach. We make use of the production-level cross-validation again as the result would be hopelessly too optimistic, for now not only the input parameters, but also target values share up to six sevenths if adjacent. The smoothed result can be predicted with a RMSE of 96.27 on average, which is a little more than half of the error reached before for all productions.

As regards phase two, we now must reapply the weekly deviations that could be observed previously already. On average, Monday contributes 16.7, Tuesday 18, Wednesday 17.1, Thursday 17.6, Friday 16.1, Saturday 7.8 and Sunday 6.7 percent to total weekly sales. The lower values on weekends are attributable to the fact that point of sales are closed then. In a simple attempt we could multiply the respective forecast with these fractions, but as this is a data mining study we will build a model to obtain more accurate values. For this model, an input vector is created that consists of the sales fraction of each weekday for each week of each production to

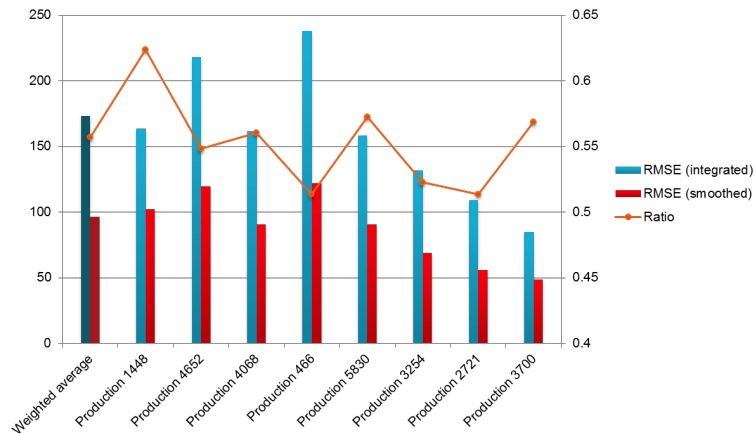


Figure 4.39: Comparison of the integrated and smoothed sliding window approach.

express the fraction of any prospective weekday. We will only consider weeks if sales are available for at least three days. Of course, we again need to take care that the production lifespan is not overshoot by requiring that the target week must contain at least one sale. Alternatively, also absolute sales, or differences of the weekly average could be used.

We now face a situation where information is very scarce making an algorithm struggle to extract patterns that are not random. However, we will not provide any other information as this is contained in the other model already. In figure 4.40 different algorithms are released on the problem, but this time we compare their root relative squared error as we want to know whether the result is better than taking the averages derived from training data, which is expressed as a value below 100%. The analysis encompasses horizons from one to 14 and we make use of k-fold cross-validation.

The only algorithm that is able to sustainably outperform the average is k-nearest-neighbor with k set to 100. The support vector machine is unable to handle the low information appropriately resulting in the worst performance by far. It is remarkable that, despite using relative values, weekend fractions are predictable significantly better. As a consequence, we are going to use k-nearest-neighbors in the following examples even though the increase in benefit is small.

We can now combine these two models. However, it must be considered that the intra-week difference model needs to be applied on calendar weeks only, i.e. the week must start with Monday and end with Sunday. So unlike before, we need to make sure that the prototype to be classified fulfills this requirement by shifting the window some days into the past if the cutoff day is no Sunday. This leads to an increase of the horizon, which in turn does not mean that the performance suffers as we have just seen.

The difference model may be applied to the weekly average of the smoothed prediction (using the six adjacent days) to prevent total sales from being biased, or to the predicted value itself. Since we would lose particular variations known to the sliding window model only in doing so, we decide upon the second option. We reuse the previous example and apply a 100 days-ahead prognosis to two phases of production 4068, as shown in figure 4.41.

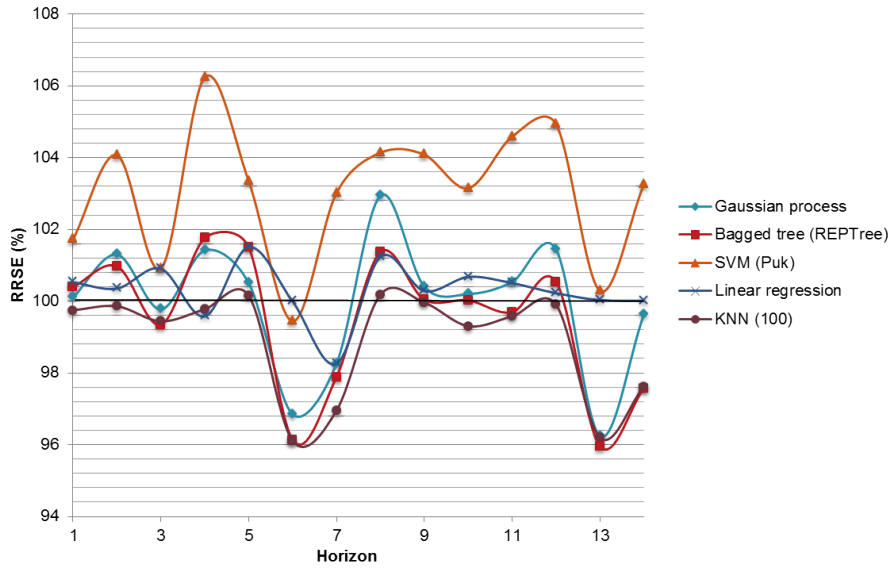


Figure 4.40: Different algorithms are applied on different horizons to project the daily fraction of total week sales.

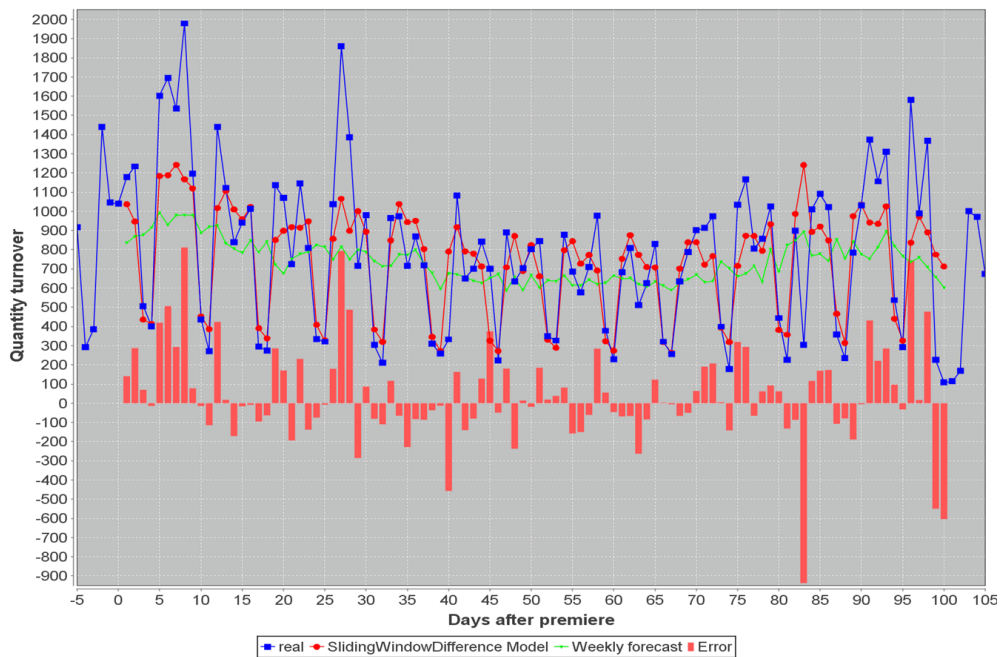
The RMSE now is 259.98 and 173.16, respectively, which is more than previously using the integrated approach. The final deviations are also bigger as they come to 4,246.5 and -856.1. Hence, it seems that structural shifts cannot be reproduced as good as before, but of course this example is too small to draw a general conclusion thereof. We omitted plotting the confidence intervals for the sake of clarity. They can easily be calculated by multiplying the confidence of the sliding window prognosis with the one of the daily difference prognosis.

Again, we included further examples in appendix B.6.

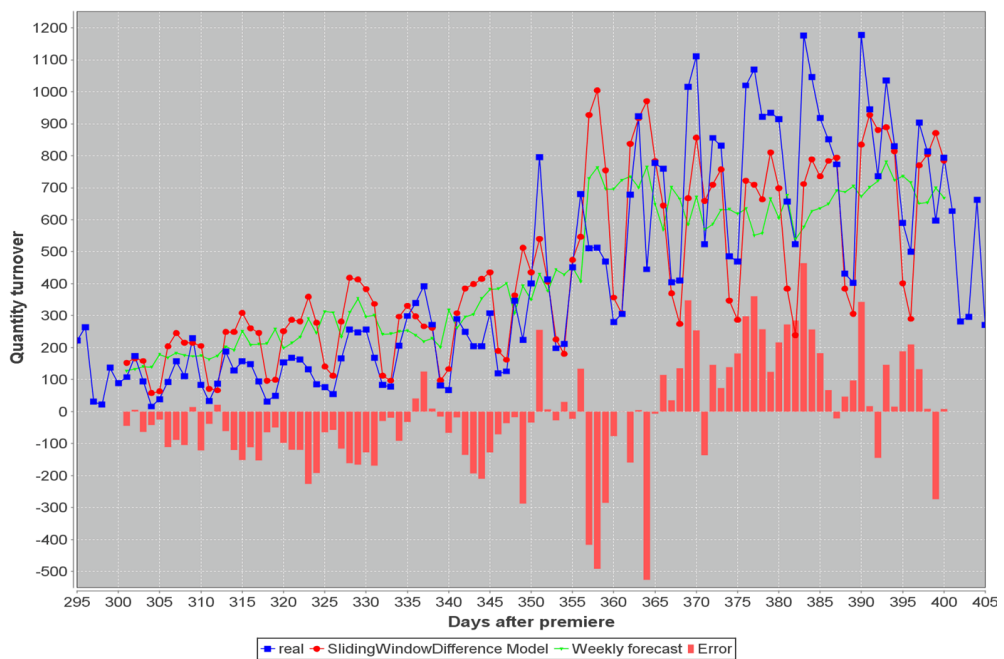
COMPARISON OF THE APPROACHES

Basically, these two approaches are rather similar, but how do they perform compared to each other – or perhaps even more interesting: how do they perform compared to classical approaches? These questions are addressed in this subsection.

As classical approaches we will use the figures of the cutoff day as the forecast (called “previous day”), two moving average predictors, one with seven and one with 14 lags and exponential smoothing with alpha set to 0.2. We will refrain from employing a classical ARIMA(X)-class time series approach for reasons already mentioned. Beyond that, finding optimal configurations by calibrating parameters and coefficients and using various extensions necessary would go far beyond the scope of this work. These simple methods were applied using previous values of the same production as calibration base and going through all days of a production for each horizon. Moving average and exponential smoothing need a certain lead time, as do our approaches. For this reason, a kickoff day is defined for each production in order to preserve comparability, usually two or three weeks after sales become significant. That is why the results differ slightly



(a) Cutoff day is set to the day of premiere.



(b) Cutoff day is set to 300 days after premiere.

Figure 4.41: Sales forecast of production 4068 using the sliding window difference approach at two different cutoff days. The blue line again stands for actual sales figures, the green one for the smoothed prediction on a weekly basis and the red one for the final combination.

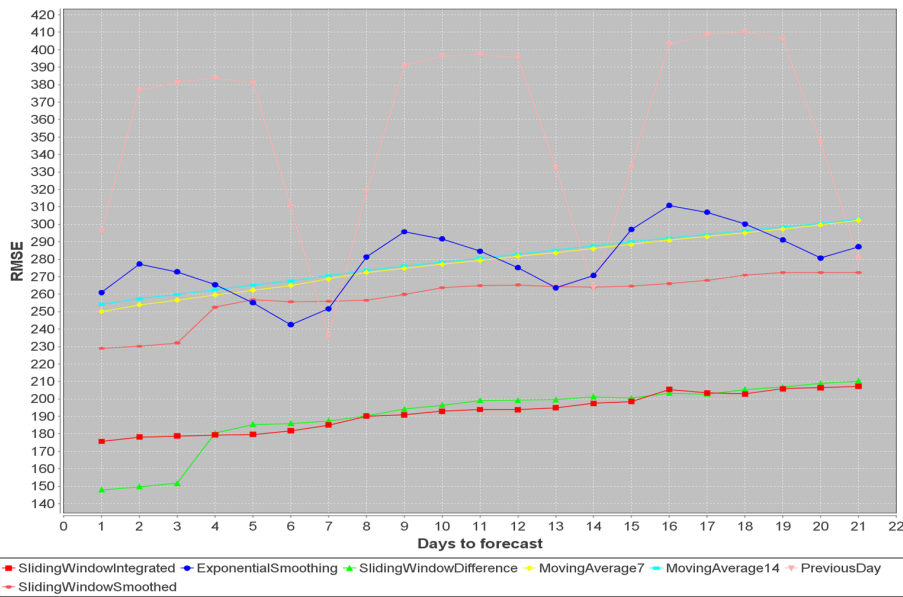


Figure 4.42: Comparison of the sliding window approaches and classical forecasting methods with the horizon reaching from 1 to 21.

from results of previous analyses. Our sliding window approaches were executed as usual – using a production-based cross-validation with the optimal configuration. For the sake of runtime we refrained from using observations of the same production for model building as this would result in as many models as days available, improving performance in a negligible magnitude only.

Figure 4.42 visualizes these seven forecast methods for one to 21 days after the cutoff day. As can be seen, the two sliding window approaches producing predictions on a daily basis outperform all classical approaches right from the beginning, while the difference between themselves is negligible. It is visible now, that the sliding window difference approach pokes out for the first three days as a part of the solution is contained in the input vector. The performance of the previous day predictor redemonstrates differences within a week as horizons of multiples of seven result in lower deviations. As it was to be expected that individual day predictors outperform simple weekly aggregate predictors, also the smoothed method outperforms those approaches. This becomes even more obvious when increasing the observation window to one year, as depicted in figure 4.43.

The curves of all sliding window approaches apparently follow a logarithmic course. Starting with a horizon of approximately 100 days, the average deviations stop from becoming significantly larger. This is attributable to three factors in all likelihood: firstly, the amount of evaluation, but also the number of training instances decreases as indicated by the straight blue line, blinding out various situations of interest. Secondly, the amount of tickets sold each day usually does not exceed 1,000. To be exact, 415.79 tickets are sold in average. As a consequence,

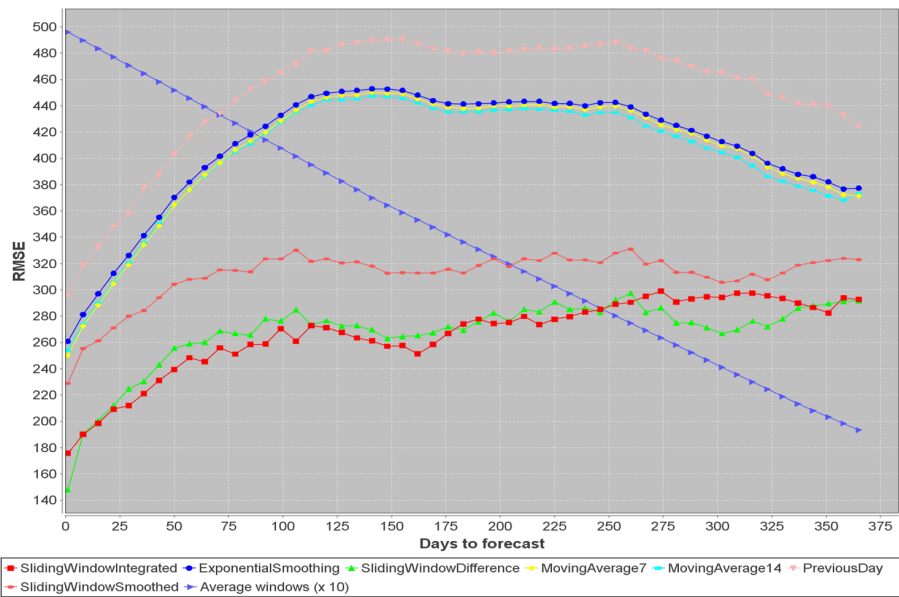


Figure 4.43: Comparison of the sliding window approaches and classical forecasting methods for the long term with the horizon reaching from one to 365. For the sake of clarity and runtime only multiples of seven were selected as horizons. The straight blue line stands for the average amount of instances, or windows, that were available for all methods at the respective horizon.

all methods, but especially the previous day-method reach their natural limit as deviations cannot become any bigger on average. The structural superiority of our approaches is the reason why they reach a lower plateau. Thirdly, the decline towards the end, which is observable at the classical approaches, evinces the presence of a yearly seasonality. Last, but not least, we want to highlight the fact that both daily forecasts are caught up more or less by the weekly ones as on the long term intra-weekly differences are becoming harder to foresee.

Hence, instead of evaluating on a daily basis, we are going to take the weekly aggregated deviations now, i.e. the total deviation at the end of each week. As a consequence, the sliding window difference approach becomes irrelevant. The result is depicted in figure 4.44, looking rather similar as the advance between our and the classical approaches is quite big again. When comparing the relative advantage at week ten, which is day 64 to 70, the value raises from 1.55 before to approximately 1.65. Note that on average 2,933.025 tickets are sold weekly over all productions. By normalizing the RMSE on that basis, the resulting fraction is 44% for the sliding windows and 73% for the classical approaches. When looking back at the daily prognosis again, the fractions are 61 and 95 percent there, respectively.

Beyond comparing different methods and horizons, one could also assess the performance in two other dimensions, that is the algorithm and data base configuration. The amount of possibilities is endless, but we conducted two such comparisons in appendix B.7.

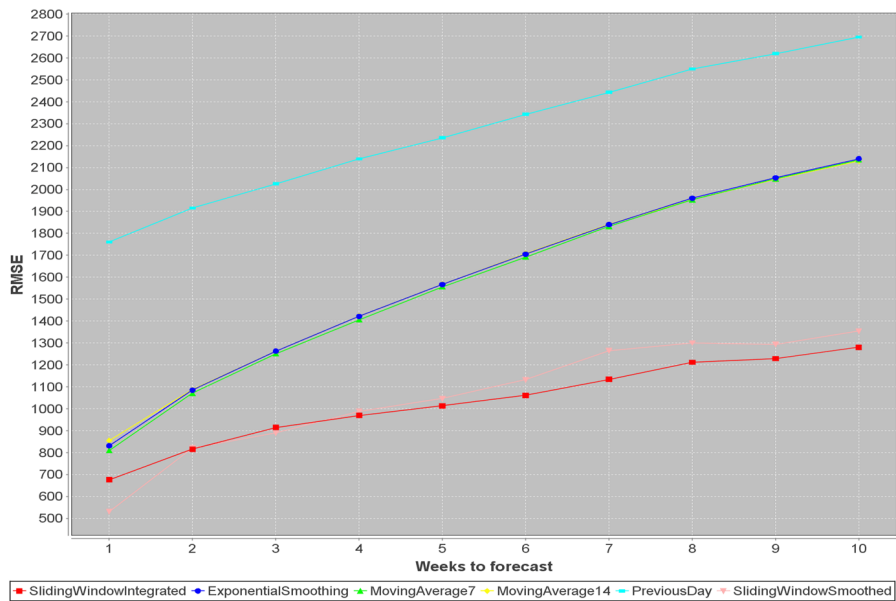


Figure 4.44: Comparison of the sliding window approaches and classical forecasting methods on a weekly basis.

MICRO-FOUNDED PROGNOSIS

The last approach to answer this domain question is proposed as the micro-founded prognosis. This term is lent from the economics field and constitutes the connection between micro and macro. That implies that behavioral aspects of the macro-level basically are the result of behavioral aspects on the micro level, derivable by aggregation. This idea can be translated to our problem quite well as total sales of a production are made up of sales of the respective events. For DQ2 we provided a solution to forecast individual event sales. These forecasts can now be combined to describe the sales trend of a production. In comparison with the sliding window approaches, the resulting figures are justified on structural grounds now, for they result from real events and their capacities. As we have seen, sliding windows incorporate the amount of events as well, but just in the form of influencing factors, which can shift the forecast upwards or downwards.

To put this idea into practice, we first of all need to determine all events that are scheduled and hence have to be considered at any moment of a production. They need to be iterated through then by each time determining a) the amount of days until the performance from the current date and b) the amount of days that remain as from the target date. Hence, we are even more interested in intermediate results than in DQ2 here. We then project the sales from day x to day y, translate the relative occupancy rate back to absolute numbers, aggregate the values and obtain the total turnover in the end. For the sake of calculatory simplicity, we are going to ignore the production plan and use events of other productions for model building only.

As described in DQ2, we further need to distinguish between normal and vague forecasts in case there are not enough tickets available to make use of their properties. Depending on the

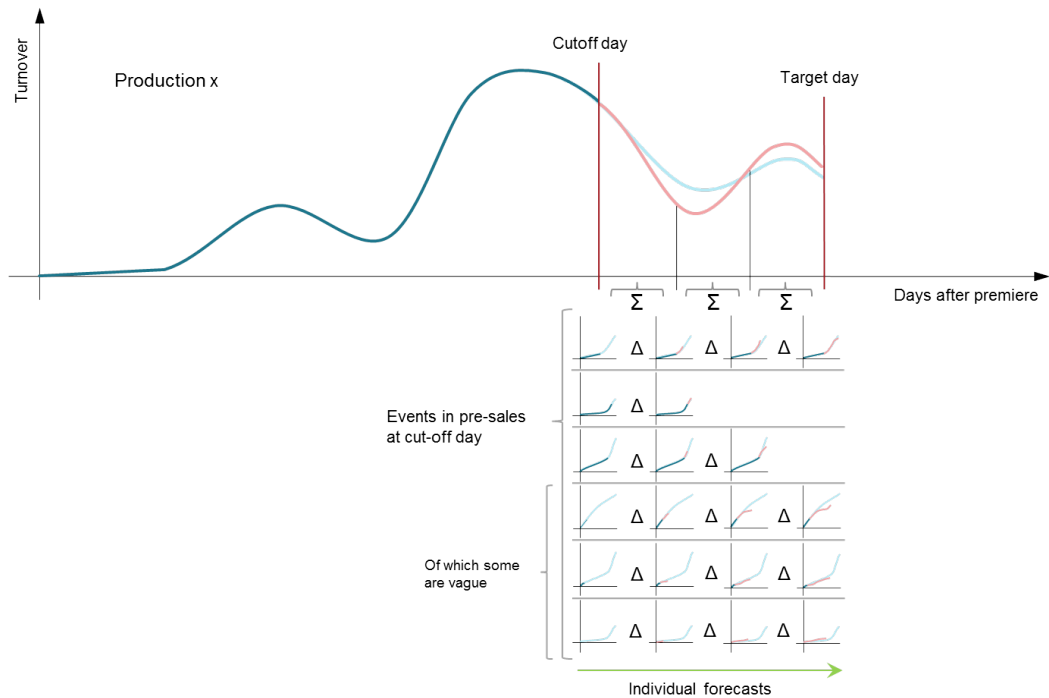


Figure 4.45: The idea of the micro-founded prognosis. The aggregate turnover of a production consists of the individual sales figures forecasted for all events that are in pre-sales during cutoff day. Some events might drop out as they take place over the time.

horizon and length of the event schedule, we might even face a major proportion of such long term forecasts. Thus, we need to keep that in mind and select the appropriate approach for each event. Figure 4.45 abstractly summarizes the idea described so far.

We also need to remember that event forecasts of different horizons are statistically independent from each other, just as the sliding window ones are. Moreover, not just the target day, but the whole timespan between the cutoff and target day is predicted. For this reason, we need to subtract the previous prognosis of each event when forecasting multiple days to obtain daily values. However, as we are interested in sales figures trends and final deviations only, this is not an issue. But what definitely is an issue is the fact that there might be a negative turnover for some days, resulting from declining occupancy rates due to instability. Another drawback is that intra-week deviations cannot be mapped as these characteristics simply get lost in the end. Both these problems can be solved by applying the weekly difference model again, but this time on a prediction that has been smoothed on a weekly basis.

We apply this approach to the second scenario of production 4068 used before in figure 4.46. This time a 200 day prediction is created to get a better overview. As we can see, some of the individual predictions (green line) fall below the line of zero but all final predictions are reasonable. The amount of normal forecasts remains stable at 69 until day 360. Afterwards, it steadily declines to reach zero at day 443. The amount of vague prognoses remains stable for a

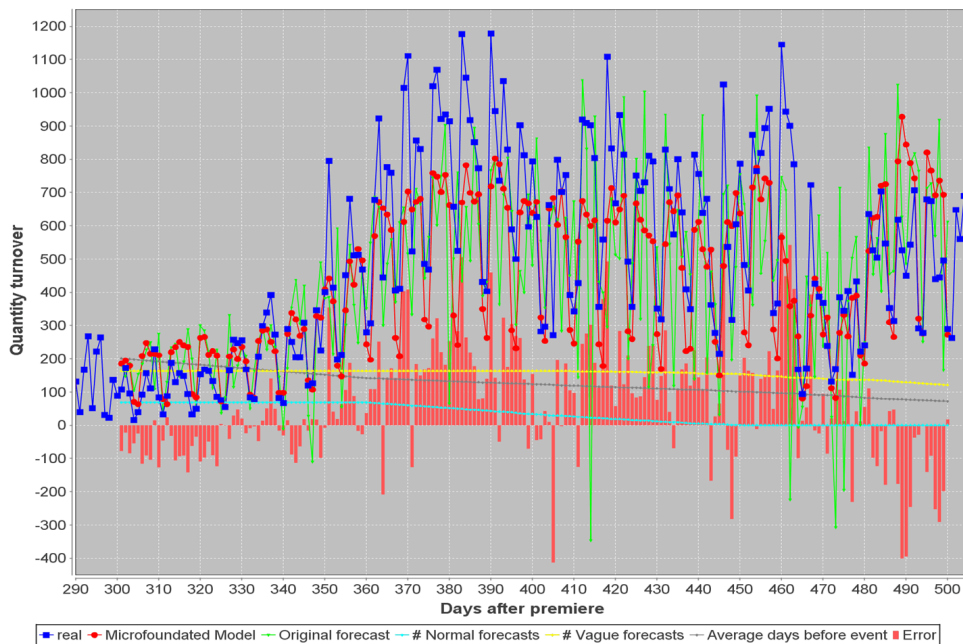


Figure 4.46: Sales forecast of production 4068 using the micro-founded approach. The blue line again stands for actual sales figures, the green one for the original prediction and the red one for the final combination. The turquoise line represents the amount of normal forecasts that were made for the respective day and the yellow one for the vague ones. The grey line represents the average amount of days that remain until the performance date.

longer period, as one would expect, starting to descend as from day 410. The resulting RMSE is 186.57 with a total deviation of 13,642 tickets. Compared to the sliding window approaches now an RMSE of 165.96 is reached for the first half of the prediction, which is higher than the integrated approach but less than the difference one.

Creating this prognosis took approximately 8.5 hours of CPU time (again measured using a 3 GHz AMD processor) and resulted in 37.517 models (of which 29,161 are vague and 8,356 normal), consuming almost 11 gigabytes of disk space. This enormous amount results from the fact that events are scheduled almost every day in this production. Depending on the horizon, the amount of events to be forecasted for each day may reach several hundreds, as in our example, yet it decreases because events take place as time goes by. Hence, we face a problem of quadratic runtime complexity while it was only linear for the sliding window solutions. Beyond that, the model is context-sensitive, i.e. models created for a specific period may not be usable for another one of the same production.

Given these drawbacks, we will not perform an evaluation as before. The framework we developed does indeed use massive multithreading and caching of instances in order to simply update target related values to increase performance. However, the quadratic runtime can only be reduced by a factor eventually. Alternatively, we could follow the idea presented for the sliding windows, that is to put all information into a single model only. This model would then consist

of a vast amount of instances, that is 2,500 base instances (coming from all productions except one) times 200 as cutoff days (depending on the horizon) times 199 as target days, making up about 100 million data items. (We deliberately left out the demand for a separation of normal and vague forecasts.) As a consequence, the amount of models can be drastically reduced to the amount of productions, but the drawbacks are obvious as well. Firstly, creating a model with this magnitude of data would take a huge amount of time. Secondly, memory demands would be tremendous as well, most likely exceeding the capabilities of a normal computer. Thirdly, it is questionable whether an algorithm is able to extract useful meaning in the light of the hotchpotch it has to face. As we have seen, this indeed is an issue. Last but not least, we could only spare a part of the work as instances need to be created either way.

Hence, we leave the approach at this experimental state and confine ourselves to a visual analysis of different scenarios, of which some are included in the appendix B.8 again.

SUMMARY

In this chapter we presented three different approaches to predict sales figures of a production. The first two are an implementation of the ML-NARX idea. We used sales figures of the past as the autoregressive part and qualitative attributes about the tickets, the week observed, the cutoff day and the target day as exogenous inputs. This and the fact that data from different time series are used as calibration is why they are to be distinguished from classical time series models. From a methodological point of view, the problem and its solution was quite similar to the one of DQ1 and DQ2.

We differentiated between the integrated sliding window approach that is capable of predicting daily values inherently and the difference approach, which is a combination of a weekly sliding window forecast and an intra-week difference model. This difference model predicts the fraction of a certain weekday of the sales from a whole week on the basis of the cutoff week. In the end, both approaches perform equally well. As there is no forecast method available that is used in the establishment by now providing a comparison as in DQ2, we had to confine ourselves to use some classical simple forecast methods. The sliding window approaches outdo them by far.

The third method makes use of the approach presented in DQ2 to forecast sales on the micro level, i.e. the event level. These predictions are then aggregated to reproduce the global picture on the macro layer, i.e. a production. This micro-founded approach proved promising, but due to its complexity an automated evaluation is not possible.

4.6 Domain Question 6 – Extension of a Running Production

Using the solution proposals presented in DQ5, we now address the question whether it makes sense to extend a production currently active beyond the already scheduled period. In order to do so, we simply need to set the forecast period accordingly to project prospective sales and then manually assess whether it would be financially rewarding.

As we have seen, these approaches cannot be applied to forecast periods beyond schedule without further ado as they rely on prospective event information. This especially goes for

the micro-founded approach for no events mean no sales there. On the other hand, the sliding window approaches are basically able to cast a prediction as only one attribute disappears, that is the information about the amount of events that is to take place on the target day or in the target week. As a consequence, uncertainty increases by a small degree. To overcome these problems, different event schedule scenarios should be used.

CASE STUDY

Any such decision is generally not taken at the very end of the production schedule, but much earlier already, when there are still enough events in the pipeline. This is also the case in a small case study that shall be done in order to answer this question. It is based on a real problem from the past. The initial situation is a production that has already been playing for about 200 days and has a schedule that is still six months long. The question at hand then is whether the market potential is sufficient to ensure adequate event capacity utilization rates if the season was further extended. The production concerned (6422) has been excluded in all previous questions. As a result, we can evaluate our approach by the means of a completely novel manifestation.

The forecast scenario looks as follows: the cutoff day is set to 207 days after the premiere. We use one event schedule only, that is the one that was realized in the end, lasting approximately ten months until day 513 after the premiere. Nevertheless, the forecast shall be done for one year into the future, reaching beyond the scenario and thus enabling us to assess the behavior of the approaches in such a situation. In order to evaluate at least a part of the solution, we have the real sales figures of 203 days after the decision date at hand. The training base was formed using all other productions and the sales of production 6422 until the cutoff day.

We start with sliding window integrated in figure 4.47. After day 513, the amount of prospective events must be set missing. In this period predicted values do not differ much from before. The picture is much different, however, if the value is set to zero instead of missing, as it squeezes down the figures. Before that, when data is still available, the performance is quite respectable. The RMSE is 124.94, representing an RRSE of 60.7%, which is comparable to the performance reached previously. But the final deviation is just 277.3 tickets that were sold less than predicted, coming down to just 1.37 per day. This remarkable result is probably due to fortunate circumstances as different phases are passed through more or less equally. Accordingly, there is the summer break where sales are underestimated, while the subsequent autumn is overestimated, compensating the backlog.

The results are rather similar when using the sliding window difference approach, as depicted in figure 4.48. There, the RMSE and the RRSE are a little lower with 118.92 and 57.78%, respectively. The final deviation is -3,712.26 tickets, however, as the compensation of over- and underestimation turns out more unfavourable in this case. The model remains stable as well after the production schedule ends.

Both sliding window approaches suffer from lacking training examples for long-term horizons as we have seen in figure 4.43, making such predictions rather vague and to be treated with caution. Due to their architectural similarities many characteristics are rendered the same, such as the rise from day 400 to 450 or the small drop between days 480 and 500. Interestingly, the overall sales level is rather different though, emphasizing uncertainty.

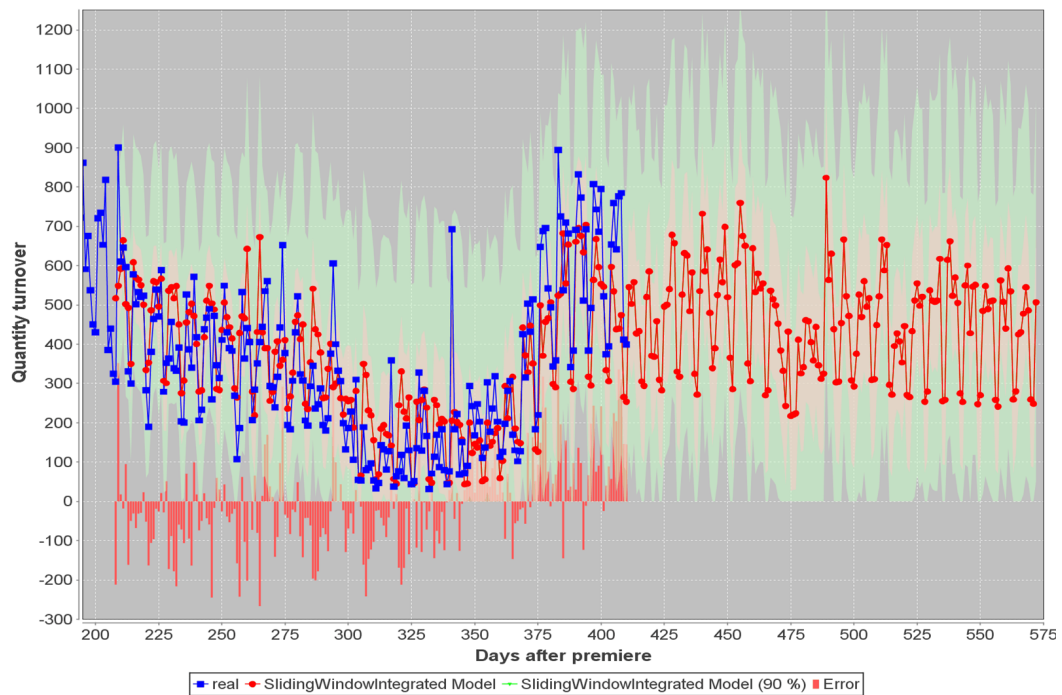


Figure 4.47: Sales figures forecast to decide upon the extension of production 6422 using the sliding window integrated approach.

We now apply the micro-founded approach, where uncertainty of long-term prognoses results from the accumulated uncertainty of long-term event forecasts. The prediction is limited to the duration of the production schedule here, which is 306 days. The result can be seen in figure 4.49. Performance is much worse than before as the RMSE is 210.57 and the RRSE 102.3%, or 19,663.6 tickets final deviation. It is visible that the summer break sales are kept up quite well, but as soon as this break is over, approximately at day 360, actual sales are much lower than predicted. We increased the window of observation to cover a longer period in this case to demonstrate that those figures are far from being visionary, but revert to levels reached previously. The algorithm performs a conclusion by analogy – it assumes unaltered circumstances and draws its predictions despite the fact that market saturation has indeed set in, leading to this overestimation. We are going to discuss this issue later in chapter 5.1 as it is relevant for all domain questions. Despite this, sales steadily decline towards the end as the production schedule is incorporated structurally.

Compared to the other approaches, predictions are rather different here, even contrary. So can we provide an answer to the question of the extension in the end? At least in this case it would be certainly not satisfactory, despite the good accuracy achieved by the first approach.

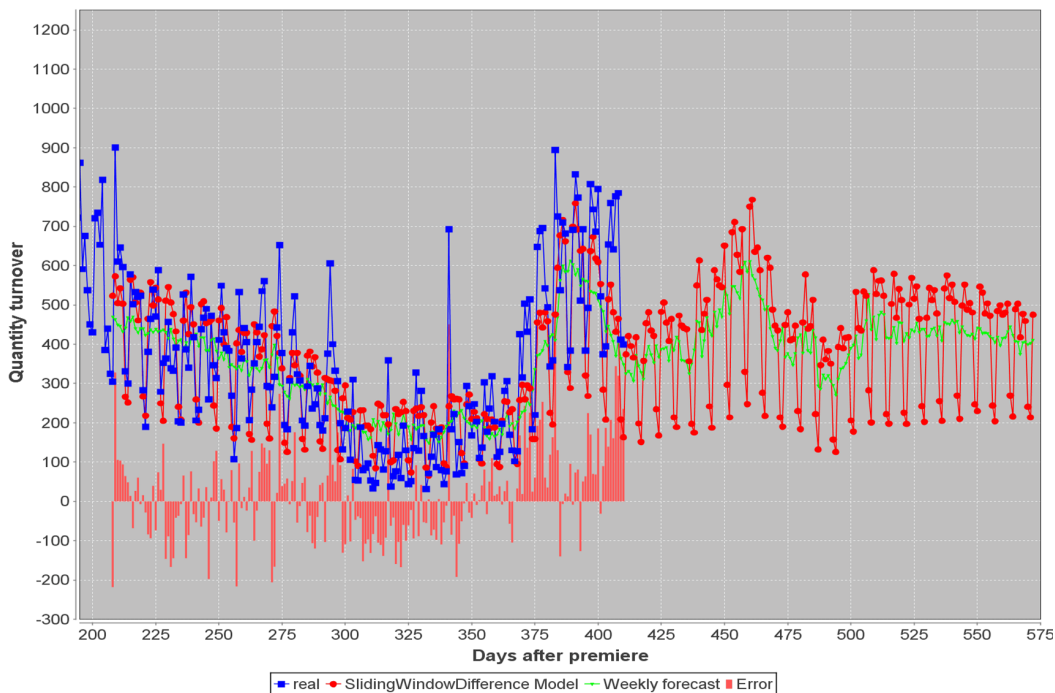


Figure 4.48: Sales figures forecast to decide upon the extension of production 6422 using the sliding window difference approach.

SUMMARY

By using the approaches to predict production sales figures as presented in DQ5, only insufficient results concerning this domain question can be obtained. The reason is that decisions about the extension are usually made far in advance, requiring long-term predictions which heavily suffer from uncertainty. This results in a great instability and great variability of the numbers returned even if the model or data base configuration is altered just slightly. As a consequence, we recommend using a different approach to address this issue, or at least incorporate some information about market potential in order to be aware market saturation effects.

4.7 Domain Question 7 – Success of a Production

Classifying the success of a production at an early stage is comparable to the vague event capacity utilization rate prediction introduced in chapter 4.2, as the information situation is rather scarce there as well. Similarly, one could make use of miscellaneous qualitative attributes about the production itself, for instance whether it is an in-house production or whether it is a remake of a production already played. Further, information about the prominence and popularity of the leading actors could prove useful, the production costs or the success of the underlying base works. Also marketing-related characteristics could contribute effectively such as marketing costs and activities, presence of other establishments performing similar productions, market

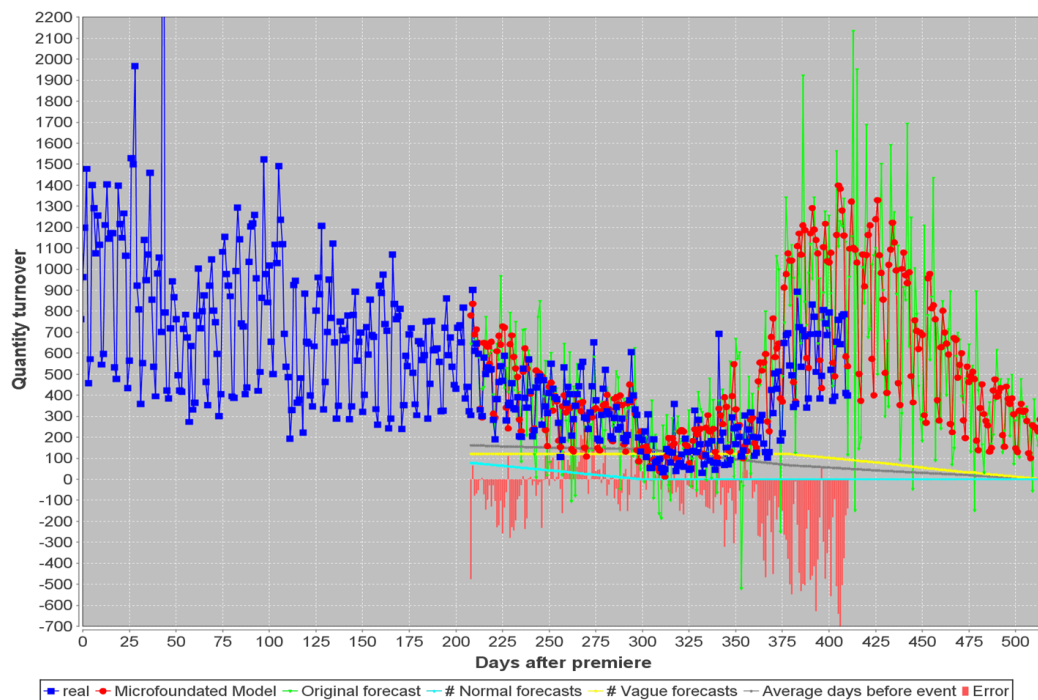


Figure 4.49: Sales figures forecast to decide upon the extension of production 6422 using the micro-founded approach.

potential figures, a differentiation in terms of target audiences and many others. This approach is quite similar to what was done in Stimpert et al. [2011] and Dhar and Chang [2009] to predict cinema and music sales.

In addition to this, we could again make use of tickets already sold as far as they are sufficiently representative. Hence, it would be interesting how early how many tickets have been sold, the categories they have and other features. In this context, the focus could be put on the buying behavior of key accounts and big partners buying large amounts of tickets, for if they get involved very early one could conclude that they expect the production to become a success and ensure that they get enough tickets. In this way, we could take over their planning and forecasting efforts quite easily which could turn out effective, yet risky. Despite that, such sales would have to be reclaimed as they were filtered out in the beginning because they do not represent end customer behavior.

Of course, a reasonable target variable needs to be defined as well. This could be for instance the average utilization rate of all events, the average turnover per event or total sales as in Stimpert et al. [2011]. In this case, a production schedule or different scenarios thereof would be necessary just as in DQ6. Eventually, we can apply different algorithms on this configuration draft to create forecast models.

As we have seen multiple times already, there are eight major productions in our data base, which means that we could provide as much as seven training instances for each model, which is certainly not enough. But it is even more grave that we do not have qualitative attributes of

productions to the extent that would be necessary to implement an expedient analysis by far. As a consequence, data mining is the wrong means to address this question, given the data base that we have. If it encompassed such features, however, and quantities of productions were bigger, it could be, just as it did for DQ2. Hence, we need to reference to alternative approaches such as the bass-model for instance, or the spatial diffusion-based approach as briefly described in chapter 2.3, which would in turn require some other characteristics to be present, however.

4.8 Summary

In this chapter we addressed all domain related questions that are subject of this case study. We revealed all steps that are necessary in order to provide an answer, from deriving attributes and instances as the learning base, via building the models, comparing different solutions through to employing and interpreting the results. In the course of doing so, we also floated and verified several data related hypotheses.

All event related questions, DQ1 to DQ4 could be answered overly well. For DQ2 we could even prove the supremacy of our solution compared to the forecast approach that is used currently. Moreover, one question concerning the prediction of productions, DQ5, could be solved using our ML-NARX-based approach. We also developed a micro-founded method in this context, which makes use of predictions as defined in DQ2. The last two questions, however, could not be answered satisfactorily. The reasons for that, other shortcomings relevant in almost all cases and some general issues are subject of the following chapter.

Discussion and Limitations

In the course of answering the domain questions as far as feasible in the previous chapter, we evaluated the presented approach and its results in the respective section by providing comparisons of different kind. What remains for this chapter now is to discuss some supplementary issues that are relevant for all problems and putting the results into the context of using them on a global level.

5.1 Analogism and Quantitative Limitations – Concept Drift

In the course of the case study conducted in DQ6, a circumstance was recognized that needs to be borne in mind in all situations when making use of data mining. That is its inability to appropriately react to novel situations. We pointed out in chapter 2.1 already that past data and the contained situations and constellations are the only source to learn and eventually derive conclusions for classification and regression tasks. In DQ6, the novelty and unknown concept was the market saturation that set in, but it could also be any other kind of concept drift – i.e. a change in prevailing circumstances and structures. This inability was especially observable at the micro-founded approach. It equally holds true for the sliding window variants which, however, might even partially contain market saturation information as long term models are based on productions that have been played for a long time already.

An algorithm should in general be able to recognize such aspects up to a certain degree implicitly from data, for instance by relying on proportionally less early bookings or any pattern that might not even be known. It universally applies that the more training instances are available, the better and the more likely it is to detect and reproduce such nonlinearities and nonstationarities. This is a big issue for our case study because our data base does in fact not provide the quantities that would be necessary for a big data project. Indeed, it consists of three million tickets sold, but the quantities of transformed aggregations used for the respective question are much lower, and this is what counts in the end. Hence, there are about 3,000 instances for the models of DQ1 to DQ4 and about 5,500 for DQ5 and DQ6, depending on the horizon.

Nevertheless, we could observe that patterns are exploitable sufficiently to allow for suitable predictions to be drawn. Still, in order to detect patterns of high complexity in high-dimensional attribute space like market saturation effects, data demands are higher by several orders of magnitude. We have also seen this when performing feature-selection.

An alternative to bypass this quantitative data demand is to directly incorporate such factors and hence increase data quality. Nevertheless, this not necessarily simplifies matters. For this purpose, influences that are assumed to cause such drift could be made explicit by adding them as parameters. Sticking to the market saturation example we could add the general market potential of a production, the probability of a customer to visit the performance twice or even more often, etc. From a different point of view, incorporating such factors represents a decrease in uncertainty, for these inputs explain and reduce the remaining noise and deviations of the predictions. Statistically speaking the coefficient of determination is increased.

Anyway, all predictions draw a picture that is sufficient to get an impression of future sales under the condition that circumstances do not change considerably, e.g. market saturation does not become an issue yet. As long as the respective observer is aware of that, benefit can still be drawn by mentally multiplying factors that are unknown to the model. In the end, the very goal is to reduce the necessity of this workaround. With this in mind we want to stress once again that our predictions can never serve as the ultimate picture, for they a) ignore many important influence factors and b) they address the real problems in a rather concise way only, as a more precise definition of the objective is necessary *ex ante*. They rather serve as an indication about future developments that can be used for informed decision making.

5.2 Control Measure Interference

There is another problem that is equally important and universally valid. It is a result of control measure interferences and ultimately attributable to missing input parameters to represent them and feed them into the model.

Let us draw an example for illustration purposes. The sales department identifies a set of events which fall short of expectations by exhibiting an utilization rate that is much lower than of comparable events in comparable situations. As a reaction, measures will be taken to correct this condition such as launching promotions, issuing coupons and other means to drive sales of these events. It is left undecided whether this goal is accomplished or not, but what counts is that sales from this point in time onwards differ from “ordinary” sales in terms of causality and their properties by implication, for they are a result of deliberate control.

Such control measures can also occur in rather different situations. A real-world example is the substitution of the leading actor of a production with an actor popular among the target audience in order to drive sales. In other situations, marketing efforts are simply increased. We also encountered that in the form of free tickets in the very beginning already.

For an approach based on data mining such actions represent disturbances which are practically undetectable and difficult-to-treat in most cases. The reason is that the only input an algorithm gets is the sales that actually happened, and not their causes. For the example depicted in the first section this means that as soon as a poorly performing event is identified, the model assumes that sales accelerate from a certain point on, when such measures are usually

taken, for it has always been like that. A possible indication of such situation could be derived when looking at intercept points of the typical sales trends depicted in figure 4.3.

We need to keep in mind that everything the model predicts is based on the assumption that behavioral patterns do not shift. This is exactly the point of contact of both problems, for in the end conclusions can only base on parallelisms. The example of stock price prediction depicted in the introduction of chapter 2.1 bears some similarities to that as well as there it is the predictions themselves that change behavioral patterns. Given that our predictions are exploited in the end, we will also be affected by that.

A solution to overcome this problem partially, yet effectively, is to create models that make use of the monetary equivalents of the quantitative numbers used so far as prediction targets. For event predictions this would be the financial occupancy rate, i.e. the actual turnover against the maximum limit determined using the list price and maximum capacity of the different price categories. For production sales predictions the quantitative turnover is simply replaced by the monetary turnover. The input space for all problems could then be extended by this monetary dimension as well. Especially for production related questions, retaining quantitative forecasts could turn out to be appropriate, however, as word-of-mouth and its multiplier effects must not be neglected. As a result of this measure, control interferences caused by discounting can be masked out. Due to its importance, this represents a high-priority issue for further developments. Alternatively, also the attribute space might be extended to cover factors like active marketing measures, which are treated just below.

5.3 Expert Discussion and Feedback

In order to augment our evaluation that was only quantitative so far in a qualitative way as well, we arranged two discussion rounds with representatives of five specialty departments – sales, finance, marketing, controlling and press – to present our results, discuss them and get some feedback. The atmosphere was very positive and results, especially the comparison between the old and new approach of DQ2 were embraced.

The focus at these discussions was the integration of various exogenous sources, of which some were mentioned already. In this context, assessing economic efficiency by taking a look at the cost-benefit relation of each of them is of great importance. Methodologically this means to construct a hypothesis concerning the influence in the first place. This hypothesis is then to be verified using some data and following an approach that is comparable to what has been shown to assess the influence of the Google Trend. However, it is important to keep in mind that data dredging is a fallacy that might be encountered rather quickly in doing so.

Returning to our problem of market saturation, data acquisition involves a great deal of expense, let alone that it is almost impossible to obtain figures for productions of the past. Hence, only use data of present and prospective productions could be used, drastically reducing the amount of training observations. The picture looks different for data from Google Analytics of the homepage of the establishment however, as the cost-benefit ratio is more favorable there. One could for instance easily assess the influence of click rates per day and qualitative attributes as the exit pages or duration of stay, as they are extractable with low effort and available far

into the past. Apart from this, these figures could also be used to ascertain market potential in a limited way.

Marketing activities were another discussion issue. They are available in an extensive form as data of a media observation institute have been collected long since, capturing media penetration of each measure. Hence, we know the production and medium concerned, a coarse classification of the configuration of the measure itself, its duration of effect and duration of validity, i.e. the events concerned. As a consequence, finely-grained information can be provided, leading to the possibility to assess the effectiveness of each form, for instance a newspaper ad of different sizes or with and without a picture. It remains theoretical, however, as we again lack data quantities and many such activities were performed just once at all. Instead, we could perform an aggregation just as for the pre-sales intervals of DQ1 or the past weeks of DQ5, by considering the total media penetration of all campaigns currently active as well as a rough capture of the media concerned. Hence, the influence in general can be captured, making the control measure explicit. Just as done at the micro-founded model, one could then simulate and assess the effectiveness of different marketing action scenarios.

Also the discount system could be extended to capture deliberate interferences. Therefore, a classification that is finer than in TICKET_Ermäßigungsgruppe, which basically distinguishes between full price and reduced price tickets, and coarser than in TICKET_Ermäßigungsbezeichnung, containing 1,499 different values, as well as the duration of effect would be necessary. Some further ideas were born during discussion such as gathering data about activities of competing establishments, press commentaries or demographic data of the customer, e.g. whether it is a family, the age and the like.

5.4 Generalizability

As we remember, the aim of this thesis from a scientific point of view is to work out a general methodology for a data mining project that can be applied in a similar context. The previous chapter serves as template for that, representing a prototypical approach for each respective domain related question. In this subchapter we want to go into detail of this generalization idea and examine how the proceedings shown could be applied to such similar applications.

In the first step we assume an opera house, i.e. a repertory institution that wants to employ such forecasting system. The data base format therefore is comparable to ours, which is not unlikely, or at least could be achieved with minor effort. Also, the questions to be answered shall be similar. Firstly, the general preparation steps are reapplied without major changes. A difference could be, however, that contingent- and partner sales make up a larger proportion than in our case, resulting in the need to keep them in the data base for they could contain interesting and useful patterns. Hence, the focus is shifted away from end customer behavior. Alternatively, one could also try to “convert” these sales by breaking them down according to predefined rules, leading to a huge increase of complexity though.

Further, the fact that the production schedule differs considerably from ours, that follows an en-suite style, leads to several adaptations concerning the domain questions. Accordingly, despite the retention of the hierarchy between events and productions, the latter are not organized in distinct seasons but in groups of a few events occurring multiple times a year, possibly even

overlapping with other plays. As a consequence, DQ5 drops away completely. For the others, both concepts could be linked together more or less. We could for instance extend DQ1 and perform a clustering of productions as well using the typical median sales trend for each of them. Of course, a preliminary analysis would be recommended to determine if such trend even exists or whether it is independent from a production. Moreover, due to the fact that a play has most likely been performed already, the success of such short series and the respective events can be predicted more effectively. Accordingly, DQ2 could be adapted to only consider sales data of the same production if enough data is available or put more weight on them if not, and DQ7 could now even be answered at all in analogy to DQ2.

The basic procedure to create instances by transforming the input base would remain the same, for the goal is to express the target variable, regardless of whether quantitatively or monetarily, by the characteristics occurring at a specific point in time. Depending on the availability, appropriate attributes are to be derived either way. The steps following this preparation, e.g. attribute selection or the choice of the best performing algorithm can be conducted in analogy to here as well.

However, our methodology may also be applied to cinemas for instance, which basically differ substantially from theatres or opera houses, but not from a technical perspective concerning the application of data mining. A production is now called “movie” and performed multiple times a day for a duration of several weeks, usually unrepeatably. Unlike the opera context, questions related to productions are relevant now, for forecasting single events could turn out to be difficult as tickets are sold on the same day, yet even just before the show starts in most cases. In turn, DQ6 and DQ7, i.e. deciding how long a movie should be shown, could be even more interesting.

On the other hand, making conclusions based on sales data of other films might be fraught with problems as the target audience is much more inhomogeneous now. Hence, it is necessary to select films to be used as training examples based on several characteristics, such as romanticism or creepiness, more or less representing the approach followed in Žliobaitė et al. [2009] and Žliobaitė et al. [2012]. Furthermore, sales related information will probably be sparser, making it even more necessary to fall back on film related data. In the end, however, the data base will be bigger in quantitative terms as the frequency of events is higher.

By taking a step back, one could also consider a totally different area, such as hotel business, for there, sales time series and occupancy rates need to be predicted based on bookings as well.

In this brief glimpse we could point out that the methodology proposed allows for a huge flexibility as steps are arbitrarily adaptable, replaceable and combinable depending on the respective requirements. While technical circumstances change, technological proceedings remain the same as we still follow the CRISP-DM procedure model. The same goes for data base extensions in our context. New information such as marketing activities discussed before could be incorporated in analogy to Google Trend. Also, swapping the target measures from quantitative to monetary measures would not have a big impact on the methodology.

Conclusion

6.1 Summary

In this feasibility study we were able to demonstrate that data mining is a suitable technique to address the domain related questions in essence. This especially goes for DQ1 to DQ5 where expectations were even exceeded. On the other hand, DQ6 and DQ7 could not be solved in a satisfactory manner using the data that is available. In the following, we will provide a short achievement-summary for each question, mainly leaving out methodological issues as they were already summarized at the end of each section in the methodology.

DQ1 Is it possible to reduce the sales trends of single events to distinct patterns and how accurately can such pattern be predicted for an event prior to its performance? We demonstrated that separating reoccurring sales trend patterns is perfectly feasible. Therefore, k-means was found to be the optimal clustering algorithm. The resulting prototypes are seemingly allocated to the productions up to a certain extent, but when taking a closer look structural properties such as the time of year play an equal role. Forecasting the prospective trend pattern of events that have not been performed yet works fine as well – one day before the show the accuracy is about 94%, one month before about 80% and two months before 66.5%. The wrapper approach for feature selection was confirmed optimal, selecting a very small subset only. The optimal classification algorithm was identified as the logistic regression.

DQ2 How accurate are predictions of the final capacity utilization rate of an event in general and how does the confidence change over the time? The data base prepared for DQ1 could perfectly be used for this question as well as it covers past sales figures, properties about the event, some characteristics of the tickets already sold and some further indicators. In this regression setting, the bagged tree and support vector machine with PUK-kernel were performing best. We showed that substituting the regression with different classification approaches leads to an inevitable loss of performance. The average

absolute deviation in terms of percentage of utilization rate is 2.3 when predicting at the day of the performance, 5.75 for one month before and 6.9 for two. Using this approach and by appropriately specifying the target variable, even the whole prospective sales trend can be projected, just as in DQ1, but more fine-grained. Compared to the approach currently used in the establishment, our data mining based solution improves the result by 1.2 to 10.8% depending on the horizon, again in absolute terms.

- DQ3 What are the influencing factors that play an important role for these problems?** The identification of influential factors and their effects was shown to be nontrivial. Using the selected feature sets of several filter and wrapper approaches, we found out that indications about previous sales figures play an important role for both questions. However, for the former nonlinear relationships cannot be regarded, while for the latter influence directions are concealed. Instead, by looking behind the scenes of some simple algorithms themselves, sometimes a clearer picture about relationships and interdependencies can be obtained. We illustrated this by means of a simple decision tree and the linear regression.
- DQ4 How do predictions of sales trends and final capacity utilization rates look like when breaking them down to the individual price categories?** By reducing the variety of price categories and condensing them to their roots, a simple data set selection can be performed to obtain the tickets of a specific category only. With this result, all previous steps and also those following can be reapplied with almost no adaptations necessary, resulting in a breakdown of total sales to the respective category. We could observe that homogeneity varies and just one category could be predicted better than the model that uses all of them.
- DQ5 How can sales trends of entire productions be predicted and can simple prediction approaches be outperformed?** Predicting sales figures of running productions can be done by using a time series approach which we solved by means of data mining. We proposed the idea of **ML-NARX** therefore, that is a model that contains an autoregressive and an exogenous part, calibrated using a machine learning algorithm. On closer consideration, also DQ1 and especially DQ2 turned out to be a time series approach if not that proper. We presented two such sliding window variants, one predicting daily values inherently and one by applying intra-week differences onto a weekly prediction. Both performed more or less the same, outdoing classic approaches such as moving average or exponential smoothing by far. For them, the daily deviation reaches from 110 to 125 tickets in the first three weeks on average, while they add up to 170 to 215 for the classical ones. In addition to this and not related to time series, we proposed a **micro-founded approach**, which draws individual event predictions à la DQ2 to aggregate them and hence obtain a picture on the macro-level, i.e. the production. Due to its complexity and time requirements we had to drop an extensive evaluation for it, however.
- DQ6 Is it possible to provide decision support for questions concerning the extension horizon of an ongoing production?** In order to answer this question we applied the concepts proposed in DQ5 to a real world problem. We emphasized the need for a production schedule therefore, as all approaches, but especially the micro-founded one, make use of that information. Both sliding window approaches drew a picture that was rather similar

concerning the characteristics, but the total sales level was quite different. While sales remained stable beyond the schedule, they gradually declined in the micro-founded solution due to its nature. In the end, forecasts were too inhomogeneous to draw a clear conclusion from them, leaving this question unanswered.

DQ7 How can the success of a production be predicted very early in its life cycle, possibly even before its premiere? Answering this question turned out difficult as well. We briefly pointed out that in order to cast such prognosis, quantitative data demands are much higher as we only have eight major productions to derive conclusions from in our base up to now. Additionally, we would need many different characteristics about them, which are not available at all. Hence, data mining turned out to be unsuitable in this context.

In the course of working out solutions for these questions following the CRISP-DM process model, we pointed out the steps that are necessary in general and the most important aspects that need to be considered. Concerning the technical questions, we for instance showed how transactional data can be transformed into a form that is appropriate for the respective problem (for instance ML-NARX), that classification substitutions of regression problems are likely to deliver inferior results, that normalization does not necessarily lead to an increase in performance and eventually how parameters and configurations can be compared and optimized.

These findings can be applied to any data mining project exhibiting comparable characteristics as we have pointed out with three brief examples. In general, the Pareto principle was found to be valid as minor effort can effect results that are suitable already, but efforts required to obtain a result that is virtually optimal are practically unlimited.

In our context, we found two technical issues to be crucial and difficult to implement when preparing the data for data mining. The first concerns the fact that a considerable proportion of sales in our data base is due to key accounts and partner companies which buy huge amounts of tickets in batches. Although these tickets are sold to end customers eventually, we lose some important information. As our ultimate goal is to predict end customer behavior, these sales are irrelevant for us – they even need to be dismissed as they distract behavioral patterns. However, identifying them is nontrivial due to the structure of the data base.

The second issue concerns sales which are the result of non-natural circumstances, i.e. a deliberate manipulation of the buying behavior which exceeds usual marketing efforts such as billboard or newspaper advertising. We pointed out the problems that result from such control interferences, as it distorts relationships identified by a data mining approach by way of breaking up natural processes – a form of concept drift. As a consequence, events that underperform might be assumed to accelerate inherently due to the measures taken in the past, for instance. However, as long as one is aware of this, useful insights can still be gained.

We became witnesses of another manifestation of concept drift in the course of DQ6 by the example of market saturation. But certainly there are some others hidden in the models as well. We pointed out that observation quantities are insufficient to detect such complex adaptations, as explicit indicators are not available. Moreover, numbers are too low to even appropriately detect coherences between features that are available. This finding is based on the fact that optimal feature subsets contain a handful characteristics only, improving the performance in many cases. Hence, it can be concluded that current data availability suffices for simple problems

and analyses as shown here, even yielding remarkable results, but in order to issue sound and professional recommendations ready for the market, a quantitative increase is highly desirable.

From a technological point of view we want to stress that data mining still is a rather vague and elusive technology. This especially comes into the picture when facing the instability of many solutions, which is due to the sensitivity of the algorithms involved. Additionally, the possibilities of realizations are vast – be it the general data representation of the problem, or more specifically the definition of attributes, filtering of observations, or the combination of data sets for model building. To find the ideal configuration is an NP-hard optimization problem which can be effectively solved by means of heuristics only. Even though the basic procedure was formalized making it easier to reproduce such approach, we cannot provide a solution to the problem in that respect. Hence, the core principles of applying data mining remain what they are – art as much as science.

6.2 Future Work

The core purpose of this feasibility study by its very nature is to pave the way for further steps. Although the whole data mining process is pervaded and driven by ideas, many of them could not be implemented due to time constraints. These primarily include transforming all problems to use monetary dimensions as targets to reduce the impact of control measure inferences. Another ambition is to increase the size of the data basis to include further productions. As we have seen, there are some available which can be used without any considerable conversion effort due to different structural properties, but were held back by the establishment deliberately. Further, in order to improve the accuracy of forecasts by increasing the coefficient of determination and to prevent concept drift, extending the data base by exogenous influence factors is a logical consequence. We already visualized some examples in chapter 5.3. In this respect, it is important to keep in mind that any such extension leads to higher quantitative data demands.

One source that has not been analyzed here are social media. This widely available, extensively used and simple-to-tap information-source has been employed for marketing purposes for years now and is rather convenient in this context as well. For instance, one could use quantitative numbers such as the amount of tweets or blog entries of a specific topic, the amount of likes or comments made on Facebook or the amount of bookmark entries. However, also qualitative measures are possible such as a sentiment analysis conducted on the set of tweets or any other written material, as has been done extensively already, resulting in a barometer of public opinion. In general, there is a myriad of scientific work available about how social media can be used in a data mining context. As an example, Abel et al. [2010] shows how information extracted from blog posts can be used to predict the success of music or films, or even of books as done in Moon et al. [2010]. We have also seen such an example already in chapter 2.3, as Dhar and Chang [2009] uses blog chatter and the amount of Myspace friends to predict music sales. The range of applications is manifold.

By leaving our context, one could also think about changing the orientation from ticket sales to customers in order to predict behavioral aspects there. This is actually one of the most prominent fields of application of data mining. With this, data mining could be used to establish a recommendation system, to improve customer relationship management by enabling target

group specific measures, or also to expedite demographic analysis. In further consequence, a price discrimination system could be created as well, as it has been done by many airlines already. At last, we want to mention the application of approaches that have nothing to do with data mining at all in order to obtain answers to questions that remained unanswered.

Concerning the scientific continuation, any follow-on project could concentrate on improving and refining the models. Top candidates therefore would be the incorporation of new exogenous sources, but also the derivation of other attributes, the improvement of filtering mechanisms or optimization of algorithm parameters, which has been left out almost completely here. The result would be a methodology on a finer-grained level, again to facilitate the reapplication of the practices shown. On the other hand, concerning further steps of the forecast-project itself, a concretization of the questions as well as involving specialty departments and persons that are going to use the system eventually is advised. For at the end of the day the data mining approach should be turned into a professional decision support system. But this is a different story.

Data Description

This chapter takes a brief look at all attributes that are contained in the raw export data. There are more available in the ERP system, but these are the ones that are used for this study, proving more or less valuable in the end. Numeric attributes are accompanied with a boxplot and histogram, whereas nominal ones comes with a frequency diagram of the most common values given that there are not too many of them and no sensible information is concerned.

Table A.1: VORST_ID attribute summary.

VORST_ID		
Nominal	No missing	3,240 distinct
Simple random numeric identifier for a particular event. It is used for administrative tasks only, such as matching of data sets.		

Table A.2: KUNDE_Postleitzahl attribute summary.

KUNDE_Postleitzahl		
Nominal	867,150 (27%) missing	14,043 distinct
Contains the zip code of the respective customer. There are many misentries such as salutations, special characters, addresses, countries, phone numbers and a lot of obviously invalid codes. The reason for that is that this field is filled manually by employees or the customer himself, depending on the medium. Moreover, it is an optional field obviously.		

Table A.3: KUNDE_Land attribute summary.

KUNDE_Land																		
Nominal	1,089,380 (34%) missing	319 distinct																
<p>Just like the zip code, the country of the customer is recorded. Interestingly, this attribute has much more values missing. Those that are not are heavily multilabeled as country codes are used side by side with full country names. There are many misentries as well such as arbitrary labels. In about a thousand cases this field contains an email-address, indicating that there has been a change in the user interface or it has been misinterpreted. In the graph, GS stands for German speaking country.</p>	<table border="1"> <caption>Data for KUNDE_Land Distribution</caption> <thead> <tr> <th>Category</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>GS1</td> <td>88.27%</td> </tr> <tr> <td>GS2</td> <td>6.59%</td> </tr> <tr> <td>GS3</td> <td>0.66%</td> </tr> <tr> <td>JP</td> <td>0.17%</td> </tr> <tr> <td>SI</td> <td>0.16%</td> </tr> <tr> <td>IT</td> <td>0.13%</td> </tr> <tr> <td>Others</td> <td>4.03%</td> </tr> </tbody> </table>		Category	Percentage	GS1	88.27%	GS2	6.59%	GS3	0.66%	JP	0.17%	SI	0.16%	IT	0.13%	Others	4.03%
Category	Percentage																	
GS1	88.27%																	
GS2	6.59%																	
GS3	0.66%																	
JP	0.17%																	
SI	0.16%																	
IT	0.13%																	
Others	4.03%																	

Table A.4: KUNDE_Geschlecht attribute summary.

KUNDE_Geschlecht												
Nominal	1,439,934 (45%) missing	4 distinct										
<p>Gender of the customer, i.e. the purchaser of the ticket. If a purchase comprises multiple tickets, this field does not differ. “s” stands for contingent or similar purchases that do not allow for a determination of the gender. “f” and “w” stand for female and “m” for male.</p>	<table border="1"> <caption>Data for KUNDE_Geschlecht Distribution</caption> <thead> <tr> <th>Category</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>s</td> <td>53.09%</td> </tr> <tr> <td>w</td> <td>32.47%</td> </tr> <tr> <td>m</td> <td>14.44%</td> </tr> <tr> <td>f</td> <td>0%</td> </tr> </tbody> </table>		Category	Percentage	s	53.09%	w	32.47%	m	14.44%	f	0%
Category	Percentage											
s	53.09%											
w	32.47%											
m	14.44%											
f	0%											

Table A.5: KUNDE_Flag_Firma attribute summary.

KUNDE_Flag_Firma										
Nominal	309 (0%) missing	3 distinct								
<p>This is a simple indication for the customer being a company or an individual. The value “0” is an old representation of “y”.</p>		<table border="1"> <caption>Data for KUNDE_Flag_Firma Distribution</caption> <thead> <tr> <th>Value</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>n</td> <td>72.7%</td> </tr> <tr> <td>y</td> <td>26.92%</td> </tr> <tr> <td>0</td> <td>0.38%</td> </tr> </tbody> </table>	Value	Percentage	n	72.7%	y	26.92%	0	0.38%
Value	Percentage									
n	72.7%									
y	26.92%									
0	0.38%									

Table A.6: POS_Hierarchie0 attribute summary.

POS_Hierarchie0												
Nominal	10,346 (0%) missing	4 distinct										
<p>The point of sale information is structured in a hierarchical way, where this attribute represents the top aggregation.</p>		<table border="1"> <caption>Data for POS_Hierarchie0 Distribution</caption> <thead> <tr> <th>Value</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Self</td> <td>67.86%</td> </tr> <tr> <td>Direct partner</td> <td>12.85%</td> </tr> <tr> <td>Self online</td> <td>9.67%</td> </tr> <tr> <td>Extern</td> <td>9.61%</td> </tr> </tbody> </table>	Value	Percentage	Self	67.86%	Direct partner	12.85%	Self online	9.67%	Extern	9.61%
Value	Percentage											
Self	67.86%											
Direct partner	12.85%											
Self online	9.67%											
Extern	9.61%											

Table A.7: POS_Hierarchie1 attribute summary.

POS_Hierarchie1		
Nominal	10,346 (0%) missing	6 distinct
<p>This second level of the point of sale hierarchy contains six distinct attributes. For the sake of anonymization, we refrain from revealing the value distribution here.</p>		

Table A.8: POS_Hierarchie2 attribute summary.

POS_Hierarchie2		
Nominal	10,346 (0%) missing	14 distinct
<p>This third level of the point of sale hierarchy contains 14 distinct attributes. For the sake of anonymization, we refrain from revealing the value distribution here.</p>		

Table A.9: TICKET_Ermäßigungsnummer summary.

TICKET_Ermäßigungsnummer		
Nominal	4,325 (0%) missing	8,061 distinct
Identification of the discount of a ticket. In this context, discount is to be seen as an umbrella term for all price categories, also those that have no discount at all. Each discount number is valid for one event only. It is used for data set matching.		

Table A.10: TICKET_Ermäßigungsbezeichnung attribute summary.

TICKET_Ermäßigungsbezeichnung		
Nominal	15 (0%) missing	1,935 distinct
The label of the discount, condensed in groups. The most common ones are the main categories one to seven, full price, miscellaneous campaigns of collaboration partners, early bird discounts, subscriptions, Mother's Day, warm up events and the like.		

Table A.11: TICKET_Ermäßigungsgruppe attribute summary.

TICKET_Ermäßigungsgruppe		
Nominal	209,121 (7%) missing	5 distinct
A grouping of the allowed discount. As already mentioned, full price is also treated as discount group.		

Table A.12: TICKET_Preiskategorie_ID attribute summary.

TICKET_Preiskategorie_ID		
Nominal	No missing	695 distinct
Identification of the price category of the ticket. Price category and discount are two rather similar concepts that have also been mixed up regularly in the past. This, however, is not that obstructive for our purpose. In essence, the price category represents a coarse classification of tickets on the basis of their price, whereas the discount considers the kind of the final price and how it is made up. Again, this identification is used for matching only.		

Table A.13: TICKET_Preiskategorie attribute summary.

TICKET_Preiskategorie		
Nominal	No missing	105 distinct
<p>This attribute holds the label of the price category. As there are less distinct values, categories with the same name may be of a different kind. The most important categories are “Blau” (blue), “Gelb” (yellow), “Grün” (green), “Rot” (red), “Rosa” (pink), “Orange” (orange), “Stehplatz” (standing) and special seats. For the colors, there are also many different sub-categories such as the discount rates or price ranges. Blau has 12 different sub entries of that kind for instance. There is also a considerable amount of multilabeling in this attribute such as “Stehplatz/Stehplätze” or textual appendices for the color categories.</p>		

Table A.14: PREIS_Platzkapazität attribute summary.

PREIS_Platzkapazität		
Numeric	No missing	Mean: -
<p>Each price category has a certain amount of seats at each event, resulting in this capacity attribute.</p>		

Table A.15: PREIS_Verkaufte_Tickets attribute summary.

PREIS_Verkaufte_Tickets		
Numeric	No missing	Mean: -
<p>The amount of seats of the price category concerned that have been sold for a specific event in the end.</p>		

Table A.16: TICKET_Vollpreis_It_Liste attribute summary.

TICKET_Vollpreis_It_Liste		
Numeric	3,442 (0%) missing	Mean: -
<p>The list price of the ticket.</p>		

Table A.17: TICKET_Tatsaechlicher_Preis attribute summary.

TICKET_Tatsaechlicher_Preis		
Numeric	13,642 (0%) missing	Mean: -
<p>The price that was actually paid for the ticket. Interestingly, there are some cases where this price is higher than the list price. This is attributed to errors in acquisition and technical peculiarities that we will not discuss here.</p>		

Table A.18: TICKET_Reihe attribute summary.

TICKET_Reihe		
Numeric	No missing	Mean: -
The row of the respective ticket.		

Table A.19: TICKET_Sitz attribute summary.

TICKET_Sitz		
Numeric	No missing	Mean: -
The number of the seat in the respective row of a ticket.		

Table A.20: TICKET_Verkaufstyp attribute summary.

TICKET_Verkaufstyp		
Nominal	5 (0%) missing	6 distinct
A classification of tickets in reference to the vending type that we will not explain due to technical reasons.		

Table A.21: TICKET_Rechnungsnummer attribute summary.

TICKET_Rechnungsnummer		
Nominal	No missing	744,524 distinct
The invoice number that this ticket belongs to. Accordingly, approximately 4.28 tickets have been sold at once on average.		

Table A.22: TICKET_Buchungsdatum attribute summary.

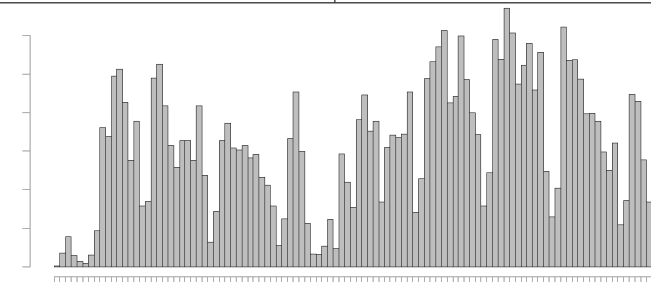
TICKET_Buchungsdatum		
Date	No missing	3,110 distinct
<p>This attribute holds the date of purchase. As already mentioned, its timespan represents approximately 8.5 years. In the diagram, each bar represents one month, allowing to get an idea of annual variations.</p>		

Table A.23: TICKET_Zahlart attribute summary.

TICKET_Zahlart		
Nominal	683,563 (21%) missing	9 distinct
The payment type this ticket was purchased by. Many of the missing values concern box office sales in early years.		

Table A.24: TICKET_Versandart attribute summary.

TICKET_Versandart		
Nominal	No missing	8 distinct
No matter how a ticket was purchased, this attribute describes the shipping method. As a consequence, the most common value is “cash”, which means that there was no delivery involved.		

Table A.25: TICKET_Kauf_Tage_vor_Vorstellung attribute summary.

TICKET_Kauf_Tage_vor_Vorstellung		
Numeric	No missing	Mean: 56.85
Date of purchase relative to the day of the performance. Some of the values are negative, as there is a considerable amount of tickets that are booked into the system belatedly. This attribute will be used to create time series for event related questions.		

Table A.26: TICKET_Kauf_Tage_nach_Premiere attribute summary.

TICKET_Kauf_Tage_nach_Premiere		
Numeric	84,970 (3%) missing	Mean: 207.32
<p>Date of purchase relative to the premiere of the respective production. Missing values come from small and short productions that do not have a premiere. This attribute will be used to create time series for production related questions.</p>		

Table A.27: KONTINGENT_ID attribute summary.

KONTINGENT_ID		
Nominal	2,869,798 (90%) missing	29,359 distinct
<p>The identifier of the contingent this ticket belongs to, provided there is one. A contingent sale is handled by a partner company. The majority of the tickets (indicated by a missing value), have been sold without a partner as a consequence. A contingent is valid for one single event.</p>		

Table A.28: KONTINGENT_Max_Tickets attribute summary.

KONTINGENT_Max_Tickets		
Numeric	2,870,366 (90%) missing	Mean: -
<p>Just like a price category, each contingent has a maximum number of seats that can be sold.</p>		

Table A.29: KONTINGENT_Verkaufte_Tickets attribute summary.

KONTINGENT_Verkaufte_Tickets		
Numeric	2,870,366 (90%) missing	Mean: -
<p>The amount of seats of the contingent concerned that have been sold for a specific event in the end.</p>		

Table A.30: VERANST_ID attribute summary.

VERANST_ID		
Nominal	No missing	55 distinct
Simple random identification of a particular production. Used for matching and administration tasks only.		

Table A.31: VORST_Veranstaltungsreihe attribute summary.

VORST_Veranstaltungsreihe		
Nominal	No missing	16 distinct
Identification of a yearly aggregation of productions into groups.		

Table A.32: VORST_Datum attribute summary.

VORST_Datum		
Date	No missing	2,118 distinct
This attribute holds the date of the event. As already mentioned, this timespan represents exactly 8 years.		

Table A.33: VORST_Wochentag attribute summary.

VORST_Wochentag																		
Nominal	No missing	7 distinct																
The weekday the event is scheduled.	<table border="1"> <caption>Weekday Distribution Data</caption> <thead> <tr> <th>Weekday</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Saturday</td> <td>19.05%</td> </tr> <tr> <td>Thursday</td> <td>16.5%</td> </tr> <tr> <td>Sunday</td> <td>15.72%</td> </tr> <tr> <td>Tuesday</td> <td>15.35%</td> </tr> <tr> <td>Friday</td> <td>15.34%</td> </tr> <tr> <td>Wednesday</td> <td>11.3%</td> </tr> <tr> <td>Monday</td> <td>6.75%</td> </tr> </tbody> </table>		Weekday	Percentage	Saturday	19.05%	Thursday	16.5%	Sunday	15.72%	Tuesday	15.35%	Friday	15.34%	Wednesday	11.3%	Monday	6.75%
	Weekday	Percentage																
Saturday	19.05%																	
Thursday	16.5%																	
Sunday	15.72%																	
Tuesday	15.35%																	
Friday	15.34%																	
Wednesday	11.3%																	
Monday	6.75%																	

Table A.34: VORST_Startzeit attribute summary.

VORST_Startzeit		
Nominal	No missing	9 distinct
The time the event is scheduled.		

Table A.35: VORST_Spielort attribute summary.

VORST_Spielort		
Nominal	No missing	16 distinct
An administrative capture of the venue of the performance. As we have seen there are just three physical theatres, so multiple identifiers represent the same one.		

Table A.36: VORST_Platzkapazitaet attribute summary.

VORST_Platzkapazitaet		
Numeric	No missing	Mean: -
The total capacity of an event, which of course depends on the theatre at which it is to be performed at.		

Additional Figures

B.1 Attribute correlations of DQ2

To provide an insight about the model data of DQ2, three correlograms of selected attribute sets are depicted here. All of them have the target value as the last variable. The graphs below the diagonal depict the simplified correlation curve, and the pie charts above the sign and strength of correlation.

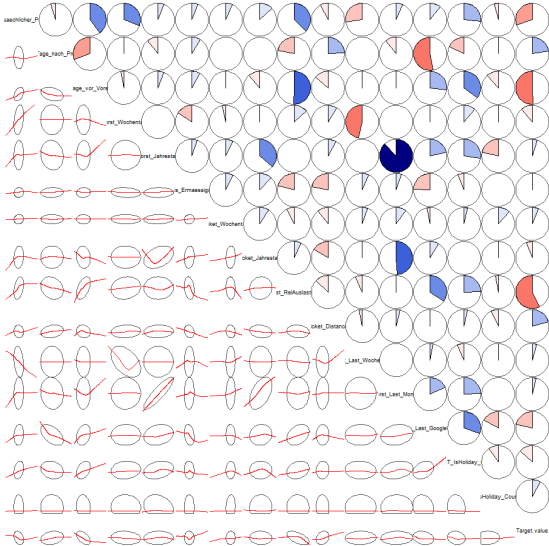


Figure B.1: Correlogram of selected qualitative attributes and the target variable of DQ2.

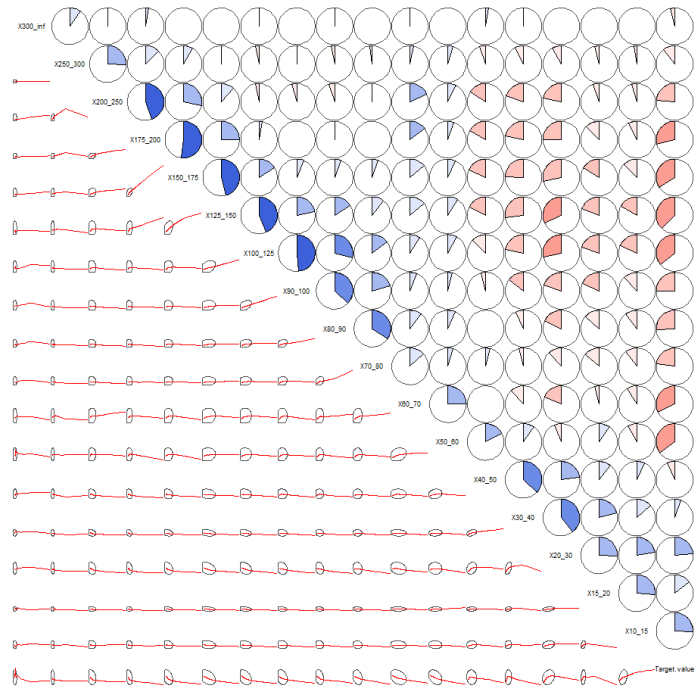


Figure B.2: Correlogram of the sales figures attributes and the target variable of DQ2.

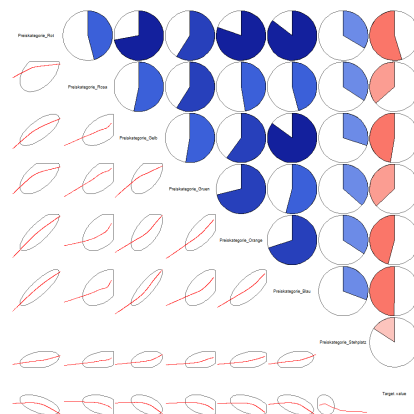


Figure B.3: Correlogram of price category sales data and the target variable of DQ2.

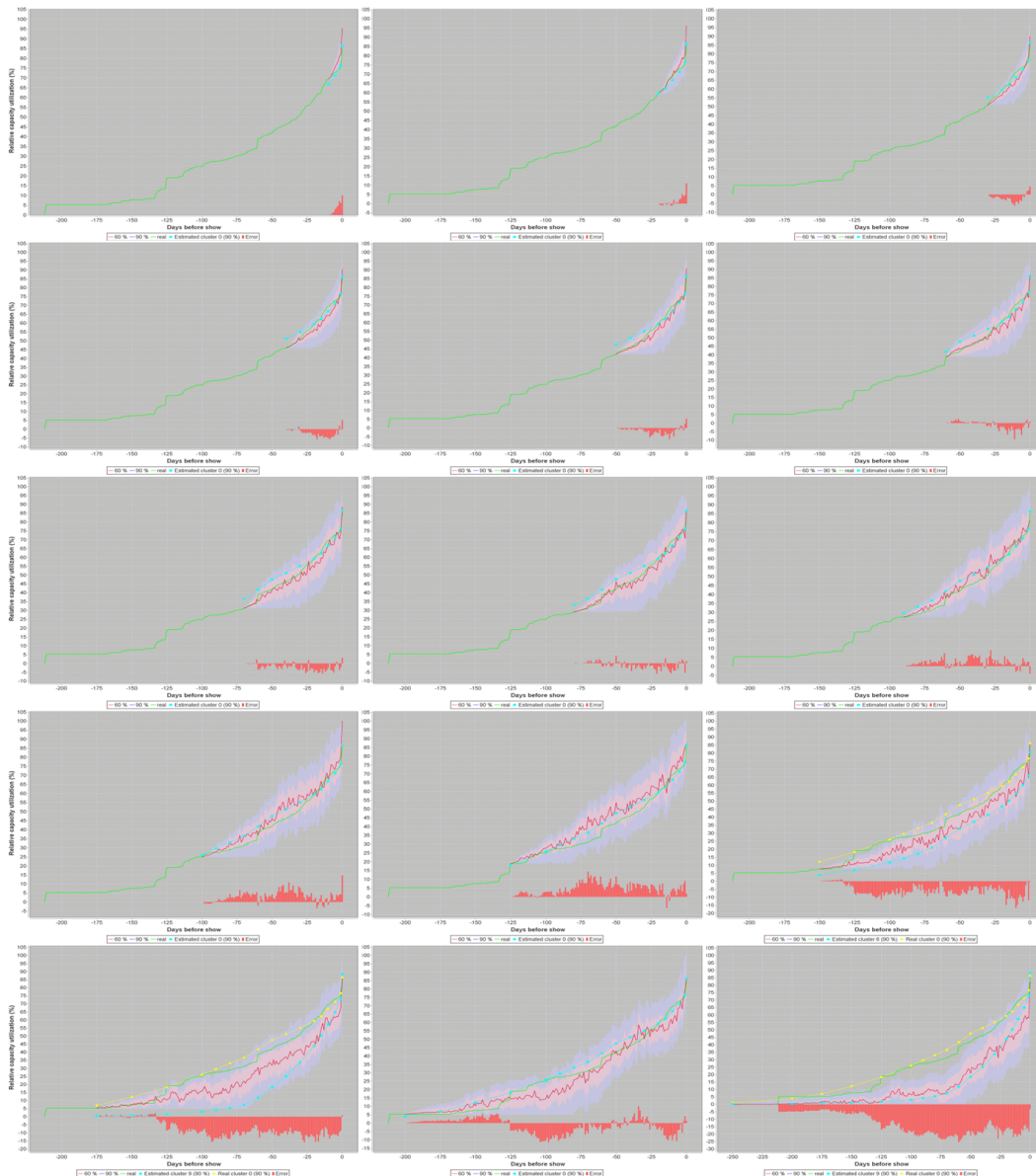


Figure B.4: Prospective sales trends of event 53412 using DQ1 and DQ2 models. Altogether, 15 different cutoff days are involved.

B.2 Capacity Utilization Rate Prediction at Different Cutoff Days

Figure B.4 displays the prospective sales trends using models defined in DQ1 and DQ2 for event 53412 of production 4652.

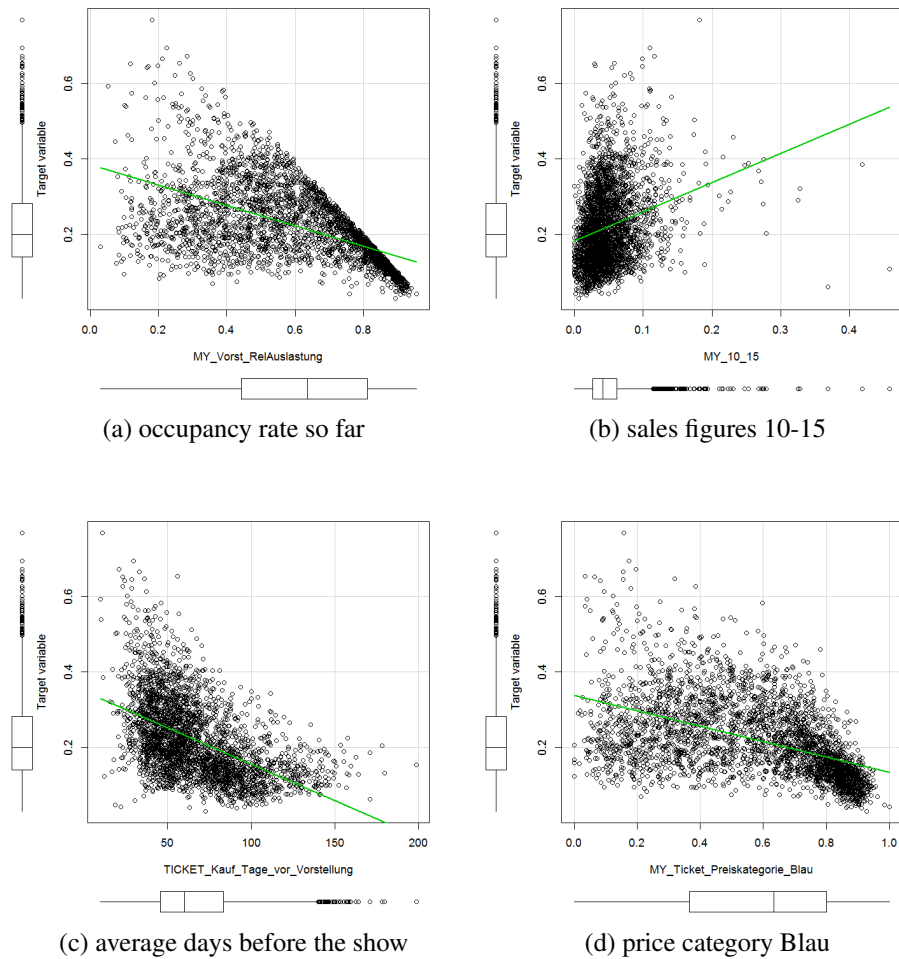


Figure B.5: Scatterplots of the most important attributes against the target of DQ2.

B.3 Scatterplots of Influential Attributes (DQ2)

The four scatterplots in figure B.5 draw the most influential attributes against the target variable.

B.4 Capacity Utilization Rate Prediction at Different Cutoff Days (Blau)

Figure B.6 again displays the prospective sales trends using models defined in DQ1 and DQ2 for event 53412 of production 4652, but now for the price category Blau only.

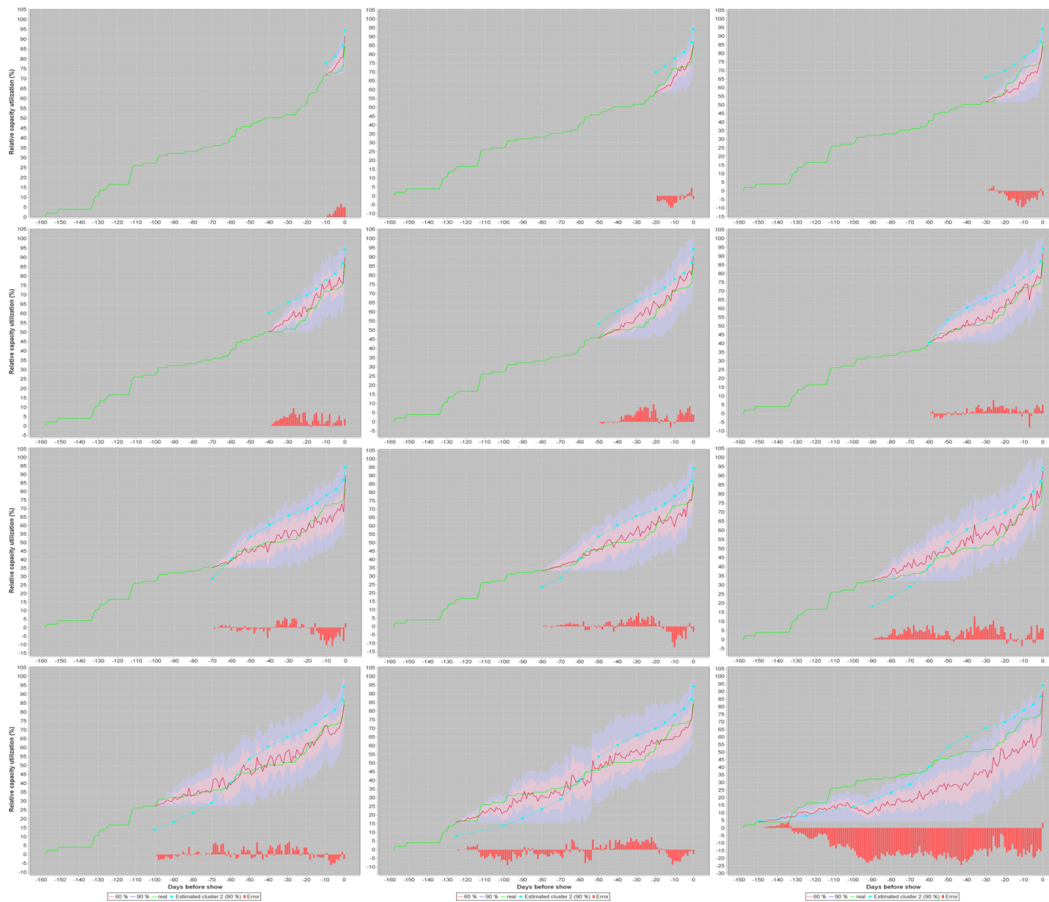


Figure B.6: Prospective sales trends of event 53412 using DQ1 and DQ2 models for price category Blau. Now, 12 different cutoff days are involved. The cluster is predicted correctly in all cases.

B.5 Sales Prediction with Sliding Window Integrated

Figure B.7 depicts a collage of six sliding window integrated forecasts of a length of 100 days at different cutoff days at production 4652. The respective RMSE values are 247.51, 339.19, 263.2, 153.04, 227.1 and 121.76.

B.6 Sales Prediction with Sliding Window Difference

Figure B.8 depicts a collage of six sliding window integrated forecasts of a length of 100 days at different cutoff days at production 4652. The respective RMSE values are 279.52, 352.92, 277.73, 176.6, 222.18 and 126.97.

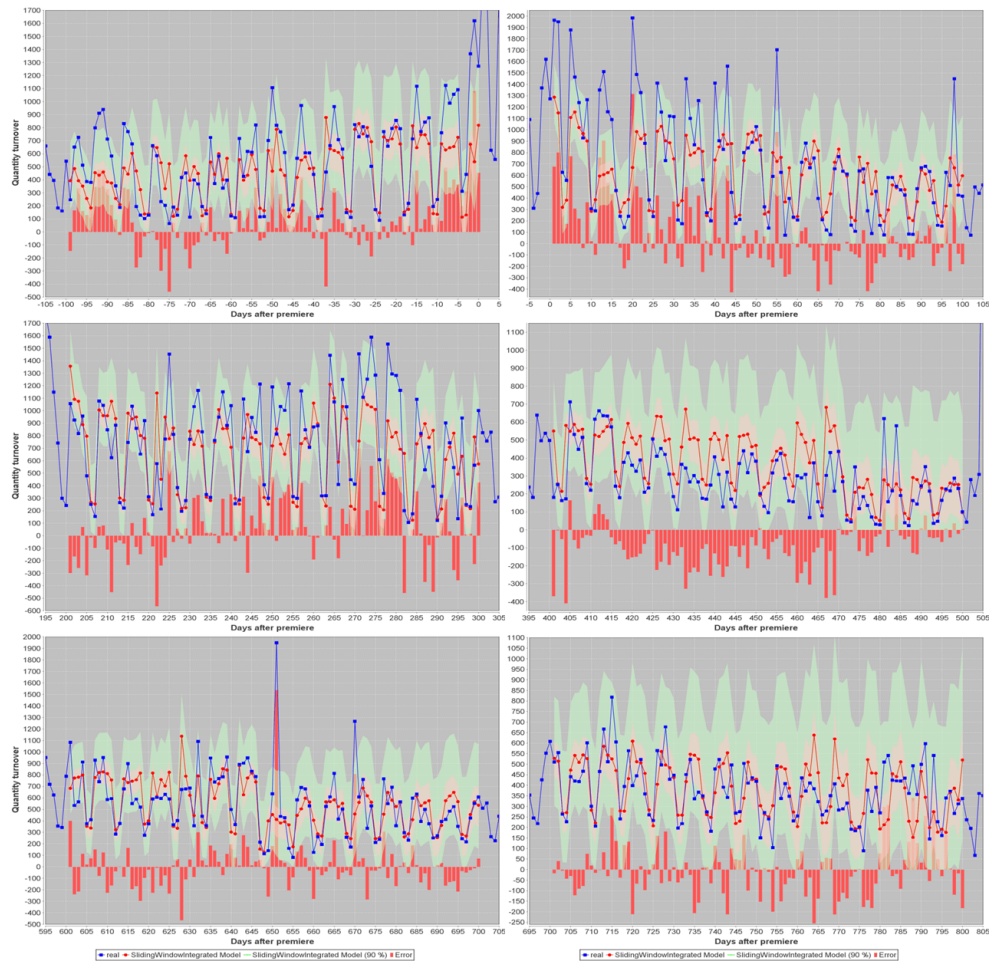


Figure B.7: Sales prediction of production 4652 using the sliding window integrated approach.

B.7 Sliding Window Evaluation Examples

In figure B.9 we compare different algorithms on different horizons by means of the sliding window integrated approach. It can be seen that the bagged tree outperforms all other algorithms and that the linear regression performs respectfully well. Moreover, the bagged tree using a model tree as base learner (M5P) reacts comparably instable which is due to the fact that for some performances the rules learned lead to vast deviations, whereas they are normal for the others. This is the reason why we have chosen the simpler and more robust regression tree REPTree.

In place of algorithms different data base configurations are compared in figure B.10. We fix the algorithm dimension at the bagged tree with REPTree.

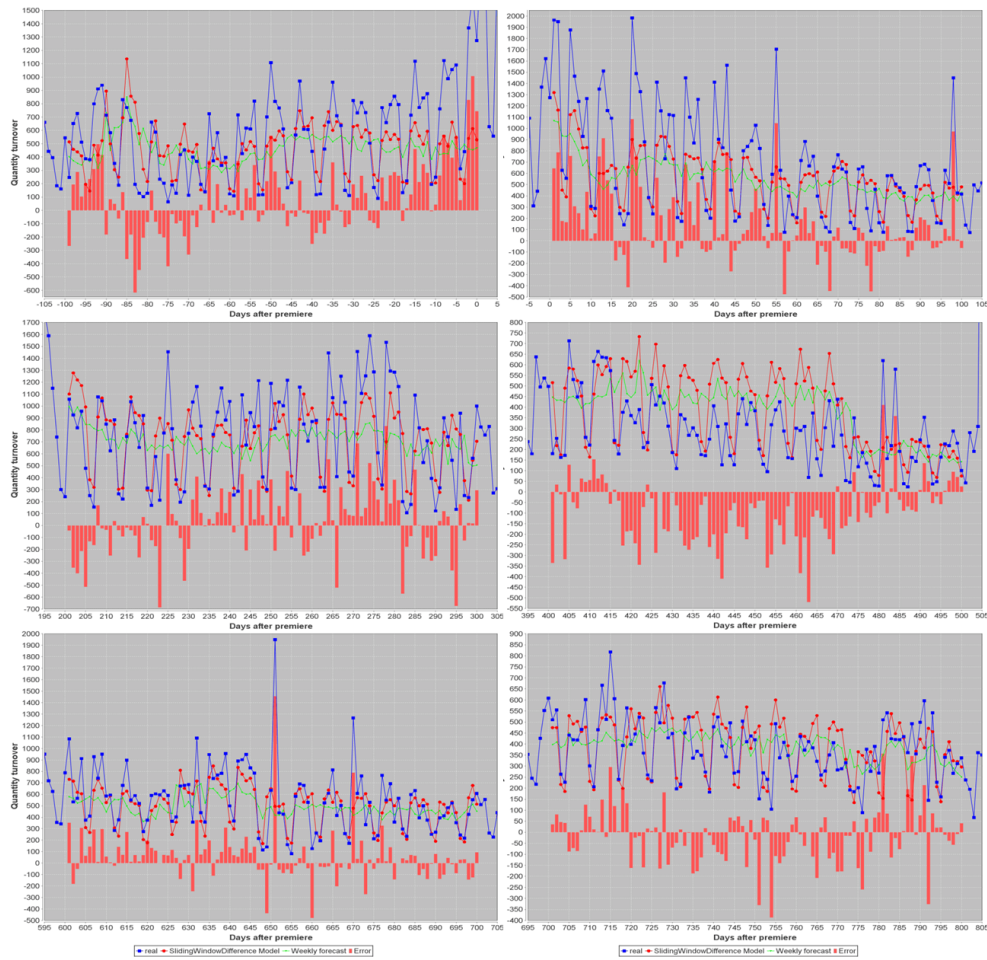


Figure B.8: Sales prediction of production 4652 using the sliding window difference approach.

B.8 Sales Prediction with the Micro-founded Approach

We first take a look at some other scenarios of production 4068 depicted in figures B.11, B.12 and B.13. The first one is the same used for the sliding window approaches in the methodology. There, the end of the summer break (also indicated by the ceasing amount of events involved) and the resulting increase in sales is reproduced quite well. Nevertheless, the level reached stays behind real sales, more than in the other two approaches. In the second figure sales are underestimated as well. The resulting RMSE is 186.57 there, resulting in an RRSE of 62.41% and a total deviation of 13,642 tickets as over- and underestimation balance out to some degree. Finally, in the last scenario showing the end of the life cycle (and as a consequence performs normal forecasts only), figures are still underestimated as the final deviation is 8,285.1 tickets. The RMSE is 139.65 and the RRSE 71.1%.

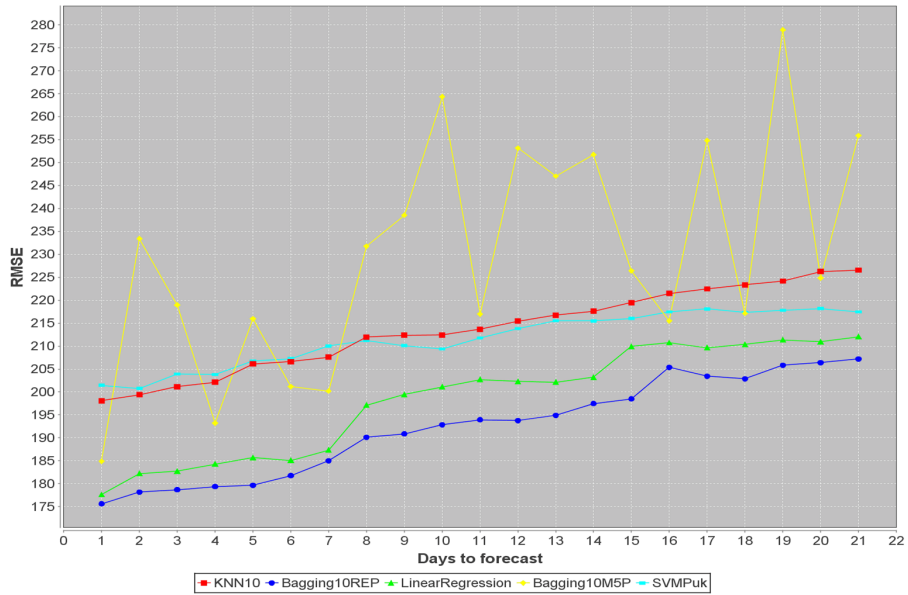


Figure B.9: Comparison of different algorithms using the sliding window integrated approach.

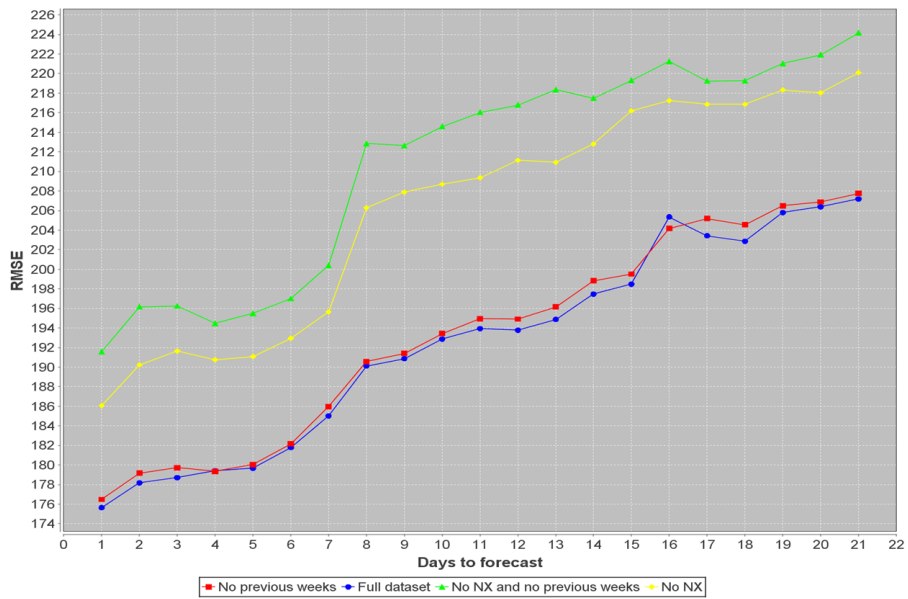


Figure B.10: Comparison of different attribute configurations by means of the sliding window integrated approach. The red line ignores the four attributes that describe sales of the weeks prior to the week of observation. “No NX” stands for no information about the target day, i.e. the amount of events to take place and date related information.

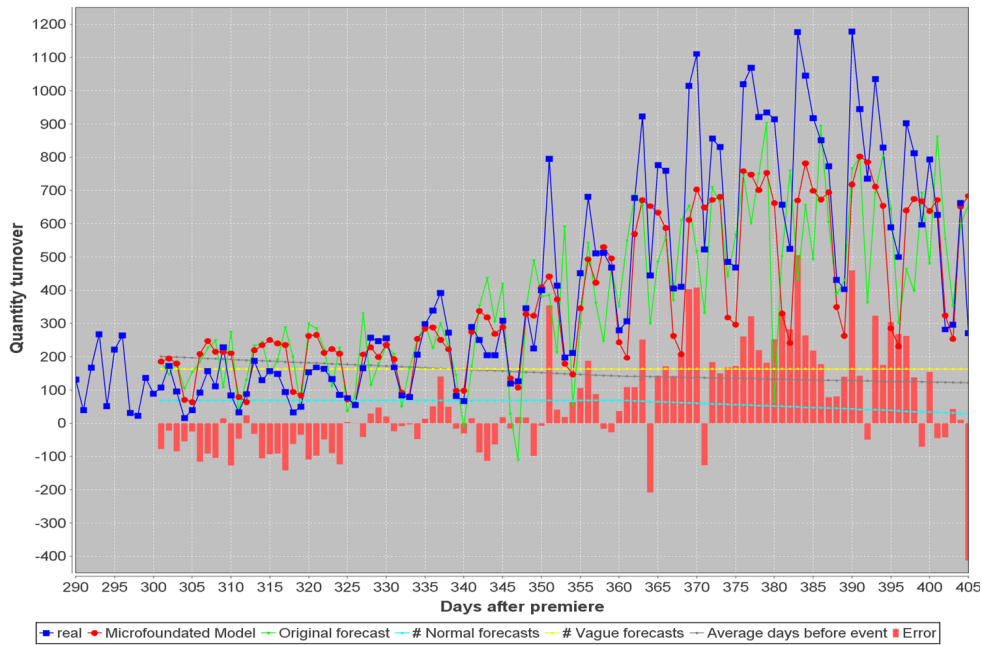


Figure B.11: Sales prediction of production 4068 using the micro-founded approach with the forecast period set to 300 – 400.

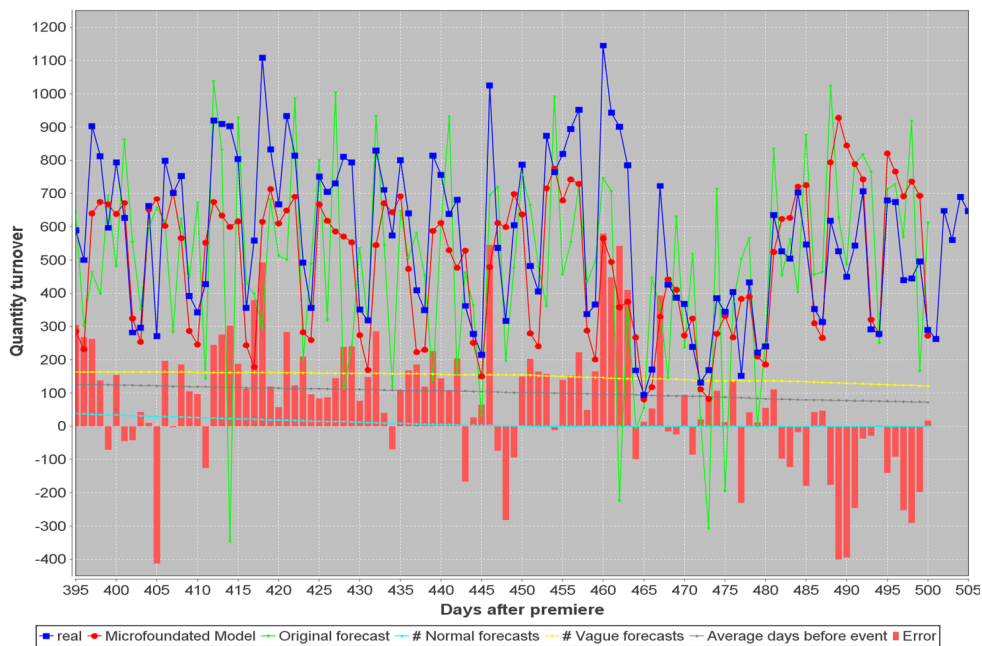


Figure B.12: Sales prediction of production 4068 using the micro-founded approach with the forecast period set to 400 – 500.

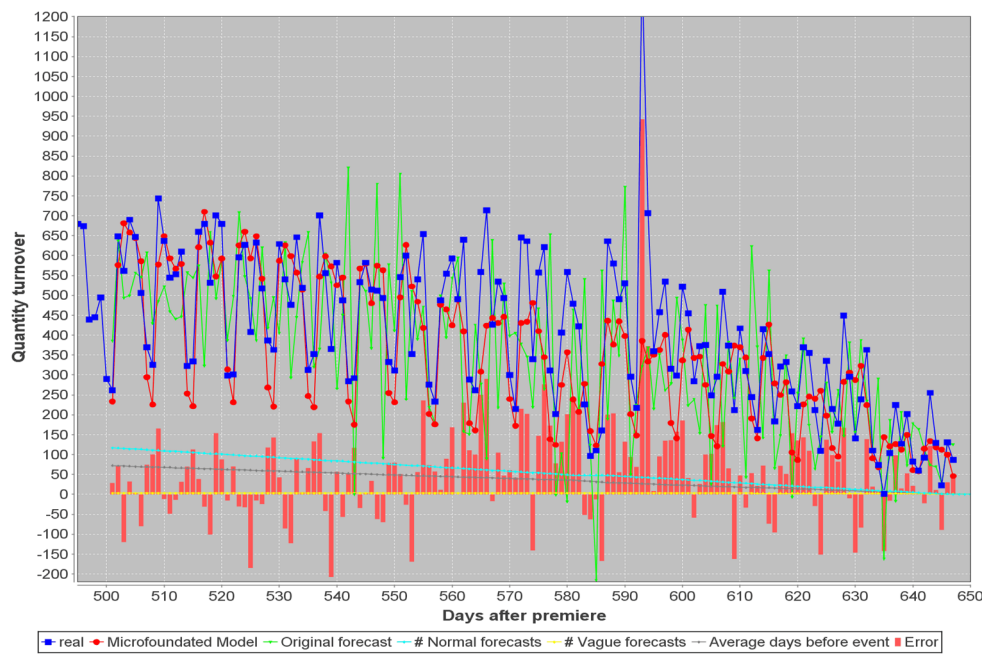


Figure B.13: Sales prediction of production 4068 using the micro-founded approach regarding the end of life cycle.

In figure B.14 we take a look at a different production, that is 5830 from day 100 to 200. We created 9,626 normal and 23,764 vague models therefore. The RMSE is 175.5, the RRSE 84.75% and the total deviation 11,736.

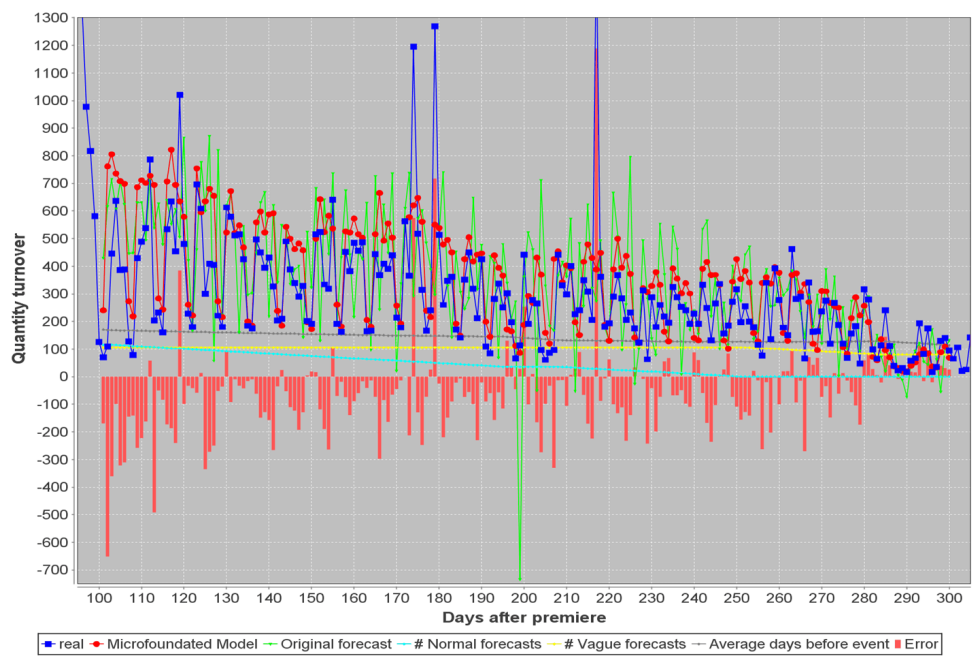


Figure B.14: Sales prediction of production 4068 using the micro-founded approach with the forecast period set to 100 – 200.

Attribute Groups

This chapter list all attributes that are used in the methodology to be referenced from the respective model that makes use of it.

SF – Sales figures interval attributes:

- infinity–300
- 300–250
- 250–200
- 200–175
- 175–150
- 150–125
- 125–100
- 100–90
- 90–80
- 80–70
- 70–60
- 60–50
- 50–40
- 40–30
- 30–20
- 20–15
- 15–10
- 10–5
- 5–1
- 1–0

EQ – Event related qualitative attributes:

- weekday
- month
- day of year
- holiday?

TQ – Ticket related qualitative attributes (all of them are averaged):

- actual price
- days after the premiere
- days before the show
- absolute discount
- relative discount
- weekday
- month
- day of year
- distance
- temperature
- precipitation
- fraction of male customers
- fraction of female customers
- fraction of neuter customers
- fraction of local customers
- fraction of contingent sales
- fraction of price category Rot
- fraction of price category Rosa
- fraction of price category Gelb
- fraction of price category Grün
- fraction of price category Orange
- fraction of price category Blau
- fraction of price category Stehplatz

DC – Decile attributes:

- decile 1
- decile 2
- decile 3
- decile 4
- decile 5
- decile 6
- decile 7
- decile 8
- decile 9

CD – Cutoff day related attributes:

- weekday
- month
- Google Trend
- temperature
- precipitation

NX1 – Target day related attributes of DQ1 and DQ2:

- day after the premiere
- amount of holidays as from cutoff day

LG – Lag attributes:

- lag 0
- lag 1
- lag 2
- lag 3
- lag 4
- lag 5
- lag 6

PW – Sales figures of past weeks:

- past four weeks
- past four weekdays
- all past weeks
- all past weekdays

EX – Exogenous qualitative attributes (all of them are averaged):

- actual price
- full price
- days after the premiere
- days before the show
- relative discount
- weekday of events
- month of events
- month of ticket
- is the event date a holiday?
- is the purchase date a holiday?
- ticket weight
- purchaser's distance
- temperature
- precipitation
- Google Trend of cutoff day

SS – Sales situation attributes:

- amount of events concerned
- amount of events happening
- amount of prospective events
- sales figures for this week
- sales figures for future weeks

NX5 – Target day/week related attributes of DQ5:

- amount of events
- is the target day a holiday/the amount of holidays
- weekday
- month

Other attributes directly related:

- x1 total occupancy rate until cutoff day
- x2 amount of holidays until cutoff day

Bibliography

- Fabian Abel, Ernesto Diaz-Aviles, Nicola Henze, Daniel Krause, and Patrick Siehndel. Analyzing the blogosphere for predicting the success of music and movie products. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 276–280. IEEE, 2010.
- Alan Agresti. *Categorical Data Analysis*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series. John Wiley & Sons, 2002.
- Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.
- Victoria Ateca-Amestoy and Juan Prieto-Rodriguez. Forecasting accuracy of behavioural models for participation in the arts. *European Journal of Operational Research*, 229(1):124–131, 2013.
- Richard Bellman. *Adaptive Control Processes: A Guided Tour*, volume 4. Princeton University Press, 1961.
- Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8):7067–7083, 2012.
- Yevgeniy Bodyanskiy and Sergiy Popov. Neural network approach to forecasting of quasiperiodic financial time series. *European Journal of Operational Research*, 175(3):1357–1366, 2006.
- Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In *Business Intelligence*, pages 62–77. Springer, 2013.
- George E.P. Box and Gwilym M. Jenkins. Time series analysis: Forecasting and control. *Holden-Day series in time series analysis*, 1976.
- Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2002.

- Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. CRISP-DM 1.0 step-by-step data mining guide. *CRISPWP-0800*, 2000.
- Chen-Yuan Chen, Wan-I Lee, Hui-Ming Kuo, Cheng-Wu Chen, and Kung-Hsing Chen. The study of a forecasting sales model for fresh food. *Expert Systems with Applications*, 37(12):7696–7702, 2010.
- Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- Jan G. De Gooijer and Rob J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006.
- Vasant Dhar and Elaine A. Chang. Does chatter matter? The impact of user-generated content on music sales. *Journal of Interactive Marketing*, 23(4):300–307, 2009.
- Philip Doganis, Alex Alexandridis, Panagiotis Patrinos, and Haralambos Sarimveis. Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, 75(2):196–204, 2006.
- James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2012.
- Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, 1996.
- Eibe Frank and Remco R. Bouckaert. Conditional density estimation with class probability estimators. In *Advances in Machine Learning*, volume 5828 of *Lecture Notes in Computer Science*, pages 65–81. Springer, 2009.
- Tal Garber, Jacob Goldenberg, Barak Libai, and Eitan Muller. From density to destiny: Using spatial dimension of sales data for early prediction of new product success. *Marketing Science*, 23(3):419–428, 2004.
- Mike Gualtieri. The forrester wave™: Big data predictive analytics solutions, Q1 2013. *Forrester Research Inc. Report*, 2013.
- G.K. Gupta. *Introduction to Data Mining with Case Studies*. PHI Learning Pvt. Ltd., 2006.
- Chris Hand and Guy Judge. Searching for the picture: Forecasting UK cinema admissions using Google Trends data. *Applied Economics Letters*, 19(11):1051–1055, 2012.
- Anthony J.G. Hey, Stewart Tansley, and Kristin Michele Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research Redmond, WA, 2009.
- Martin Hilbert and Priscila López. The world’s technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011.

- George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the 11th International Conference*, pages 121–129. Morgan Kaufmann, 1994.
- V. Kecman. Support vector machines—an introduction. In *Support Vector Machines: Theory and Applications*, volume 177 of *Studies in Fuzziness and Soft Computing*, pages 1–47. Springer, 2005.
- Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319, 2003.
- Teuvo Kohonen. *Self-Organizing Maps*, volume 30. Springer, 2001.
- Indrė Žliobaitė, Jorn Bakker, and Mykola Pechenizkiy. Towards context aware food sales prediction. In *IEEE International Conference on Data Mining Workshops*, pages 94–99. IEEE, 2009.
- Indrė Žliobaitė, Jorn Bakker, and Mykola Pechenizkiy. Beating the baseline prediction in food sales: How intelligent an intelligent predictor is? *Expert Systems with Applications*, 39(1):806–815, 2012.
- Sheng Lu and Ki H. Chon. Nonlinear autoregressive and nonlinear autoregressive moving average model parameter estimation by minimizing hypersurface distance. *Signal Processing, IEEE Transactions on*, 51(12):3020–3026, 2003.
- Patricia E.N. Lutu and Andries P. Engelbrecht. A decision rule-based method for feature selection in predictive data mining. *Expert Systems with Applications*, 37(1):602–609, 2010.
- Óscar Marbán, Gonzalo Mariscal, and Javier Segovia. A data mining and knowledge discovery process model. *Data Mining and Knowledge Discovery in Real Life Applications. InTech*, 2009:8, 2009.
- Pablo Marshall, Monika Dockendorff, and Soledad Ibáñez. A forecasting system for movie attendance. *Journal of Business Research*, 66(10):1800–1806, 2013.
- Fernando Mateo, Juan José Carrasco, Abderrahim Sellami, Mónica Millán-Giraldo, Manuel Domínguez, and Emilio Soria-Olivas. Machine learning methods to forecast temperature in buildings. *Expert Systems with Applications*, 40(4):1061–1068, 2012.
- Helmut A. Mayer and Roland Schwaiger. Evolutionary and coevolutionary approaches to time series prediction using generalized multi-layer perceptrons. In *Proceedings of the 1999 Congress on Evolutionary Computation*, volume 1. IEEE, 1999.
- Viktor Mayer-Schönberger and Kenneth Cukier. *Big Data: A Revolution that Will Transform how We Live, Work, and Think*. Eamon Dolan/Houghton Mifflin Harcourt, 2013.
- Ronen Meiri and Jacob Zahavi. Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171(3):842–858, 2006.

- Patrick Meulstee and Mykola Pechenizkiy. Food sales prediction: “if only it knew what we know”. In *IEEE International Conference on Data Mining Workshops*, pages 134–143. IEEE, 2008.
- Elena Montañés, José R. Quevedo, Maria M. Prieto, and César O. Menéndez. Forecasting time series combining machine learning and box-jenkins time series. In *Advances in Artificial Intelligence — IBERAMIA 2002*, volume 2527 of *Lecture Notes in Computer Science*, pages 491–499. Springer, 2002.
- Geun Choi Moon, Go Kikuta, Takashi Yamada, Atsushi Yoshikawa, and Takao Terano. Blog information considered useful for book sales prediction. In *7th International Conference on Service Systems and Service Management*, pages 1–5. IEEE, 2010.
- Glenn J. Myatt and Wayne P. Johnson. *Making Sense of Data II: A Practical Guide to Data Visualization, Advanced Data Mining Methods, and Applications*. John Wiley & Sons, 2009.
- John Naisbitt. *Megatrends: Ten New Directions Transforming our Lives*. Warner Books, 1982.
- Chia Huey Ooi, Madhu Chetty, and Shyh Wei Teng. Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets. *Data Mining and Knowledge Discovery*, 14(3):329–366, 2007.
- Sonja Ostendorf-Rupp. Datenanalyse: Erkenntnisse aus der Anwendung eines Verkaufsvorhersagemodells im Orchesterbetrieb. <http://kulturmanagementusa.blogspot.com/2013/10/datenanalyse-erkenntnisse-aus-der.html>, 2013.
- Ajoy K. Palit and Dobrivoje Popovic. *Computational Intelligence in Time Series Forecasting*. Springer, 2005.
- Juan Peralta, Xiaodong Li, German Gutierrez, and Araceli Sanchis. Time series forecasting by evolving artificial neural networks using genetic algorithms and differential evolution. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.
- Selwyn Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156(2):483–494, 2004.
- Dorian Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.
- Thomas Ragg, Wolfram Menzel, Walter Baum, and Michael Wigbers. Bayesian learning for sales rate prediction for thousands of retailers. *Neurocomputing*, 43(1):127–144, 2002.
- Hanan C. Selvin and Alan Stuart. Data-dredging procedures in survey analysis. *The American Statistician*, 20(3):20–23, 1966.
- Robert H. Shumway and David S. Stoffer. *Time Series Analysis and its Applications: With R Examples*. Springer, 2011.

- Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.
- Antti Sorjamaa and Amaury Lendasse. Time series prediction using DirRec strategy. In *European Symposium on Artificial Neural Networks*, pages 143–148, 2006.
- J.L. Stimpert, Judith A. Laux, Coyote Marino, and George Gleason. Factors influencing motion picture success: Empirical review and update. *Journal of Business & Economics Research*, 6(11), 2011.
- B. Üstün, W.J. Melssen, and L.M.C. Buydens. Facilitating the application of support vector regression by using a universal pearson VII function based kernel. *Chemometrics and Intelligent Laboratory Systems*, 81(1):29–40, 2006.
- Sébastien Thomassey. Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics*, 128(2):470–483, 2010.
- Hastie Trevor, Tibshirani Robert, and Jerome Friedman. *The Elements of Statistical Learning*, volume 1. Springer, 2001.
- Peter van der Putten, Joost N. Kok, and Amar Gupta. Why the information explosion can be bad for data mining, and how data fusion provides a way out. In *Proceedings of the Second SIAM International Conference on Data Mining*, pages 128–138. Society for Industrial and Applied Mathematics, 2002.
- Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *Computational Intelligence and Bioinspired Systems*, volume 3512 of *Lecture Notes in Computer Science*, pages 758–770. Springer, 2005.
- Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.
- Xindong Wu, Kumar Vipin, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu Yu, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.
- Zhiyuan Yao, Tomas Eklund, and Barbro Back. Using SOM-ward clustering and predictive analytics for conducting customer segmentation. In *IEEE International Conference on Data Mining Workshops*, pages 639–646. IEEE, 2010.
- Guoqiang Zhang, Eddy B. Patuwo, and Michael Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 1998.
- Shichao Zhang, Chengqi Zhang, and Qiang Yang. Data preparation for data mining. *Applied Artificial Intelligence*, 17(5-6):375–381, 2003.