FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Optimization of Real-Time Auction Bidding Strategies in Mobile Advertisement

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Alexander Dür
Matrikelnummer 0927362

an der
Fakultät für Informatik der Technischen Universität Wien

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

Wien, 25.09.2014

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Optimization of Real-Time Auction Bidding Strategies in Mobile Advertisement

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Alexander Dür
Registration Number 0927362

at the Faculty of Informatics
at the Vienna University of Technology

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

Vienna, 25.09.2014

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Alexander Dür
Jahngasse 5/9, 1050 Wien

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 Verfasser)

Abstract

Today many smartphone applications are free for the user and financed by advertisements, also called ads. This trend leads to a larger volume of advertisement space sold each year over so called ad exchanges. On an ad exchange advertisement space is sold live to the highest bidder. The bidders see only certain characteristics of the smartphone user like the app he is using right now, the country he is in and his device. Based on this information they make an offer on how much they are willing to pay to show an advertisement of their choice right now to that user. Most advertisements are bought by big agencies that bid on behalf of customers and are paid to optimize click and conversion performance for them. They have to decide how much an impression is worth to them and the ad of which customer they want to show. This is a complex problem as there are large amounts of data to with and decisions have to be made within milliseconds.

The problem which should be solved in this thesis is finding a good bidding strategy from the perspective of a company that optimizes the click or conversion performance of campaigns on behalf of their customers. Advertisements for web browsers heavily rely on cookies and user tracking while mobile advertisements that are shown within apps have to rely on other information. We will look at all available features that are known before one has to make a bidding decision and test them for their association with click and conversion performance, which are by far the most important metrics used to measure the success of an advertisement opportunity. After finding a set of attributes that is suitable to make predictions various machine learning techniques will be used to learn which prediction method performs best for the problem at hand. Then we will show how to implement a predictor that can make good predictions about click and conversion performance considering constraints such as having to learn from large datasets and make extremely fast predictions. Our approach relies on a Naive Bayes classifier which makes simplifying assumptions that do not hurt the classification performance but reduce the computational power needed for training the classifier and making predictions to a minimum. This is important to be able to consistently respond with very low latency.

One key finding also is that the naive strategy of estimating click and conversion performance through the overall click or conversion rate of the app during an individual campaign has already performed surprisingly well due to the fact that the specific app

used already includes a lot of information about the advertising opportunity. The specific application used is by far the most important attribute when evaluating the value of an impression. Most applications are used only in a few countries and predominantly at specific times of day. Additionally our approach can also handle common problems like missing attributes or unknown attribute values. Our strategy also allows for modifications to adapt itself to different markets and competitors that determine whether it is more important to choose one's advertisement opportunities very carefully or whether one should bid on as many impressions as possible as the most important thing is to not miss any clicks. With the right settings our improved strategy heavily outperforms the reference strategy. In a direct comparison without any other bidding agents our improved strategy results in 250% increased profit.

Kurzfassung

Viele Smartphone-Applikationen die man heutzutage herunterlädt sind für den Benutzer gratis und finanzieren sich über Werbeeinsschaltungen. Dieser Trend führt dazu dass die Anzahl der Werbeplätze für mobile Geräte die auf sogenannten Ad-Exchanges verkauft wird Jahr für Jahr steigt. Auf Ad-Exchanges werden Werbeplätze live versteigert (Anmerkung "Ad" ist die Kurzform von Advertisement). Die Teilnehmer sehen gewisse Eigenschaften der Werbemöglichkeit wie z.B. das Land in dem sich der Smartphone-Benutzer gerade befindet, die App die er gerade verwendet und sein Gerät. Auf Basis dieser Informationen müssen die Auktionsteilnehmer ein Gebot abgeben. Die meisten Werbungen auf solchen Ad-Exchanges werden von großen Agenturen gekauft, die die Werbekampagnen ihrer Kunden bezüglich Clicks oder Conversions optimieren. Sie müssen entscheiden wie viel eine Impression für sie wert ist und falls sie die Auktion gewinnen für welchen ihrer zahlreichen Kunden sie eine Werbeanzeige schalten. Die Herausforderungen dieses Problems liegen in den großen Datenmengen die zur Entscheidungsfindung herangezogen werden müssen und dass innerhalb nur weniger Millisekunden ein Gebot abgegeben werden muss.

Das Ziel dieser Diplomarbeit ist es eine gute Bietstrategie für Unternehmen zu finden, die für ihre Kunden Click- und Conversion-Optimierung anbieten. Ähnliche Optimierungsprobleme gibt es bereits für herkömmliche Internetwerbung im Browser, allerdings beruhen momentan fast alle eingesetzte Lösungen dieser Probleme auf Usertracking mit Hilfe von Cookies. Für Werbung innerhalb von Apps muss man allerdings auf kontextuelle Informationen zurückgreifen. Wir werden alle unterschiedlichen Attribute, die uns im Rahmen einer Ad-Exchange zur Verfügung stehen auf ihre Assoziation mit Click und Conversion-Performance untersuchen, da dies die zwei wichtigsten Metriken sind, um die Qualität von Werbeeinblendungen zu beurteilen. Nachdem wir herausgefunden haben welche Attribute am stärksten mit Click und Conversion-Raten assoziert sind, werden wir unterschiedliche Maschinlernverfahren anwenden um zu sehen welche Verfahren die Qualtität von Werbeeinschaltungen am besten vorsagen können. Der nächste Schritt ist eine Lösungsmöglichkeit mit der man ein System implementieren kann, das gute Vorhersagen produziert und trotzdem allen Herausforderungen in Bezug auf Geschwindigkeit und große Datenmengen gewachsen ist. Unser Ansatz basiert auf einem Naive Bayes Classifier, der stark vereinfachende Annahmen macht, welche die

Klassifizierungsperformance nicht beeinträchtigen, allerdings den Rechenaufwand pro Vorhersage stark reduzieren. Dies ist eine wichtige Vorraussetzung um konsistent mit niedriger Latenz zu antworten.

Ein sehr interessantes Ergebnis der Arbeit ist, dass die naive Strategie, die Click- und Conversion-Raten anhand der historischen Click und Conversion-Raten der App errechnet, sehr gute Schätzungen liefert. Die spezifische Applikation, in der eine Werbeeinblendung gezeigt wird, spielt mit Abstand die größte Rolle, wenn es darum geht den Wert einer Impression zu bestimmen. Die meisten Applikationen werden nur in wenigen Länder und zu gewissen Zeiten verwendet. Ein weiterer Vorteil unserer Methode ist, dass sie auch mit häufigen Problemen wie fehlenden Attributen oder unbekannten Attributwerten (z.B. eine neue Applikation) umgehen kann. Außerdem erlaubt die Strategie Modifikationen, um sich an die Marktgegebenheiten anzupassen. Je nach Marktgegebenheiten ist es wichtiger seine Gebote so zu wählen, dass nur wenige Auktionen gewonnen werden, die eine sehr hohe Wahrscheinlichkeit haben, in einem Click oder einer Conversion zu enden oder es ist wichtiger ja keine Gelegenheit für einen Click oder eine Conversion auszulassen. Richtig verwendet liefert unsere neue Strategie sehr viel bessere Ergebnisse als die Referenzstrategie. In einem direkten Vergleich ohne andere Auktionsteilnehmer macht die verbesserte Strategie um 250% mehr Gewinn.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The broad topic of my master's thesis is mobile advertisement, more specifically ad space allocation in applications for smartphones. A newly emerging trend in mobile advertisement is that advertisement space is sold on so called ad exchanges instead of directly to an ad agency for a flat fee. On an ad exchange each time a customer opens the smartphone application an auction is created on an ad exchange, where multiple potential buyers can bid. The highest bid wins and the smartphone application user sees the winner's advertisement. The bidders have to decide on their bid based on information about the user like country, local time and smartphone model and the application where the advertisement is shown. All this happens within a time frame of roughly a second where the bidding decision itself has to be made within 200 milliseconds. [24] [25]

Mobile devices in general are one of the fastest evolving technology sectors right now and advertisement plays a key role in that as it is one of the main financing sources for software on mobile devices. Smartphones change how we interact with each other, offer new economic opportunities and are driven by technical advancement in the areas of processor technology and wireless networks. Mobile advertisement is challenging on the technical side due to the constantly changing devices and new technology and on the other hand still requires social considerations in regard to privacy and usage patterns. That leads to a great opportunity to improve the scientific understanding of the mobile advertising market. [15]

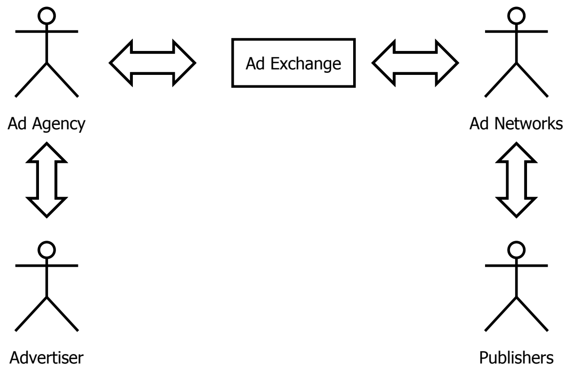

Figure 1.1: Ad exchange market overview

1.2 Problem Domain

Each of the actors of the part of the mobile advertisement market that we are interested in falls into at least one of the following categories: *advertiser*, *ad agency*, *ad network*, *publisher* or *ad exchange* (cf. Figure 1.1). Advertisers who want to promote their brand or product pay ad agencies to position their advertisements optimally. Publishers who own a smartphone application which they want to monetize through advertisement on the other hand sell their advertisement space to an ad network. On the ad exchange the ad agencies buy and ad networks sell advertisement space on behalf of their customers. The roles of the ad network and the ad agency are not mandatory. In theory advertisers and publishers can directly use the ad exchange and can skip the middle men, which are ad agencies and ad networks in our case, but this is rarely seen in practice due to many ad exchanges being private contracts between multiple companies.
[37]

As the online advertisement market is very different from the traditional advertisement market it has developed its own vocabulary. It is mandatory to understand what certain words in context of the mobile advertisement industry mean. Below you can find the most important definitions which we will use throughout this work:

ad The term *ad* is an abbreviation for advertisement and we will use the two terms synonymously.

ad agency The term *ad agency* is used for every actor that directly buys ad space on an ad exchange.

ad exchange The term *ad exchange* is used for auction platforms on which advertisement space is sold in real time.

advertiser The *advertiser* is an actor whose brand or product is promoted by the ads sold at the ad exchange.

campaign Advertisement efforts are structured into *campaigns*. Each *campaign* has a goal (e.g. promoting a product or brand) and is limited in time, budget or cost per click and conversion or a combination of the three.

click The *click* is an event where a user clicks an advertisement presented to him. In practice users have to open an application on their smartphones which present the ads to them and they have to tap with their fingers onto the display onto the spot where the ad is displayed for a *click* to be registered.

conversion Some advertisements have an additional goal besides getting clicked. As soon as users have clicked the advertisement and completed all required additional actions (e.g. entering their email address into a form) the click becomes a conversion.

CPC The acronym *CPC* stands for cost per click. It is calculated over a certain reference frame (e.g. 24 hours, advertisement campaign) using the following formula:

$$\frac{total\ amount\ of\ money\ spent}{total\ number\ of\ clicks} \tag{1.1}$$

CTR The *click-through-rate* is the ratio of clicks to impressions. It is calculated by the following formula:

$$\frac{number\ of\ clicks}{number\ of\ impressions} \tag{1.2}$$

impression The *impression* is an event of a user seeing a certain advertisement.

publication The *publication* is the specific application where the advertisement is shown. Synonyms are: sub, app and application. The word sub is used because as we see later we can subdivide all the impressions allocated to a specific publisher into smaller sets of impressions allocated to this publisher's applications.

publisher The *publisher* is the owner of an application where ads are shown.

When users opens their mobile application which had ad space included by the publisher through the API of an ad network, a notification is sent to the ad exchange. The ad exchange now extracts all important information from the notification and sends a predefined part of this information to all registered ad agencies for a bid-request. After a certain time period has elapsed or all parties have answered the ad agency with the highest bid wins and an advertisement of its choice is shown. The winning agency pays the price of the second highest bid or the value of the so called price floor, the absolute minimum bid, for which the publisher or ad network wants to sell the inventory.

The reason why this system is used by all its participants is because of its advantages over traditional inventory selling. The ad exchange takes a spot in the mobile advertisement market similar to a stock exchange in the financial markets. [23] Like a financial exchange it provides efficient prices under the general assumption that no participant has any secret knowledge. Due to the fact that impressions are auctioned in a Vickrey auction market participants are encouraged to bid the actual amount of money that they assume a given impression to be worth. [31] This plays a major role in the design of an optimal bidding strategy. While providing the buying side with fair prices it also allows the selling side to get the actual worth of its inventory and additionally mitigates the problem of unsold inventory. Advertisement impressions are a good that has to be sold just-in-time as it cannot be stored in any way - as soon as customers opens the application they either have to see an ad or the opportunity is gone.

The system of the ad exchange is already heavily used in the online advertisement market but is still relatively new in the mobile market. While there are certainly a lot of aspects that remain the same there are also slight differences. The main differences are what data are available to the bidders and the way in which mobile devices are used differently from desktop computers. Due to the situations in which smartphones are used and the different perception and usage patterns between mobile and desktop devices the mobile market becomes increasingly attractive. People often spend more time on their smartphone or tablet than their computer and additionally it is a lot easier to get people to install software immediately through a renowned app store (e.g. Google Play). The total as well as relative amount of mobile ad inventory sold on ad exchanges is significantly increasing at the moment and there is currently no sign of this trend to stop. [14]

1.3 Problem Description

The thesis will focus on a problem for ad agencies. Ad agencies are trusted by their customers with the task to optimally buy impressions on an ad exchange. Normally ad agencies manage different campaigns for different customers. Campaigns that have to be optimized are always limited in one way or another. Financially there is either a total budget or a goal cost per click or conversion and there also will be a certain time window in which the campaign runs. Success of such a campaign is measured in the number of acquired clicks or conversions in relation to the money and time spent. The target of the thesis is the optimization for clicks and conversions although conversion rates are extremely low. The main reason is that whether an impression was clicked or not is directly reported by the ad exchange on which the impression was acquired. Conversions on the other hand are not centrally handled and often ignored especially when payment is per click or impression and not per conversion. As agencies have to rely on their customers to report back conversions there is a much larger technical overhead on both sides involved to make sure all conversions are reported back properly. Especially when customers pay per conversion they have an incentive not to report all conversions. In respect to campaigns we will consider the following scenario:

- limited number of impressions to bid on
- goal cost per click

Under these conditions the profit for the ad agency is the difference between the money they get paid based on the amount of clicks generated and the total amount of money spent. This scenario is preferred as it is currently most often used in reality, the reason being that in practice most campaigns are very selective with their target customers which reduces the need to regulate spending. We will ignore the problem of selecting the right advertisement as this is not part of the ad agency's task. The ad agency can only bid on behalf of its customers on potential impressions and serve a predefined advertisement of a customer of its choice in case it wins the auction. The main problem statement therefore is:

- What is a good bidding strategy for an ad agency managing a given number of campaigns having a limited number of impressions to bid on?

This problem statement is on a very high level and we will now break it down into smaller sub problems which we are able to solve more efficiently. When we start from the bottom up we have to start with a single bid request from an ad exchange. This is the most basic decision unit of our strategy. Our goal is to obtain clicks. Clicks can only be obtained through impressions and impressions can in our scenario only be attained by buying them on an ad exchange. Therefore an optimal bidding strategy has to answer the following three questions for every incoming request:

- Do we bid on this impression?
- If yes, how much do we bid?
- If we win, which campaign gets the impression?

By splitting our complex decision problem into these smaller and easier tasks we can start with a very simple strategy and continuously improve it step by step. Of course, we will later continue to further break down the solution to some of those problems but they form a good basis for the logical understanding of the optimal decision process which we want to find. We will discuss the first two questions in great detail while the answer to the third one depends heavily on the business model of the ad exchange and there are already solutions described in existing literature as we will see.

1.4 Aim of the Work

The aim of the work is to look at what is possible in terms of optimal decision making in regard to bidding on an ad exchange in the role of an ad agency. In reality the bidding decision has to be made in the fraction of a second. The aim of this thesis is to provide a template to implement such a bidding algorithm in practice while still exploring whether computationally more expensive options that one would generally not consider for a production system perform significantly better. Furthermore we also want to give a good overview about the different areas in which one can aim for optimization in practice. Due to the fact that the auction system used is the Vickrey auction there is less additional benefit in analyzing competitorś bidding strategies than in optimizing the own strategy. In reality this is a topic that should not be underestimated but is very complex and requires multiple reference strategies to compare against each other. It is an interesting research topic that can be tackled once a variety of efficient strategies have been established and mobile ad exchanges operate in a similar manner to each other. [31]

1.5 Methodological Approach

The methodological approach will incorporate a number of different techniques as we try to separate the subproblems from one another as much as possible. Each problem will require its own tools in order to be solved.

For the decision whether we will bid on a certain impression we use standard machine learning techniques for classification. For finding the best price to bid we will look at possible predictors of a fair price. The sub-problems of bidding and distributing the ad impressions amongst different campaigns will not be handled in detail as already mentioned before. We will focus more on the topics where we have to make fewer assumptions about the market and business model of us as an ad agency.

CHAPTER 2

State of the Art

In the last chapter our research problem has been broken down into three sub-problems which are key questions for the thesis:

- Do we bid on this impression?
- If yes, how much do we bid?
- If we win, which campaign gets the impression?

This chapter is about existing research that helps us to tackle these problems. Mobile advertisement, especially ad exchanges for mobile advertisement, are not around for a very long time but have significant parallels with areas of online advertisement. There are differences in the data available when buying ad impressions on web sites instead of mobile applications but the general structure stays the same. No matter where you look in online advertisement the vocabulary is similar and the distinction between clicks and conversion exists. Grouping of advertisement tasks into campaigns can be found everywhere, as well as the requirements on the computational speed and the data available are similar in the whole field of online advertisement. The main difference is that the tracking of individual users has not been a feasible option on mobile phones yet.

2.1 The Datamining Process

The CRISP-DM methodology will be used as a guideline for this thesis as it imposes a structure on the project that makes it easier to dissect into a series of steps that the solution has to cover.

The CRISP-DM [9] methodology splits the data mining process into a series of six steps. It is based on the cyclical nature of data mining as it is a continuous process that is repeated over and over again as the data changes. We will go through the process once during the thesis and have a working prediction model and strategy in end which can be used as the basis of the next iteration of the here proposed datamining process.

In the CRISP-DM methodology there are six steps in the process of data mining, which will all be covered in the following chapters.

1. Business Understanding: This point consists of an initial analysis of the project objectives and their translation into a data mining problem.
2. Data Understanding: This stage is concerned with gaining insights into all the information contained within our data that might not be visible at first glance. It is also concerned with quality issues such missing values.
3. Data Preparation: The data preparation phase constructs the final dataset out of the raw data available. In our case this includes tasks such as extracting features like the weekday out of the timestamp of our impression or removing attributes.
4. Modeling: This covers the training of our classifier.
5. Evaluation: In this step one should check whether the model holds up to the expected results and whether the model addresses all our business needs.
6. Deployment: The last stage is the deployment of the model that just passed the evaluation stage.

2.2 Online Ad Exchange Bidding Strategies

The first logical step in solving a problem is to look at how it is currently being handled. In the case of mobile advertisement there is only a very limited amount of research but in the Internet advertisements have existed a lot longer. Many techniques in mobile advertisement have been copied from online advertisement like the ad exchange model, for example. Similar models are heavily used throughout the whole Internet. Even *Facebook* started using the ad exchange model due to its exceptional performance in bringing advertisements to a target customer group.

The ad exchanges used in online advertisement operate very different from the ones used in mobile advertisement due to the fact that the interaction with the user occurs within the web browser and not within an application. Successful optimization approaches like the one from *media6degrees* focus on tracking individual users by saving a cookie within their browser. [13] Information about the user is stored within the cookie including internet usage behavior and recent search queries on various sites. The content of these cookies is synchronized with internal databases whenever possible. Due to the advanced ad exchanges used in online advertisement it is possible to specifically receive only bid requests from users that are tracked this way. The actions taken in case of an incoming bid request are partly precomputed for each campaign and are matched against the data of the users and contextual information about the web site they are currently visiting. In this approach the campaign has certain parameters that need to be calibrated in order to be able to effectively match impressions and campaigns.

What is interesting is also the way impressions are assigned to campaigns. The past browsing behavior monitored through the use of cookies and partnered web pages is used to check whether a potential target qualifies for a segment of a currently running campaign. If the target user qualifies this is communicated to the ad exchange and from now on bid requests are received for that particular user. The first filter through which only roughly 1% of all observed users qualify is solely based on browsing history. In the case that a target user is in segments of two or more different campaigns the ad agency first checks against frequency limits. If the users see an advertisement more than once within a very short time frame they will probably react similar to it every time. After all eligible segments have been made out the one for which the impression has the highest value gets the impression. It is important to observe that the value of a certain ad opportunity is a function of the ad shown, the user and the circumstances under which it is shown, the properties of the impression (e.g. web site). It is also implied that the value of an impression is only determined by its ability to convert.

There exists a unique probability estimation procedure for every ad campaign. Browser URLs are mapped onto about 5000 unique inventories to meaningfully aggregate single pages from the same web site. User specific data is already aggregated in a way that the user can be immediately assigned to a target segment of the campaign, when visiting a certain web page. Through using a different estimation procedure for each campaign and filtering users depending on the properties of the impression the ad agency, that wants to buy the best advertisement space for its customers, arrives at a conversion probability for each campaign and advertisement that only depends on the user. For the probability estimation itself logistic regression is used. Also inventory features are only included if sufficient volume is acquired for estimating its impact.

The experiments performed by *media6degrees* in respect to the amount to bid for a certain conversion goal give interesting insights into bidding strategies on ad exchanges. The first strategy involved setting a base price for each inventory which is always bid when the opportunity arises to bid on an impression from this particular web site. This is the strategy which served as baseline. In a second step this strategy was improved by multiplying the base price by the score ratio of the inventory which is a measure for conversion probability. The third strategy also involves the score ratio. A ratio below 0.8

results in no bid and a score ratio above 1.2 results in twice the normal bid, while a ratio in between results in a normal bid of the base price. While the only real advantage of the second strategy over the first is a slightly better conversion rate, they both dominate the second in means of conversion rate. The advantage these two strategies have over the reference strategy is that they bid more money on impressions with a higher score. This obviously results in them winning more often in this higher quality segment. The results of the paper show that it pays off to focus on the quality of the impressions rather than only the pure number of impressions bought. The reason why the third strategy is slightly better than the second in terms of conversion performance is that the third strategy completely ignores the lower quality segment of impressions. Another advantage of the third strategy is, that it brings by far the best long term benefits for the customer as the target users which converted revisit the site of the customer over-proportionally often.

The most important differences between probability estimation models in ad exchanges for web browsers and our scenario which involves mobile applications is that because of the reduced threshold on mobile devices to download an application the meaning of click and conversion has slightly changed. While on a normal ad exchange often a page visit is counted as conversion and long term customer acquisition is the goal, the mobile ad campaigns which are the topic of this thesis have different conversion goals as it is a lot harder to track individual users without cookies across many different applications. [13]

Often ad networks which are losing relevancy with the rise of ad exchanges decide to buy additional inventory from ad exchanges which they resell to their customers. A challenge in this field is again the highly differentiated inventory. A huge problem with this very diverse data is that while the structure of the data is constant over time often new web pages or new apps are introduced.It takes a while to be able to reliably evaluate these new attribute values. One way to effectively circumvent this problem is by introducing a weighted average of the new value and all values on the same hierarchical level. It is possible to use the average conversion rate of all chat applications to estimate the conversion rate of a new one. With an increased number of observations weight of the domain hierarchy decreases and the weight of the specific app increases. [36]

2.3 Click and Conversion Probability Prediction

Keyword based auctions are still the main form of targeted advertisement on the Internet. The reason being that by looking at the keywords users searched for one gets a very clear picture of what they are looking for right now. When users search for a car brand there is a relatively high chance that they are thinking about buying a car of this brand right now. While this area of advertisement is quite different from the ad exchange model it is far better researched and has enough parallels to allow us to draw valuable conclusions as it requires a CTR estimation on side of the search engine. The reason for this is that search engines prefer contextual relevant advertisements so that the advertisement is actually seen as benefit. Google uses a so called quality score which reflects the contextual relevance to the users for the key words you chose. [3] A bid from a campaign with a bad quality score actually leads to a devaluation of the bid. This means when comparing bids from multiple campaigns, bids from campaigns with lower quality scores are discounted. Campaigns with extremely bad quality scores actually don't get any impressions at all. One very important factor for the quality score is the CTR prediction.

The first important parallel is in the way the auctions are conducted. Basically the participants set how much they are willing to pay for a single ad impression for a search query matching certain keywords. [16] This first part is also known as the *query features*. Additionally features of the user can be specified, like geographic location or time. This second group of information specifically about the user is called *context features*. So like in our case the place where the ad is shown is a seperate logical entity from the information about the user.

In a paper published by *Microsoft Research* a *Probit* regression model is used to predict the click-through-rate of sponsored search advertising. [33] While in reality advertisements shown in search engines are selected not only by price but also by certain quality measures defined by the search engine, the prediction strategy takes a completely isolated view on the topic of *CTR* prediction. [4] The most important finding is the fact that handling high cardinality features in a way where dependencies and covariances between features are ignored still produces good results. But it also shows that the classical Naive Bayes classifier does not give very accurate probability predictions.

Another interesting approach presented by Moira Regelson and Daniel C. Fain from Yahoo involves clustering of keywords to increase the amount of material from which the CTR can be estimated. [28] In keyword advertisements, the keywords themselves play a huge role in the prediction of click or conversion probability as the contextual relevance on a search engine is defined by them. Two main problems are that certain search terms show cyclic behavior (e.g. Halloween costume) or there are only very few observations for some key words. While there are only relatively few search terms that change strongly over time there is a large amount with only a handful observations. The research showed that the CTR prediction could be significantly improved by clustering keywords according to the textual similarity according to the texts of advertisements shown. By predicting click through rates for the whole cluster in case there are not enough observations in the historical data at hand, has significantly improved accuracy of the predictions.

When predicting probabilities it is not only important to look at the way data can be pre-processed or aggregated but also what techniques can use this data best to predict not only an outcome but also a probability that this prediction is right. Alexandru Niculescu-Mizil et al. examine standard machine learning techniques and their performance in predicting probabilities. [27] Some methods like SVMs push probabilities away from 0 and 1 and others such as Naive Bayes push them in the opposite direction and most probabilities end up near 0 or 1. The paper examines two techniques to transform the probabilities and remove any bias that is introduced by the characteristics of the method. In the experiments conducted calibrated boosted trees, calibrated random forests, calibrated SVMs, uncalibrated bagged trees and uncalibrated neural nets predict the best probabilities. Calibration in this context means that additional data besides the training data was used to calibrate the transformation that mitigates the bias.

2.4 General Auction Strategies

Auctions are an essential part of our economy. The way stock exchanges operate is called a continuous double auction, because the buyer as well as the seller side can continuously make offers which are then matched. While this type of auction works differently from the second price auctions found in online advertisement, the strategies of other market participants are far more important than on ad exchanges. This thesis' focus lies on predicting the performance of potential impressions as it does not require knowledge about other market participant's strategies and is by far the most important aspect of second price auctions. Still we cannot ignore competition as it is a central point in auction theory that there are other bidders as well.

A notable difference from the scenario in the paper from Tesauro is the way prices are handled. [32] The authors assume as in most markets for undifferentiated products unlike advertisement impressions that the buy and the sell price are publicly visible and that selling is also possible. In the scenario used for this thesis the goal is to estimate prices which is possible due to the high volume of the ad market but it is not possible to resell advertisements bought before. There are two very notable algorithms presented in the paper that outperform the rest. The first one is the *Sniping strategy* by Kaplan. [34] The strategy uses different heuristics to find good offers. One is based on buying every item that is sold for less than its minimum trade price in the previous period. Another relevant heuristic of this first strategy is to buy goods when the expected profit is high and the bid-ask spread is small as an undervalued asset in a liquid market corrects its price very fast. In the case of advertisement markets could adapt these mechanisms to bid increasingly aggressive in time periods with below average prices or above average click rates.

The second interesting strategy is a modified version of the *Gjerstad-Dickhaut strategy*. This method estimates the probability for a certain price to be accepted or not. In the second step it calculates the average surplus for the price which is of course the higher the lower the price and the higher the probability that the price will be accepted. But to maximize payoff the price has to be as high as possible which is in contrast to the first statement where a low price corresponds to a higher acceptance rate. This contradiction creates an optimization problem for the price. The goal is to find a middle price that creates an equilibrium between acceptance rate and payoff for the won auction which subsequently maximizes the total payoff. The concept of total payoff maximization is usable for us in a scenario where a flat amount is payed for each click or conversion. [32]

2.5 Starting a New Campaign

A big problem for all machine learning systems is the start. When a new campaign is started there is no actual data to train and calibrate a prediction mechanism. Most of the time it is impossible to produce data as the underlying probability distribution is obviously not well known or trivial, otherwise there would be no need for further analysis. The next best thing is to use historical data but only when the campaign has already started. In case the campaign has not started yet this is not an option. Thus the problem can be reduced to selecting the right subset of the existing data to train our prediction algorithm.

One approach in key word advertising found in literature is based on such a method. The goal for this approach is to maximize conversions. Like many online advertisement approaches it is very user centric and uses past search history of users to create profiles. User profiles are an n-dimensional vector with each dimension representing the weight in respect to a certain key word. [35] This way of modeling the user profiles allows for easy extension in case there is additional information about the user (e.g. location). As similarity measure between two users the Cartesian product is used. The clustering algorithm is a very interesting hybrid between supervised and unsupervised learning. Due to the fact that there is a very large number of users and little information about many of them only the most active users are used for clustering. After the clustering a set of centroids is calculated. The rest of the users can then be classified in a highly parallelized fashion by calculating the distance to the centroids of the different clusters.

After all users have been clustered the actual campaign optimization starts. At first a new campaign is shown to users of all clusters in equal parts. After a time the campaign is optimized by only showing the ad to the users from the clusters performing best. The advantage of this approach is that it can be used with basically any kind of data model as long as it can be clustered. Depending on the granularity of the clustering algorithm it is also possible to balance quality of the predictions with computational expenses. Another obvious advantage is also that by not using human intuition when classifying web pages or users there is no bias towards superficial categories that do not reflect the actual segmentation of users or web pages.

2.6 Pacing

In an auction environment it is not only important to show that a strategy is theoretically feasible but it is also important to know the strategies of competitors or at least how they influence the market over time. Even when the strategies of all competitors stay the same over a certain time frame, time may be part of their strategy. For this reason it is important to know when to spend most of the money. The main point of interest is research that tells us on what factors it depends on whether one should spend all money evenly over the whole period or wait for a period with a lower price level.

Current research has identified different issues for campaign optimization to increase performance. The first issue is that campaigns should not run out of budget prematurely in order not to miss any extremely good opportunities on the market. Another issue that has to be somehow handled is the fluctuation in budget spending. When the budget spending varies too strongly over time it is not possible to continuously analyze the campaign to report to the customer and to further optimize it while running. [21]

While strong fluctuations are generally not good when managing a campaign there are also problems with just completely uniform distribution of spending over time. Some campaigns target primarily audiences which are available at a particular time of day. A campaign that targets school kids for example will not get many clicks and conversions during school time or late at night. Very strongly connected to the availability of target audiences is the quality of traffic which is available at any given time. It is only logical to allocate more budget to times where the campaign performs better. Existing approaches are computationally expensive or cannot function properly in the fast paced RTB environment.

An approach found in literature proposes to first split the whole time window for the campaign into time slots. [21] Each time slot gets assigned a part of the budget and the goal is to maximize the value we get out of each part of the budget. The time slots have to be chosen in a way so that the variance of the impression price stays roughly constant over the duration of the whole time slot. The total number of impressions bought is calculated by multiplying the number of requests with the pacing rate, which tells us on which ratio of the auctions to bid on and the ratio of the auctions won:

$$s(t) = requests(t) \cdot pacing\_rate(t) \cdot win\_rate(t) \tag{2.1}$$

The number of requests to expect and the win rate for the next time window are predicted through historical. The pacing rate on the other is calculated recursively from the pacing rate of the previous period and the deviations during the previous time window from the predicted number of requests and win rate. The difficulty with this approach lies in having a good plan on how to spend the budget over the whole day.

Under the assumption of a constant price for each time slot one can forecast the amount of money that will be spent for the whole period. In the case at hand the goal is to predict the click or conversion probability for each impression. Therefore it is possible to adapt the exact prediction mechanism to fit our needs and forecast a probability distribution instead of a flat probability for all impressions or use a machine learning technique to forecast the click probability. In this paper the probability function only depends on time and does not explain what other factors might be important for the scenario (e.g. country, daytime). [21] While the exact estimation procedure is not explained in detail in this paper, in reality the estimation can be radically improved by considering information about the user, publisher and advertiser as those attributes are thought to be of high importance in most other research. [13] A possibility is to combine this information with the approach just described and use multiple estimates together to provide a boosted result which incorporates all information available. An implementation of the strategy yields very good results when the actual budget spent over time is compared with the ex-post calculated budget optimally spent over time based on actual performance during each of the time slots.

Another important aspect of budget pacing is the exploration of new advertisement opportunities. The authors of the strategy presented above propose to allocate part of the budget to uniform spending over all ad impressions in order to gather data over all different kinds of advertisement opportunities. [21] They also recommend to continuously reduce exploration as the campaign progresses and optimize stronger on already gathered data. [21] Another important topic regarding pacing that is only partly covered in the above strategy is price levels in general. Many bidders on ad exchanges use inferior pacing strategies. One example is that many bidders just divide their budget over multiple days and stop bidding when the daily budget is depleted. When many competitors do this and come from the same region and thus reset their budget roughly at the same time this can lead to price levels continously dropping over the course of a day and skyrocketing immediately after midnight. [26]

2.7 Campaign Matching

One of the less researched topics in scientific literature is the matching of impressions to campaigns.This is basically just an allocation problem and is only relevant to people managing multiple, very similar campaigns as most campaigns in practice only target a very specific demographic group. Allocation is a very basic optimization problem and methods are usually universally usable. A very good example in literature is from Ye Chen et. al. who used linear programming to optimize display ad allocation. [36] The scenario was different from ours but the presented method can be used for optimization in the exact same way for the problems in this thesis. The problem statement in the paper is formulated from the perspective of publishers who are payed for advertisement space in a performance based way. This means they have to show the advertisement of the right campaign to the right user. For campaigns with unlimited budget this assignment would be trivial as the optimal way is to just always sell the impression to the campaign with the highest conversion or click probability. But under demand side constraints this mechanism is suboptimal.

The approach chosen by Ye Chen et. al. is based on linear programming while it mitigates the main flaws with a standard linear programming approach. [17] One of the main problems with standard approaches is that they often miss very strong indicators like past browsing history due to the high number of constraints. [12] Other common challenges are the intractability of budget spending in cases where there is a large number of campaigns to optimize within budget constraints and of course offline techniques do not adapt to marketplace dynamics but rather hold on to a pre-calculated strategy as it is computationally very expensive to recalculate the solution to a large linear program. [24] The adaptive algorithm described assumes that one has a procedure to estimate the click or conversion rates of ad impressions and that the estimates of the procedure converge exponentially with the number of training examples to the true value. Other important considerations are the amounts of money received from each customer when his campaign receives a click or conversion. A training phase is used to learn the empirical distribution of the impressions' click and conversion rates. An additional value α is used to control the distribution of impressions among campaigns and adaption to the price level of the market. Each campaign has an parameter α which continuously updated and always subtracted from the valuation of the impression for this specific campaign. α is positive if the campaign gets too many impressions and negative if it gets too few. Bids can only be assigned to a campaign if the resulting value obtained by subtracting α from the valuation of an impression for a campaign is positive.

The interesting part about the model is the way α is adjusted. For this purpose a proportional-integral-derivative (PID) controller is used as it has been shown that this kind of controller is optimal when there is no knowledge about the underlying process. [2] [5] What the controller does is that it adjusts α based on two different expressions which both depend on the difference between desired and observed bid winning probability. One expression only depends on the deviation for the last time window the other is an integral over the whole time frame from campaign start up until now. There are two parameters for tuning the adjustment speed due to each of the two deviation measures. A second strategy adapts the so called *Waterlevel* algorithm in which the error term is used together with a control variable as a parameter of the exponential function. [10] A third presented option is based on a statistical model to estimate the chances of winning the auction with a certain bid. This model based approach continuously updates the parameters of a probability distribution to estimate a valueα so that the campaign gets exactly the amount of impressions it needs. This strategy is highly price level oriented.

In the empirical evaluation of the three algorithms one can see that the PID and Waterlevel controller produce nearly identical results and their α values are consistently very near the optimal value of α which was computed ex ante. The probability model based strategy produces a very uneven distribution of bids over the day. Experiments also show that the start value of α that adjusts the bids in a way that the campaign budget goal is reached does not have a significant influence if the campaign budget is reasonably large. But when cutting the budget of the test campaigns by 50% it becomes apparent that historical values and other well grounded estimations for the value α outperform a starting value of 0 by a huge amount and make the difference between profit and loss. [11]

2.8 Summary

This chapter gives an overview of existing literature on which a the solution for our research problem can be built. It is meant to introduce various techniques that have already been successfully used. Later chapters will reference the literature presented here in a more detailed way where it is used by making adaptions to be usable in the scenario at hand. In this chapter we also presented the CRISP-DM framework which will guide the datamining process.

The most important points were that CTR prediction is an already well researched topic and there exist multiple approaches that either focus on the data that is straight up available or cluster the data on attributes that have high cardinality with some underrepresented values. By using clustering one can also mitigate the cold start problem and assemble a pool of training data for values that have never been encountered before or campaigns that have not started yet. The literature shows that many standard machine learning techniques can be used to predict probabilities. Interestingly all approaches we looked at made their probability estimation dependent on the campaign.

When we look at campaigns and try to make the step from a forecasted probability to a bid we saw that the valuation of an impression is directly connected with campaign pacing and coordination between campaigns. The approaches found in literature focused on simply getting a fair valuation of an impression by multiplying the click or conversion probability, depending on the campaign goal, with valuation of a click or conversion. But it is important to keep in mind that one of the most successful techniques only distinguished three different cases: bid nothing, bid the base price and bid twice the base price. Then always some kind of controller like for example a PID controller was used to bias the estimation in order to spend the whole budget smoothly over time as planned. Coordination between campaigns, like splitting impressions among them, can then be reduced to comparing these biased valuations. The campaign with the highest valuation then gets the impression.

Now the thesis will focus on what data is available and how one can make the most out of this data in order to predict good click and conversion probabilities or create a useful classification scheme. This knowledge can then use these to implement a bidding strategy. First only for a single campaign, later an approach is introduced that extends this strategy to multiple campaigns.

CHAPTER 3

The Data Set

The last chapter was about current research on mobile advertisement and bidding strategies. Classifying impressions as clicks or conversions or giving them some kind of quality score can be separated from the rest of our problem statement. We also learned that prediction performance can be forecasted using historical data. In our case we have various contextual clues which we can use for making predictions. In the literature information about the user's browsing behavior and information about the publisher have already been successfully used. [33] It will be interesting to see how good it is possible to predict click and conversion performance without any past information about the user, but methods that use only contextual clues have already shown promising results in the past. [21] We will use general statistical techniques and look for good click and performance predictors using χ^2-tests and a related statistical measure called Cramer's V. [1] This statistic measures how strongly linked two categorical values are.

We will cover the first three stages of the CRISP-DM framework in this chapter. First we will talk about the business environment in which our strategy has to function. Afterwards we will discuss how the data we work with has been gathered and how the attributes relate to each other. The next step is an evaluation of all the different attributes in the dataset. For the third step in the CRISP-DM framework which is the data preparation step it is mandatory that we identify a subset of the data that we can use to make good predictions.

In a paper published by *Microsoft Research* a *Probit* regression model is used to predict the click-rate of sponsored search advertising. [33] The *Probit* model is used for making predictions as it is able to consider the influence of multiple attributes. This regression approach indicates that it is an advantage to use as many attributes as possible to make good predictions, because all relevant features have to be included in a regression model, due to the assumptions made by such a model.

This chapter will start with an overview about the dataset we will work with and how it is structured. This is important as it determines how we need to pre-process our data for different machine learning techniques and which parts of the information is most important and which parts are redundant. For this reason we will visualize the data where possible and give descriptive statistics to better understand the structure of the data. Additionally, we will perform statistical tests to check different hypotheses on which features of an impression are good indicators for click and conversion performance.

The main hypotheses which will be tested in the first part of the chapter are:

H1 Publisher and publication ID are the best predictors for click and conversion performance.

H2 Time, specifically the weekday and the hour play a major role in click and conversion performance.

H3 Geographical features of the data set like the user's country or city have significant influence on click and conversion performance.

H4 The specific ad shown to the users influences their click and conversion behavior.

H5 The click and conversion performance is influenced by the actual device used.

H6 Click performance of an attribute's identifier (e.g. a specific city) is strongly correlated with its conversion performance.

In the second part the correlation between different attributes will be explored through a series of cross-correlation plots. This is important to gain a deeper understanding about the data set at hand and how we can best predict clicks and conversions.

Information about how data usually comes from an ad exchange and what interfaces are defined for transmitting this information will also be provided. Other important topics such as missing data which is very common as not all ad exchanges and publishers use the same conventions for some tasks and data items like tracking conversions or handling user specific data will also be considered

3.1 OpenRTB

In order to facilitate the communication for bidders and operators of real time auction platforms for advertisement impressions *iab* created an industry standard with *OpenRTB*. This general model for ad exchanges defines how communication between bidders and the ad exchange takes place. To organize communication two message objects are defined to facilitate the exchange of information. The first object is the bid request object which holds all the information a potential bidder needs to know in order to be able to make an informed bid on an impression. [20] The second one is the bid response object which is sent by the bidders who want to place a bid on an impression they received a bid request for. Both of these objects are on top of their own respective hierarchy and contain sub-objects for different types of information. Generally these objects are exchanged in the *JSON* format through web-services. In the table below you can find a summary about the most important aspects of these two objects. The purpose of this short summary is to get an overview about the type of information that can at least theoretically be distributed by an ad exchange. Irrelevant details will be left out, especially when certain information is further enclosed in sub objects. No ad exchange completely fills every single object with information as only a handful of data items are mandatory.

The tables should give an idea of what is possible within an ad exchange and give additional insight into the way a real time bidding platform operates. (cf. Table 3.1) (cf. Table 3.2) It is important to note that this specification tries to give the most general framework for exchanging bidding information possible. In reality only a fraction of the possibilities are really used in every single impression. Some construct might look overly complex but are in place in order to support the exchange of multiple bid requests or the bidding on behalf of different advertisers within a single response.

3.2 Attributes of the Data Set

Our data set is from a campaign for the Internet browser *Opera* targeted at Android users which ran over the course of 1 week. The campaign was run by the company *MobFox* on behalf of a customer. In our data set there are roughly 14.6 million impressions, 40,000 clicks and 695 conversions. Instances where the auction was lost or the ad was not shown for any reason are not included in our data set. The campaign was also already partly optimized for publishers. This means there were no bids made for impressions which would have been shown on apps that have been manually blacklisted by *MobFox* due to bad performance.

The structure of the data set with all non constant and ex-ante known data items can be seen in the following table. (cf. Table 3.3) Most attributes have a few missing values which are marked specifically in the dataset. Additionally to the data shown we also know about each impression whether it resulted in a click or conversion.

Bid Request Object	This is the top level object that contains all other objects. Additionally to all the other objects discussed below it contains a unique *bid request ID* and important meta information about the auction itself like *allowed currencies* and *maximum allowed response time*.
Impres-sion Object	This object holds all information that directly concerns the impression but none of the other categories. For example the *size of the banner* would be saved within this object. In case that the impression is within a video or has other technological dependencies like a specific *display manager* that is used, information about the technical specification of the impression opportunity can be found here. In case there is a *bid floor* in place which is a minimum amount one can bid, it can also be found in this object. Like most objects within this framework the impression can be extended by custom *JSON* code.
Site Object / App Object	All data related to the site or app on which the impression is shown is represented within this part of the bid request. Instead of the site object an app object has to be transmitted instead on mobile exchanges, which is nearly identical to a site object. Embedded in the site or app object the bidder can find the *name* of the site, the *domain*, the *category* of the app or site the impression is shown on and eventually a *search string* if the current page is the result of a search query. For an app we additionally know whether the app was *paid* for. For certain websites or apps an additional content object is embedded with information about the content shown on the page. In this object we get information about the content like the *type* (e.g. game, video, text), the *user rating* and the *producer*. Another object that resides within the app or site object is the publisher object which contains information to uniquely identify the *publisher*.
	If the impression is from a mobile application an app object is included instead of an site object which contains data about the mobile application. As this object is a nearly identical copy of the site object we can find again various categorizations of the content currently shown within the app, the ID number of the publisher and a keyword list if relevant.
Device Object	All information available about the device and its location will be transmitted within this part of the bid request. Most of this information is retrieved from the *ip-address* of the browser or mobile application. In case of mobile apps there can be additional information like a *unique device identifier* or *location* retrieved through a GPS signal included. In case of a mobile device we find the *make and model* of the device used and its *operating system*.
Geo Object	This object contains all available geographic information about the user. This can include *GPS coordinates*, the *country*, the *city*, the *region* or even the *zip code*.
User Object	User related data is enclosed inside the user object. Most fields within the user object are related to user tracking. The ad exchange can for example map IDs specified by the buyer onto users. When additional information is known because the user registered or otherwise disclosed it, like *gender* or *interests* for a social network, it is saved here.
Exten-sions Object	The extension object gives ad exchange operators a compatible way to extend the standard *OpenRTB* specification. Within this object lies a *JSON* string that can contain any additional information the exchange operator wants to include.

Table 3.1: OpenRTB Bid Request Specification

Bid Response Object	This is again the top-most object in the hierarchy which contains all other objects and information but it is a lot smaller than its request pendant. The bid response object holds the ID of the bid request it answers and it allows to set a cookie within the exchanges cookie.
Bid Object	This is one of the smallest objects in the whole specification. It only contains a *unique ID of the bid request* and the bids in the form of *seatbid* objects. This construct allows one bid response object to contain the bids of multiple bidders. An ad agency can use this to bid on behalf of their customers on an ad exchange without handling any payments to the ad exchange as their customers are directly billed by the exchange. The bid object also supports a *currency* string and setting cookies in the user's web-browser.

Table 3.2: OpenRTB Bid Response Specification

3.3 Pre-Processing

In order draw useful conclusions about the data we need to prepocess it to make more of the underlying structure visible. (cf. Table 3.3)

Our first data set is already very well prepared. The only very high dimensional data items are the publisher ID, publication ID and the time stamp. Last chapter we saw that most approaches use a different prediction model based on the publisher. Generalization is useful to generate additional data when there is not enough for a reliable model. As publication IDs are a sub set of publisher IDs they are already grouped by the publisher IDs. Where we definitely can reduce cardinality is at the time stamp. We will split the time into four new data items: weekday, hour, minute and second. It is much more probable that there are recurring patterns every day or week than a pattern that can be described by an ever increasing stream of numbers. [21]

3.4 Feature Selection

The next step is to look at the influence of the different features on the outcome of an impression, clicks in our case. To do this we ran a χ^2-test with a null hypothesis that the feature has an influence on whether there was a click or not and with the alternative of the features value distribution being independent of whether there was a click or not.

In the table below we can see the confidence level on which the null hypotheses of dependence holds. All values have been rounded to six digits after the comma.

Attribute	Description	Value Range
pub ID (also referred to as publisher ID)	This is the unique ID of the publisher on whose app the impression is shown.	1119 different identifiers
sub ID (also referred to as publication ID)	This is an identifier for the exact application the impression is shown on. Publication IDs are unique so different applications can be distinguished even without the publisher ID. Another common word used for publication ID is sub ID as publications are a subset of publishers.	5330 different identifiers
time	*EPOCH* time stamp from the moment the impression was shown.	The Campaign ran from Mon, 28 Oct 2013 00:00:00 GMT until Sun, 03 Nov 2013 23:59:59 GMT.
country code	The country in which the user was at the time of the impression	11 different identifiers
city code	The city in which the user was at the time of the impression. Only available for certain cities.	4630 different identifiers
state code	This identifies the province where the impression was shown. (e.g. Tirol, Hessen)	1734 different identifiers
carrier ID	Network carrier of the user.	54 different identifiers
ad ID	The exact advertisement that is shown. A campaign could have two different banner sizes depending on the device the advertisement is shown on.	11 different identifiers
device ID	Unique identifier of the device used by the user who saw the impression.	1108 different identifiers

Table 3.3: Description of Attributes

Attribute	p-Valuel (Clicks)	p-Value (Conversion)
pub ID	0.0	0.0
sub ID	0.0	0.0
weekday	0.0	0.247216
hour	0.0	0.0
minute	0.003472	0.727783
second	0.987536	0.152416
country code	0.0	0.0
city code	0.0	0.0002
state code	0.0	0.0
carrier ID	0.0	0.0
ad ID	0.0	0.0
device ID	0.0	0.0

Table 3.4: χ^2 Significance Tests under the Null Hypothesis of no Association between the Attributes and Click respectively Conversion Performance

As we can see in the table nearly all variables seem to have significant influence when their influence in isolation from all other attributes is measured. (cf. Table 3.4) But due to these results we can also safely remove minutes and seconds from our data set in both cases and weekdays in the case of conversions. While seconds seem to be an important factor in the conversion case this is most likely due to the fact that conversions are only reported every fully minute or sometimes even hour. As we want to reduce dimensionality even further to make the handling of the data as easy as possible we are also interested in the relative influence of the different attributes. A good measure for the strength of association between categorical variables is Cramer's V. [1] In the following table we will calculate it for each attribute and in respect to clicks and conversions. The measure itself is always between 0 and 1. Where 0 corresponds to no association at all and 1 corresponds to both variables being identical.

In the next two tables one can see the levels of association between the attributes and clicks and conversion ranked. (cf. Table 3.5) (cf. Table 3.6) While minute and second are insignificant in comparison to the other predictors for clicks as well as conversions, the weekday is only insignificant for conversions. Insignificant variables will not be used in further research.

When we look at the table regarding clicks we can observe that the strength of association slowly decreases from roughly 0.09 to 0.008 when we go from the strongest association down to the weakest. (cf. Table 3.5) When we combine this with the fact that certain attributes already contain information about others we can probably drastically reduce the number of attributes. With one attribute containing information about another we refer to the fact that certain apps already contain information about the loca-

Rank	**Attribute**	Cramer's V (Clicks)
1	sub ID	0.0907538291
2	pub ID	0.0771358208
3	city code	0.0308939453
4	state code	0.0242179301
5	device ID	0.0228596514
6	ad ID	0.0209446686
7	country code	0.0177305298
8	carrier ID	0.0152869518
9	weekday	0.012691969
10	hour	0.0081797739

Table 3.5: Cramer's V for Attributes in Relation to Clicks

Rank	**Attribute**	Cramer's V (Clicks)
1	sub ID	0.2857136058
2	device ID	0.2839416486
3	country code	0.2838646141
4	carrier ID	0.28386192
5	city code	0.0184389784
6	pub ID	0.0160796574
7	state code	0.0061697818
8	ad ID	0.0054486016
9	hour	0.0033926943

Table 3.6: Cramer's V for Attributes in Relation to Conversions

tion as they are predominantly used in only a few countries and at special times. There also exist certain subset relations, like between country and region, which are sometimes incomplete if there is no data available for one of the two sets. When looking at the table that looks at Cramer's V for conversions and the attributes from our data set we see a very different picture. (cf. Table 3.6) The first four elements, namely sub, device, country and carrier have near identical and extremely high association compared to what we saw in the click table. This might indicate that there is some kind of correlation between these four attributes. But it also tells us that it is very unlikely that we will improve our prediction much by including stat, ad ID or hour in our calculations as they are rated extremely low and their strength of association is two orders of magnitude below the first four attributes.

To gain a better understanding on how strong associations really are we will take

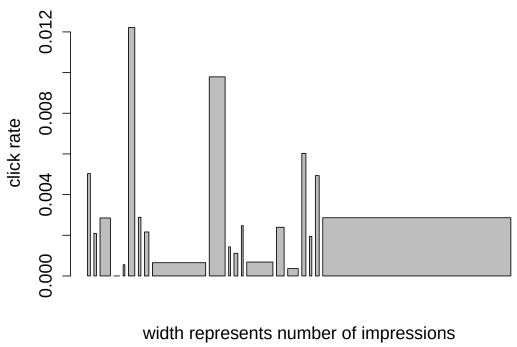

Figure 3.1: Publisher against Click Rate

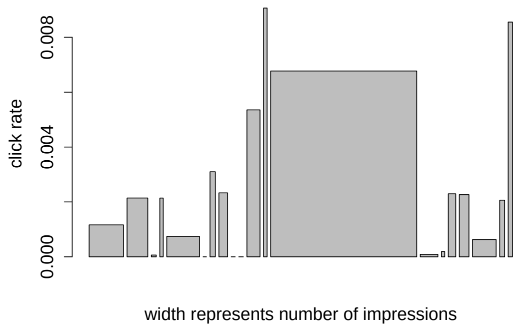

Figure 3.2: Publication against Click Rate

a look at diagrams illustrate the connection between the various features. Our focus will lie on the more significant variables as they should give us better insights in how to differentiate between impressions where users will click on the advertisements and those where they will not. The first diagrams are plots of 20 random publishers and 20 random publications with more than 10,000 observations plotted against their respective click or conversion rate. We will continue to use this scheme and always plot 20 random identifiers with over 10,000 impressions against their click and conversion rate. The only exception is the case when there are less than 20 such identifiers. In this case all of them are used. It should be noted that there are only 695 conversions in the whole data set and only 14 subs have more than 5 conversions. We will always select the same identifiers for click and conversion analysis to be able to test our hypothesis that click performance is a good indicator for conversion performance.

The click rates vary quite a bit and certain publishers have far higher click rates than others if publishers and publications are compared with their respective click rates. (cf. Figure 3.1) It is important to note that the average click rate of the campaign is 0.0027. Looking at the relation of different applications and their click rate we see again much variation. (cf. Figure 3.2) It also seems important to us that there is no connection between the amount of impressions from publishers we have in our dataset and their average click rate. This speaks for the representativeness of our dataset. The fact that some publishers from which we have very few impressions have such high click rates is most likely due to our specific sample but all in all there are very different click rates. This supports our hypothesis **H1** that publisher is one of the strongest indicators of click performance.

The next two diagrams show the conversion rate for different publishers and appli-

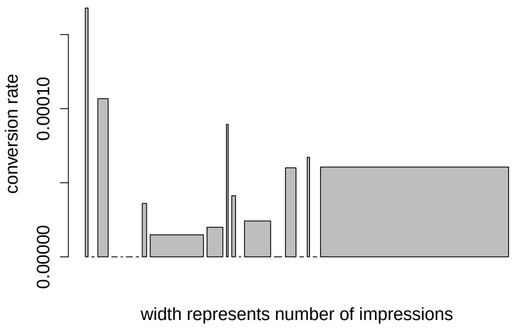

Figure 3.3: Publisher against Conversion Rate

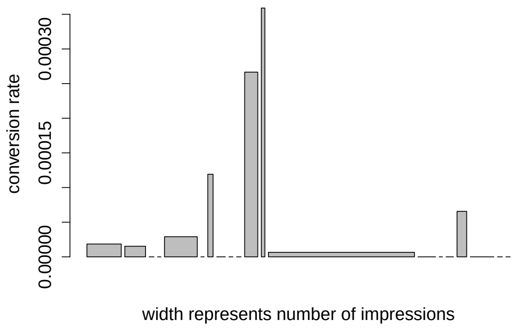

Figure 3.4: Publication against Conversion Rate

cations. (cf. Figure 3.3) (cf. Figure 3.4) We can again observe that there is no size effect. The only thing to keep in mind is that the there are only 695 conversions in total in the whole data set which amounts to an average conversion rate of 0.000047. It is very likely that the extremely high peaks are only there due to the small sample size.

To check whether the distinction into different publications makes sense we will also look at the click and conversion rates from different subs that have the same publisher. The distinction between publisher and publication makes sense as there are vast differences between different apps from the same publisher. The distinction between B1 and B2 for click rates for example makes much sense as the difference is vast and significant in terms of sample size. (cf. Figure 3.5) For conversions we can observe a similar pattern of very different conversion rates within different apps of the same publisher. (cf. Figure 3.6)

The next step is a look at other attributes and their relationship to click and conversion performance. Looking at the differences between countries you can see a click rate in Brazil which is three times higher than in Germany. (cf. Figure 3.7) It is hard to make definite statements on basis of the data from 10 countries but it seems that especially in Africa and South America click rates are higher than in Western countries, like Great Britain and Germany. It is surprising that Japan has a significant higher click rate than Germany and Great Britain as, in our case at least, countries with higher click rates are generally poorer. The fact that Japan has such a high click rate supports the theory that smartphone usage is driven by cultural factors and wealth does not play an important role. On the other hand the conversion rate of Nigeria is far higher than the conversion rate of other countries. (cf. Figure 3.8) While this may be only due to the specific sample the observation that developing countries have a higher click rate than western countries can be explained by the lack of Internet access via Internet browser.

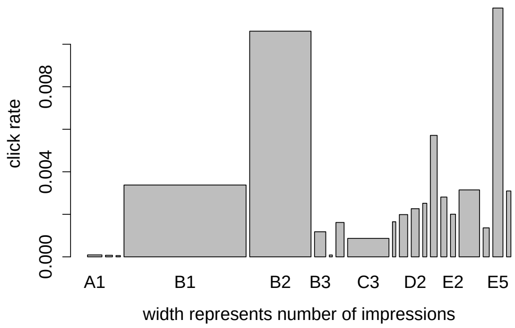

Figure 3.5: Publisher and Publication against Click Rate

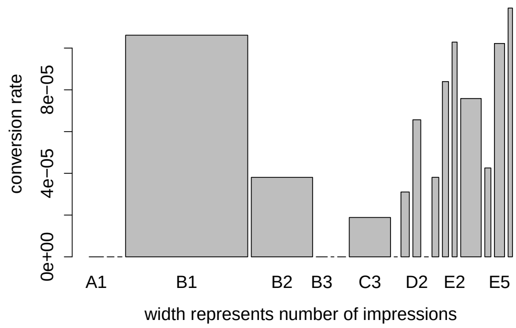

Figure 3.6: Publisher and Publication against Conversion Rate

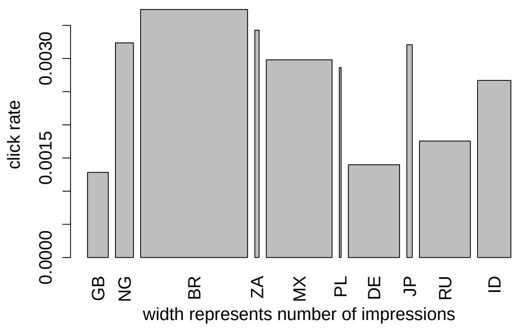

Figure 3.7: Country against Click Rate

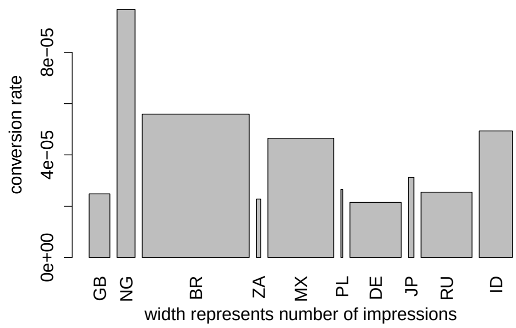

Figure 3.8: Country against Conversion Rate

So most of the buying activities are made with the smartphone. Also in this specific instance of an ad campaign for the Opera Internet browser it is important to note that Opera advertises with making browsing on slow Internet connections faster. [19] This value proposition is of course more interesting in countries with slow Internet speeds.

It might be interesting to look also at subsets of country like city and state. The diagram that compares states and their click rates is ordered by the respective countries. (cf. Figure 3.9) The first few columns all belong to Mexican states and their click rates are very different from each other. Comparing Hessen and Frankfurt in Germany it becomes clear that there has to be some level of difference in the click rates of different states in the same country. Both states have a similar and rather large sample size but vastly different click rates. It is quite interesting that all of Brazil's states have nearly

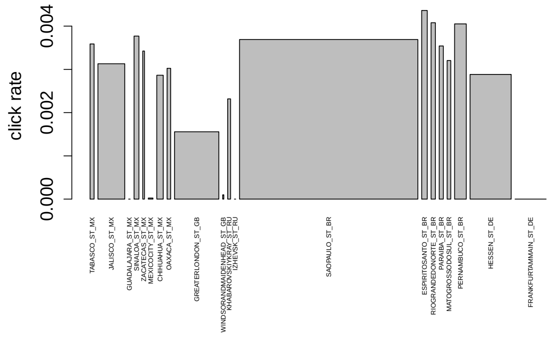

Figure 3.9: State against Click Rate

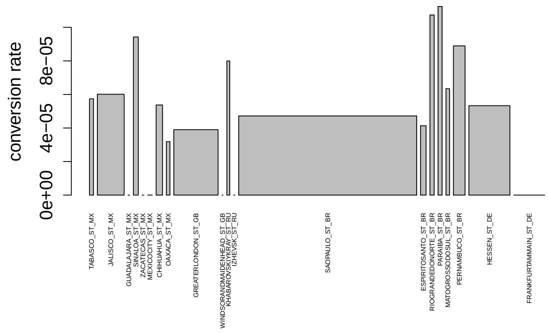

Figure 3.10: State against Conversion Rate

identical click rates. The conversion rates of different states show a similar picture as before. (cf. Figure 3.10) Conversion rates seem highly correlated with click rates when we compare this diagram with the previous one. Even though there is a higher variance due to the small number of conversions we have in our data set our observations we made about clicks also hold here.

A more specific attribute is the city as it encompasses a very small geographic region compared to a country or a state where it is not unlikely that smartphone usage patterns are very similar in the whole region. When we compare the click rate of different cities from the same country we see that there are clear differences. (cf. Figure 3.11) In Russia for example there are cities with no clicks at all and cities with a very decent click rate. The same holds for Great Britain where Slough has a very average click rate but London did not record even a single click. For Germany, Mexico and Brazil on the other hand click rates seem constant except for some minor disturbances which is expected. The conversion rates look again strongly correlated with the click rates. (cf. Figure 3.12) In Russia and Great Britain they are again very different while they are very similar in the cities of Germany, Mexico and Brazil. The disturbances are again expected to be larger due to the small number of conversion in total.

It would be interesting to look at another subset of country, the carrier of the mobile network, the user uses. Maybe the branding of specific carriers already attracts a specific kind of customer which clicks on a certain advertisement more often. Looking at the actual diagrams depicting the carrier against the click rate there is surprisingly low variance among the different click and conversion rates. (cf. Figure 3.13) (cf. Figure 3.14) Carriers from the same country have nearly identical performance. The carriers in Great Britain for example have very similar click rates and conversion rates. They only differ slightly as one or two conversions is enough for a small carrier to have an

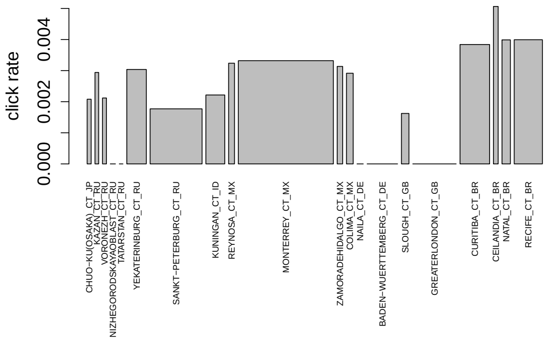

Figure 3.11: City against Click Rate

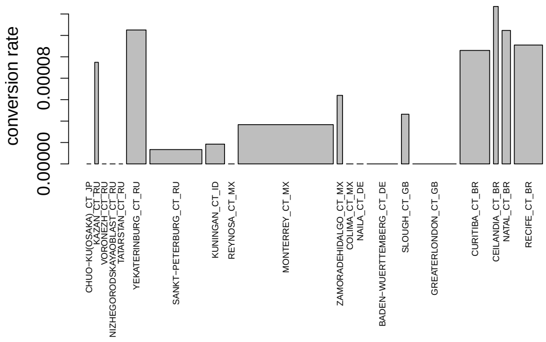

Figure 3.12: City against Conversion Rate

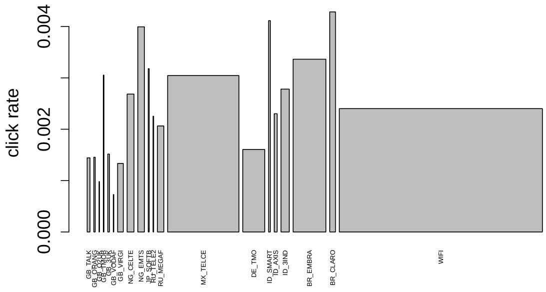

Figure 3.13: Carrier against Click Rate

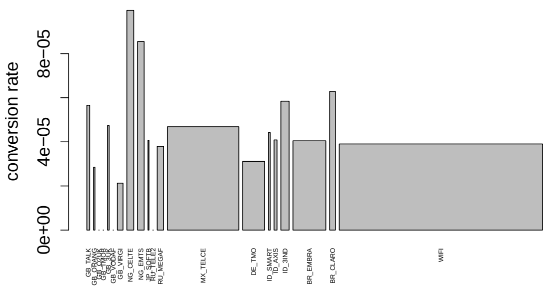

Figure 3.14: Carrier against Conversion Rate

above average conversion rate. In Nigeria, Russia and Indonesia we can see that the differences in click and conversion rates between different carriers from the same country are negligible. Compared to cities and states carriers are actually not a good way to split mobile phone users of the same country into smaller more homogeneous groups. Which carrier somebody chooses does not seem to have any influence on their click and conversion behavior at least in our case. But this might be the case due to the particular product advertised. Internet browser users are a very broad category of people. The results might be very different if we had a product like a luxury car brand's application, a game or a dating application.

Another interesting attribute is the device used while the impression is seen. Devices might contain valuable information about the user. Certain devices are targeted at wealthier individuals which demand high processing power and large screens for their

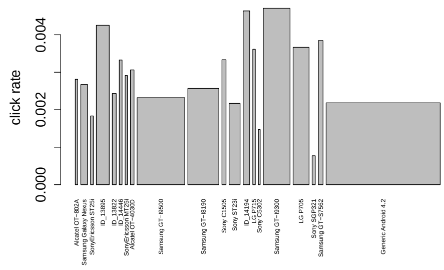

Figure 3.15: Device against Click Rate

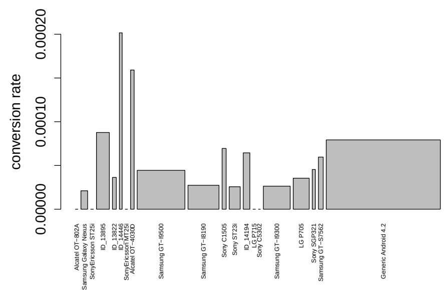

Figure 3.16: Device against Conversion Rate

money while other devices are made to be as cheap as possible. It would not be surprising to see that the device influences consumer behavior significantly. By looking at the diagram that compares different devices and their click rates, observe that most devices' click rates are very close to the average click rate with a few exceptions. (cf. Figure 3.15) The phones that have exceptionally low click rates are cheaper ones while the Samsung GT-I9300 was Samsung's outdated flagship phone at the time the campaign was shown and still cost over 300 Euros. Its successor the GT-I9500 on the other hand which cost around 500 Euros at the time the campaign ran has a very average click rate like all other Samsung phone in our sample. There seems a rather complex pattern going on if there is a pattern at all. The two Samsung phones mentioned before have near identical features, are the same brand and have still very different click rates. Without seeing a clear pattern that makes sense in the real world it is dangerous to use this association as this could be just a pattern of random correlation that changes drastically over a short time frame and is therefore unusable in a real world scenario. The relation between conversion rates looks again very random. (cf. Figure 3.16) The high peaks in the beginning can again be explained by the low amount of conversions in the data set. The only really interesting thing is that the Generic Android 4.2 seems to have a large amount of impressions and thus a smaller variance and is still above average. Generic Android versions are generally found in unbranded smart phones that are predominantly produced and sold in fast developing countries like China or Brazil. These phones use high end components but are sold for roughly half the price of a comparable phone from a well known brand like Samsung, HTC or Sony but they obviously often do not fulfill the same quality standards that branded phones have to lie up to. This result might actually be the result of the fact that countries like Brazil, Mexico, Nigeria and Indonesia which are expected to be the main users of such phones from the countries in our dataset, have higher conversion rates.

Two other features that we expected to play a major role in click and conversion

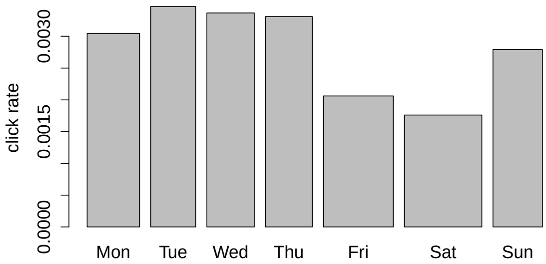

Figure 3.17: Weekday against Click Rate

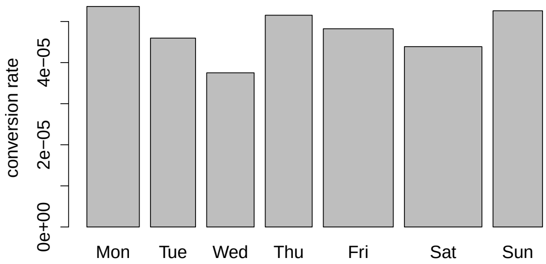

Figure 3.18: Weekday against Conversion Rate

behavior are the weekday and the actual hour when the impression was shown. During working hours for example one would generally think that people click less often on advertisements. Or late at night people might actually click more often on ads as they miss-click because they are already tired. Looking at the distribution of click and conversion rates over weekdays it is obvious that people actually click their phones less often on Fridays and Saturdays which are generally the days most people go out and are social. (cf. Figure 3.17) But on the other hand there are one and a half times as many impressions from those days, too. This might indicate that on these days people actually use their phones more often but not so much for killing time but rather to do something specific. In other words people have their standard usage for their phones but on Fridays and Saturdays there are additional things they use their phone for where they do not want to be disturbed or have time to click on something. For conversions we get a different picture. (cf. Figure 3.18) The graph of different weekdays plotted against the conversion rate seems rather constant. Even on Saturdays and Sundays there is no significant difference compared to other weekdays. This might indicate that conversions are not very context dependent but rather depend highly on the perceived actual product value that customers see in the advertised product. For conversion optimization this might mean that is increasingly important to build actual user profiles for targeted advertising like in online advertising. [13]

Surprisingly the attribute that is associated weakest with click as well as conversion performance is daytime. The exact hour at which a person sees the ad has little to no effect on the click rate. (cf. Figure 3.19) There seems to be a slight trend upwards the later it gets but compared to the other diagrams we saw the minor differences are negligible. For conversions, on the other hand, there is a more noticeable trend upward and there are some peak hours where conversions are through the roof. (cf. Figure 3.20) But it looks like the exceptional high conversion rates at 18 and 19 are just due to a sampling effect. Still there is a clear noticeable upward trend. Maybe time based effects can be observed more clearly when we filter them by country or the application used.

When evaluating our initial hypotheses we come to the following conclusions:

H1 The first hypothesis stated that publisher and publication ID are the best predictors for click and conversion performance. Our statistical tests as well as the graphs **support** this hypothesis. The only exception is that publisher and conversion performance showed rather weak association in our dataset.

H2 Initially we thought that time would play an important role in predicting click or conversion performance. Based on the low Cramer's V and the χ^2-tests we have to **reject** the hypothesis that the weekday or the specific time are important for click or conversion performance. Weekday, hours, minutes and seconds are the most weakly associated attributes in our whole dataset for clicks as well as conversions.

H3 The hypothesis regarding the significance of geographical features is definitely **supported** by the data. We saw that some countries have far higher click rates than other countries. Brazil for example has three times the click rate of Great Britain and more than twice the click rate of Germany.

H4 The specific ad we choose to show due to the banner place and device used seems to be nearly irrelevant for conversions and has an only limited influence on clicks. The influence on clicks could be due to the fact that large advertisements register more miss-clicks where the user clicked by mistake. All in all the evidence we found is **inconclusive**.

H5 The click and conversion performance is definitely influenced by the actual device used. The device ID shows high association with the conversion rate and mediocre association with the click rate. This hypothesis is **supported** by the data.

H6 As already mentioned before conversion tracking is a lot more error prone than click tracking as every customer is responsible for reporting its own conversions. We only looked at the click and conversion rate pairs for publishers and publications and there were some very strong outliers if our hypothesis was true. On the other hand we only have a handful of conversions and therefore more extreme results become more probable due to the law of small numbers. [18] Based on the low number of conversions in our dataset we can neither support nor reject the hypothesis as the evidence is **inconclusive**.

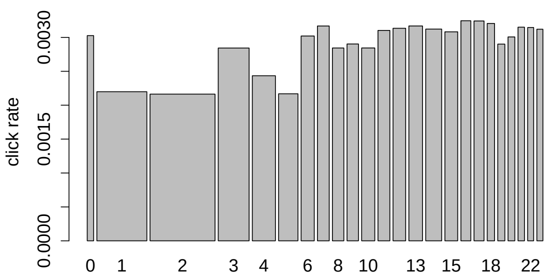

Figure 3.19: Hour against Click Rate

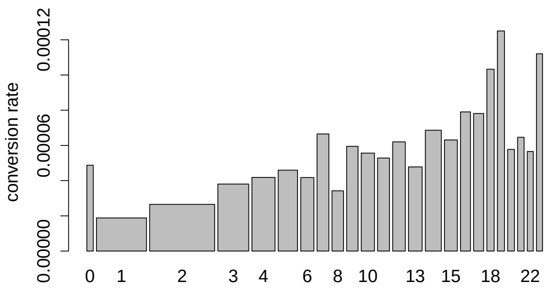

Figure 3.20: Hour against Conversion Rate

3.5 Correlation

The different attributes and their association with click and conversion performance are evident. In a dataset like ours with very low click and conversion rates we have to expect to not find any strong associations. There are valuable insights in how strong the influence of different apps, countries and devices really is.

The next step is the investigation of the correlation between the variables. One correlation can be that one variable might actually contain most of the information about another variable. An example would be that certain apps are mostly used at certain times and the fluctuating click rate over the day can be explained by the fact that at different hours different apps with different click rates are used. In this case it would be sufficient to include only the specific apps in our prediction model.

The hypotheses we have about correlations build on the observations we made in the last section. Previously we only considered a single attribute and its association with click and conversion performance. This time we will add an additional dimension as two attributes combined might explain more than each of the two attributes individually combined. We saw for example that click and conversion rates stay nearly the same over the course of the day. Maybe people just click on different apps over the course of a day, but the overall click rate stays the same. It might improve our predictions to divide the observations we have for each publication into 24 groups, one for each hour:

H7 Subs have different click rates over the course of the week.

H8 Subs have different click rates over the course of the day.

H9 Subs have different click rates in different countries.

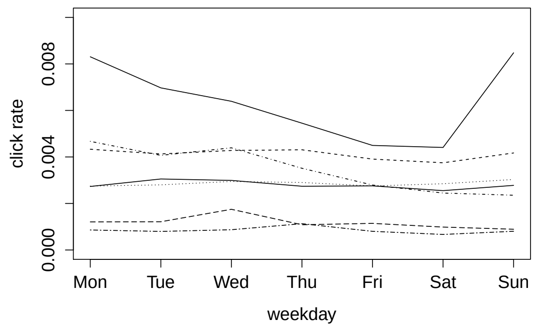

Figure 3.21: Change in the Click Rate of different Publications over the Week

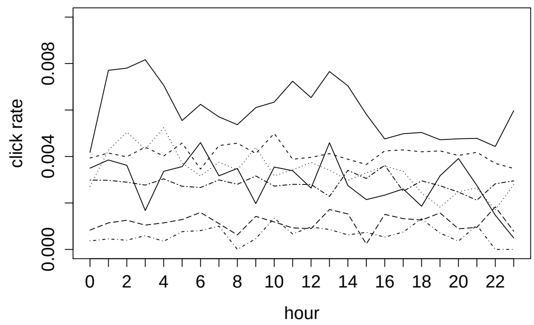

Figure 3.22: Change in the Click Rate of different Publications over the Day

H10 Click rates differ over the course of the week depending on the country.

H11 Click rates differ over the course of the day depending on the country.

As publisher and publication showed the most promising results in the first half of this chapter, the second half is concerned with their click rates in combination with additional attributes. It does not make sense to look at the same graphs for conversions as you there are only 15 subs that have more than 5 conversions. In a graph for 10 subs and their performance per weekday we would get none or only one conversion in most cases. All assertions would be pure speculation. But one should keep in mind that many attributes showed strong correlation between clicks and conversions in the first part of the chapter.

The graph shows how the click rate changes over the week. (cf. Figure 3.21) There are only seven different applications that have more than 10000 impressions on every day of the week. We see that most publications are relatively stable across the week. One exception is the app which consistently has the highest click rate as it shows a significant lower click rate at Fridays and Saturdays. All other subs only show slight variations. It definitely would be optimal to differ between weekdays for certain apps but as we already saw in the previous section there are other attributes that show greater variance in click performance.

Looking again at 7 subs which have more than 2000 impressions for each hour of the day and the development of their click rates across the day it is expected that different apps are used at different times throughout the day which may cause an ad to be shown to different people across the day. People also might not download a new

Internet browser first thing in the morning but rather later in the afternoon when they have free time. Surprisingly there is no real trend observable that certain apps lead to more clicks at certain times. (cf. Figure 3.22) For most apps there are some ups and downs involved but there are no general patterns that hold for all publications and even for single publications trends are rather weak. Taking the number of clicks for each measurement point into account none of the trends are clearly significant. When we assume a Binomial distribution with the parameters $n = 2000$ and average click rate of $p = 0.04$ we would expect the click rates to fall between roughly 0.0315 and 0.0485 on the 95% confidence level. This gives us an idea of what a significant outlier is. Most points fall within such a range of their respective means. With more data we might be able to observe more subtle trends but compared to the variation of other attributes the subdivision of publications into 1 hour time windows does not pay off.

It might also be interesting to see whether certain apps perform better in specific countries. Looking at the data the first thing one can observe is that most apps only have a significant number of impressions in two or three countries. For this reason we only selected applications which had more than 10,000 impressions in at least three countries. In the diagram we can see that again most of the complexity in the different click rates can be explained by considering the country and the app separately. (cf. Figure 3.23) There is no app in our sample that performs obviously better or worse in one specific country than what we would expect when comparing this graph to the different click rates across countries. (cf. Figure 3.7) But an important learning point was that certain countries were preselected when bidding on an impression in a specific publication as publications generally get most of their users from just a few countries.

Another hypothesis we have is that countries have different click rates over the course of the week. For this reason we will look at all countries which have more than 10,000 impressions on each day of the week. Because some countries like Japan have a very high work ethic there might be almost no people that have time to click on an advertisement during the week. This might differentiate them from other countries where people work less hours and have more time to spend on clicking on advertisements. The graph shows that there is again some degree of random variation but it is again very weak compared to what we have seen before. (cf. Figure 3.24) We have already discovered other attributes that have a stronger varying click rate and therefore will most likely be better predictors than an attribute daytime combination. The only thing in favor of actually making use of this correlation is that lots of data per measurement point are available even after dividing our impressions per country and weekday.

As with different weekdays having higher click rates in specific countries there is a similar hypothesis about daytime. Southern countries for example are well known to

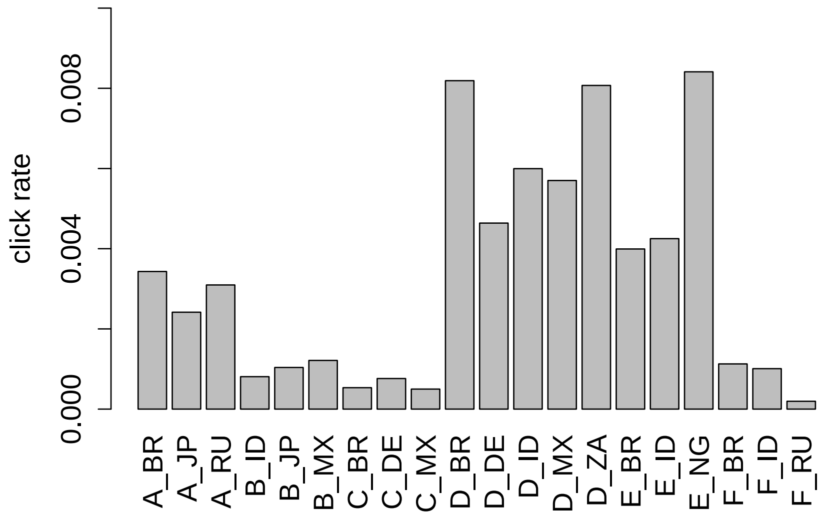

Figure 3.23: Click Rate of different Publications in different Countries

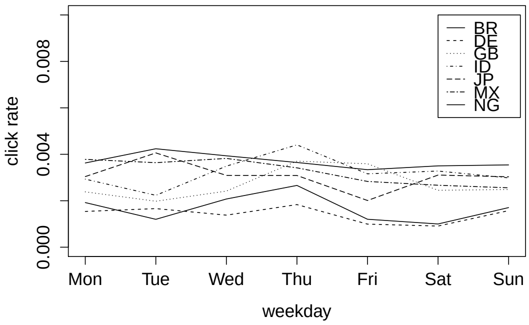

Figure 3.24: Click Rate in different Countries across the Week

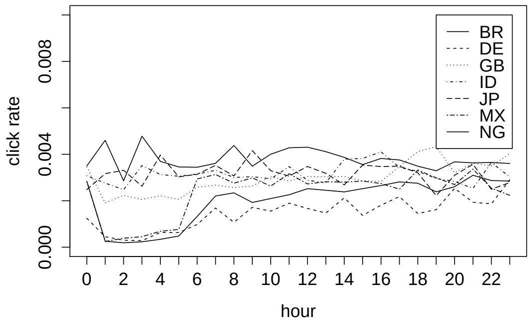

Figure 3.25: Click Rate in different Countries across the Day

start the day a few hours later such that their work and leisure time are pushed back further into the afternoon and evening. The graph shows the development of the click rate over the day in different countries. (cf. Figure 3.25) The click rate is again rather constant across time except for some countries which have significantly less clicks between one and five o'clock in the morning. It would be interesting to find an indicator that can tell us for which country this occurs and also why. Important reasons that people are asleep during those hours might be the working culture or the law.

When we look at our initial hypotheses for this section we come to the conclusion that they might hold if we had more data but on the evidence we have now we have to **reject** all seven hypotheses. Even with more data available we would be able to observe very minor trends at best.

3.6 Summary

After looking at all the different attributes and their influence on the click and conversion rate as well as different correlations between the attributes we can say that there is definitely some systematic component to click and conversion performance. Many attributes when looked at in isolation make it possible to distinguish between higher and lower probabilities of a click or conversion. Especially publisher, publication, geographic attributes and device ID showed high association with click and conversion performance. Correlations are rather weak and do not seem to offer much additional explanatory power with the data we have. Many hypothesis we had in the beginning are supported by the data we have while some are not. All our hypothesis that assumed a strong influence of time on click and conversion performance should be discarded based on what we could observe. The χ^2-tests were significant for nearly all attributes but that was to be expected in such a large sample in which even smallest systematic associations lead to very small p-values. Hypotheses that make sense in a real world scenario should be the backbone of our attempts to find a way to predict click and conversion performance, not pure statistical correlation which can change in an instant.
We also covered the first three steps in the CRISP-DM methodology. We explored the ad exchange protocol and gained a deeper understanding about the dataset when we looked at graphs comparing the influence of different attributes on click and conversion performance. By using Cramer's V we ranked the different attributes by their association with click and conversion rates. This will become important when we select subsets of the data as training data. In the next chapter we target ways to predict click and conversion performance on the basis of the data set we investigated in this chapter. A variety of machine learning methods will be used to see which methods perform best in classification. The sampling mechanism will also be varied as we are dealing with a highly unbalanced data set when we compare clicks and non-clicks. We will have to use many different tools to optimize our click and conversion prediction performance on this complex dataset.

CHAPTER 4

Prediction Model

In the previous chapter descriptive statistics were used to describe the data set at hand and gain insight in the dynamics behind click and conversion performance. The attributes with the highest influence on clicks and conversions were determined. This chapter is about using this newly gained knowledge to construct a machine learning model with which the bidding on impressions will be more efficient. For building such a model various machine learning algorithms will be discussed and also different training methods will be used with those algorithms. The emphasis of this chapter lies in the exploration of which methods yield the best results in click and conversion forecasting and which attributes yield the best results when used in combination with these methods. Looking at the strength of the association of different attributes in isolation there is still the possibility that including many weakly associated attributes improves prediction performance beyond what is possible with only a few strong ones. Dealing with real world data there are also many associations between the different data items. As discussed already in the last chapter, attributes like country, region, city and the app used are strongly dependent on one another. We will see which method uses these associations best.

For classification many different approaches have been used successfully. One of them is the Naive Bayes method which shows extremely good performance in tasks such as spam filtering which is similar to our problem. We also have to deal with many categorical variables, some with only a handful of observations. Support Vector machines have already been used for many different tasks like handwriting recognition, but heavily depend on data preparation. We will use these techniques to solve our classification problem. [29]

We are now at the beginning of the modelling stage of the CRISP-DM methodology. This stage consists of the exploration of different modelling techniques. A variety of different algorithms that can work with categorical data like ours will be used:

The first algorithm is a *Naive Bayes* classifier. [22] It is based on Bayes Theorem which when applied to our problem looks like this:

$$P(Click|Information)P(Information) = P(Information|Click)P(Click) \tag{4.1}$$

$$P(C|I) = \frac{P(I|C)P(C)}{P(I)} \tag{4.2}$$

$$P(C|I) = \frac{P(I|C)P(C)}{P(I|C)P(C) + P(I|\neg C)P(\neg C)} \tag{4.3}$$

The term *Information* stands for all the information we get ex-ante about the impression we are about to bid on like the user's device, his country or the app. The reason this theorem is so powerful is that it allows us to relate the probabilities we are interested in but which are difficult to directly estimate to probabilities we can estimate a lot easier. The probability of having a click ($P(C)$) and the probability of not having a click ($P(\neg C)$) can be easily estimated by using the average click rate of our data set. Similar methods can be applied for estimating the probability of observing this exactly same set of information where we can look what fraction of impressions with this exactly same set of information resulted in a click. This procedure alone would not make things easier to estimate as we still would need impressions with that exactly same information set. For this reason the classifier makes a *naive* assumption and assumes that all attributes are independent from one another. This leads to the following formula which is easier to compute:

$$P(C|I) = \frac{P(I_1|C)P(I_2|C)...P(I_n|C)P(C)}{P(I_1|C)P(I_2|C)...P(I_n|C)P(C) + P(I_1|\neg C)P(I_2|\neg C)...P(I_n|\neg C)P(\neg C)} \tag{4.4}$$

The I_is stand for the different pieces of information like country or weekday. As we assumed independence of those different features the probability of having a click from Brazil on a Tuesday is the same as the product of the probability of having a click on Tuesday and having a click from Brazil. This means that we can forecast for impressions where we have never seen this exact set of features and it also drastically simplifies our model. Whether this is good or bad will be shown later when we evaluate the different algorithms.

Next we will discuss the *Random Forest* classifier. A random forest is based on a collection of decision trees. [6] Therefore to understand how a Random Forest works one has first to understand how a decision tree works. A decision tree splits samples at each node depending on a specific attribute until the sample reached a leaf node. In our case all samples reaching a specific leaf node would be classified as either a click or not a click. There are variants that allow only binary trees but that does not matter as every decision tree can also be rewritten as a binary decision tree. [7] The idea behind a Random forest is to train multiple decision trees and let them vote for a class label. The problem is that when all decision trees are trained on the same dataset with the same method we will get the same tree over and over again. Therefore either the feature selection is randomized or only a sub sample of the whole dataset is used to train each like in our implementation where two thirds of the dataset are randomly chosen for the training of each tree. Generally the goal is to have many well diversified trees that generalize well on unseen data.

The last method we will consider is the *Support Vector Machine*. A *SVM* sees each sample as a data point in a very high dimensional space. In our case binary encoding is used to transform the samples into a series of 0s and 1s. Each dimension is 1 if our impression belongs to a certain category at a certain attribute. The first number might indicate whether our impression comes from Brasil, the second whether it comes from Germany and so on until the eleventh indicates whether it was recorded on a Monday, the twelfth whether it occurred on a Tuesday until we have one number for each value of each attribute. This way of preparing data is also known as the Kernel Trick. [30] The SVM then fits an $n-1$ dimensional hyperplane through this n dimensional space to divide the space into two half-spaces where one contains mostly clicks and the other all the rest of the impressions.

The next important aspect are the different subsets of the dataset we are going to use. A bigger dataset does not always yield better results. We will vary two things in our subsets:

- Attributes: We will use different combinations of attributes to find a set of attributes that works well with the different classifiers.
- Number of Clicks: We have a click rate of roughly half a percent. Obviously one of the best strategies would be to always predict a non-click and one gets over 99.5% of all impressions right. Different training datasets will be used which vary how of often the different clicks and non-clicks are included. The following different sampling methods will be used. Their names will be used throughout the rest of the thesis for better distinction:
  - Normal: In this sampling procedure all impressions will be included exactly as they are a single time.

- Reduced: In this sampling procedure clicks or conversions are included exactly once and all non-clicks or non-conversions are removed except every nth one such that there are as many clicks as non-clicks or as many conversion as non-conversions.
- Balanced: In this sampling procedure all non-clicks will be included one time and all clicks exactly as often such that we get the same amount of clicks and non-clicks in our training set. The same hold for conversions. For the dataset at hand this means a weight of 368 for clicks and a weight of over $21,000$ for conversions.

4.1 Classification

In this section the predictive power of the different classifiers mentioned before will be tested. Unless specified otherwise all experiments performed will be using 10-fold cross-validation. The 5 subsets shown in the table below will be used. (cf. Table 4.1) The thought process behind those specific subsets is that they capture different strategies in feature selection. The first data set uses all information available. It would not be unexpected if machine learning algorithms actually performed best on the set with the most information. The next dataset is included as *MobFox* currently estimates their click probabilities only by using the average click rate of each app. This simple strategy produces suprisingly good results in reality. The third dataset uses the six attributes that are most strongly associated with click performance. (cf. Table 3.5) There is no sudden drop in strength of association after six attributes but the rest of the attributes' information is already implied by other attributes or is very weak as we could see in the diagrams in the last chapter. The next dataset only uses five attributes as publisher and publication have a 1 to n relationship. This means there is an extremely strong dependence between the two attributes and some algorithms might perform worse in the presence of highly correlated variables. These two data sets will only be used for click performance forecasting. The last dataset covers the four attributes that are most strongly associated with conversion performance. (cf. Table 3.6) These 4 attributes have a near identical, very high association with respect to Cramer's V. The fifth strongest association is already weaker by more than a factor of 10. This vast difference presents a good cut-off criterion.

At first the different algorithms on each dataset are compared. The goal is to get an understanding on which algorithms perform best. But also how the algorithms compare against each other. The two most important statistics to look at are precision and recall. Precision refers to the ratio between the correctly predicted clicks and the total number of impressions predicted to be clicks, while recall refers to the ratio between correctly predicted clicks and total amount of clicks. Both values are between 0 and 1 where 1

Attributes Used	Short Notation
pub ID, sub ID, country code, state code, city code, carrier ID, device ID, ad ID, weekday, hour	full
sub ID	sub_only
pub ID, sub ID, state code, city code, device ID, ad ID	click_top6
sub ID, state code, city code, device ID, ad ID	click_top5
sub ID, country code, carrier ID, device ID	conversion_top4

Table 4.1: Attribute Subsets used

would be the optimum. Obviously it would be perfect to find a prediction method that reaches high precision and recall and thus can perfectly predict clicks. It is much more likely that we find strategies with either high precision or high recall. We should not forget that we want as many clicks as possible for the best price possible. It is bad if we bid on too few clicks but it is also bad if we are not selective enough with the impressions we bid on. We have to make a trade-off between precision and recall.

If a classifier could predict with high precision for example this knowledge can be used to filter out a group of impressions that has very high potential of being clicks and one can bid higher on them. If on the other hand the classifier has high recall it is possible to not bid on the impressions classified as non-clicks or non-conversions and bid higher on those classified as clicks or conversions. By looking at the table below one can see how the different algorithms perform when predicting clicks on a dataset only consisting of sub IDs. (cf. Table 4.2) In our case true positives are clicks correctly classified as clicks, true negatives non-clicks correctly classified as non-clicks, false positives are non-clicks classified as clicks and false negatives clicks not classified as such. It is important to note that we cannot calculate the precision when both true as well as false positives are 0 as the ratio $\frac{0}{0}$ is undefined.

When comparing the three algorithms for the dataset that used normal sampling and thus only had very few clicks compared to non clicks one can observe that all three algorithms come to the roughly same result. Extremely few impressions classified as clicks and none of them correctly and the vast majority of impressions classified as non-clicks. It has already been mentioned that this strategy is obviously the best to classify the highest percentage of impressions correctly but useless for our goals. The reduced dataset where only a very small subset of non-clicks are present on the other hand shows very different results for the different methods. The Naive Bayes method, for example, has rather low precision and a pretty high recall. This finding makes sense as it recognizes most clicks but has some trouble differentiating between clicks and non-clicks due to the lack of non-clicks in the sample. The Random Forest on the other hand shows similar results as with the first sampling method but has higher precision as the Naive Bayes method or the SVM. But it also has far worse recall. The SVM seems to win this round as over 91.8% of clicks are classified as such and the precision is the highest we have seen yet. The last sampling method includes all clicks multiple times such that there is an equal amount of clicks and non-clicks. The Naive Bayes classifier has roughly the same performance as before and thus does not seem to suffer from the reduced amount of information from the last dataset. The Random Forest on the other hand finally performs on a level with the other classifiers and has near identical performance with the Naive Bayes method. The SVM surprisingly performs worse than the other two methods due to its worse recall. Compared to the previous sampling method the SVM gains precision, which was to expect due to the many new examples of non-clicks to learn from but it lost recall at the same time. When only using a single attribute the SVM functions best when using a reduced sampling method and the other two algorithms perform best with a balanced sampling method. Also the SVM outperforms the other two when one looks only at the recall. The Naive Bayes classifier and the Random Forest seem to give the best balance between precision and recall.

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	0	14599938	6	39536	0.00000	0.00000
RF	N	0	14599924	20	39536	0.00000	0.00000
SVM	N	0	14599944	0	39536	NA	0.00000
NB	R	33270	8073826	6526118	6266	0.00507	0.84151
RF	R	93	14597010	2934	39443	0.03072	0.00235
SVM	R	36296	6039297	8560647	3240	0.00422	**0.91805**
NB	B	33086	8182433	6417511	6450	0.00513	0.83686
RF	B	33174	8148943	6451001	6362	0.00512	0.83908
SVM	B	25739	9613591	4986353	13797	**0.00514**	0.65103

Table 4.2: Sub-only Dataset Prediction Performance

Next one should look at the performance of our three algorithms when using the dataset that contains the top 6 attributes in our dataset. (cf. Table 4.3) When the normal sampling method, which just takes the dataset as it is, is used the only algorithm that actually produces something useful is the Naive Bayes. The other two algorithms both have recall 0 which makes them infeasible in this case. When we look at the reduced sampling method the Naive Bayes method performs nearly identical to when we used only sub IDs for prediction. It has again decent recall and low precision. Also the Random Forest predicts again nearly everything as non-click. The SVM increases its precision compared to the previous dataset but reduces its recall slightly. Finally when using the balanced sampling method for this dataset we can observe that the Naive Bayes method again performs nearly identical to when we used the previous dataset. The Random Forest has reduced recall and increased precision compared to when we used only the sub IDs for prediction. The SVM reacted similar. It has significantly reduced recall from all previous useful experiments but shows extremely high precision compared to everything else we saw before. While this dataset did not significantly improve all our algorithms like we expected it gave a few interesting results. The SVM configuration found in this part of the experiment has high precision while still having nearly 18.4% recall.

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	1405	14572505	27439	38131	**0.04871**	0.03554
RF	N	0	14599796	148	39536	0.00000	0.00000
SVM	N	0	14599944	0	39536	NA	0.00000
NB	R	33087	8037618	6562326	6449	0.00502	0.83688
RF	R	6	14599446	498	39530	0.01190	0.00015
SVM	R	34612	7589901	7010043	4924	0.00491	**0.87546**
NB	B	32329	8241294	6358650	7207	0.00506	0.81771
RF	B	23311	11072935	3527009	16225	0.00657	0.58961
SVM	B	7256	13943749	656195	32280	0.01094	0.18353

Table 4.3: Click-Top6 Dataset Click Prediction Performance

One hypothesis to check is that by not including the publisher ID in the data set one can improve the performance of some classifiers as publication is a subset of publisher. Therefore the dataset used in this part of the experiment has the exact same attributes as the one before except that it is missing the publisher ID. (cf. Table 4.4) When looking at the results of the normal sampling method one can again observe that the Random Forest and the SVM again classify nearly all impressions as non-clicks. The Naive Bayes classifier on the other hand actually performs worse than when we included the publisher ID. For the reduced sampling method all three algorithms performed similar to the case where we used all 6 attributes. The Naive Bayes classifier has slightly reduced precision and recall. The Random Forest does still not produce good results and the SVM has slightly reduced precision but also slightly increased recall. For the last sampling method the Naive Bayes method has near identical results compared to the performance on the previous dataset but the classifier still performs a bit worse. The same holds for the Random Forest. The SVM gives us again high precision like with the previous dataset but a little bit less than before but on the other hand it has higher recall than before. The dataset consisting of the top 5 attribute actually does not change our results much when compared to the dataset that used the top 6 attributes. All changes are very small and more negative than positive.

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	34	14598052	1892	39502	**0.01765**	0.00086
RF	N	0	14599765	179	39536	0.00000	0.00000
SVM	N	0	14599944	0	39536	NA	0.00000
NB	R	32749	7828412	6771532	6787	0.00481	0.82833
RF	R	7	14599255	689	39529	0.01006	0.00018
SVM	R	35083	6954088	7645856	4453	0.00457	**0.88737**
NB	B	31978	8014639	6585305	7558	0.00483	0.80883
RF	B	21370	11231200	3368744	18166	0.00630	0.54052
SVM	B	8257	13760245	839699	31279	0.00974	0.20885

Table 4.4: Click-Top5 Dataset Click Prediction Performance

Last but no least the whole dataset will be used for training our classifier. (cf. Table 4.5) Again for the normal sampling method the Naive Bayes classifier is the only method to produce acceptable results. The other two algorithms put nearly everything into the non-click category. For the reduced sampling method the Naive Bayes classifier and the SVM perform very similar to what we saw with the last datasets. Compared to each other the SVM is more precise and recalls more clicks and it also has slightly lower recall compared to when used with the dataset consisting of only the top 6 attributes but with slightly higher precision. The Random Forest is again not making useful prediction when using this sampling method. When looking at the balanced sampling method it should be noted that the Naive Bayes classifier has the highest recall and the lowest precision. The SVM has the highest precision and the lowest recall and the Random Forest is in between the other two algorithms when considering precision and recall. Here one has a clear trade-off between precision and recall and should choose the algorithm accordingly. All in all our findings with the last 4 datasets showed that the Naive Bayes method works equally well with reduced as well as balanced sampling, while the SVM has the highest recall with reduced sampling and the highest precision with balanced sampling. The Random Forest only works reliably with balanced sampling in the two other cases we just get 99.9% non-clicks and a few false positives. Generally it can be assumed that Naive Bayes with either balanced or reduced sampling gives us good recall, while the SVM with reduced sampling gives us even slightly better recall and with balanced sampling produces high precision results. While looking at attributes other than sub ID our predictions improved for the SVM but not for the Naive Bayes classifier and the Random Forest. This supports our hypothesis that the sub ID already contains a very large amount of information about click performance. When searching for the best classifier it depends strongly on the trade-off between precision and recall one is willing to make and how much computational power is at hand. Generally the Naive Bayes method gives us the best recall with a reduced or balanced training set, only the SVM with the reduced sampling method is still a bit better but needs far more time to train the model. The highest precision results are produced by the SVM and a balanced sampling method. The Random Forest gives us a trade-off between precision and recall and is in between the other two algorithms.

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	1516	14566931	33013	38020	**0.04391**	0.03834
RF	N	0	14599700	244	39536	0.00000	0.00000
SVM	N	0	14599944	0	39536	NA	0.00000
NB	R	32402	7913492	6686452	7134	0.00482	0.81956
RF	R	0	14599624	320	39536	0.00000	0.00000
SVM	R	34010	7893023	6706921	5526	0.00505	**0.86023**
NB	B	31738	8084170	6515774	7798	0.00485	0.80276
RF	B	19007	11918244	2681700	20529	0.00704	0.48075
SVM	B	4389	14309072	290872	35147	0.01486	0.11101

Table 4.5: Full Dataset Click Prediction Performance

The next thing to look at is how the different algorithms perform when used for conversion performance forecasting. The first dataset again only contains the sub IDs. Maybe this simple strategy can be improved later on better than for clicks. When looking at normal sampling one can see that the Naive Bayes method and the Random Forest perform identically. (cf. Table 4.6) They only find roughly 8% of all conversions but with precision 1 which is extremely useful for filtering out very high potential conversions. The SVM machine does not classify any impressions as conversions which is consistent with our observations earlier when trying to classify into clicks and non-clicks. Reduced sampling again leads to the Random Forest classifying everything as a non-conversion. The Naive Bayes and SVM both perform with relatively high recall and extremely low precision which was to expect as the training samples only contained a tiny fraction of non-conversions. Surprisingly the Naive Bayes classifier outperforms the SVM in both precision and recall. The balanced sampling method leads to the SVM performing identically to the Naive Bayes and Random Forest with normal sampling. When considering the computational effort needed to train a SVM this is a disappointing result. The Naive Bayes classifier and the Random Forest perform nearly identically. They both have slightly reduced recall compared to the Naive Bayes with reduced sampling but they also have slightly increased the precision.

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	56	14638785	0	639	**1.00000**	0.08058
RF	N	56	14638785	0	639	**1.00000**	0.08058
SVM	N	0	14638785	0	695	NA	0.00000
NB	R	553	8468513	6170272	142	0.00009	**0.79568**
RF	R	0	14638785	0	695	NA	0.00000
SVM	R	526	7506693	7132092	169	0.00007	0.75683
NB	B	538	9758904	4879881	157	0.00011	0.77410
RF	B	536	9600155	5038630	159	0.00011	0.77122
SVM	B	56	14638785	0	639	1.00000	0.08058

Table 4.6: Sub-only Dataset Conversion Prediction Performance

When looking at the top 4 attributes one can observe that for normal sampling all three algorithms have identical recall. (cf. Table 4.7) But they differ significantly in precision. The Naive Bayes has the lowest precision while the SVM has the highest. It is interesting to see that the three algorithms probably again found all the same 56 conversions which seem easiest to distinguish from the rest. When comparing the performance of the three methods on the reduced samples the Random Forest again does not produce meaningful results. The Naive Bayes classifier and the SVM both perform very well. The Naive Bayes method performs even a bit better. While these results are pretty good they are not as good as the performances accomplished with the previous dataset and the balanced sampling method. With the top 4 dataset the SVM again finds 56 conversions with precision 1. The Naive Bayes on the other hand shows relatively high precision and acceptable recall. The Random Forest on the other hand has even higher precision but drastically worse recall. Again the pattern emerges that the Naive Bayes classifier has the highest recall but the worst precision and the SVM the highest precision and lowest recall. Generally this larger dataset produced better results than the previous one.

Finally the three algorithms will work again with the full dataset and hopefully show their best performance yet. (cf. Table 4.8) For the normal sampling method the results are similar as before. All three classifiers recognize 56 conversions and thus have identical recall. The SVM has again the highest and the Naive Bayes classifier the lowest precision. When considering the Naive Bayes performance for the reduced samples one can see that precision and recall are very high. While previous experiments produced similarly high or even higher precision no other experiment yet recalled such a high absolute number of conversions. The Random Forest again cannot work with this dataset. The SVM performs a little bit worse than the Naive Bayes classifier but still very decent. When looking at the performance for the balanced dataset one gets the same picture as

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	56	14638695	90	639	0.38356	0.08058
RF	N	56	14638777	8	639	0.87500	0.08058
SVM	N	56	14638785	0	639	**1.00000**	0.08058
NB	R	481	10188175	4450610	214	0.00011	**0.69209**
RF	R	0	14638599	186	695	0.00000	0.00000
SVM	R	462	8822842	5815943	233	0.00008	0.66475
NB	B	437	11097459	3541326	258	0.00012	0.62878
RF	B	223	13137933	1500852	472	0.00015	0.32086
SVM	B	56	14638785	0	639	**1.00000**	0.08058

Table 4.7: Conversion-Top4 Dataset Conversion Prediction Performance

before. The Naive Bayes has again the highest recall and lowest precision and vice versa for the SVM. For predicting conversions the SVM machine performed surprisingly badly. It only ever found the presumably same 56 impressions and also only with a balanced training set. The results of the Naive Bayes classifier and the Random Forest that using a normally sampled training set containing only the sub IDs were identical to the results seen here. Disregarding the case where only 56 conversions where correctly classified as such with precision 1 the Random Forest with the top 4 dataset and balanced sampling gives us the highest precision. But the Naive Bayes gives us nearly the same precision with highly increased recall on the same data.

Algorithm	Sampling	TP	TN	FP	FN	Precision	Recall
NB	N	56	14635856	2929	639	0.01876	0.08058
RF	N	56	14638772	13	639	0.81159	0.08058
SVM	N	56	14638785	0	639	**1.00000**	0.08058
NB	R	542	9374853	5263932	153	0.00010	**0.77986**
RF	R	0	14638771	14	695	0.00000	0.00000
SVM	R	471	9890309	4748476	224	0.00010	0.67770
NB	B	385	11504486	3134299	310	0.00012	0.55396
RF	B	82	14526873	111912	613	0.00073	0.11799
SVM	B	56	14638782	3	639	0.94915	0.08058

Table 4.8: Full Dataset Conversion Prediction Performance

For our purposes the best algorithm is a Naive Bayes classifier. It shows consistently very good results and balances precision and recall very well. The SVM shows consistently high precision and also high recall under the right settings. While the SVM beats the Naive Bayes method under the right settings the differences are very minor and the SVM shows very high dependence on the exact input data. The Naive Bayes classifier's overall performance was the best or close to the best with all different datasets. It also has the advantage that it scales very well with more complex datasets and can be trained faster than the SVM. In the next section we will use the Naive Bayes classifier to distinguish incoming impressions into a high performance and a low performance group.

4.2 The Bidding Strategy

After the characteristics of the different training methods have been discovered we are in phase five of the CRISP-DM process. The goal now is to find a good bidding strategy that achieves the business objective of being profitable. A bidding strategy can only be evaluated when compared with another bidding strategy and under certain assumptions about the overall goals of the bidder. The start is improving our reference strategy and change it step-wise to create a new improved bidding strategy. Additionally different modifications will be presented that can be made to react to different optimization goals. One assumption is that the bidder gets 1 € per successfully bought click. Another assumption is that the bidder only has one campaign to optimize for simplicity. Our reference strategy always bids the average click-rate (or conversion-rate) of the sub ID of the current impression in Euros. That means that it would bid 0.05 € on an impressions whose publication has an average click-rate of 5%.

In the first experiment the baseline strategy will be changed by only bidding on a selected few impressions. That means a modification on which impressions to bid but not how much is bid on each. This will be accomplished by training a Naive Bayes classifier with a dataset that contains all attributes except publisher and publication as these attributes are used to calculate the amount to bid. The Naive Bayes classifier was chosen as it gives us the best trade-off between precision and recall. As will be clear in a moment, recall is one of the most important criteria for a good classifier. By continuously increasing the weights of the clicks and looking at the total profit and the profit per click the changes in profit and recall with changes in click weight can be observed. We increase the weight of clicks by including each click multiple times in the training dataset. For a click weight of 10 we include each click 10 times in the training dataset instead of one time. When using a Naive Bayes classifier we only need to readjust the two class probabilities as all other expressions only concern the relative frequencies within each class, which are not changed by including all data within a class an additional time. (cf. Equation **??**) It is important to keep in mind that this simulation assumes a first price auction and that each auction one places a bid on is won. In reality a second price auction is used and one does not win every impression one bids on.

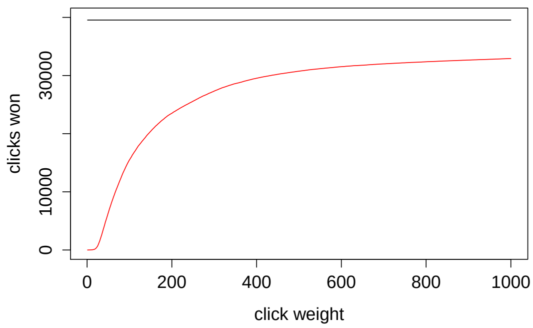

Figure 4.1: Number of Clicks Recalled against Click Weight

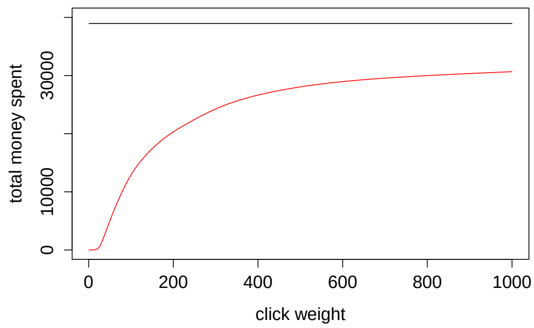

Figure 4.2: Total Money Spent against Click Weight

As expected the number of recalled clicks rises as the click weight increases as can be seen by the red line representing the modified bidding method compared to the standard bidding method indicated by the black line. (cf. Figure 4.1) Interestingly, increasing the click weight slightly has close to no effect on the recalled clicks in the beginning. Only after increasing the click weight to around 20 the number of recalled clicks rises sharply. It can also be seen that it converges in the direction of the black line which represents the standard algorithm which bids on all impressions and therefore has a recall of 1. But it is also important to observe that the invisible line the red line converges to is actually not the black line. There seems to be a slight gap of unforeseeable clicks from the dataset. All in all this diagram confirms that increasing the weight of clicks increases recall although it is impossible to make any statement about the precision yet.

Next up we look at the total profit made with different click weights. Remember the total profit made is:

$$profit = clicks\_recalled - money\_spent \tag{4.5}$$

For very small click weights the profit is close to zero. (cf. Figure 5.1) It climbs very fast over the reference strategy's total profit and reaches its peak at a click weight of just below 200. This means that one definitely needs to increase the click weight far beyond 1 to maximize the total profit. What is interesting is that even with click weights far beyond 1000 one still does better than the standard strategy. When compared with the profit per click one can see that the profit per click rises much faster with the weight of clicks but also falls off much faster. (cf. Figure 4.4) It reaches its peak at roughly 10 to 20. This indicates that it is a good idea to be very selective about the impressions to

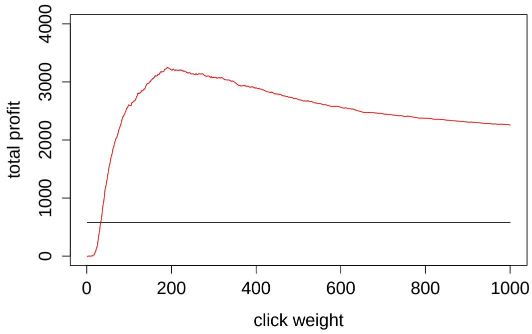

Figure 4.3: Total Profit against Click Weight

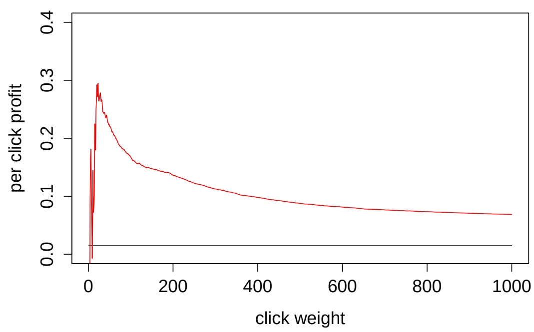

Figure 4.4: Profit per Click against Click Weight

bid on when one has a very low limit on the number of impressions one can buy.

When looking at the graph of conversions recalled against conversion weight it becomes clear that the recall is nowhere near where it was for clicks. (cf. Figure 4.5) This means that conversions are a lot less predictable than clicks and one has to be willing to accept a lower precision than with clicks. The total profit as well as the profit per conversion only climb very slowly as well. (cf. Figure 4.7) In combination these two graphs show us that optimizing for conversions is probably not very profitable as we have at most a 4 times bigger profit margin when we compare the per click and per conversion profit but a lot fewer conversions. This might only hold for our sample data but optimizing for conversions is a lot harder and seems to require different methods than optimizing for clicks due to the extreme rarity of conversion events.

The next issue of this simulation to address is that in reality one has to face a second price auction and not a first price auction like it was the case in the experiments until now. This means that one does not pay the price which was bid but the price the second highest bidder is willing to pay. This can be simulated by assuming a percentage margin upon our turnover. This tries to simulate the fact that we will never actually pay the price we bid as this is a second price auction. The amount we have to pay is always the second highest bid. By assuming a percentage margin upon our turnover we simulate that the final price we pay is always a certain percentage below our bid. This assumption allows us to continue our evaluation without making any assumptions about other bidders' strategies. One can clearly see that as soon as the winning bid is on average some fixed percentage over the second highest bid it is optimal to increase the click weight. This obviously only holds for a constant highest bid and a dropping second highest bid. The reason is that by paying only second highest price it depends

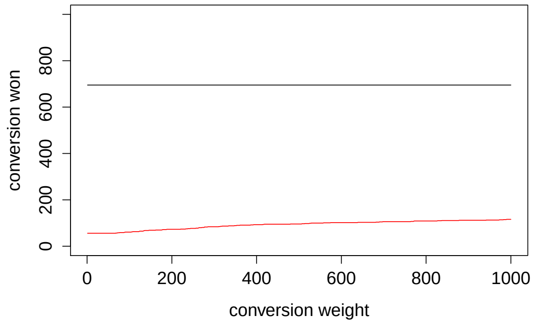

Figure 4.5: Number of Conversions Recalled against Conversion Weight

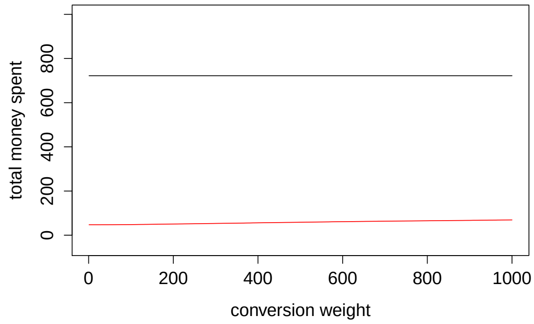

Figure 4.6: Total Money Spent against Conversion Weight

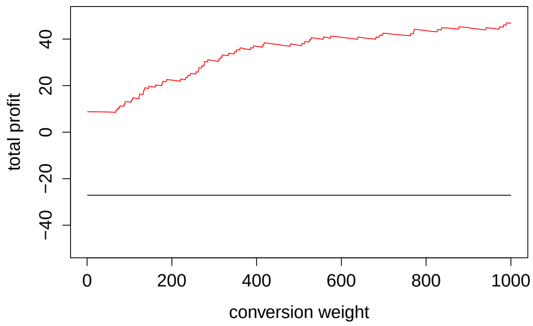

Figure 4.7: Total Profit against Conversion Weight

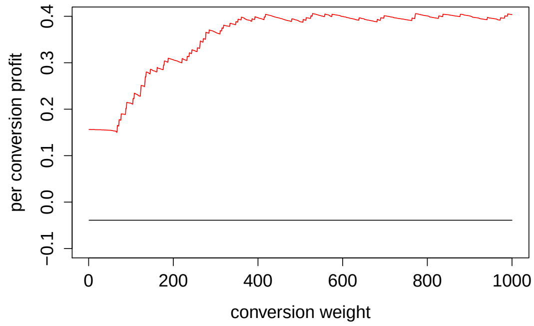

Figure 4.8: Profit per Conversion against Conversion Weight

on the second highest price how low the expected performance of an impression can be while it is still profitable to bid on it. (cf. Figure 4.9) The surprising thing is that already at an average difference of 20% between the two highest bids the reward for increased recall far outweighs the reduced precision one gets from bidding on more and more clicks. After 20% the maximum click weight for which the statistics have been computed is already reached. At a certain point it will be most profitable to bid on all impressions one gets a bid request for. When thinking about our bidding strategy this means that a central parameter would be the average difference between the price bid and the price paid which influences the selectivity concerning impressions.

There are now two cases, either one wants to be selective with clicks or recall is rewarded so heavily that it is optimal to get as many clicks as possible without looking

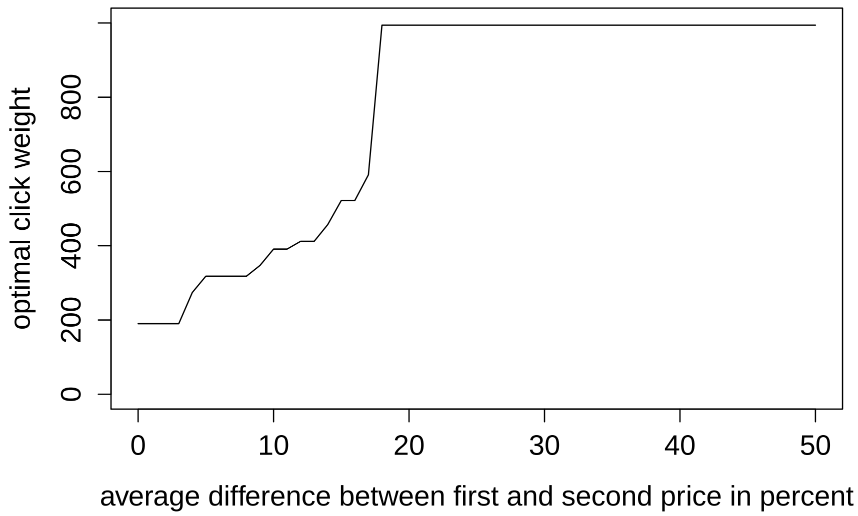

Figure 4.9: Optimal Click Weight against Percentage of Turnover additionally Gained as Profit

much at the costs. In the first case it is possible to vary the bid weight to select only the best clicks. In the second case one can also bid on the impressions classified as non-clicks or non-conversion but bid only a reduced price similar to what is done in online advertising. [13] The training is then performed as follows:

- Train a Naive Bayes classifier with a balanced training set. This means that each click is included multiple times in the training dataset such that in total there are as many clicks as non-clicks. One can of course vary the number of times that clicks are included in the training data to adapt to other market participants.

- After training the classifier, classify the whole training set and record the total amount of money one would pay for the upper split (m_u), the impressions classified as clicks, and the total money paid for the lower split (m_l), the impressions classified as non-clicks. We will also record the number of clicks falling into the upper and lower split (c_u and c_l).

- The following two ratios should be equal as the money that is spent on a set of impressions is directly proportional to the number of clicks that are in the set:

$$\frac{m_u}{m_l} = \frac{c_u}{c_l} \tag{4.6}$$

- As has been shown earlier there is additional information in the other attributes which means that this ratio will not actually be equal. The modified prices for the upper and respectively lower split are given by the equation below. The lower fraction of the double fraction in the first equation is the ratio of money currently spent on the upper split and the upper fraction is the ratio of clicks in the upper split.

$$p'_u = p_u \frac{\frac{c_u}{c_u+c_l}}{\frac{m_u}{m_u+m_l}} \tag{4.7}$$

$$p'_l = p_l \frac{\frac{c_l}{c_u+c_l}}{\frac{m_l}{m_u+m_l}} \tag{4.8}$$

This advanced bidding strategy allows one to bid more money than the reference strategy would on higher-potential impressions and less on lower-potential impressions. In the table below one can see the results of a direct comparison between the just described advanced strategy and the reference strategy. (cf. Table 4.8) The abbreviations *FP* and *SP* stand for the case of a first or second price auction. The statistics about the splits only refer to the first price, which is the price each of the algorithms was willing to pay as the improved strategy wins all bids in the upper split and the baseline strategy wins all lower split auctions. This is due to the fact that the improved strategy increases the base price used by the standard method for impressions in the upper split and lowers them for impressions that fall into the lower split. Looking at the total number of clicks each strategy won in auctions one can observe that our improved strategy outperforms the standard strategy and gets around 2.5 times more clicks. At the same time there is only a very small difference in the non-clicks which have been bought. The standard strategy has less than half the number of clicks but only roughly 10% more non-clicks. This is reflected by the fact that the profit is more than three times as high for our improved strategy. One can see that still both algorithms spend nearly the same amount of money per click in our simulated second price auction. Still the advanced strategy is a bit better and has a profit margin which is more than 25% larger. Even though this is a second price auction the prices the strategies would pay per click in a first price auction have been included as well. This price indicates how much more the two algorithms would have been willing to pay on average in the instances where they won an auction. A higher average price one is willing to pay per click indicates that there is more potential for losses when a better algorithm comes along. The higher turnover also means that the advanced strategy has more potential for increased profit in case the average second price, which is payed when the auction is won, goes down by a certain percentage.

We implemented this strategy from the perspective of a big ad agency that has to win a significant part of all bids to be able to satisfy all their clients. Smaller ad agencies

	Standard	Improved
Clicks won	11210	28326
Non-Clicks won	6917246	7682698
Turnover	10329.75	25239.42
Total Profit	880.25	3086.58
Average Cost per Click (FP)	1.22	1.02
Average Cost per Click (SP)	0.92	0.89
Average Bid per Upper-Split Impression (FP)	0.0033	0.0037
Average Bid per Lower-Split Impression (FP)	0.002	0.0015

Table 4.9: Comparison Improved Strategy and Reference Strategy

might be better off using a SVM and try to get only the best deals. The only problem with this approach is that high precision can only be achieved in combination with extremely low recall, which is not acceptable for an agency that controls half the market. An acceptable solution would be a combination of the two methods that creates three different splits. It is also possible to define multiple cut-off point for the Naive Bayes method and create additional splits with higher precision in this way. With a normal training set the Naive Bayes classifier showed extremely high precision very similar to the SVM. 4.5 Issues with implementing an SVM based approach would be that it requires a lot more computational power for training and server costs are actually another large cost driver for ad agencies that participate in auctions on ad exchanges because each successful auction brings only a fraction of cent in revenue. Further research is needed to evaluate the consequences of other market participants focusing only on the top segment of impressions.

4.3 Summary

In this chapter it has been shown that the Naive Bayes classifier, SVM and Random Forest all have different properties when used for click and conversion forecasting. For high precision forecasting the SVM is superior to the other two methods. If one wants high recall the SVM is still the best but the Naive Bayes classifier comes very close. The Random Forest is always somewhere in between the other algorithms when looking at precision and recall. The best training set for a SVM seems to be a reduced training set where only a fraction of non-clicks or non-conversions are included such that each click or conversion is exactly once in the training set and an equal number of non-clicks and non-conversions. The Naive Bayes classifier functions with both reduced and balanced training sets although a balanced training set increases its precision. The Random Forest needs the balanced training set to make useful predictions. It was also interesting to observe that clicks seem to be much better forecastable than conversions but this might be due to the limited amount of conversions we had for training. A low number of positives in the training set obviously makes it hard for the algorithms to detect the few conversions that are in the test set. I was also shown that there is no reason to not take all the data we have and select only a few attributes. In the cases considered it does not make a measurable difference. Additionally the publication ID was found to already contain most of the information regarding click and conversion performance. Therefore the decision was made to use all attributes except pub and sub ID for classification and to use the average click rate of a publication as a base price one is willing to bid. In the next chapter we will start with the sixth phase of the CRISP-DM datamining process and look at a possible implementation of such a prediction system. The best classifier for such a system is the Naive Bayes classifier as it is fast, has high recall which is by far the most important feature as has been shown in the comparison of the improved strategy to the reference strategy. It is also is easy to update and extremely fast to train and thus perfect for our purposes.

CHAPTER 5

Implementation

The last chapter was about different classifiers and their performance in click and conversion prediction. It was shown that high recall is one of the most important characteristics when one is not severely limited in the number of clicks one can purchase because as soon as there is a gap between the price bid and the price payed each time, recall is rewarded much stronger than precision. Two strategies for bid optimization were presented. One uses a classifier to separate impressions with higher click potential from the rest and only bid on them. The other algorithm that should be used when profit margins are a large percentage of total turnover. It divides impressions into two groups. One with higher and one with lower potential and bids on both groups. Both strategies use the average click rate of the publication ID of the impression as base price. The second strategy also changes this base price based on the specific characteristics of the classifier split.

In this chapter the focus lies on the structure of a good implementation of our two proposed strategies. Only the most important parts of the implementation will be covered. This includes the structure of the classifier, how it is updated and how it is trained. There exists also a recommended architecture for the production environment and a list of things one has to consider when selecting the data to use in the process. We are now in the final phase of the CRISP-DM methodology which is concerned with the deployment of our model.

From the business side the main objective of the data mining process was to answer three questions for each impressions:

- Do we bid on this impressions?
- If we do bid on this impression, how much should we bid?

- If we won this impression, which campaign's banner should we show?

The first two questions were already covered in the last chapters in great detail. There exists the possibility to either bid only on selected impressions or to bid all impressions but adjust the price one is willing to pay based on the classification result. Which of the strategies one should use depends on for how many clicks customers are willing to pay. For the implementation presented here the difference are minor code changes. For the third question a policy is proposed where the campaign that values the impression the highest gets the impression. In reality this might lead to a situation where one campaign gets all or none of the impressions. For this reason a reference is made again to the strategy already presented in the second chapter where an adjustment factor is included in the final valuation of an impression for a campaign and again the campaign with the highest adjusted valuation gets the impression. [17]

To be able to talk about where the different steps are executed it is mandatory to establish an architecture that determines the data flow in the implementation. In the graph below one can see the general structure of the proposed production environment. (cf. Figure 5.1) All business concerns about the data like validity of conversion confirmations have to be addressed before the data is written into the data warehouse. But generally the only issue one has to take care of is that all conversions are confirmed by the customer as it is the only step where the ad agency relies on third parties and there is no clear incentive for said third party to complete this step. For the customer each conversion means that he has to pay more if such a contract is in place. It is therefore important to set the right incentives from the business side. Other data items are received from the ad exchange and clicks can be confirmed by the ad agency itself through the banner.

Next it is mandatory to understand the meaning behind each data item received from the ad exchange and how it is extracted from the user's device. This helps in estimating the reliability of the different attributes. Generally this step only has to be completed once when the decision is made about which features to include into the machine learning model. Ad Exchanges normally always send the same attributes and rarely change the data items one receives in the bid request. In our case this step was discussed extensively in the third chapter. As this step defines big parts of the implementation it has to be conducted beforehand.

After gaining insights into the data at hand one has to convert all the data from the data warehouse into a dataset to work with. This is the first step covered in our solution model. The ingestion node selects all relevant data from the data warehouse. The database query itself is already able to make all required transformations like extracting the week out of a timestamp or removing attributes. Then the ingestion node writes the data into the database in an aggregated format. The format chosen aggregates clicks and

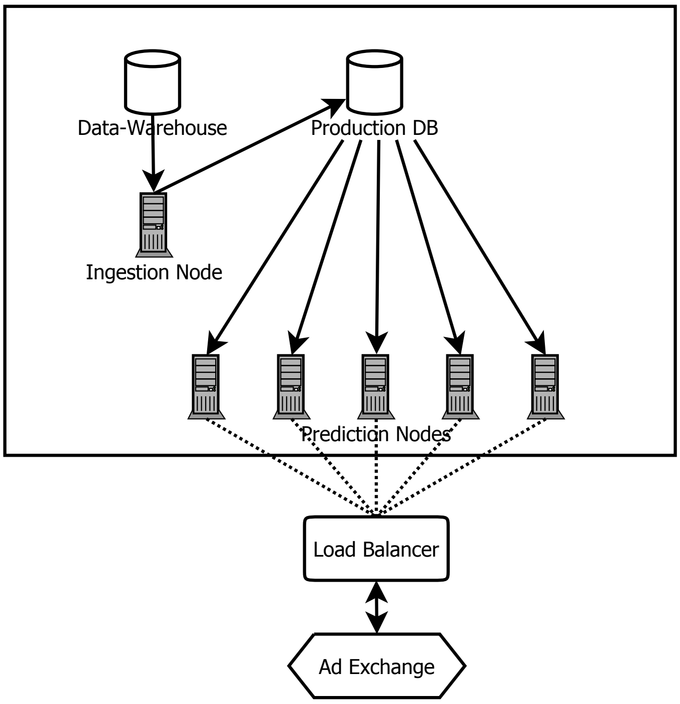

Figure 5.1: Production Environment

non-clicks by campaign, attribute and attribute value. These are sufficient statistics for the calculation of the relative frequencies of attribute values in clicks and non-clicks as well as average click rates in campaigns. They can also easily be updated with new data as one just has to increment the numbers already saved in the database.

If one wants to use the strategy presented earlier where the impressions are split into two groups and one bids on both but a bit higher on the upper split and bit lower on the lower split one has to additionally calibrate this model. To do this calibration it is required that the ingestion node also holds an instance of the machine learning model. As the calibration only requires to classify each new impression and count the clicks and non-clicks in the upper and lower split as well as the money spent on each. Thus new data can be used calibrate the already existing model. How one can handle the cold start problem will be discussed in the next section. This step requires our program to run through a second time after model building step is complete. The first time to build the standard model and the second time to estimate the click rate in the two splits.

The model creation process is guided by the need of extremely fast response times. It is important to periodically update the model with new data from the data warehouse. Considering the formula from last chapter the result of the classifier is calculated the following way for an impression with n attributes each with its own distinct values:

$$P(C|I) = \frac{P(C)\prod_{k=1}^{n} P(V_{k,i_k}|C)}{P(C)\prod_{k=1}^{n} P(V_{k,i_k}|C) + P(\neg C)\prod_{k=1}^{n} P(V_{k,i_k}|\neg C)} \tag{5.1}$$

$P(C)$ and $P(\neg C)$ can be estimated by the average click rate in a campaign. If one wants to simulate the balanced sampling method for example one can just set both to 0.5 because as will be explained later all other expressions stay the same. $P(V_{k,i_k}|C)$ stands for the probability that a click has the value with the index i_k for the k^{th} attribute. Therefore this probability can be estimated by the relative frequency of each attribute in the set of all clicks which obviously does not change if all clicks are included an additional time.

Earlier it was mentioned that each campaign should be optimized individually as different products will obviously attract different customers. One main problem with this approach is that there are many smaller campaigns and there are problems for making predictions for completely new campaigns and in the case that a lot less data is available in general. A solution to this problem is a fallback model that is calculated over all campaigns. In case there is not enough data for a certain country, for example, the model creation step skips this country and at runtime when the classifier cannot find the frequency of this specific country it falls back onto the aggregated model over all campaigns.

Additional numerical optimization can be made in accord with Laplace's Rule of Succession. [8] Two common problems can mitigated by assuming that there is already one click and non-click for every attribute value. This mitigates the case when there are absolutely no clicks or no non-clicks in all campaigns for a certain attribute value. In this case the frequency of the respective attribute value would be 0 and therefore the whole expression presented above is either 1 or 0 if there is only 1 click or non-click in our database. When one uses this rule or makes somehow sure that there is no probability of 0 one can additionally use the sum of the logarithms which helps with the numerical representation of very small probabilities. This is only possible when one can rule out the case of a 0 probability as one can get undefined expressions when one allows for taking the logarithm of 0 which is $-\inf$. One then gets the following improved formula:

$$P'(C|I) = \frac{e^{log(P'(C))+\sum_{k=1}^{n} log(P'(V_{k,i_k}|C))}}{e^{log(P'(C))+\sum_{k=1}^{n} log(P'(V_{k,i_k}|C))} + e^{log(P'(\neg C))+\sum_{k=1}^{n} log(P'(V_{k,i_k}|\neg C))}} \tag{5.2}$$

All frequencies should be calculated beforehand and stored in a data structure like a hashmap to allow for near instantaneous retrieval of thousands of values per second. Alternatively one could use an additional database table which is optimized for fast read access of specific values instead of every prediction node having its own copy of the model. Each prediction node holds its own model to be able to make near instantaneous predictions. It should be noted that the size of the data structure used to store the values required for the calculation of the formula above grows only linearly with the number of attribute values due to the independence assumption.

5.1 Evaluation and Deployment

A Naive Bayes model is is very simple to evaluate compared with the two other models presented in the last chapter as every single model component which consists of a set of two frequencies can be looked at in isolation due to the assumptions made by the model. Additionally all values are single numbers that stand for frequencies which can be easily interpreted by a human reader. One thing one has to check in the model is the case of 0 probabilities and whether one wants to accept them or handle in the way presented above or some other way. It is important to review the classification results of the classifier as one might get too many or not enough impressions. We talked earlier about a method where one has to record how many clicks and non-clicks are classified as click respectively non-clicks to adjust the base prices using the information not available from only looking at the sub ID. It is useful to make such calibration tables also when only bidding on impressions classified as clicks or conversions. With such statistics one can change the weight of clicks in the model by changing the click-rate the classifier assumes for each campaign or the cutoff value above which impressions are classified as clicks or conversions. This way one can manage on how many clicks or conversions one bids and also look at the expected results beforehand.

Deployment is not a separate step from building the model in our solution. As model building is computationally inexpensive it is easier to rebuild the model on each prediction node separately than save and transfer it somehow. Evaluation can be done on a specific prediction mode reserved for evaluation and afterwards one can centrally change settings like click weights per campaign in the database. When the prediction nodes then create the new model from the data in the database they each have an identical model. This is by far the best model as it makes it easy to test new settings and makes it easy to scale up the network by adding new prediction nodes. All parts of the solution we proposed scales linearly at worst. Especially an increased amount of bid requests can be handled by a proportional increase in the number of prediction nodes. Increased amounts of data mostly concern the ingestion node which can either be scaled up or also be replicated and then ingest a smaller timeslice from the data warehouse.

Currently the proposed strategy is already being tested in the production environment of *MobFox*. One of the biggest strengths of the solution is how well it handles large data streams as the architecture. Predictions are near instantaneous as the whole model can be held in memory and a prediction consists only of a handful of additions and multiplication. We are confident that at the end of the year all of *MobFox* ad servers will be based on the our proposed implementation.

5.2 Conclusion

In this chapter the proposed solution has been discussed in detail. The recommendation is an architecture for the production environment including data flow that splits the different required parts in a way where it is easy to scale the system up. Important challenges and optimizations to overcome them were discussed like Laplace's rule to handle 0 probabilities and taking logarithms to handle numerically very small numbers. Concerns about scalability were addressed and it was shown that each type of increased demand, be it more data or an increased number of requests, can be mitigated by scaling up some component linearly.

The next chapter will be about open issues that our approach does not handle or where additional research is needed to find a satisfying solution. While the proposed solution produces good results improvements can be made.

CHAPTER 6

Conclusion and Open Issues

In the last chapter an implementation of the bidding strategy, that uses the strongly simplifying assumptions or a Naive Bayes classifier to make good predictions was presented. This chapter summarizes all the findings made within this thesis and lists the topics where additional research might further improve the presented strategy.

In this thesis we discussed the challenges that arise with the increasing importance of mobile advertising. Most well-known smartphone applications rely on advertisements for monetization and there is no end of this trend in sight. A better match between the user and advertisement benefits both the user which may actually be interested in the promoted product and the product owner who wants to reach interested customers. We analyzed the problem from the perspective of an ad agency that buys advertisement inventory on an ad exchange and optimizes click and conversion performance on behalf of their customers.

First it was shown that there exists already a huge body of knowledge that can be used to solve this problem. Many techniques used in online advertising can be adapted to mobile advertising. Many problems have already been solved individually. The challenge was to combine all this existing knowledge into a coherent strategy that one can follow in order to consistently make good bidding decisions on ad exchanges.

The first step in creating a bidding strategy was to analyze the ad exchange framework and our dataset. We found out when inspecting the documentation of the OpenRTB protocol and our sample data that ad exchanges consistently send the same data. Therefore the different attributes of our dataset were analyzed at how good they are fitted for click and conversion performance forecasting. One could see that time plays only a very small role in click and conversion performance. The best predictor by far was the specific app used. For this reason the proposed strategy was based on a classifier that splits impressions into a high and low performance group and a base price given by the average click or conversion rate of the publication. Depending on the properties of the market one can either only bid on the high performance group or bid on all impressions but adjust the bidding price depending on the group. Our benchmark strategy that bid on every impression and used the same base prices was clearly outperformed by both our new improved strategies.

A possible implementation of our strategy consisting of a data warehouse, a production database and two node types was proposed. One node type that continuously ingests new data into the production database and one node type that makes predictions based on the model. With the right adjustments our implementation can make numerically stable predictions and is easily adjustable for a specific precision and recall. Due to the modularity of our system it is easy to scale up individual components to keep up with increased requirements either from the data warehouse side or a higher number bid requests.

One way to improve the algorithm is to find additional data about consumers. More useful data let us make better predictions and achieve higher recall and precision in our classification scheme. Especially techniques that can track individual users over many different applications would add a vast amount of new optimization options. The big advantage of such a method would be that in online advertisements such techniques are already being used with huge success. [13] A much easier way to get additional information would be for ad exchanges to form partnerships with apps that already have a lot of user information. When one has an individual identifier for each smartphone and can connect this ID with personal information from applications that let the user order food, buy clothes or share his interests with his friends, one can create very coherent pictures of one's target users. Recent developments point in this direction and open up a big market which is obviously very valuable. Especially for conversions this approach promises vast improvements with respect to the solution proposed in this thesis.

Another big topic not discussed in detail in our strategy is the handling of multiple campaigns at the same time. In practice bidders at an ad exchange have many different campaigns to optimize and also goals of how much money to spend or how many clicks and impressions to buy. An improved strategy should also incorporate different phases of the campaign. While in the beginning it might be important to spread out the impressions over the day and different countries and apps when the campaign comes to an end one should not prioritize diversity as much as in the beginning. [21] The earlier in the lifetime of a campaign the more different impressions one should explore to find things that work exceptionally well for this campaign. But when one has already spent most of the budget one should focus on the experience already gathered and bid on impressions where it is sure that one has a high probability of getting a click or conversion. Other related topics are budget spending and campaign pacing. Especially for small campaigns where there is little room for error it is important to spend the budget in an intelligent way over he whole timeframe. [11] One should incorporate knowledge about the times with the highest click rates as well as price levels at different points in time. Even more interesting is the fact that an adaptive pacing technique of varying the amount one is prepared to pay at a certain point in time which is based on the budget already spent and the total budget can be used for managing the click allocation between different campaigns. Campaigns with more budget still available, are willing to pay more for an impressions than those that have already spent most of theirs. Internally one can distribute the impressions according to which campaign values the impression highest.

One main point of research that would give us new interesting insights would be a comparison between many different campaigns. It might be that the click-rate over week fluctuates the same way in every campaign. It is possible to find very similar types of campaigns through clustering. Generally this type of research would help especially when optimizing new campaigns or very small campaigns that do not collect much data over their lifetime. It might be helpful to have lots of reference campaigns with similar click behavior on the user side. A similar approach is already used at YAHOO to cluster keywords. [28] This allows the company to make much more accurate predictions about very rare keywords.

This approach can be extended to cover not only campaigns but also countries, publishers and publications. By using huge amounts of data to cluster all attribute values it is possible to significantly reduce the cardinality of different attributes. If all the data can be reduced to a handful of attributes with only 10 to 100 values each it becomes possible to estimate the true multivariate distribution which could incorporate dependencies between different attributes. Such a distribution would also give us far superior insights into the data and how click and conversion-rates are determined. The difficulty of this research topic and the clustering of campaigns which was mentioned before would be the handling of such large amounts of data and processing it. It also would be interesting to see how much the data actually changes over time. If the data was highly volatile it would mean that all types of analysis would have to be repeated again and again. Certain procedures like clustering over many different campaigns or clustering different apps might either be not meaningful if they change very fast over time or the computational power needed to periodically repeat such procedures would be more expensive than just paying a bit more for the impressions.

While there is still room for additional research our proposed strategy offers a large improvement over the standard strategy. In a direct comparison the improved strategy made 250% more profit than the standard strategy. By adapting existing literature or doing further research this performance can be even further increased. Possible improvements either create new data, let one extract more information out of the existing data or focus on improving the bidding process. The main challenges lie in the handling of large amounts of data.

It will be interesting to see how the mobile advertisement market continues to change. The more competitive the market and the lower the profit margin are getting the better the strategies will have to get. If the trend of financing free software with advertisements continues and ad exchanges get to know more and more data about consumers then the market may become as big as classical online advertising one day. But even today there are lots of opportunities for research to better understand mobile data, get more mobile data or just use it better in one's bidding strategy.

As already mentioned before a running implementation of our proposed strategy is already being tested by *MobFox*. Due to its excellent performance and scalability characteristics of the implementation and infrastructure we are confident that we will soon be able to quantify the advantage of using our improved strategy in a real world scenario.

It will also be interesting to see how the mobile advertisement market continues to change and the strategies used with it. The more competitive the market and the lower the profit margin are getting the better the strategies will have to get. If the trend of financing free software with advertisements continues and ad exchanges get to know more and more data about consumers then the market may become as big as classical online advertising one day. But even today there are lots of opportunities for research to better understand mobile data, get more mobile data or just use it better in one's bidding strategy.

Bibliography

[1] Alan Acock and Gordon Stavig. A measure of association for nonparametric statistics. *Social Forces*, 57(4), June 1979.

[2] Karl Johan Aström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton University Press, 2010.

[3] Alexander Beck. *Google AdWords*. Hüthig Jehle Rehm, 2011.

[4] Michael Schwarz Benjamin Edelman, Michael Ostrovsky. Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords. *NBER Working Papers*, (11765), 2005.

[5] Stuart Bennett. *A history of control engineering*. IET, 1930.

[6] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[8] Rudolf Carnap. On the application of inductive logic. *Philosophy and phenomenological research*, 8(1):133–148, 1947.

[9] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. 2000.

[10] Denis Charles, Max Chickering, Nikhil R Devanur, Kamal Jain, and Manan Sanghi. Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 121–128. ACM, 2010.

[11] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pages 1307–1315. ACM, 2011.

[12] Ye Chen, Dmitry Pavlov, and John F Canny. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 209–218. ACM, 2009.

[13] Rod Hook Ori Stitelman Troy Raeder Foster Provost Claudia Perlich, Brian Dalessandro. Bid Optimizing and Inventory Scoring in Targeted Online Advertising. In *Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'12), Beijing, China*. ACM, August 2012.

[14] Peter Cohan. Growing At 100%, Nexage Gobbles $7 Billion+ Mobile Ad Market, http://www.forbes.com/sites/petercohan/2013/09/05/growing-at-100-nexage-gobbles-7-billion-mobile-ad-market/, 2013. Accessed: 2013-10-08.

[15] Subhankar Dhar and Upkar Varshney. Challenges and business models for mobile location-based services and advertising. *Communications of the ACM*, 54(5):121–128, 2011.

[16] Evgeniy Gabrilovich, Vanja Josifovski, and Bo Pang. Introduction to computational advertising. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Tutorial Abstracts*, HLT-Tutorials '08, pages 1–1, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[17] Arpita Ghosh, Benjamin IP Rubinstein, Sergei Vassilvitskii, and Martin Zinkevich. Adaptive bidding for display advertising. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*, pages 251–260. ACM, 2009.

[18] Richard K Guy. The strong law of small numbers. *American Mathematical Monthly*, pages 697–712, 1988.

[19] Edwin A Hernandez. War of the mobile browsers. *IEEE Pervasive computing*, 8(1):0082, 2009.

[20] iab. *OpenRTB API Specification Version 2.0*, January 2012.

[21] Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. Real time bid optimization with smooth budget delivery in online advertising. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, page 1. ACM, 2013.

[22] DavidD. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*, pages 4–15. Springer Berlin Heidelberg, 1998.

[23] Markus Mobius, Hamid Nazerzadeh, Gregory Lewis, and Elisa Celis. Buy-it-now or take-a-chance: A new pricing mechanism for online advertising. In *2012 Meeting Papers*, number 443. Society for Economic Dynamics, 2012.

[24] S. Muthukrishnan. Ad Exchanges: Research Issues. *ACM SIGecom Exchanges*, 8(2), 2009.

[25] S. Muthukrishnan. AdX: A Model for Ad Exchanges. *ACM SIGecom Exchanges*, 8(2), 2009.

[26] Nicolas Kruchten. http://datacratic.com/site/blog/peeking-black-box-part-2-algorithm-meets-world. Accessed: 2013-10-16.

[27] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine learning (ICML'05)*, pages 625–632.

[28] Moira Regelson and D Fain. Predicting click-through rate using keyword clusters. In *Proceedings of the Second Workshop on Sponsored Search Auctions*, 2006.

[29] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[30] Bernhard Scholkopf. The kernel trick for distances. *Advances in neural information processing systems*, pages 301–307, 2001.

[31] Mark Klein Simon Parsons, Juan A. Rodriguez-Aguilar. Auctions and Bidding: A Guide for Computer Scientists. *ACM Computing Surveys*, 43(2), 2011.

[32] Gerald Tesauro and Rajarshi Das. High-performance bidding agents for the continuous double auction. In *Proceedings of the 3rd ACM Conference on Electronic Commerce (CEC'01)*, pages 206–209. ACM, 2001.

[33] Thomas Borchert Ralf Herbich Thore Graepel, Joaquin Quinonero Candela. Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. In *Proceedings of the 27th International Conference on Machine Learning (ICML'09), Haifa, Israel*. eMarketer, April 2009.

[34] IE Toft and Anthony J Bagnall. Adaptive sniping for volatile and stable continuous double auction markets. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, pages 119–134. Springer, 2009.

[35] Antonio Tomarchio, Francesco Bellacci, and Filippo Privitera. Data-driven behavioural algorithms for online advertising. In *Proceedings of the International Conference on Multimedia*, pages 1533–1534. ACM, 2010.

[36] Jie Li Sharon Wan Tak W. Yan Ye Chen, Pavel Berkhin. Fast and Cost-efficient Bid Estimation for Contextual Ads. In *Proceedings of the 21st World Wide Web Conference (WWW 20012), Lyon, France*. ACM, April 2012.

[37] Tom Minka Chenguang Zhu Zeyuan Allen Zhu, Weizhu Chen and Zheng Chen. A Novel Click Model and Its Applications to Online Advertising. In *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE 2009), December 14-18, 2009, Rome, Italy*, pages 1–12. ACM, 2010.