FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Analyse der Auswirkungen künstlicher Intelligenz im digitalen Marketing auf das personalisierte Kundenerlebnis

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

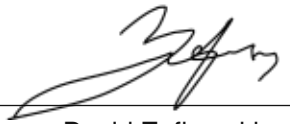
Wirtschaftsinformatik

eingereicht von

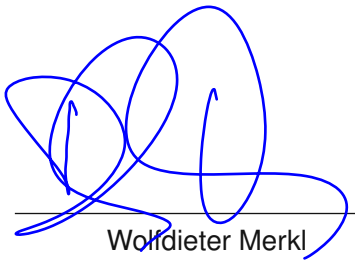
David Zafirovski
Matrikelnummer 01614836

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Ao.Univ.Prof. Mag. Dr. Wolfdieter Merkl

Wien, 1. März 2021



David Zafirovski



Wolfdieter Merkl

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Analyzing the impact of Artificial Intelligence in Digital Marketing on personalized customer experience

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

David Zafirovski
Registration Number 01614836

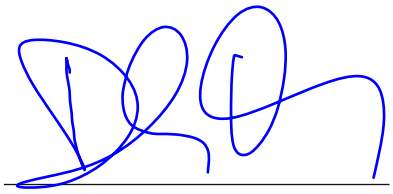
to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Mag. Dr. Wolfdieter Merkl

Vienna, 1st March, 2021

David Zafirovski



Wolfdieter Merkl

Erklärung zur Verfassung der Arbeit

David Zafirovski
Nikola Rusinski 6/1-21, 1000 Skopje

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.

Wien, 1. März 2021

David Zafirovski

Danksagung

Während des Schreibens dieser Masterarbeit habe ich enorme Unterstützung und Unterstützung erhalten.

Zunächst möchte ich meinem Betreuer, Ao.Univ.Prof. Mag. Dr. Wolfdieter Merkl, für sein Fachwissen über digitales Marketing und seine Inspiration bei der Formulierung des Masterthemas danken.

Darüber hinaus möchte ich allen, die mich bei der Durchführung des Masterprojekts unterstützt haben, meinen tiefsten Dank aussprechen, insbesondere meinen Eltern für ihre moralische Unterstützung während all der Jahre, die ich in Wien studiert habe.

Acknowledgements

During the writing of this master thesis I have received enormous support and assistance.

Firstly, I would like to thank my supervisor, Ao.Univ.Prof. Mag. Dr. Wolfdieter Merkl, for his expertise about Digital Marketing and inspiration in formulating the master topic.

Furthermore, I wish to express my deepest gratitude to everyone who supported me while conducting the master project, especially to my parents for their moral support during all the years I studied in Vienna.

Kurzfassung

Ziel der Masterarbeit ist es, anhand von Kundendaten ein Klassifizierungsmodell zu erstellen, das klassifiziert, welche Kunden eine Marketingkampagne akzeptieren und welche Kunden ablehnen. Die Ergebnisse des Vorhersagemodells können verwendet werden, um die Kunden, die positiv reagieren und die Marketingkampagne akzeptieren, korrekt zu segmentieren und zu zielen und um zu vermeiden, dass die Kunden angesprochen werden, die negativ reagieren und das Marketingangebot ablehnen. Dies führt zu einer effizienten Arbeit des Unternehmens, reduzierten Kosten und einer Maximierung des Unternehmensgewinns. Das Klassifizierungsmodell wurde unter Verwendung der branchenübergreifenden Standardmethode für Data Mining (CRISP-DM) erstellt. Ein Datensatz über Kundenverhalten und Kaufgewohnheiten wurde von der Online-Community von Kaggle Data Science gesammelt. Wir haben uns für diesen Datensatz entschieden, da er unseren Problembereich für personalisiertes Marketing abdeckt und Funktionen verschiedener Typen enthält, die das personalisierte Kundenverhalten in Bezug auf Antworten auf vergangene Marketingkampagnen, den für verschiedene Produkttypen ausgegebenen Geldbetrag und den über verschiedene Geschäfte getätigten Deal beschreiben Kanäle und persönliche Kundendaten (Geburtsjahr, Art der Ausbildung, Registrierungsdatum des Kunden in der Firma, Anzahl der Besuche auf der Webseite der Firma im letzten Monat, Anzahl der Tage ab dem Kauf des vorherigen Kunden, materieller Status, Anzahl der Kinder, Einkommen). Die Zielvariable stellt die Antwort des Kunden auf die Marketingkampagne dar, unabhängig davon, ob der Kunde den Vorschlag in der letzten Marketingkampagne angenommen oder abgelehnt hat. Die Forschungsfragen sind, welche der Merkmale bei der Modellkonstruktion für die Vorhersage der Reaktion des Kunden auf die Marketingkampagne auf der Grundlage einer personalisierten Kundenerfahrung am wichtigsten sind und welcher Typ der ausgewählten Klassifizierungsalgorithmen die genaueste Vorhersage der Reaktion des Kunden auf das Marketing liefert Kampagne basierend auf personalisierter Kundenerfahrung. Folgende Algorithmen für maschinelles Lernen wurden verwendet: Logistische Regression, Random Forest, Support Vector Machine, Naive Bayes und Multi-Level-Perceptron. Die Ergebnisse zeigten, dass der Logistic Regression Classifier die höchste durchschnittliche Genauigkeit von 0,87, die höchste durchschnittliche Präzision von 0,61, den höchsten durchschnittlichen F1-Wert von 0,57 und den höchsten durchschnittlichen ROC-AUC-Wert von 0,88 aufweist. Es wurde der Schluss gezogen, dass die logistische Regression die genaueste Vorhersage der Reaktion des Kunden auf die Marketingkampagne liefert.

Abstract

The objective of the master thesis is to build a classification model using customer data, that classifies which customers will accept and which customers will reject a marketing campaign. The results of the predictive model can be used to segment and target correctly the customers who will respond positive and accept to the marketing campaign and to avoid targeting the customers who will respond negative and reject the marketing offer. This leads to an efficient work of the company, reduced costs and maximization of the company's profit. The classification model was constructed using Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology. A dataset about customer behavior and purchasing habits was collected from Kaggle data science online community. We have decided to choose this dataset because it covers our problem domain about personalized marketing and contains features from different types that describe personalized customer behavior about responses to past marketing campaigns, amount of money spent on different types of products, amount of deals conducted via various channels and personal customer data (year of birth, type of education, client's registration date in the firm, amount visits to the firm's web page in the past month, amount of days from the previous client's purchase, material status, number of children, income). The target variable represents the customer's response to the marketing campaign, whether the customer accepted or rejected the proposal in the latest marketing campaign. The research questions are, which of the features are the most important in the model construction for predicting the customer's response to the marketing campaign based on personalized customer experience and which type of the selected classification algorithms provides the most precise prediction of customer's response to the marketing campaign based on personalized customer experience. The machine learning algorithms that were used are: Logistic Regression, Random Forest, Support Vector Machine, Naïve Bayes and Multi-level Perceptron. The results showed that Logistic Regression classifier has the highest average Accuracy 0.87, highest average Precision 0.61, highest average F1 score 0.57 and highest average ROC AUC score 0.88. It was concluded that logistic regression provides the most precise prediction of customer's response to the marketing campaign.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Artificial Intelligence (AI) is the intelligence demonstrated by machines with the ability to tackle problems that are usually done by human intelligence such as learning, knowledge representation, perception, motion etc. The goal of AI is to create systems that can function intelligently and independently. It has the potential to revolutionize every aspect of our daily life, work, society, and it is transforming many industries, starting with retail, finance, technology, marketing and healthcare [SSSK19] [Dir15]. Using the advances of AI, businesses are able to save time and money by optimization and automation of routine and repetitive processes and tasks, increase productivity, make faster business decisions, reduce operational costs and use insight to predict customer's preferences in order to offer them better, personalized customer experience.

Digital marketing is part of marketing that exploits the Internet, social media and other online technologies to promote brand's products and services. Companies use digital marketing to connect with their customers, to optimize user experience, expand their influence and consequently increase their net profitability i.e. their Return On Investment (ROI). The increasing use of Internet and popularity of social media has motivated the marketers to change their marketing strategies and to use a various digital features like websites, emails, videos, blog posts and social media to attract customers and raise brand awareness. There are many marketing strategies for businesses to promote their product or service: Content Marketing, Mobile Marketing, Integrated Digital Marketing, Personalized Marketing etc. Content marketing is a marketing strategy based on generating an appropriate content to gain and maintain a particular customer audience. Mobile marketing is a marketing strategy designed for getting a specific customer audience who uses smart phones and mobile devices. The channels through which the target customer audience is reached can be SMS/MMS messages, websites which are adapted to mobile screen size or other mobile applications. Integrated digital

marketing is a marketing strategy that combines several marketing strategies in order to create a consistent approach for achieving the business goals. It consists of Search Engine Optimization (SEO), content marketing, social media marketing, paid advertising campaigns etc. The main concept of integrated digital marketing is that when different individual marketing strategies are combined together, they make higher impact, rather than using a single marketing strategy separately. Personalized marketing is a marketing strategy where tailored and personalized advertisements are delivered to the individual customers. This kind of marketing approach improves the customer experience and it increases brand awareness and loyalty.

Digital marketers are aware that the customer experience could be a driver of growth when customers are satisfied, but also could be a source of risk when the experience is negative. Therefore, to achieve their goals, marketers are engaging with AI technologies.Some of the benefits that AI brings to the customers are: 24/7 support and assistance, handling several customer's requests in parallel, maintaining customer data etc. In addition, there are other advantages like minimal manual work, promotion of brand image and awareness, personalized advertisements that recommend the right content to the right customer by tracking customer's purchase patterns and preferences [Thi18].

Examining patterns and behavior from customer data can help businesses to enhance their customer's experience. Artificial Intelligence helps companies to segment and target their audience base and accelerate sales with tailored products/services [DGGB20]. Nonetheless, AI-powered solutions using machine learning algorithms and customer data gathered from various sources can lead to prediction of the behavior of the customer and success of the company's marketing campaign for new products/services [KS16][BE15][Bal15].

AI in digital marketing improves the target audience and by using machine learning algorithms so that the right audience can be reach out. AI systems can process customer's location, gender, age, interests, and visited pages, purchasing habits. These insights can be used to create accurately 360 degree profiles of existing and future customers and their behavior, ensuring that only customers who are willing to buy the product are targeted with most relevant advertisements.

1.2 Problem Statement

The problem statement is that in the marketing campaign, all the customers are targeted with advertisements including the ones who will not respond positive to the marketing campaign and reject the offer. This means that the company is not working efficiently, its marketing campaign is not optimized because the customers are not segmented and targeted correctly. As a result, the costs are increased and the company's profit is not maximized. This can lead to a failure of company's marketing campaign. Machine learning as a subset of AI, covers several broad categories: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning [AOOM18]. The problem that is examined in the master thesis belongs to supervised learning category of machine learning approaches. Supervised learning is a machine learning approach that makes

prediction using a training dataset. It can be categorized into classification or regression problems. Regression models are used for predicting continuous values like “salary” or “price”, while classification models are used for predicting discrete variables that can be labels or categories like “spam” or “not spam”, “yes” or “no” [XLQ]. In our case, we cover a classification problem domain that will be explained in the following section.

1.3 Aim of the work

Artificial Intelligence helps companies to segment and target correctly their customers and this can lead to a success of the company’s marketing campaign. The goal is to build a classification model using customer data, that classifies which customers will accept and which customers will reject the offer. After that, these insights can be used to target the customers who will respond positive and accept to the marketing campaign and to avoid targeting the customers who will respond negative and reject the marketing offer. This leads to an efficient work of the company, reduced costs and maximization of the company’s profit. By implementing AI-powered solutions to their business issues, companies can minimize their expenses and increase their product sales and profit through improved marketing campaigns that offer personalized advertisements that enhance user experience.

The classification model was constructed using Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology [WH00]. CRISP-DM provides structural methodology for data science projects.

A dataset about customer behavior and purchasing habits was collected from Kaggle [1]. Kaggle is a data science online community that enables individuals to discover datasets and create machine learning models. We have decided to choose this dataset [2] because it covers our problem domain about personalized marketing and contains features from different types that describe personalized customer behavior about responses to past marketing campaigns, amount of money spend on different types of products, amount of purchases conducted via different channels and personal customer data (year of birth, type of education, date of customer’s enrollment to the company, number of visits to the company’s web site in the last month, number of days since the last purchase, material status, number of children, income). The target variable represents the customer’s response to the marketing campaign, whether the customer accepted or rejected the offer in the last marketing campaign.

[1] https://www.kaggle.com
[2] https://www.kaggle.com/rodsaldanha/arketing-campaign

1.4 Research Questions

Research Question 1: Which of the features are the most important in the model construction for predicting the customer's response to the marketing campaign based on personalized customer experience?

Research Question 2: Which type of the selected classification algorithms provides the most precise prediction of customer's response to the marketing campaign based on personalized customer experience?

1.5 Methodology and Approach

In order to answer the research questions and obtain the results, a Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology was used [CCK+00]. CRISP-DM provides structural methodology for data mining projects, so the dataset regarding targeted marketing campaign was examined and a model that predicts the customer behavior was built.

The life cycle model consist of a six phases shown on Figure 1.1, where switching between different phases is allowed and recommended:

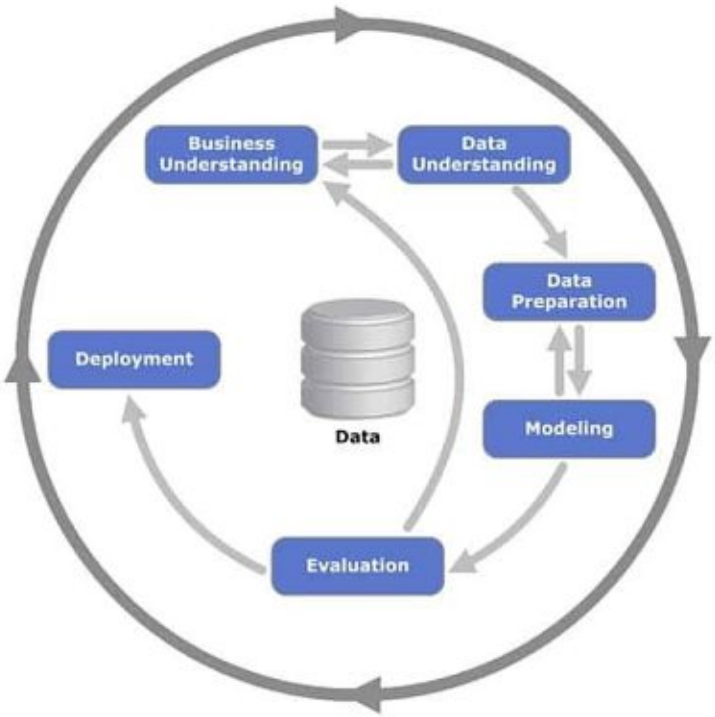


Figure 1.1: Relationship between phases of CRISP-DM, [CCK+00]

1. **Business understanding** is initial phase whose activities focus on how to determine the business objectives and translate them into data mining goals.

2. **Data understanding** focuses on how to collect, describe and explore the data, as well as verifying the data quality.

3. **Data preparation** includes every task that is necessary to be conducted in order to build the final dataset. These activities include: missing values imputation, outlier detection and removal, data visualization, feature engineering, resampling of an imbalanced dataset etc.

4. **Modeling** phase consist of selection and implementation of various modeling techniques in order to create the classification model. The selected machine learning algorithms are: Logistic Regression, Random Forest, Support Vector Machine, Naïve Bayes and Multi-level Perceptron.

5. **Evaluation** aims to estimate the generalization model accuracy on unseen data. The evaluation metrics that were used are: classification accuracy, precision, recall, F1 score and Receiver Operating Characteristic (ROC) Area Under Curve (AUC) ROC AUC score.

6. **Deployment** phase comprises of monitoring, maintenance and producing final report that supports the company to maximize the profit of the next marketing campaign.

1.6 Systematic Literature Review

Systematic Literature Review means evaluating and interpreting the available research relevant to specific research questions. Conducting a literature review includes collecting, evaluating and analyzing publications that are related to the research questions [RS04]. The well-defined methodology makes it more probable that the results of the literature study are unbiased.

In order to explore what have been already done in previous researches, we performed research on search engine and academic social network like Google Scholar[3] and Research Gate[4]. A search strategy was used to cover the relevant literature. Therefore, we defined specific keywords such as 'Artificial Intelligence', 'Digital Marketing', 'Content Marketing', 'Personalized Customer Experience', 'Targeted Customers', 'missing values imputation', 'CRISP-DM', 'outlier detection', 'visualization techniques', 'class imbalance', 'naive bayes', 'support vector machine', 'random forest', 'logistic regression', 'multi-layer perceptron','evaluation metrics'. The types of publications that were referenced are articles and conference papers.

[3] https://scholar.google.com/
[4] www.researchgate.net

1.7 Related Work

In the paper "Predicting customer response to bank direct telemarketing campaign" written by Justice Asare-Frempong and Manoj Jayabalan the research task is to predict a customer response to the marketing campaign [AFJ17]. The experiment was done using Waikato Environment for Knowledge Analysis (WEKA). Four algorithms for classification are being implemented through 10 Fold Cross-Validation: Multi- layer Perceptron Neural Network, Decision Tree, Logistic Regression and Random Forest. As evaluation metrics are used Accuracy and ROC score. The results show that Random Forest has Accuracy 86.8% and ROC AUC score 92.7%, Decision Tree Accuracy 84.7% and ROC AUC score 87.7%, Multi-layer Perceptron Neural Network Accuracy 82.9% and ROC AUC score 90.0% and Logistic Regression Accuracy 83.5% and ROC AUC score 90.9%.

Another paper that uses the same dataset is "Bank direct marketing analysis of data mining techniques" written by Hany A Elsalamony [Els14]. Hary A Elsalamony has implemented four algorithms for classification: Multilayer Perception Neural Network, tree augmented Naïve Bayes (TAN) , Logistic Regression and Decision Tree algorithm. In this paper as evaluation metrics are used Accuracy, Sensitivity, and Specificity. The dataset was split into 70% training dataset and 30% test dataset. The results show that Multilayer Perceptron Neural Network has Accuracy 90.92% on the training dataset and 90.49% on the test dataset, Sensitivity 65.66% on the training dataset and 62.2% on the test dataset, Specificity 93.28% on the training dataset and 93.12% on the test dataset. Tree augmented Naive Bayes (TAN) has Accuracy 89.16% on the training dataset and 88.75% on the test dataset, Sensitivity 55.87% on the training dataset and 52.19% on the test dataset, Specificity 91.97% on the training dataset and 91.73% on the test dataset. Logistic Regression has Accuracy 90.09% on the training dataset and 90.43% on the test dataset, Sensitivity 64.83% on the training dataset and 65.53% on the test dataset, Specificity 91.76% on the training dataset and 92.16% on the test dataset. Decision Tree has Accuracy 93.23% on the training dataset and 90.09% on the test dataset, Sensitivity 76.75% on the training dataset and 59.06% on the test dataset, Specificity 94.92% on the training dataset and 93.23% on the test dataset.

1.8 Structure of the work

The first part of the master thesis is called Implementation which is explained in Chapter 2. This chapter describes in detail the Cross-Industry Standard Process for Data Mining (CRISP-DM) phases that were conducted, including short descriptions of each phase. The obtained outcomes with figures and tables from the specific tasks and classification algorithms are given as subsections. The next Chapter 3 is named Results, where the answers from the Research Question 1 and Research Question 2 are provided. In the last chapter, Chapter 4 is explained the contribution, the progress made compared to the State of the Art.

CHAPTER 2

Implementation

In this chapter are explained in detail the phases of Cross-Industry Standard Process for Data Mining (CRISP-DM) that were undertaken. A short description of each phase is given, following the procedures that were conducted in order to obtain the specific outcomes.

2.1 Business Understanding

Business understanding is initial phase whose activities focus on how to determine the business objectives and translate them into data mining goals. In our case scenario, the main objective would be to precisely classify the customer's responses to the marketing campaign into those who accepted and rejected the company's offer for a product/service.

2.2 Data Understanding

Data understanding focuses on how to collect, describe and explore the data, as well as verifying the data quality. A dataset about customer behavior and purchasing habits was collected from Kaggle. Kaggle is a data science online community that enables individuals to discover datasets and create machine learning models. We have decided to choose this dataset because it covers our problem domain about marketing and contains features from different types that describe customer behavior and purchasing habits. Figure 2.1 displays the initial dataset that contained 29 variables, out of which 15 are numerical, 7 are categorical and 7 are Boolean type. The number of observations was 2240, with 24 missing cells which make less than 0.1 percent of the whole dataset and all the 24 missing cells were located in the "Income" variable, which make 1.1 percent missing values of the "Income" variable.

There were zero duplicate rows. There were 2 variables named 'Z_CostContact' and 'Z_Revenue' that contain constant value across all the observations, and therefore we dropped these 2 variables because they are irrelevant in the process of learning.

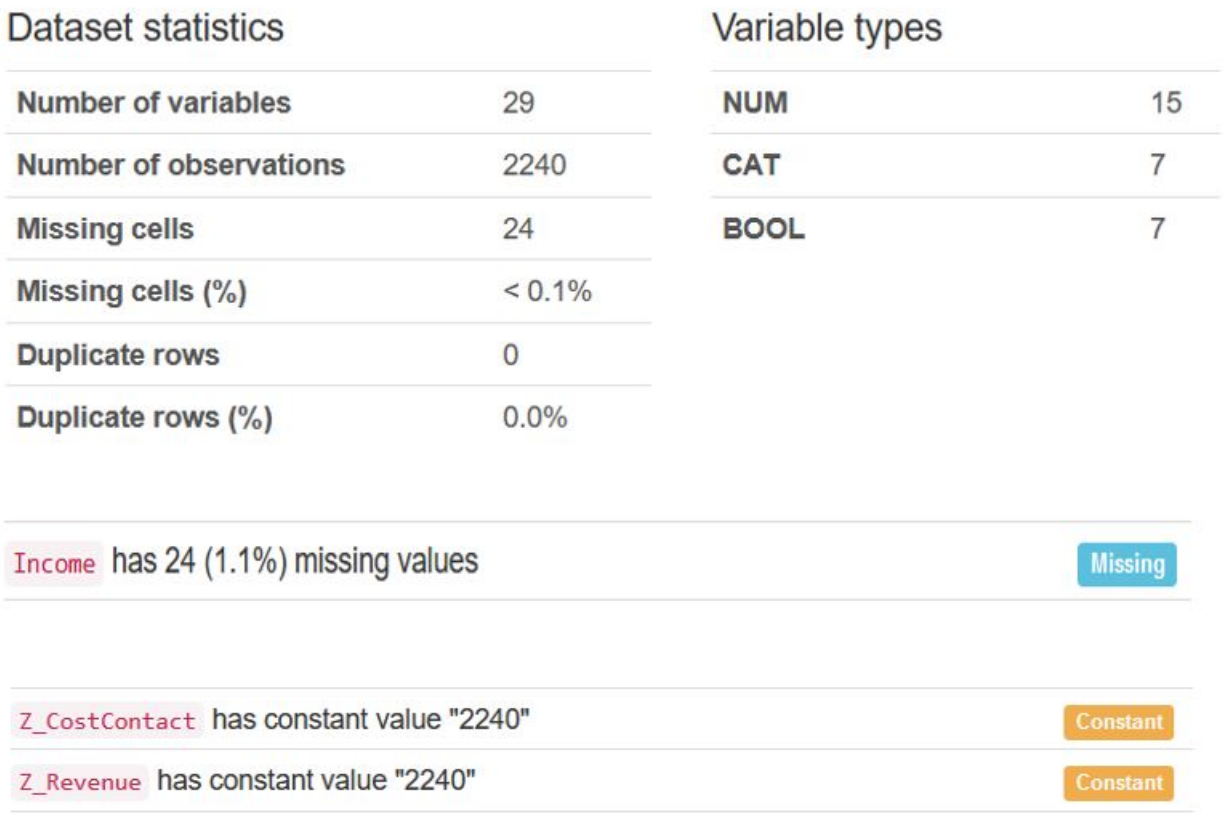


Figure 2.1: Dataset statistics and variable types

The features used for predicting the customer's response to the marketing campaign include: responses to several past marketing campaigns, amount of money spent on different types of products (Meat products, Fish products, Gold products, Fruit products, Sweet products, Wine products), amount of purchases conducted via different channels, amount of deals conducted with deduction, personal customer data (year of customer's birth, type of customer's education, client's registration date in the firm, amount of client's visits to the firm's web page in the past month, amount of days from the previous client's purchase, client's relationship status, number of children, client's annual salary). The target variable Response, is Boolean type, which represents the customer's response to the marketing campaign, whether the customer accepted or rejected the proposal in the latest marketing campaign.

In the Table 2.1, below are given all the features and their description.

ID	identifier
YearBirth	client's birth year
DtCustomer	client's registration date in the firm
Education	client's educational degree
Marital	client's relationship status
Kidhome	amount of kids in client's home
Teenhome	amount of teenagers in client's home
Income	client's annual salary
MntFishProducts	money spent on fish in the past 2 years
MntMeatProducts	money spent on meat in the past 2 years
MntFruits	money spent on fruits in the past 2 years
MntSweetProducts	money spent on sweets in the past 2 years
MntWines	money spent on wine in the past 2 years
MntGoldProducts	money spent on gold in the past 2 years
NumDealsPurchases	amount of deals conducted with deduction
NumCatalogPurchases	amount of deals conducted through the catalogue
NumStorePurchases	amount of deals conducted in shops
NumWebPurchases	amount of deals conducted via firm's web page in the past month
NumWebVisitsMonth	amount visits to the firm's web page in the past month
Recency	amount of days from the previous client's purchase
Complain	1 if a client complained in a past 2 years
AcceptedCmp1	1 if a client confirmed a marketing proposal in 1st campaign, if not 0
AcceptedCmp2	1 if a client confirmed a marketing proposal in 2nd campaign, if not 0
AcceptedCmp3	1 if a client confirmed a marketing proposal in 3rd campaign, if not 0
AcceptedCmp4	1 if a client confirmed a marketing proposal in 4th campaign, if not 0
AcceptedCmp5	1 if a client confirmed a marketing proposal in 5th campaign, if not 0
Response	1 if a client confirmed a marketing proposal in latest campaign, if not 0

Table 2.1: Features of Customer dataframe and their description

On Figure 2.2 is given a sample of the Customer dataframe, displaying some of the first and last observations and the names of the features.

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	88	546
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	1	6
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	49	127
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	4	20
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	43	118
...	...	...	...	...	...	...	...	...	...	...	...	...
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709	43	182
2236	4001	1946	PhD	Together	64014.0	2	1	2014-06-10	56	406	0	30
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908	48	217
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428	30	214
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84	3	61

Figure 2.2: Sample of the Customer dataframe

After the dataset regarding targeted marketing campaign was processed into a data frame, an exploratory data analysis was performed to summarize the main characteristics and generate descriptive statistics.

On the Figure 2.3 can be observed some of the variables of the Customer dataframe and their basic statistical details such as count, mean, standard deviation, min, max, percentiles of the numeric values in the dataframe.

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits
count	2236.000000	2236.000000	2236.000000	2236.000000	2236.000000	2236.000000	2236.00000	2236.000000
mean	5589.008497	1968.898032	51922.737120	0.444097	0.506708	49.116279	304.12746	26.275939
std	3244.826887	11.703281	21497.234199	0.538459	0.544609	28.957284	336.59181	39.724007
min	0.000000	1940.000000	1730.000000	0.000000	0.000000	0.000000	0.00000	0.000000
25%	2828.250000	1959.000000	35233.500000	0.000000	0.000000	24.000000	24.00000	1.000000
50%	5454.500000	1970.000000	51301.000000	0.000000	0.000000	49.000000	174.00000	8.000000
75%	8421.750000	1977.000000	68413.250000	1.000000	1.000000	74.000000	504.25000	33.000000
max	11191.000000	1996.000000	162397.000000	2.000000	2.000000	99.000000	1493.00000	199.000000

Figure 2.3: Descriptive statistics of Customer dataframe

2.3 Data Preparation

Data preparation includes every task that is necessary to be conducted in order to build the final dataset. These activities include:

1. Missing values imputation
2. Outlier detection and removal
3. Data visualization
4. Feature engineering
5. Resampling of an imbalanced dataset

2.3.1 Missing values imputation

Missing values happen when there isn't a specific value/values assigned to a particular feature in the dataframe. The reasons for having missing data are various from values that are not available for every observation to the approaches the data was collected. This is a common issue in data mining and it might have a substantial impact on the process of learning and training the classifiers, because it can lead to inaccurate interpretation of the data. Therefore it's very important to examine and handle appropriately this problem.

There are various types of missing values:

1. Missing Completely At Random (MCAR)
2. Missing Not At Random (MNAR)
3. Missing At Random (MAR)

Missing Completely At Random (MCAR) occurs if subjects who have missing data are a random subset of complete sample of subjects. If the likelihood that a missing cell value depends on information that is not observed, missing values are named Missing Not At Random (MNAR). If the likelihood that a missing cell value depends on information for present subject, missing values are named Missing At Random (MAR) [DVDHSM06].

The missing data in the income variable in our customer dataframe belongs to the Missing Values Not At Random (MNAR) type, because usually customers with relatively high income are not willing to expose their wage.

In general, there are 2 approaches in handling missing data: deletion and imputation. Deletion approach can remove the specific rows from the dataframe that contain one or more missing cells or remove the entire columns. Nevertheless, in majority of the cases, it is not highly recommended to use deletion approach as solution to missing values, because it may lead to loss of a valuable information that can be used in the process of learning and training the classifiers.

Imputation occurs when the missing data are replaced with calculated values. The imputation approaches can be split into 2 groups: single imputation or multiple imputation. Single imputation is when only one value for every single of the missing cells is produced. Majority of the imputation techniques are single imputation approaches, based on three core strategies for calculating the missing cells. These strategies are:

1. Substitution by existing values
2. Substitution by statistical values
3. Substitution by predicted values

Substitution by existing values can include replacement of the missing values with minimum or maximum value. Substitution by statistical values can include replacement of the missing values with mean, median or the most frequent value calculated on the entire dataset for that particular variable. Substitution by predicted values can include replacement of the missing values with predicted values based on regression algorithms using the other non-missing variables as training dataset.

Multiple imputation is when many imputed values for every single of the missing cells are produced. There are many algorithms that have been developed for multiple imputation, but the most common one is Multiple Imputation by Chained Equations (MICE). Multiple Imputation by Chained Equations (MICE) considers that the missing data are Missing At Random (MAR) or Missing Completely At Random (MCAR). Because the missing data in the income variable in our customer dataframe belongs to the Missing Values Not At Random (MNAR) type, we will not use the multiple imputation by chained equations approach.

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer
10	1994	1983	Graduation	Married	NaN	1	0	2013-11-15
27	5255	1986	Graduation	Single	NaN	1	0	2013-02-20
43	7281	1959	PhD	Single	NaN	0	0	2013-11-05
48	7244	1951	Graduation	Single	NaN	2	1	2014-01-01
58	8557	1982	Graduation	Single	NaN	1	0	2013-06-17
71	10629	1973	2n Cycle	Married	NaN	1	0	2012-09-14
90	8996	1957	PhD	Married	NaN	2	1	2012-11-19
91	9235	1957	Graduation	Single	NaN	1	1	2014-05-27
92	5798	1973	Master	Together	NaN	0	0	2013-11-23
128	8268	1961	PhD	Married	NaN	0	1	2013-07-11

Figure 2.4: Customer dataframe with missing values for Income variable

On the Figure 2.4 can be noticed some of the observations that contain missing value for the Income Variable.

As imputation approach, we chose single imputation, more specifically the substitution by predicted values using the K Nearest Neighbors (KNN) algorithm. This algorithms calculates every single missing value in the dataframe based on the mean value from the nearest neighbours in the training dataset. The instances are closer if the variables that are not missing are closer. The K Nearest Neighbors algorithm includes choosing appropriate the distance measure, setting the right the number of nearest neighbors (n_neighbours) which is the main hyperparameter and the weights parameter that is the weight function used in order to generate the predictions. KNN Imputer[1] class from scikit-learn machine learning library for Python programming language was used.

The number of nearest neighbours (n_neighbours) was set to '5', weights parameter was set to 'uniform', so all points are weighted equally, and the parameter distance metric for searching neighbours was set to 'nan_euclidean'. After that, the 'fit' and 'transform' methods from KNNImputer class were called in order to construct the imputed values for the Income variable. The 'fit' method generates a copy of the dataset where the missing cells are filled out with an estimate value. The 'transform' method then configures these missing cells and as an output a new datasest without missing values is generated.

On Figure 2.5 and Figure 2.6 is given a code snippet of the instantiation, fitting and transformation of the dataframe with missing values using KNNImputer.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

```
imputer = KNNImputer(n_neighbors=5,weights='uniform', metric='nan_euclidean')
imputer.fit(df_missing_Values_kNN[['ID', 'Year_Birth', 'Income', 'Kidhome', 'Teenhome', 'Dt_Customer',
       'Recency', 'MntWines', 'MntFruits', 'MntMeatProducts',
       'MntFishProducts', 'MntSweetProducts', 'MntGoldProds',
       'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases',
       'NumStorePurchases', 'NumWebVisitsMonth', 'AcceptedCmp3',
       'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'AcceptedCmp2',
       'Complain', 'Response', 'Education_2n Cycle', 'Education_Basic',
       'Education_Graduation', 'Education_Master', 'Education_PhD',
       'Marital_Status_Absurd', 'Marital_Status_Alone',
       'Marital_Status_Divorced', 'Marital_Status_Married',
       'Marital_Status_Single', 'Marital_Status_Together',
       'Marital_Status_Widow', 'Marital_Status_YOLO']])
```

Figure 2.5: Instantiation and fitting of KNNImputer class with dataframe containing missing values

```
df_kNN_transformed = imputer.transform(df_missing_Values_kNN[['ID', 'Year_Birth', 'Income', 'Kidhome',
       'Teenhome','Dt_Customer','Recency', 'MntWines', 'MntFruits', 'MntMeatProducts',
       'MntFishProducts', 'MntSweetProducts', 'MntGoldProds',
       'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases',
       'NumStorePurchases', 'NumWebVisitsMonth', 'AcceptedCmp3',
       'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1', 'AcceptedCmp2',
       'Complain', 'Response', 'Education_2n Cycle', 'Education_Basic',
       'Education_Graduation', 'Education_Master', 'Education_PhD',
       'Marital_Status_Absurd', 'Marital_Status_Alone',
       'Marital_Status_Divorced', 'Marital_Status_Married',
       'Marital_Status_Single', 'Marital_Status_Together',
       'Marital_Status_Widow', 'Marital_Status_YOLO']])
```

Figure 2.6: Transforming the dataframe with missing values

On the Figure 2.7 can be noticed the same observations from the previous figure with imputed values for the Income variable using KNN Imputer.

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer
10	1994	1983	Graduation	Married	41301.0	1	0	2013-11-15
27	5255	1986	Graduation	Single	32229.4	1	0	2013-02-20
43	7281	1959	PhD	Single	26582.4	0	0	2013-11-05
48	7244	1951	Graduation	Single	27132.2	2	1	2014-01-01
58	8557	1982	Graduation	Single	45152.2	1	0	2013-06-17
71	10629	1973	2n Cycle	Married	28302.8	1	0	2012-09-14
90	8996	1957	PhD	Married	49280.4	2	1	2012-11-19
91	9235	1957	Graduation	Single	39090.2	1	1	2014-05-27
92	5798	1973	Master	Together	73089.8	0	0	2013-11-23
128	8268	1961	PhD	Married	51817.2	0	1	2013-07-11

Figure 2.7: Customer dataframe with imputed values for Income variable

2.3.2 Outlier detection and removal

Outliers or anomalies are extreme values that diverge drastically from the other data points in the dataset. Generally, they are a consequence of errors in measurement, human mistakes that occurred in the data entry process or data processing errors. Outliers do not represent the overall pattern of the dataset and their presence can lead to false representation of the data, poor fitting of the predictive model and lower performance. Therefore it's essential to identify and remove them from the dataset before training the machine learning algorithms for classification or regression. Outlier detection and removal conducted before the modeling phase results in better fitting of the predictive model and more precise prediction's performance [Xi08].

The method we used for detecting outliers in our dataset is Interquartile Range (IQR) [SOA04]. Interquartile Range (IQR) is a concept from statistics used for measurement of data variability by splitting the data in quartiles. The Interquartile Range is calculated as the difference third quartile (Q3) between the first quartile (Q1). We used the method of Interquartile Range (IQR) in order to create a boxplot graphs. A boxplot aims to summarize a batch of data by displaying several main features including lower bound, first quartile, median, third quartile and upper bound [FHI89]. The median represents the middle value. First quartile points out the median of the lower half of the dataset. Third quartile points out the median of the upper half of the dataset. As outliers are considered the observations that are below the lower bound and above the upper bound. On the figure below is given a function for outlier detection that summarizes the calculations for the main features. On Figure 2.8 is given a code snippet about the implementation of a function for outlier detection.

```
def findOutliers(variable,name):

    Q1=variable.quantile(0.25)
    Q3=variable.quantile(0.75)
    IQR=Q3-Q1

    print('Quartile 1 for ',name,'is: ',Q1)
    print('Quartile 3 for ',name,'is: ',Q3)
    print('Interquartile Range for ',name,'is: ',IQR)

    Lower_Bound = Q1-(1.5*IQR)
    Upper_Bound = Q3+(1.5*IQR)
    print("Lower Bound for ",name,"is: ", Lower_Bound)
    print("Upper Bound for ",name,"is: ", Upper_Bound)

    outliers = [x for x in variable if x < Lower_Bound or x > Upper_Bound]
    print("Outlier values for ",name," are: ",outliers)
```

Figure 2.8: Function for outlier detection

On the Figure 2.9 can be noticed 3 outliers which are the customers who were born before year 1900. These outliers were removed.

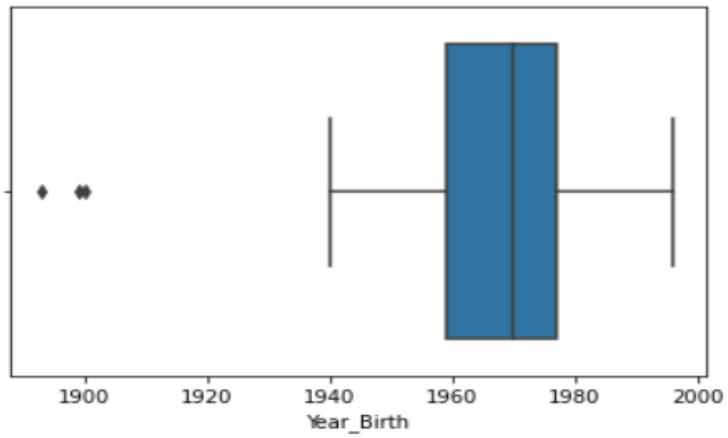


Figure 2.9: Boxplot of the Year Birth feature

On the Figure 2.10 can be noticed a small cluster of income values around 160 000, which we decided to keep them because they don't represent an extreme outlier and by deleting them we could have a loss of valuable information during the process of learning and training the classifiers. An outlier can be spotted with income value more than 600 000. Only this outlier was removed.

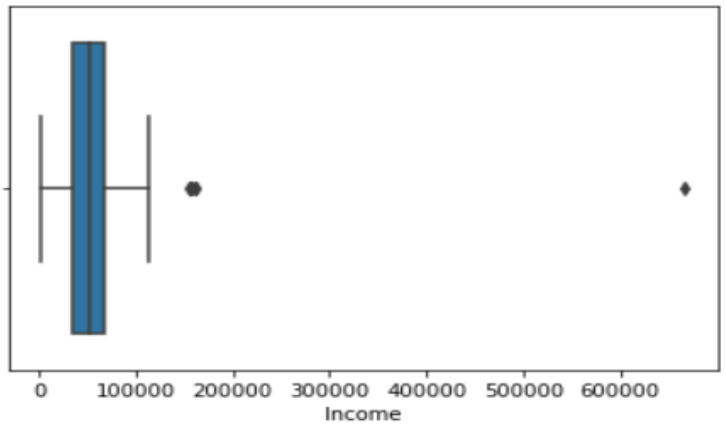


Figure 2.10: Boxplot of the Income feature

In all other features were not detected any outliers for removal.

2.3.3 Data Visualization

Data visualization is a graphical representation of data that uses visual elements (bar charts, pie charts, graphs) in order to discover patterns and spot anomalies in the dataset. Nowadays in the era of Big Data, data visualization techniques are crucial for analyzing immense amounts of data and making data-driven decisions. Data visualization assists in making faster business decisions by identifying relationships between different features quickly and improving insights. The application of data visualization is spread over several industry domains including sales and digital marketing, politics, healthcare, finance etc.

Some of the most popular visualization techniques include [SSM+16]:

1. Bar chart: This type of chart differentiates various categories of data by displaying their quantity
2. Scatter plot: This type of plot illustrates the values of two variables
3. Pie chart: This type of plot differentiates the fractions of one piece
4. Histogram: This is an approximate representation of the distribution of numerical data
5. Boxplot: This a method for representing groups of numerical data over their quartiles

The following figures show several visualization techniques that were used in order to describe visually some of the customer dataframe variables and their correlations.

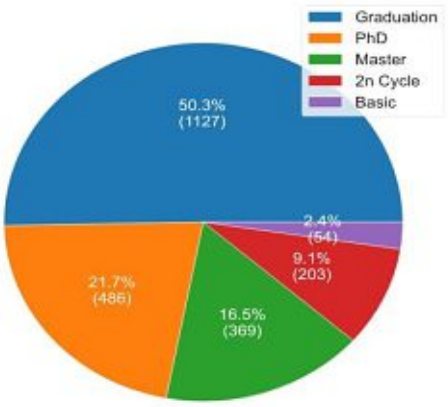


Figure 2.11: Education pie chart

On the Figure 2.11 is displayed the Educational level of the customers. It can be observed that half of the customers have Graduation type of education (50.3%), following PhD degree (21.7%), Master (16.5%), 2nd Cycle (9.1%) and Basic type of education (2.4%).

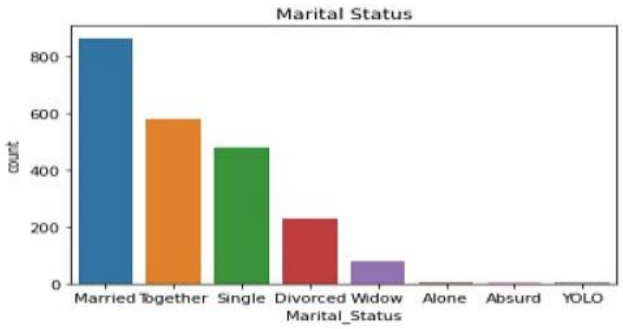


Figure 2.12: Marital Status bar chart

On the Figure 2.12 is displayed the Marital Status of the customers. It can be observed that majority of the customer are 'Married', following living 'Together', 'Single', 'Divorced' and 'Widowed'.

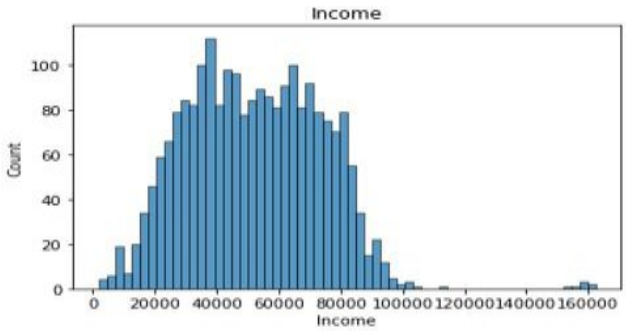


Figure 2.13: Income histogram

On the Figure 2.13 is displayed the Income of the customers. It can be observed that the majority of the customers have income between 40 000 and 80 000.

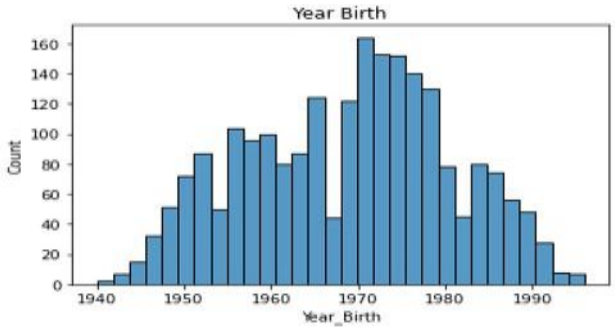


Figure 2.14: Year Birth histogram

On the Figure 2.14 is displayed the year of birth of the customers. It can be observed that majority of the customers were born between 1970 and 1980 year.

From the Figure 2.15 can be observed, that customers with Graduation - Education type have the highest both positive and negative responses to the marketing campaign. On the other side, customers with Basic - Education type have the lowest both positive and negative responses.

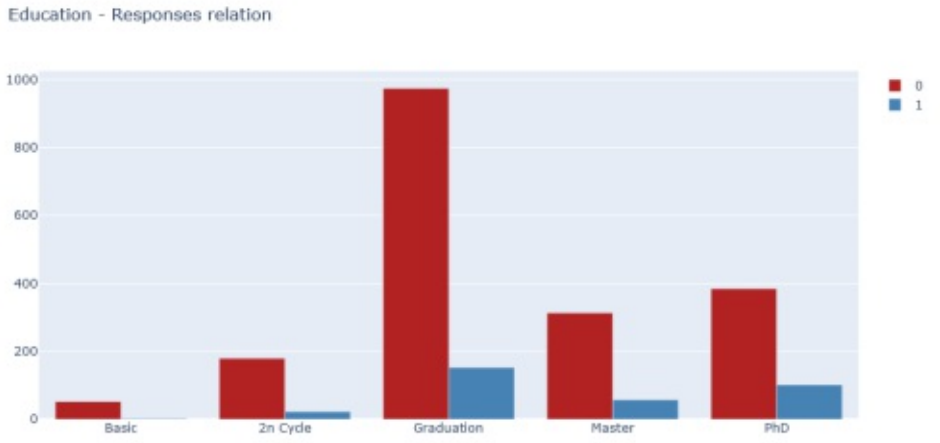


Figure 2.15: Education-Responses bar chart

From the Figure 2.16 can be observed that AcceptedCmp2 was the most rejected one (highest number of zero's) and also the least accepted (lowest number of one's). All other campaigns have almost equal number of acceptance (number of one's), but at the same time high number of zero's which means mostly rejected.

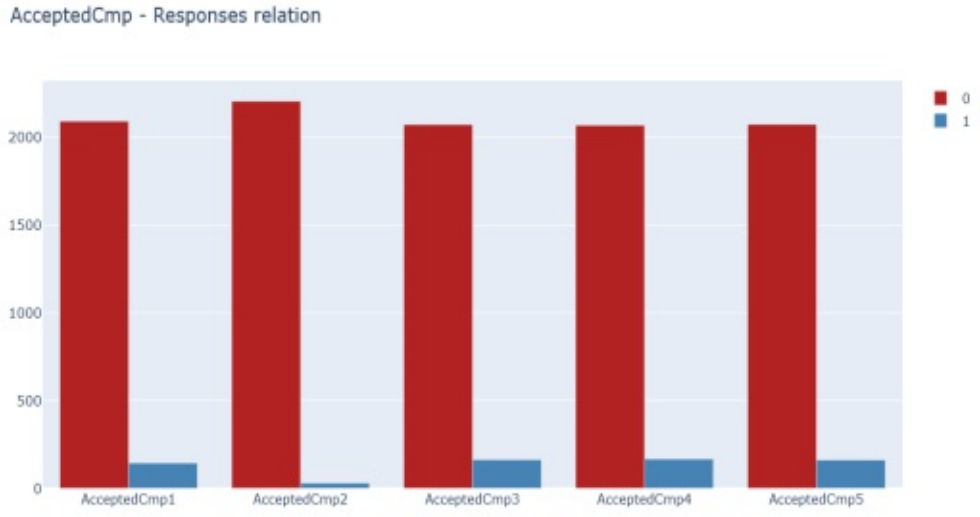


Figure 2.16: AcceptedCmp-Responses relation bar chart

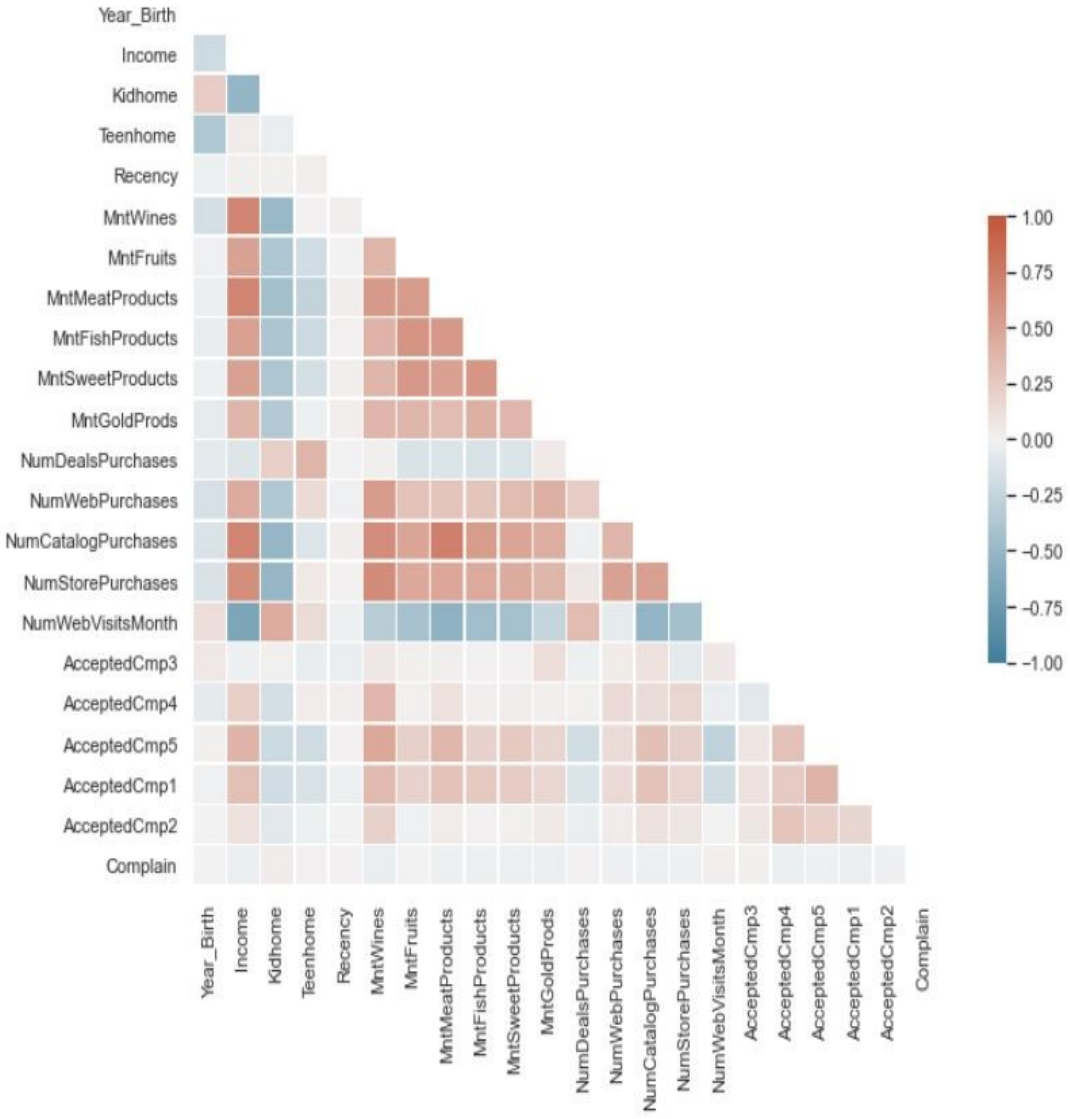


Figure 2.17: Correlation matrix of customer's features

On the Figure 2.17 is given the correlation matrix of the customer's features. Correlation matrix is a table that displays the correlation coefficients among the features. Every cell in the triangular figure displays the correlation between two corresponding features. Red color represents positive correlation between features, while blue color represents negative correlation between features. The more away from 0 the correlation coefficient is, the stronger the relationship between the two features is.

2.3.4 Feature engineering

Feature engineering is a method of generating new features from the raw data to increase the performance of the classification algorithms. It includes the application of transforming functions such as arithmetic and aggregate operations to create new features [NSK+17].

We created 15 new features from the existing features:

1. 'Total Kids' as sum of Kidhome and Teenhome features
2. 'Total_Spending_Products' as sum of amount spent on MeatProducts, FishProducts, WineProducts, GoldProducts, FruitProducts, SweetProducts
3. 'Min_Val_Spending_Products' containing the minimum value of amount spent on MeatProducts, FishProducts, WineProducts, GoldProducts, FruitProducts, SweetProducts
4. 'Max_Val_Spending_Products' containing the maximum value of amount spent on MeatProducts, FishProducts, WineProducts, GoldProducts, FruitProducts, SweetProducts
5. 'Mean_Val_Spending_Products' containing the mean value of amount spent MeatProducts, FishProducts, WineProducts, GoldProducts, FruitProducts, SweetProducts
6. 'Std_Val_Spending_Products' containing the standard deviation of amount spend on MeatProducts, FishProducts, WineProducts, GoldProducts, FruitProducts, SweetProducts
7. 'Total_Number_Purchases' as sum of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm's web page in the past month
8. 'Min_Val_Number_Purchases' containing the minimum value of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm's web page in the past month
9. 'Max_Val_Number_Purchases' containing the maximum value of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm's web page in the past month
10. 'Mean_Val_Number_Purchases' containing the mean value of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm's web page in the past month

11. ‘Std_Val_Number_Purchases’ containing the standard deviation of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm’s web page in the past month

12. ‘Total_Number_Accepted_Campaigns’ as sum of AcceptedCmp1, AcceptedCmp2, AcceptedCmp3, AcceptedCmp4 and AcceptedCmp5 features

13. ‘Sum_Last2_Accepted_Campaigns’ as sum of AcceptedCmp4 and AcceptedCmp5 features

14. ‘Sum_Last3_Accepted_Campaigns’ as sum of AcceptedCmp3, AcceptedCmp4 and AcceptedCmp5 features

15. ‘Sum_Last4_Accepted_Campaigns’ as sum of AcceptedCmp2, AcceptedCmp3, AcceptedCmp4 and AcceptedCmp5 features

On Figure 2.18 is given a code snippet of the feature engineering process.

```
customers['Total_Kids']=customers.loc[:,['Kidhome','Teenhome']].sum(axis=1)

customers['Total_Spending_Products']=customers.loc[:,['MntWines','MntFruits',
          'MntMeatProducts','MntFishProducts','MntSweetProducts','MntGoldProds']].sum(axis=1)

customers['Min_Val_Spending_Products']=customers.loc[:,['MntWines','MntFruits',
          'MntMeatProducts','MntFishProducts','MntSweetProducts','MntGoldProds']].min(axis=1)

customers['Max_Val_Spending_Products']=customers.loc[:,['MntWines','MntFruits',
          'MntMeatProducts','MntFishProducts','MntSweetProducts','MntGoldProds']].max(axis=1)

customers['Std_Val_Spending_Products']=customers.loc[:,['MntWines','MntFruits',
          'MntMeatProducts','MntFishProducts','MntSweetProducts','MntGoldProds']].std(axis=1)

customers['Mean_Val_Spending_Products']=customers.loc[:,['MntWines','MntFruits',
          'MntMeatProducts','MntFishProducts','MntSweetProducts','MntGoldProds']].mean(axis=1)

customers['Total_Number_Purchases']=customers.loc[:,['NumDealsPurchases',
          'NumCatalogPurchases','NumStorePurchases','NumWebPurchases']].sum(axis=1)

customers['Min_Val_Number_Purchases']=customers.loc[:,['NumDealsPurchases',
          'NumCatalogPurchases','NumStorePurchases','NumWebPurchases']].min(axis=1)

customers['Max_Val_Number_Purchases']=customers.loc[:,['NumDealsPurchases',
          'NumCatalogPurchases','NumStorePurchases','NumWebPurchases']].max(axis=1)

customers['Std_Val_Number_Purchases']=customers.loc[:,['NumDealsPurchases',
          'NumCatalogPurchases','NumStorePurchases','NumWebPurchases']].std(axis=1)

customers['Mean_Val_Number_Purchases']=customers.loc[:,['NumDealsPurchases',
          'NumCatalogPurchases','NumStorePurchases','NumWebPurchases']].mean(axis=1)
customers['Total_Number_Accepted_Campaigns']=customers.loc[:,['AcceptedCmp1','AcceptedCmp2',
         'AcceptedCmp3','AcceptedCmp4','AcceptedCmp5']].sum(axis=1)

customers['Sum_Last2_Accepted_Campaigns']=customers.loc[:,['AcceptedCmp4','AcceptedCmp5']].sum(axis=1)

customers['Sum_Last3_Accepted_Campaigns']= customers.loc[:,['AcceptedCmp3','AcceptedCmp4',
         'AcceptedCmp5']].sum(axis=1)

customers['Sum_Last4_Accepted_Campaigns']= customers.loc[:,['AcceptedCmp2','AcceptedCmp3',
         'AcceptedCmp4','AcceptedCmp5']].sum(axis=1)
```

Figure 2.18: Newly created features

2.3.5 Resampling of the imbalanced dataset

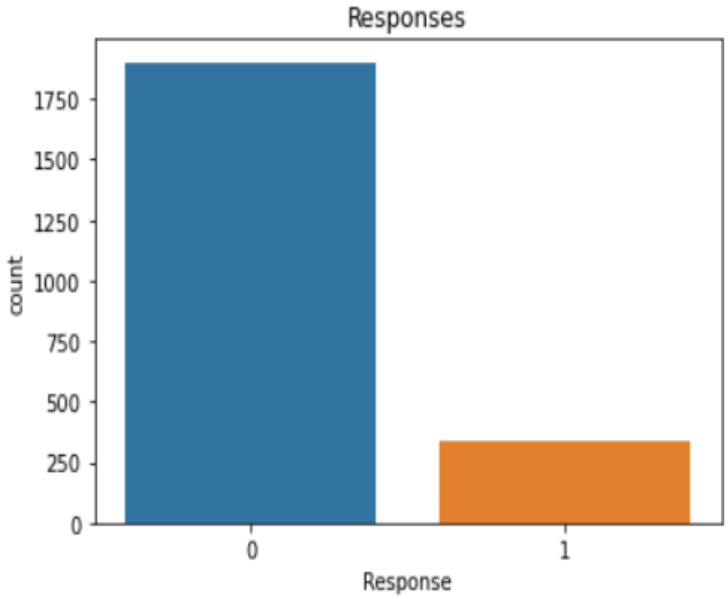


Figure 2.19: Response bar chart

As we can see from the Figure 2.19, the target variable Response has much more zero's than one's. There are 1902 zero's and 334 one's. This means that we have class imbalance.

Class imbalance occurs when one of the two classes has more instances than the other class. In imbalanced dataset the class having more number of instances is called as major class while the one having relatively less number of instances are called as minor class. The problem with class imbalance is that in most of the cases, the classifier is biased to the major class and therefore demonstrates poor classification rates to the minor class [LD13]. Resampling is the process of generating new a transformed version of the training data where the class distribution is more balanced. There are 2 resampling approaches of imbalanced datasets: oversampling and undersampling [GYD+08]. Undersampling techniques eliminate observations from the training data which belong to the major class in order to balance the class distribution. The most basic undersampling technique is random undersampling where balancing of the class distribution is conducted by randomly eliminating observations from the major class. Other undersampling techniques are based on heuristics. Oversampling techniques duplicate observations from the training data which belong to the minor class in order to balance the class distribution. The most basic oversampling technique is random oversampling where balancing the class distribution is conducted by randomly duplicating observations from the minor class. Other oversampling techniques on heuristics based on Synthetic Minority Oversampling Technique (SMOTE). SMOTE creates synthetic minority observations in increase the minor class.

Under sampling technique was not used because the dataset was small (just 2240 observations) and it will lead to big loss of valuable information since the difference between 1s and 0s is around 70 percent. Therefore we choose to perform oversampling technique.

Class SMOTE from imbalanced-learn[2] was used with sampling strategy = 0.75 which represents ratio of the number of observations in the minor class over the number of observations in the major class after resampling and K Nearest Neighbours by default was set to 5 in order to build the synthetic instances. On Figure 2.20 is given a code snippet of SMOTE oversampling of the training dataset.

```
sm = SMOTE(sampling_strategy =0.75)
X_oversampled, y_oversampled = sm.fit_resample(XX_train, yy_train)
```

Figure 2.20: SMOTE oversampling of the training dataset

On the Figure 2.21, we can notice the count of responses (1s and 0s) after SMOTE Oversampling is more balanced.

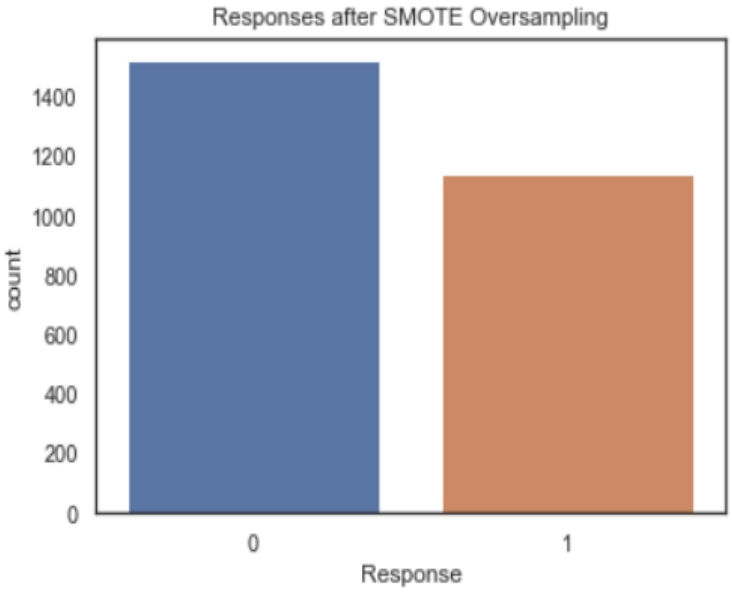


Figure 2.21: Response bar chart after SMOTE

[2]https://imbalanced-learn.org/stable/generated/imblearn.over_sampling.SMOTE

2.4 Modeling

Modeling phase consist of selection and implementation of various modeling techniques in order to build the classification model. There are several techniques for classification problems in data mining, and depending on the form of the dataset there are specific requirements [KS13]. After data preparation was completed, the final dataset was split into training and test set using stratified cross validation. Stratified cross validation splits the data into K equal folds, by preserving the percentage of samples for each class outcome value. In our case, the number of splits of the stratified cross validation was set to 5. The training data was used for learning and fitting the model, and the test data for evaluation.

Furthermore through the process of hyper parameter optimization, the parameters for each of the classifiers were adjusted to their optimal values to improve the model evaluation metrics. This process was perform through GridSearchCV[3] to find the best estimator. GridSearchCV performs in-depth extensive search over explicit values of the parameters. Some of the parameters of the classifiers that were tuned are: C, gamma, kernel, activation, solver, alpha, learning_rate, max_iter, n_estimators, hidden_layer_sizes, criterion, max_features, max_depth and penalty.

For prediction of the customer's response to the marketing campaign, several machine learning algorithms for classification were selected and implemented using scikit-learn library for Python. The selected machine learning algorithms are:

1. Naïve Bayes
2. Support Vector Machine
3. Multi-level Perceptron
4. Logistic Regression
5. Random Forest

For each of the algorithms was generated a confusion matrix and Receiver Operating Characteristic (ROC) curve figures for each of the 5 folds. Confusion matrix shows the number of True Positive, True Negative, False Positive and False Negative classified instances. Receiver operating characteristic (ROC) curve is a chart that points out the performance of a given algorithm at all classification thresholds.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

2.4.1 Naive Bayes

The Naïve Bayes algorithm trains the data presuming that the variables are independent [R+01]. This classifier applies the Bayes theorem with the 'naïve' assumption of conditional independence between every pair of features given the value of the class variable. The benefit of Naive Bayes is the fact that it only needs a small number of training data to approximate the required parameters for classification.

By means of the Bayes theorem we can find what the probability of event A to happen is, given that event B has already happened. On Figure 2.22 is given a mathematical formula of the Bayes theorem [Efr13].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 2.22: The Bayes theorem, [Efr13]

On the Figure 2.23 is given the code for Gaussian Naïve Bayes[4] implementation including the procedure for generating confusion matrices and ROC curves for the 5 Folds.

```
#Naive Bayes
gnb = GaussianNB()
yy_pred = gnb.fit(XX_train_transformed, y_oversampled).predict_proba(XX_test_transformed)[:, 1]
yy_pred_binary=gnb.predict(XX_test_transformed)

score=roc_auc_score(yy_test, yy_pred)
print('Naive Bayes ROC AUC score:',score)
scoresNB.append(score)


fpr, tpr, _ = roc_curve(yy_test, yy_pred)
roc_auc = auc(fpr, tpr)
title_roc=str(str(i)+" Fold of ROC Naive Bayes")
plot_roc_auc_curve(fpr, tpr,roc_auc,title_roc)


cf_matrix = confusion_matrix(yy_test, yy_pred_binary)
title=str(str(i)+" Fold of Naive Bayes")
make_confusion_matrix(cf_matrix,
                      group_names=labels,
                      categories=categories,
                      title=title)
```

Figure 2.23: Naive Bayes implementation

[4] https://scikit-learn.org/stable/modules/naive_bayes.html

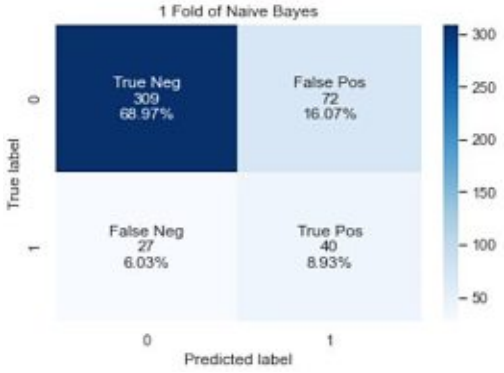


Figure 2.24: 1st Fold Naive Bayes confusion matrix

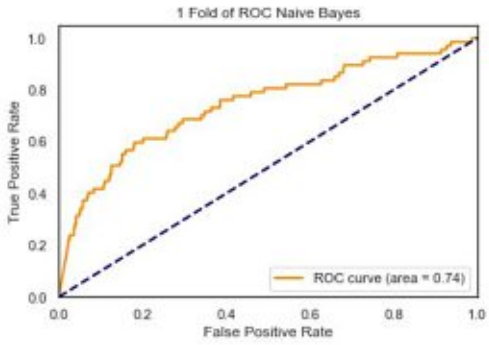


Figure 2.25: 1st Fold Naive Bayes ROC curve

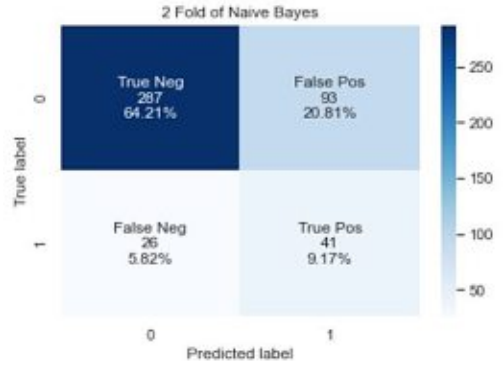


Figure 2.26: 2nd Fold Naive Bayes confusion matrix

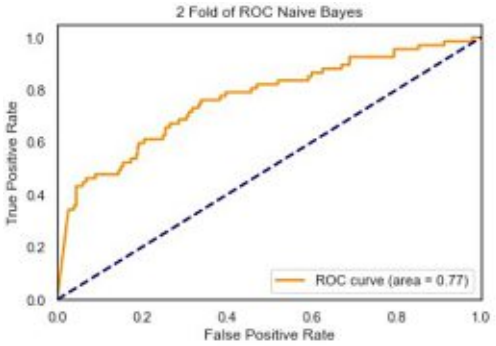


Figure 2.27: 2nd Fold Naive Bayes ROC curve

Figure 2.28: 3rd Fold Naive Bayes confusion matrix

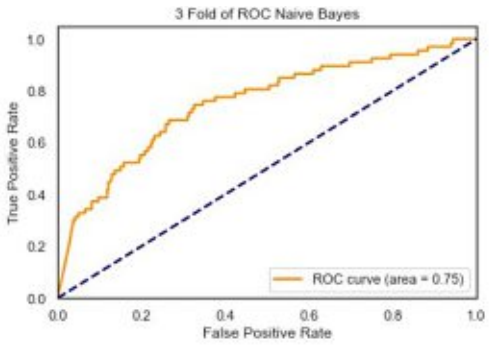


Figure 2.29: 3rd Fold Naive Bayes ROC curve

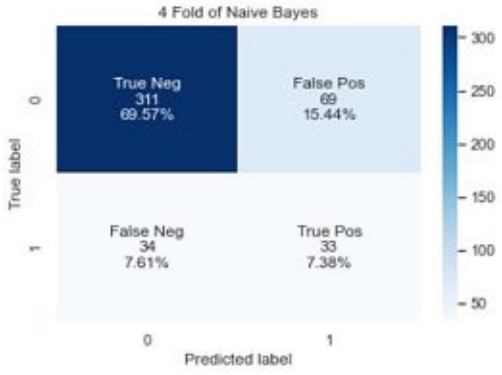


Figure 2.30: 4th Fold Naive Bayes confusion matrix

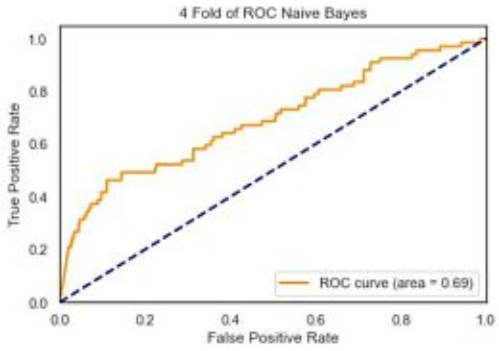


Figure 2.31: 4th Fold Naive Bayes ROC curve

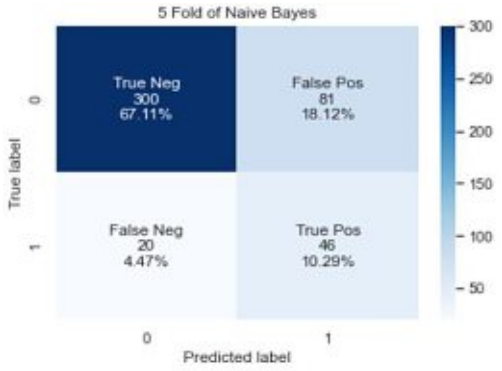


Figure 2.32: 5th Fold Naive Bayes confusion matrix

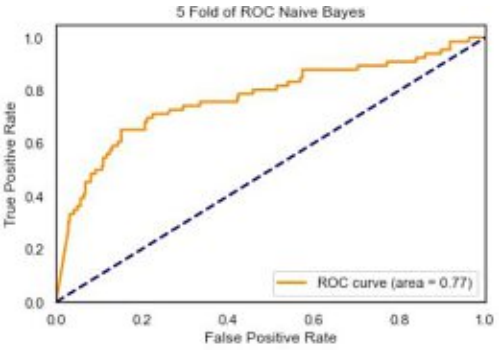


Figure 2.33: 5th Fold Naive Bayes ROC curve

On Figures 2.24, 2.26, 2.28, 2.30, 2.32 are given the confusion matrices for the 5 Folds for Naive Bayes classifier. From these figures, it can be concluded that the highest number of True Negatives (311) is in the 4th Fold, the highest number of True Positives (46) is in the 5th Fold , the highest number of False Positives (93) is in the 2nd Fold and the highest number of False Negatives (34) is in the 4th Fold.

On Figures 2.25, 2.27, 2.29, 2.31, 2.33 are given the ROC curves for the 5 Folds for Naive Bayes classifier. From these figures, it can be concluded that the area under the ROC curve was highest in the 2nd and the 5th Fold which is area = 0.77.

2.4.2 Support Vector Machine

Support Vector Machine (SVM) is an algorithm for maximizing a particular mathematical function with respect to a given collection of data [Nob06]. It learns how to assign appropriate labels to objects from examples and can be implemented for both classification and regression. The goal of Support Vector Machine algorithm is to find a hyperplane in N-dimensional space (N is the number of variables) which distinctly classifies the data points.

For implementation was used Support Vector Classification (SVC)[5] from scikit-learn library for Python. The parameters of the Support Vector Machine classifier that were tuned through GridSearchCV are 'C', 'gamma' and 'kernel'. The parameter 'C' is regularization parameter, 'gamma' is kernel coefficient and 'kernel' specifies the kernel type to be used in the algorithm. On Figure 2.34 is given a code snippet of Support Vector Machine implementation that generates also the confusion matrices and ROC curves for the 5 Folds.

```
#SVM
param_grid = {'C': [1, 10, 100, 1000],
              'gamma': [0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf', 'poly']}
grid = GridSearchCV(SVC(probability=True),param_grid,refit=True, n_jobs=-1)
grid.fit(XX_train_transformed,y_oversampled)
off_svc = grid.best_estimator_
print(off_svc)
yy_pred = off_svc.predict_proba(XX_test_transformed)[:, 1]
yy_pred_binary=off_svc.predict(XX_test_transformed)

score=roc_auc_score(yy_test, yy_pred)
print('SVM ROC AUC score:',score)
scoresSVM.append(score)

fpr, tpr, _ = roc_curve(yy_test, yy_pred)
roc_auc = auc(fpr, tpr)
title_roc=str(str(i)+" Fold of ROC SVM")
plot_roc_auc_curve(fpr, tpr,roc_auc,title_roc)

cf_matrix = confusion_matrix(yy_test, yy_pred_binary)
title=str(str(i)+" Fold of SVM")
make_confusion_matrix(cf_matrix,
                      group_names=labels,
                      categories=categories,
                      title=title)
```

Figure 2.34: Support Vector Machine implementation

[5] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

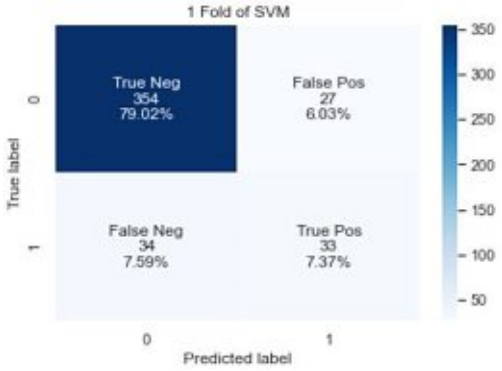


Figure 2.35: 1st Fold Support Vector Machine confusion matrix

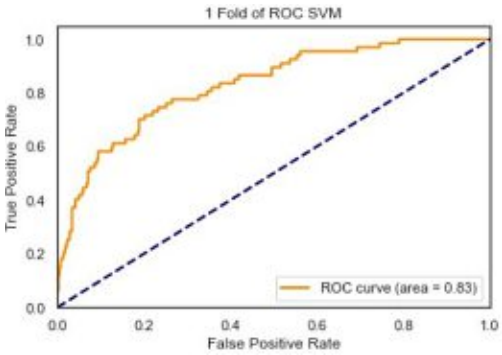


Figure 2.36: 1st Fold Support Vector Machine ROC curve

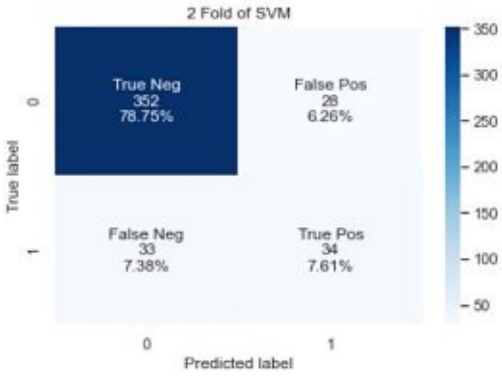


Figure 2.37: 2nd Fold Support Vector Machine confusion matrix

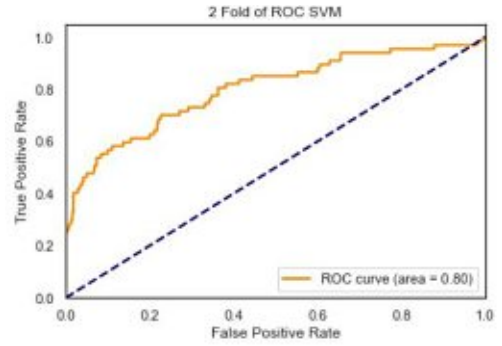


Figure 2.38: 2nd Fold Support Vector Machine ROC curve

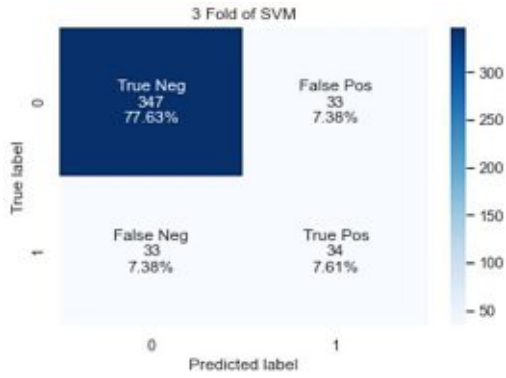


Figure 2.39: 3rd Fold Support Vector Machine confusion matrix

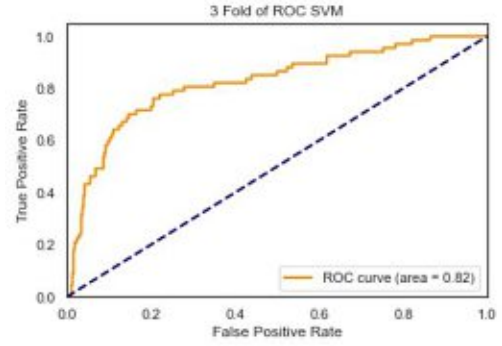


Figure 2.40: 3rd Fold Support Vector Machine ROC curve

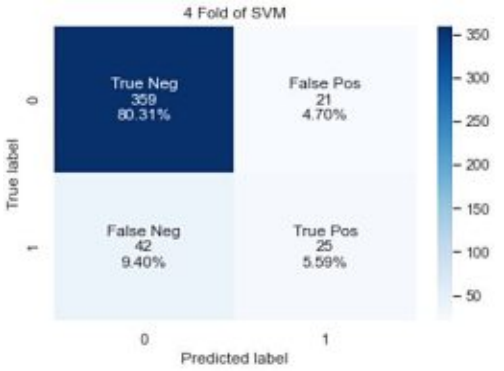


Figure 2.41: 4th Fold Support Vector Machine confusion matrix

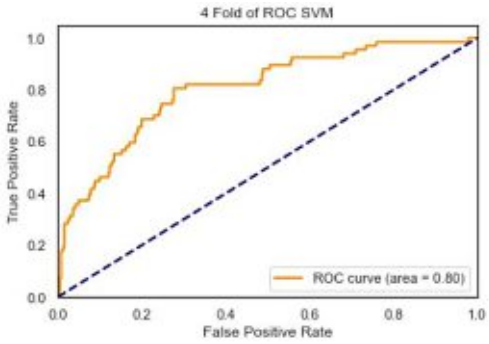


Figure 2.42: 4th Fold Support Vector Machine ROC curve

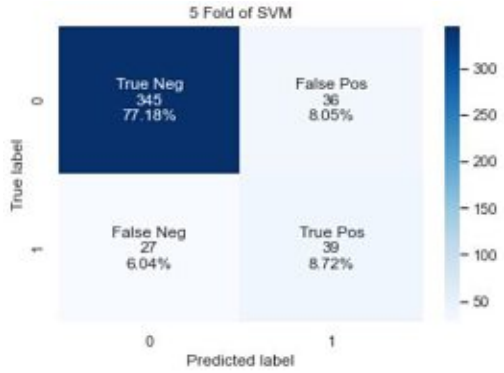


Figure 2.43: 5th Fold Support Vector Machine confusion matrix

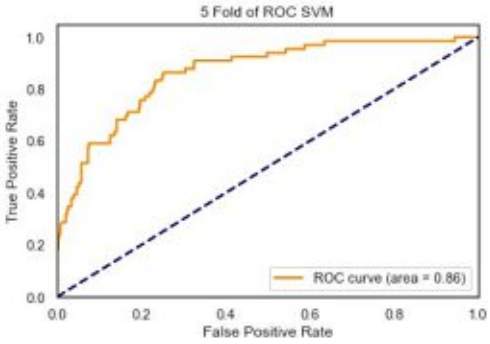


Figure 2.44: 5th Fold Support Vector Machine ROC curve

On Figures 2.35, 2.37, 2.39, 2.41, 2.43 are given the confusion matrices for the 5 Folds for Support Vector Machine classifier. From these figures, it can be concluded that the highest number of True Negatives (359) is in the 4th Fold, the highest number of True Positives (39) is in the 5th Fold , the highest number of False Positives (36) is in the 5th Fold and the highest number of False Negatives (42) is in the 4th Fold.

On Figures 2.36, 2.38, 2.40, 2.42, 2.44 are given the ROC curves for the 5 Folds for Support Vector Machine classifier. From these figures, it can be concluded that the area under the ROC curve was highest in the the 5th Fold which is area = 0.86.

2.4.3 Multi-layer Perceptron

Multi-layer Perceptron (MLP) classifier is a class of feed forward artificial neural network [Nor05]. Artificial neurons are the main components for building neural networks. They are basic computational units that have weights as input signals and generate an output using an activation function. Multi-layer Perceptron contains minimum three layers of nodes: an input layer, a hidden layer and an output layer. For training, Multi-layer Perceptron uses a technique called backpropagation.

The parameters of the Multi-layer Perceptron[6] classifier that were tuned through GridSearchCV are 'activation', 'solver', 'alpha', 'learning_rate', 'max_iter' and 'hidden_layer_sizes'. On Figure 2.45 is given a code snippet of Multi-layer Perceptron implementation that generates also the confusion matrices and ROC curves for the 5 Folds.

```
#MLP Classifier
param_grid = {'activation': ['tanh', 'relu','logistic','identity'],
              'solver': ['sgd', 'adam','lbfgs'],
              'alpha': [0.0001, 0.05],
              'learning_rate': ['constant','adaptive','invscaling'],
              'max_iter':[100,150,200],
              'hidden_layer_sizes':[(50,50,50),(50,100,50), (100,)]}
grid = GridSearchCV(MLPClassifier(),param_grid,refit=True, n_jobs=-1)
grid.fit(XX_train_transformed,y_oversampled)
off_MLP = grid.best_estimator_
print(off_MLP)
yy_pred = off_MLP.predict_proba(XX_test_transformed)[:, 1]
yy_pred_binary=off_MLP.predict(XX_test_transformed)

score=roc_auc_score(yy_test, yy_pred)
print('MLP ROC AUC score:',score)
scoresMLP.append(score)

fpr, tpr, _ = roc_curve(yy_test, yy_pred)
roc_auc = auc(fpr, tpr)
title_roc=str(str(i)+" Fold of ROC MLP")
plot_roc_auc_curve(fpr, tpr,roc_auc,title_roc)

cf_matrix = confusion_matrix(yy_test, yy_pred_binary)
title=str(str(i)+" Fold of MLP")
make_confusion_matrix(cf_matrix,
                      group_names=labels,
                      categories=categories,
                      title=title)
```

Figure 2.45: Multi-layer Perceptron implementation

[6] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

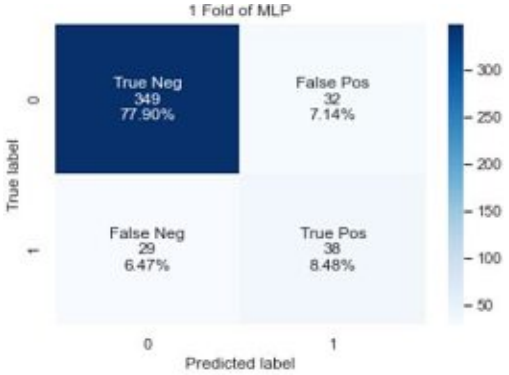


Figure 2.46: 1st Fold Multi-layer Perceptron confusion matrix

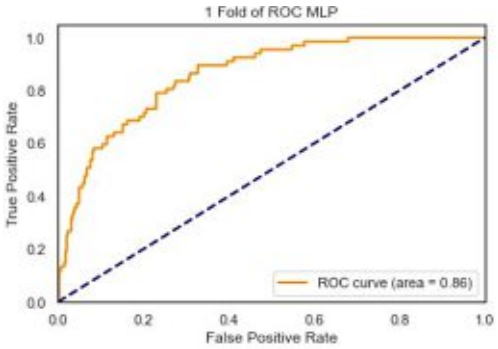


Figure 2.47: 1st Fold Multi-layer Perceptron ROC curve

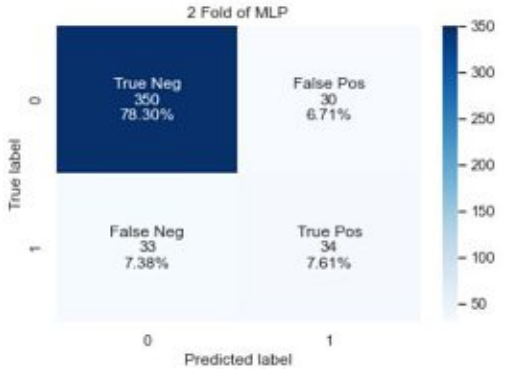


Figure 2.48: 2nd Fold Multi-layer Perceptron confusion matrix

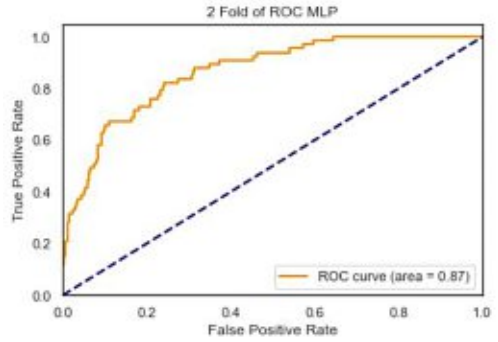


Figure 2.49: 2nd Fold Multi-layer Perceptron ROC curve

Figure 2.50: 3rd Fold Multi-layer Perceptron confusion matrix

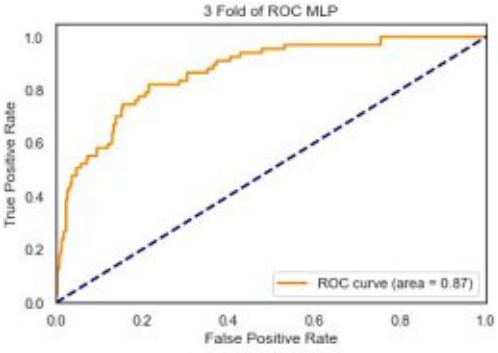


Figure 2.51: 3rd Fold Multi-layer Perceptron ROC curve

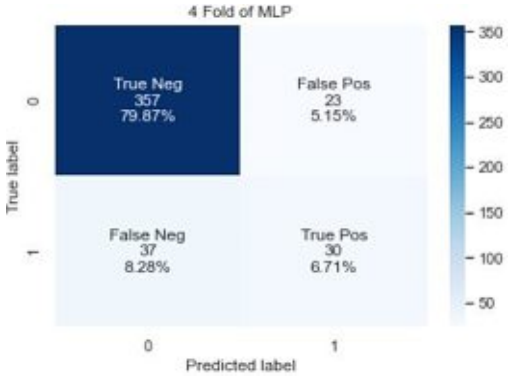


Figure 2.52: 4th Fold Multi-layer Perceptron confusion matrix

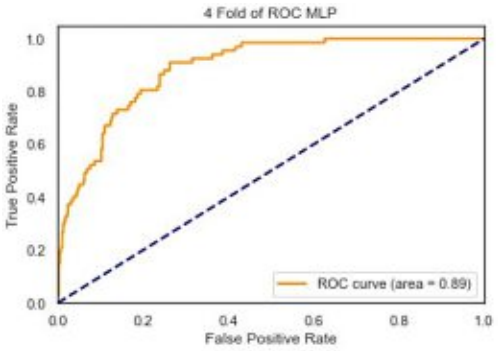


Figure 2.53: 4th Fold Multi-layer Perceptron ROC curve

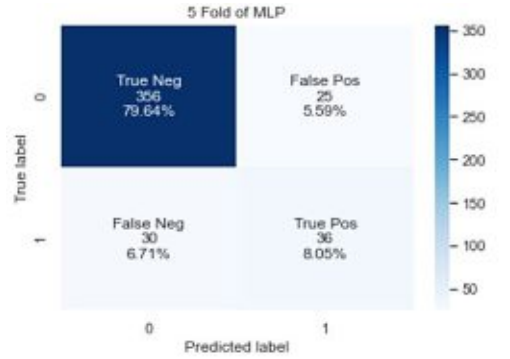


Figure 2.54: 5th Fold Multi-layer Perceptron confusion matrix

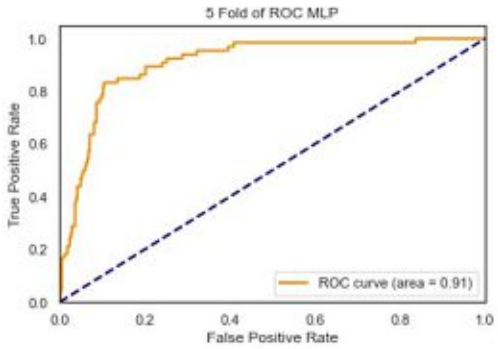


Figure 2.55: 5th Fold Multi-layer Perceptron ROC curve

On Figures 2.46, 2.48, 2.50, 2.52, 2.54 are given the confusion matrices for the 5 Folds for Multi-layer Perceptron classifier. From these figures, it can be concluded that the highest number of True Negatives (358) is in the 3rd Fold, the highest number of True Positives (38) is in the 1st Fold , the highest number of False Positives (32) is in the 1st Fold and the highest number of False Negatives (37) is in the 4th Fold.

On Figures 2.47, 2.49, 2.51, 2.53, 2.55 are given the ROC curves for the 5 Folds for Multi-layer Perceptron classifier. From these figures, it can be concluded that the area under the ROC curve was highest in the the 5th Fold which is area = 0.91.

2.4.4 Logistic Regression

Logistic regression is very prevalent classifier in the recent years. It explains the relation of dependent variable and independent variable. [SU07]. Logistic regression modifies the output using the logistic sigmoid function and as outcome returns probability value that is mapped into two discrete classes.

The parameters of Logistic Regression[7] that were tuned through GridSearchCV are 'penalty', 'max_iter', 'C' and 'solver'. The 'penalty' parameter is used to specify the norm used in penalization, the 'max_iter' parameter represents the maximum number of iterations taken for the solver to converge, 'C' is the inverse of regularization strength. On Figure 2.56 is given a code snippet of Logistic Regression implementation as well as the code for generating the confusion matrices and ROC curves for the 5 Folds.

```
#Logistic Regression
param_grid = {'penalty':['l1','l2','elasticnet'],
              'max_iter':range(20,150,10),
              'C': [1, 10, 100, 1000],
              'solver':['lbfgs','liblinear','sag','saga']}
grid = GridSearchCV(LogisticRegression(),param_grid,refit=True, n_jobs=-1)
grid.fit(XX_train_transformed,y_oversampled)
off_log_reg = grid.best_estimator_
print(off_log_reg)
yy_pred = off_log_reg.predict_proba(XX_test_transformed)[:, 1]
yy_pred_binary=off_log_reg.predict(XX_test_transformed)

score=roc_auc_score(yy_test, yy_pred)
print('Logistic Regression ROC AUC score:',score)
scoresLG.append(score)

fpr, tpr, _ = roc_curve(yy_test, yy_pred)
roc_auc = auc(fpr, tpr)
title_roc=str(str(i)+" Fold of ROC Logistic Regression")
plot_roc_auc_curve(fpr, tpr,roc_auc,title_roc)

cf_matrix = confusion_matrix(yy_test, yy_pred_binary)
title=str(str(i)+" Fold of Logistic Regression")
make_confusion_matrix(cf_matrix,
                      group_names=labels,
                      categories=categories,
                      title=title)
```

Figure 2.56: Logistic Regression implementation

[7] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

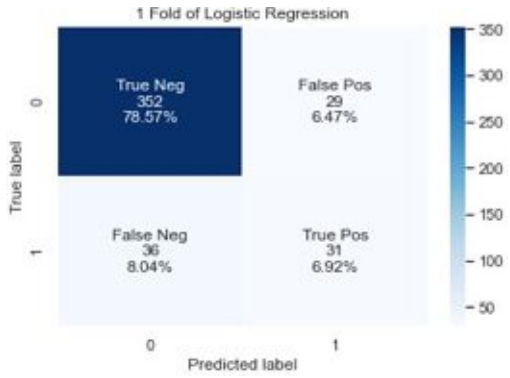


Figure 2.57: 1st Fold Logistic Regression confusion matrix

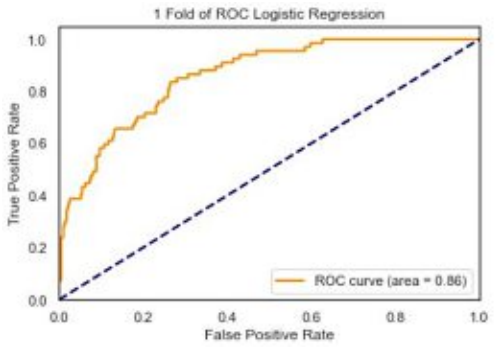


Figure 2.58: 1st Fold Logistic Regression ROC curve

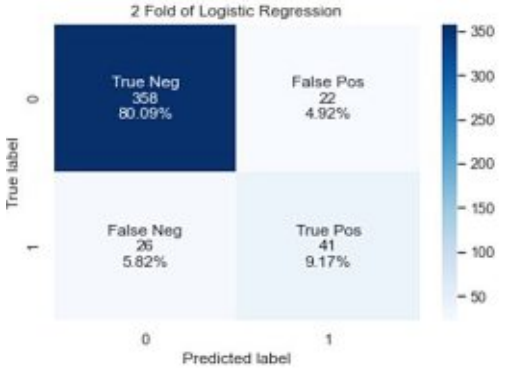


Figure 2.59: 2nd Fold Logistic Regression confusion matrix

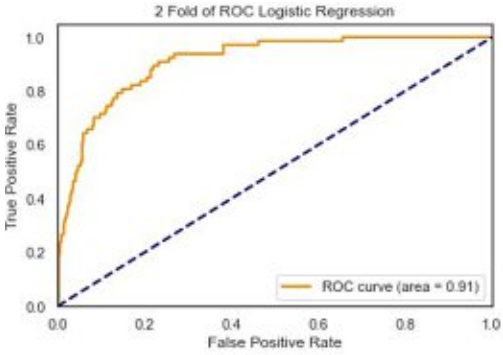


Figure 2.60: 2nd Fold Logistic Regression ROC curve

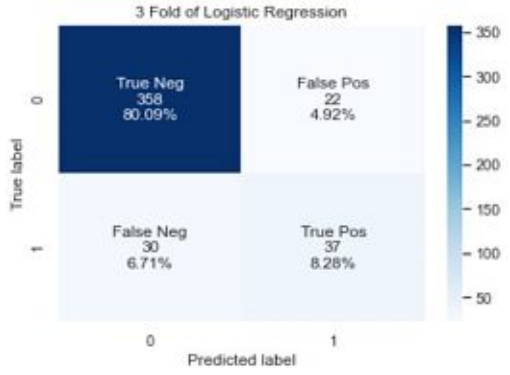


Figure 2.61: 3rd Fold Logistic Regression confusion matrix

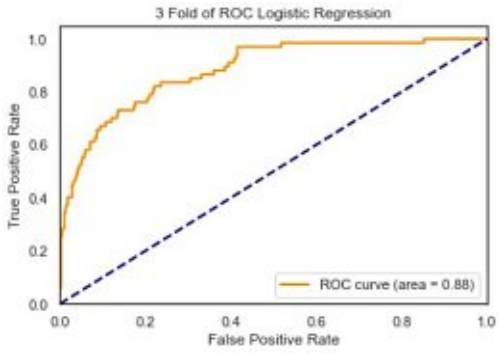


Figure 2.62: 3rd Fold Logistic Regression ROC curve

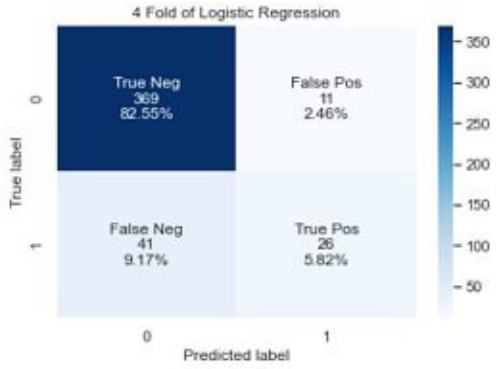


Figure 2.63: 4th Fold Logistic Regression confusion matrix

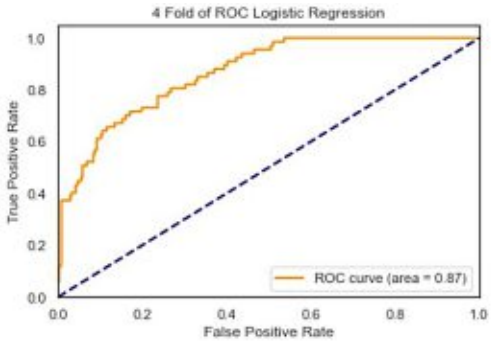


Figure 2.64: 4th Fold Logistic Regression ROC curve

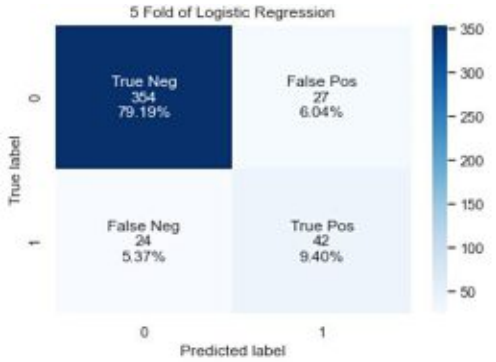


Figure 2.65: 5th Fold Logistic Regression confusion matrix

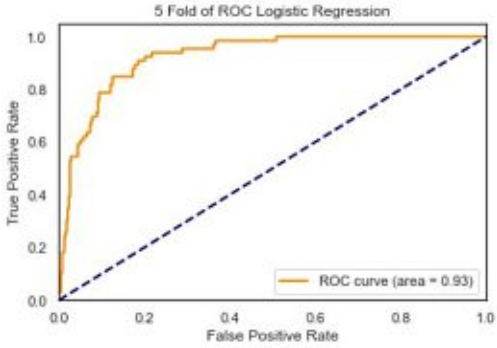


Figure 2.66: 5th Fold Logistic Regression ROC curve

On Figures 2.57, 2.59, 2.61, 2.63, 2.65 are given the confusion matrices for the 5 Folds for Logistic Regression classifier. From these figures, it can be concluded that the highest number of True Negatives (369) is in the 4th Fold, the highest number of True Positives (42) is in the 5th Fold , the highest number of False Positives (29) is in the 1st Fold and the highest number of False Negatives (41) is in the 4th Fold.

On Figures 2.58, 2.60, 2.62, 2.64, 2.66 are given the ROC curves for the 5 Folds for Logistic Regression classifier. From these figures, it can be concluded that the area under the ROC curve was highest in the the 5th Fold which is area = 0.93.

2.4.5 Random Forest

Random Forest is ensemble learning approach that generates many individual decision trees while training the data. Every of these trees creates a prediction of a class and the final model's prediction is mode of the classes of the individual decision trees [BS16]. In general, random forest classifier outperforms decision tree classifier and prevents the issue of over fitting on the training dataset.

The parameters of Random Forest[8] that were tuned through GridSearchCV are 'n_estimators', 'criterion', 'max_features', 'max_depth'. The parameter 'n_estimators' specifies the amount of trees in the forest, 'criterion' is the function to measure the quality of the split, 'max_features' represents the amount of variables to be considered when searching for the best split and 'max_depth' is the maximum depth of the tree. On Figure 2.67 is given a code snippet of Random Forest Implementation.

```
#Random Forest
param_grid = {'n_estimators':range(10,150,10),
              'criterion':['gini','entropy'],
              'max_features':['auto','sqrt','log2'],
              'max_depth':range(3,7,1)}
grid = GridSearchCV(RandomForestClassifier(),param_grid,refit=True, n_jobs=-1)
grid.fit(XX_train_transformed,y_oversampled)
off_rf = grid.best_estimator_
print(off_rf)
yy_pred = off_rf.predict_proba(XX_test_transformed)[:, 1]
yy_pred_binary=off_rf.predict(XX_test_transformed)

score=roc_auc_score(yy_test, yy_pred)
print('Random Forest ROC AUC score:',score)
scoresRF.append(score)

fpr, tpr, _ = roc_curve(yy_test, yy_pred)
roc_auc = auc(fpr, tpr)
title_roc=str(str(i)+" Fold of ROC Random Forest")
plot_roc_auc_curve(fpr, tpr,roc_auc,title_roc)

cf_matrix = confusion_matrix(yy_test, yy_pred_binary)
title=str(str(i)+" Fold of Random Forest")
make_confusion_matrix(cf_matrix,
                      group_names=labels,
                      categories=categories,
                      title=title)
```

Figure 2.67: Random Forest implementation

[8] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Figure 2.68: 1st Fold Random Forest confusion matrix

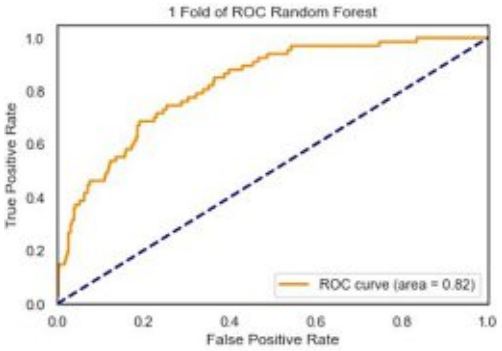


Figure 2.69: 1st Fold Random Forest ROC curve

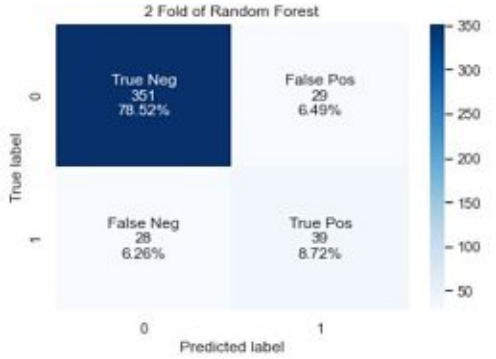


Figure 2.70: 2nd Fold Random Forest confusion matrix

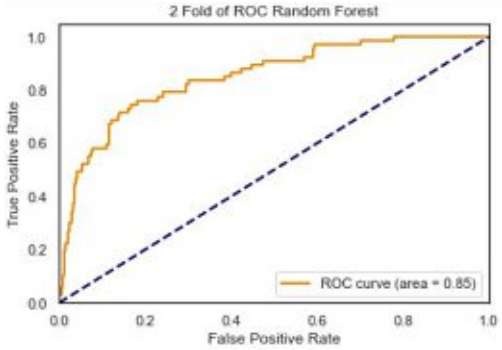


Figure 2.71: 2nd Fold Random Forest ROC curve

Figure 2.72: 3rd Fold Random Forest confusion matrix

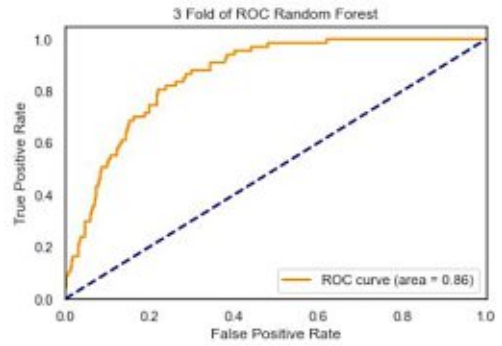


Figure 2.73: 3rd Fold Random Forest ROC curve

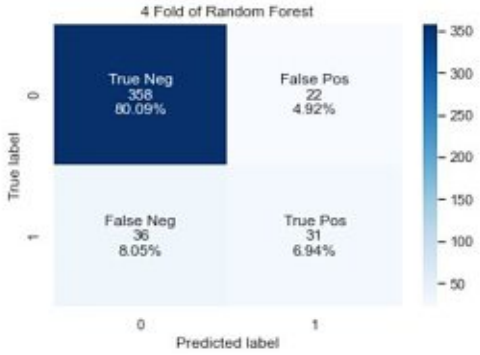


Figure 2.74: 4th Fold Random Forest confusion matrix

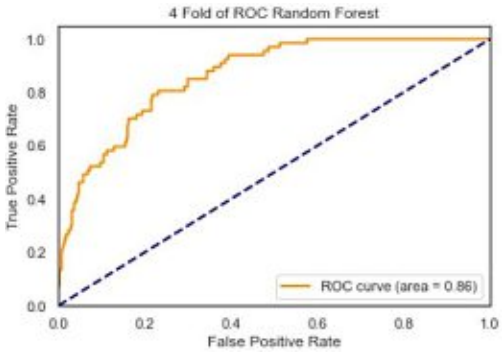


Figure 2.75: 4th Fold Random Forest ROC curve

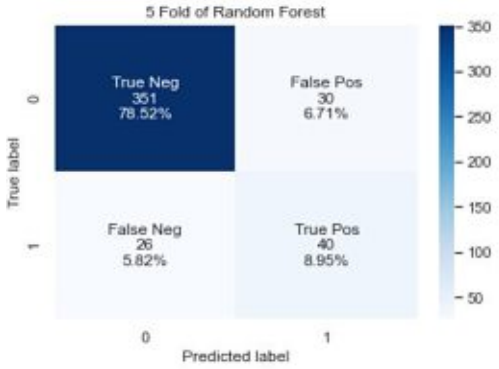


Figure 2.76: 5th Fold Random Forest confusion matrix

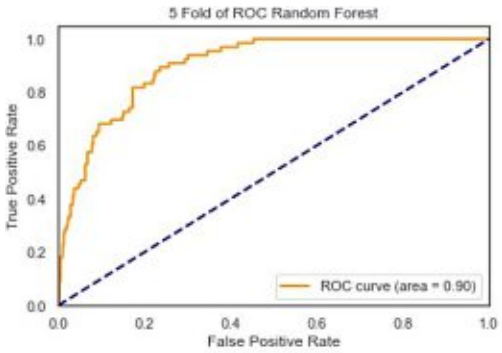


Figure 2.77: 5th Fold Random Forest ROC curve

On Figures 2.68, 2.70, 2.72, 2.74, 2.76 are given the confusion matrices for the 5 Folds for Random Forest classifier. From these figures, it can be concluded that the highest number of True Negatives (358) is in the 4th Fold, the highest number of True Positives (40) is in the 5th Fold , the highest number of False Positives (42) is in the 1st Fold and the highest number of False Negatives (36) is in the 4th Fold.

On Figures 2.69, 2.71, 2.73, 2.75, 2.77 are given the ROC curves for the 5 Folds for Random Forest classifier. From these figures, it can be concluded that the area under the ROC curve was highest in the the 5th Fold which is area = 0.90.

2.5 Evaluation

Evaluation aims to estimate the generalization model accuracy on unseen data.Evaluation metrics can be defined as measurement of the performance of a classifier by evaluating different characteristics of the classifier [HS15]. They are important in the process of model selection because the task of evaluation metrics is to find out the most precise classifier which will produce the most correct classification on unseen data. The evaluation metrics that were used are: classification accuracy, precision, recall, F1 score and ROC AUC score.

True Positive is a result when the classifier correctly predicts the positive class. True Negative is a result when the classifier correctly predicts the negative class. False Positive is a result when the classifier incorrectly predicts the positive class. False Negative is a result when the classifier incorrectly predicts the negative class.

Accuracy is the sum of true negatives and true positives classified instances divided by the sum of true negatives, true positives, false positives and false negatives [HS15].

$$Accuracy = \frac{TrueNegatives + TruePositives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives} \tag{2.1}$$

Precision is the amount of true positives classified instances divided by the sum of true positives and false positives classified instances [HS15].

$$Precision = \frac{TruePositives}{TruePositves + FalsePositives} \tag{2.2}$$

Recall is the amount of true positives classified instances divided by the sum of true positives and false negatives classified instances [HS15].

$$Recall = \frac{TruePositives}{TruePositves + FalseNegatives} \tag{2.3}$$

F1 Score can be defined as the harmonic mean of precision and recall. It ranges from 0 to 1 and demonstrates the robustness of a given classifier [HS15].

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.4}$$

ROC AUC Score is a performance metric for measuring the ability of a binary classifier to discriminate between positive and negative classes [HS15].

Below are given 5 tables for each classifier that contain every evaluation metric value for each of the 5 folds. In the last row, the average of all 5 folds for each evaluation metric is calculated.

Fold	Accuracy	Precision	Recall	F1 Score	ROC AUC Score
1	0.779	0.357	0.597	0.447	0.743075959
2	0.734	0.306	0.612	0.408	0.765278869
3	0.749	0.320	0.597	0.417	0.752022781
4	0.770	0.324	0.493	0.391	0.692046347
5	0.774	0.362	0.697	0.477	0.770878072
Average	**0.7612**	**0.3338**	**0.5992**	**0.428**	**0.744660406**

Table 2.2: Naive Bayes evaluation metrics

From the Table 2.2 can be concluded that Naive Bayes classifier has highest Accuracy 0.779 in the 1st Fold, highest Precision 0.362 in the 5th Fold, highest Recall 0.697 in the 5th Fold, highest F1 Score 0.477 in the 5th Fold and highest ROC AUC Score 0.770878072 in the 5th Fold.

Fold	Accuracy	Precision	Recall	F1 Score	ROC AUC Score
1	0.864	0.550	0.493	0.520	0.828201512
2	0.864	0.548	0.507	0.527	0.800216025
3	0.852	0.507	0.507	0.507	0.822014925
4	0.859	0.543	0.373	0.442	0.804340141
5	0.859	0.520	0.591	0.553	0.863934622
Average	**0.8596**	**0.5336**	**0.4942**	**0.5098**	**0.823741445**

Table 2.3: Support Vector Machine evaluation metrics

From the Table 2.3 can be concluded that Support Vector Machine classifier has highest Accuracy 0.864 in the 1st and 2nd Folds, highest Precision 0.550 in the 1st Fold, highest Recall 0.591 in the 5th Fold, highest F1 Score 0.553 in the 5th Fold and highest ROC AUC Score 0.863934622 in the 5th Fold.

Fold	Accuracy	Precision	Recall	F1 Score	ROC AUC Score
1	0.864	0.543	0.567	0.555	0.858835743
2	0.859	0.531	0.507	0.519	0.865357423
3	0.879	0.614	0.522	0.565	0.867458759
4	0.866	0.566	0.448	0.500	0.888275727
5	0.877	0.590	0.545	0.567	0.907758689
Average	**0.869**	**0.5688**	**0.5178**	**0.5412**	**0.877537268**

Table 2.4: Multi-layer Perceptron evaluation metrics

From the Table 2.4 can be concluded that Multi-layer Perceptron classifier has highest Accuracy 0.879 in the 3rd Fold, highest Precision 0.614 in the 3rd Fold, highest Recall 0.567 in the 1st Fold, highest F1 Score 0.567 in the 5th Fold and highest ROC AUC Score 0.907758689 in the 5th Fold.

Fold	Accuracy	Precision	Recall	F1 Score	ROC AUC Score
1	0.855	0.517	0.463	0.488	0.859580052
2	0.893	0.651	0.612	0.631	0.909465829
3	0.884	0.627	0.552	0.587	0.882227023
4	0.884	0.703	0.388	0.500	0.867026709
5	0.886	0.609	0.636	0.622	0.925693947
Average	**0.8804**	**0.6214**	**0.5302**	**0.5656**	**0.888798712**

Table 2.5: Logistic Regression evaluation metrics

From the Table 2.5 can be concluded that Logistic Regression classifier has highest Accuracy 0.893 in the 2nd Fold, highest Precision 0.703 in the 4th Fold, highest Recall 0.636 in the 5th Fold, highest F1 Score 0.631 in the 2nd Fold and highest ROC AUC Score 0.925693947 in the 5th Fold.

Fold	Accuracy	Precision	Recall	F1 Score	ROC AUC Score
1	0.826	0.425	0.463	0.443	0.824205743
2	0.872	0.574	0.582	0.578	0.850785546
3	0.841	0.472	0.507	0.489	0.859328358
4	0.870	0.585	0.463	0.517	0.863531029
5	0.875	0.571	0.606	0.588	0.90012328
Average	**0.8568**	**0.5254**	**0.5242**	**0.523**	**0.859594791**

Table 2.6: Random Forest evaluation metrics

From the Table 2.6 can be concluded that Random Forest classifier has highest Accuracy 0.875 in the 5th Fold, highest Precision 0.585 in the 4th Fold, highest Recall 0.606 in the 5th Fold, highest F1 Score 0.588 in the 5th Fold and highest ROC AUC Score 0.90012328 in the 5th Fold.

2.6 Deployment

Deployment phase comprises of monitoring, maintenance and producing final report that supports the company to maximize the profit of the next marketing campaign.

CHAPTER 3

Results

In this chapter are explained the results obtained from the research, where the answers from the Research Question 1 and Research Question 2 are presented. Graphical representation of the feature importance is interpreted as answer to the first research question and tabular form of evaluation metrics comparison among different classifiers is given as answer to the second research question.

The problem statement was that in the marketing campaign, all the customers are targeted with advertisements including the ones who will not respond positive to the marketing campaign and reject the offer. This means that the company is not working efficiently, its marketing campaign is not optimized because the customers are not segmented and targeted correctly. As a result, the costs are increased and the company's profit is not maximized. This can lead to a failure of company's marketing campaign.

The objective was to build a classification model using customer data, that classifies which customers will accept and which customers will reject the offer. The results of the most precise predictive model can be used to segment and target correctly the customers who will respond positive and accept to the marketing campaign and to avoid targeting the customers who will respond negative and reject the marketing offer. This leads to an efficient work of the company, reduced costs and maximization of the company's profit.

Research Question 1: Which of the features are the most important in the model construction for predicting the customer's response to the marketing campaign based on personalized customer experience?

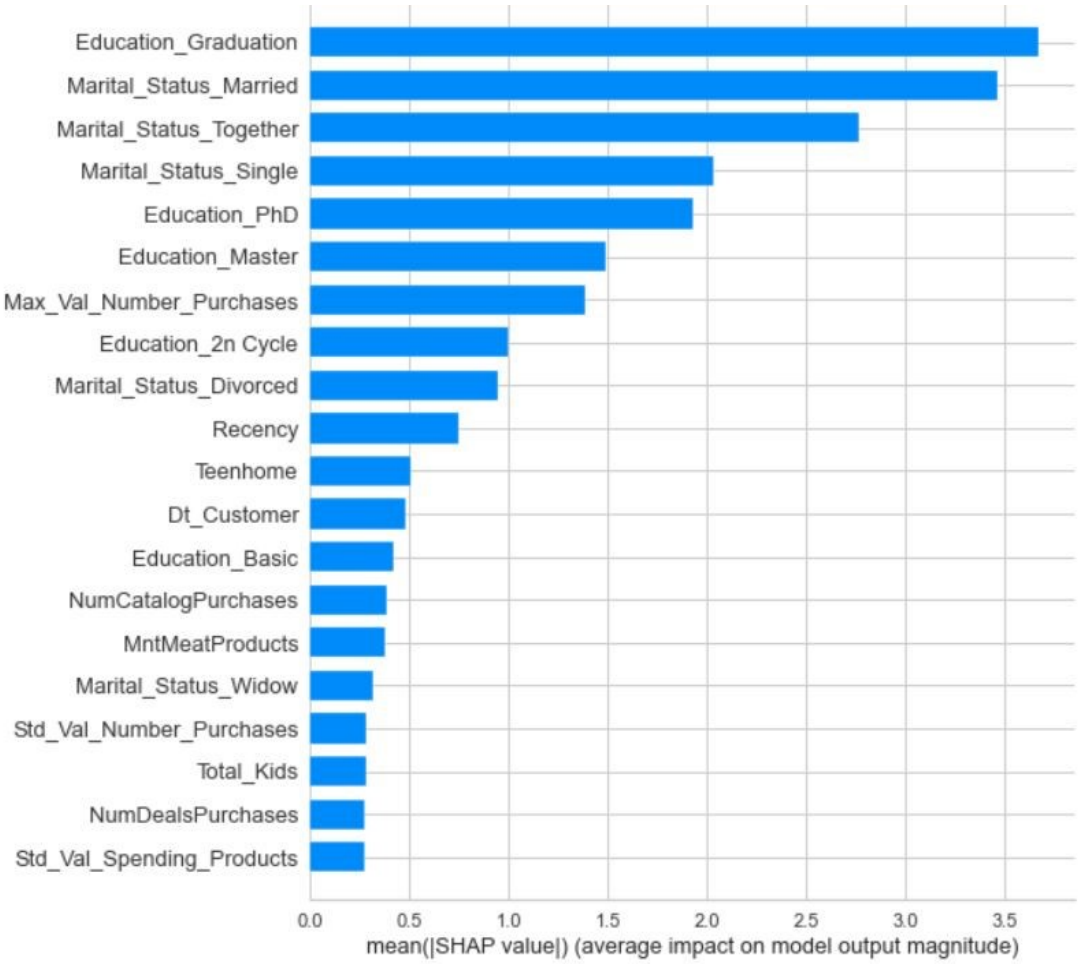


Figure 3.1: Logistic Regression feature importance

The feature importance graph was calculated based on Logistic Regression classifier. As it can be observed from Figure 3.1, the most important features used in the model construction for predicting the customer's response to the marketing campaign based on personalized customer experience are 'Education_Graduation', 'Education_PhD, 'Education_Master' whether the customer had Graduation, PhD or Master diploma, which are one-hot encoded values for Education, then 'Marital_Status_Married', 'Marital_Status_Together', 'Marital_Status_Single', 'Marital_Status_Divorced' features, whether the customer was Married, Together, Single or Divorced which are also one-hot encoded values for Marital Status. The next important features are

'Max_Val_Number_Purchases' containing the maximum value of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm's web page in the past month, 'Recency', which is the amount of days from the previous client's purchase, 'Teenhome', which is the amount of teenagers in client's home, 'Dt_Customer', which is the client's registration date in the firm, 'NumCatalogPurchases', which represents amount of deals conducted through the catalogue, 'MntMeatProducts', which is the money spent on meat in the past 2 years, 'Std_Val_Number_Purchases' containing the standard deviation of amount of deals conducted with deduction, amount of deals conducted through the catalogue, amount of deals conducted in shops and amount of deals conducted via firm's web page in the past month, 'Total_Kids' which is the sum of 'Kidhome' and 'Teenhome' features and 'NumDealsPurchases' which is amount of deals conducted with deduction, 'Std_Val_Spending_Products' containing the standard deviation of amount spend on MeatProducts, FishProducts, WineProducts, GoldProducts, FruitProducts, SweetProducts features and so on to the least important features.

Research Question 2: Which type of the selected classification algorithms provides the most precise prediction of customer's response to the marketing campaign based on personalized customer experience?

Averages	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC Score**
Naive Bayes	0.7612	0.3338	0.5992	0.4280	0.744660406
SVM	0.8596	0.5336	0.4942	0.5098	0.823741445
MLP	0.8690	0.5688	0.5178	0.5412	0.877537268
Logistic Regression	**0.8804**	**0.6214**	**0.5302**	**0.5656**	**0.888798712**
Random Forest	0.8568	0.5254	0.5242	0.5230	0.859594791

Table 3.1: Evaluation metrics comparison

As we can notice from Evaluation metrics comparison Table 3.1, can be concluded that Logistic Regression has highest average Accuracy 0.8804, highest average Precision 0.6214, highest average F1 score 0.5656 and highest average ROC AUC Score 0.888798712. This means that the Logistic Regression classification algorithm provides the most precise prediction of customer's response to the marketing campaign based on personalized customer experience.

CHAPTER 4

Contribution

In this chapter is explained the contribution, the progress made compared to the State of the Art. As State of the Art, we used the most advanced machine learning algorithms for building the classification model of customer's response to the marketing campaign based on personalized customer experience. After the results were obtained from the cutting-edge classifiers, as contribution made compared to the State of the Art, we tried to improve the performance evaluation metrics of the selected classification algorithms, by creating an ensemble of the classifiers that combines their individual predictions [DŽ04].

Stacking Classifier[1] is a stack of estimators with a final classifier. Stacking allows to use the strength of each individual estimator by using their output as input of a final estimator. In our implementation of Stacking Classifier, the stack of estimators contains the base learners' algorithms: Naïve Bayes, Support Vector Machine, Random Forest, Logistic Regression and Multi-layer Perceptron. As final estimator is used Logistic Regression classifier which will be used to combine the base estimators. On Figure 4.1 is given a code snippet of Stacking Classifier implementation as well as the code for generation of confusion matrices and ROC curves for Stacking classifier.

```
base_learners = [
             ('NB', gnb),
             ('SVC', off_svc),
             ('RF', off_rf),
             ('LG',off_log_reg),
             ('MLP',off_MLP)
            ]

# Initialize Stacking Classifier with the Meta Learner
clf = StackingClassifier(estimators=base_learners, final_estimator=LogisticRegression())
clf = clf.fit(XX_train_transformed,y_oversampled)
print(clf)

yy_pred = clf.predict_proba(XX_test_transformed)[:, 1]
yy_pred_binary=clf.predict(XX_test_transformed)

score=roc_auc_score(yy_test, yy_pred)
print('Stacking Classifier ROC AUC score:',score)
scores_Stacked.append(score)

fpr, tpr, _ = roc_curve(yy_test, yy_pred)
roc_auc = auc(fpr, tpr)
title_roc=str(str(i)+" Fold of ROC Stacking Classifier")
plot_roc_auc_curve(fpr, tpr,roc_auc,title_roc)

cf_matrix = confusion_matrix(yy_test, yy_pred_binary)
title=str(str(i)+" Fold of Stacking Classifier")
make_confusion_matrix(cf_matrix,
                      group_names=labels,
                      categories=categories,
                      title=title)
```

Figure 4.1: Stacking Classifier implementation

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html

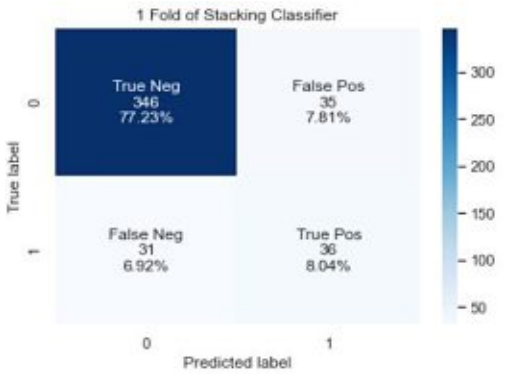


Figure 4.2: 1st Fold Stacking Classifier confusion matrix

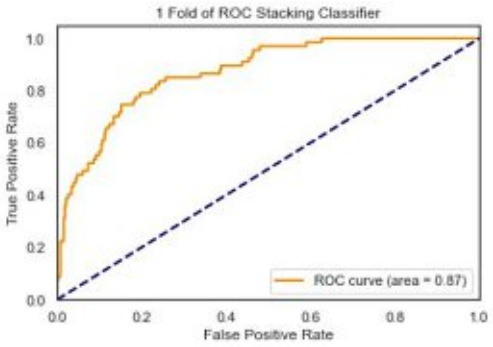


Figure 4.3: 1st Fold Stacking Classifier ROC curve

Figure 4.4: 2nd Fold Stacking Classifier confusion matrix

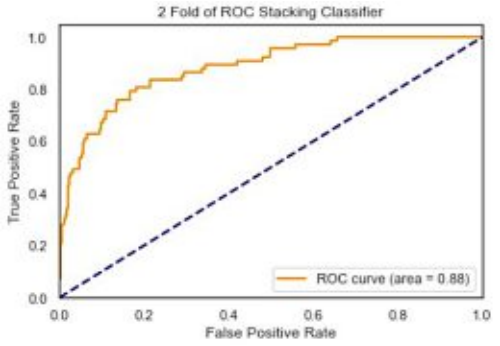


Figure 4.5: 2nd Fold Stacking Classifier ROC curve

Figure 4.6: 3rd Fold Stacking Classifier confusion matrix

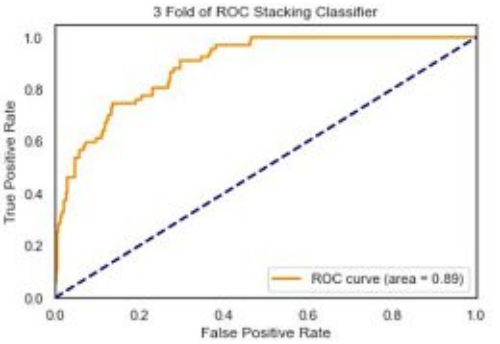


Figure 4.7: 3rd Fold Stacking Classifier ROC curve

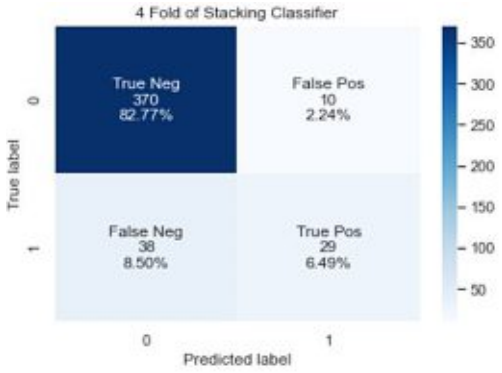


Figure 4.8: 4th Fold Stacking Classifier confusion matrix

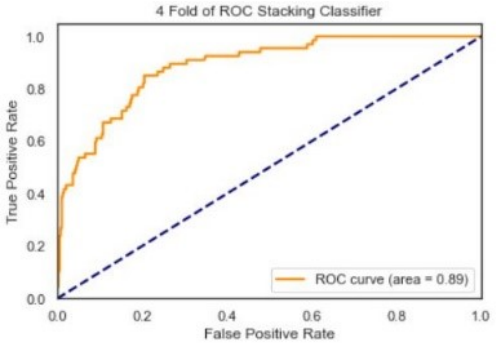


Figure 4.9: 4th Fold Stacking Classifier ROC curve

Figure 4.10: 5th Fold Stacking Classifier confusion matrix

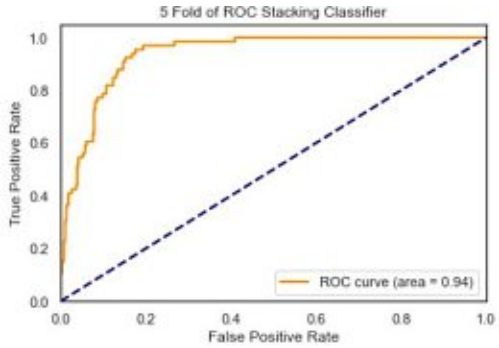


Figure 4.11: 5th Fold Stacking Classifier ROC curve

On Figures 4.2, 4.4, 4.6, 4.8, 4.10 are given the confusion matrices for the 5 Folds for Stacking Classifier. From these figures, it can be concluded that the highest number of True Negatives (370) is in the 4th Fold, the highest number of True Positives (40) is in the 5th Fold , the highest number of False Positives (35) is in the 1st Fold and the highest number of False Negatives (38) is in the 4th Fold.

On Figures 4.3, 4.5, 4.7, 4.9, 4.11 are given the ROC curves for the 5 Folds for Stacking Classfier. From these figures, it can be concluded that the area under the ROC curve was highest in the the 5th Fold which is area = 0.94.

As a next step was generated a Stacking Classifier evaluation metrics Table 4.1 that contains every evaluation metric value for each of the 5 folds, including the average of all 5 folds for each evaluation metric.

Fold	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC Score**
1	0.853	0.507	0.537	0.522	0.871214792
2	0.888	0.635	0.597	0.615	0.883110762
3	0.877	0.588	0.597	0.593	0.89141791
4	0.893	0.744	0.433	0.547	0.886783189
5	0.886	0.615	0.606	0.611	0.937226597
Average	**0.8794**	**0.6178**	**0.5440**	**0.5776**	**0.89395065**

Table 4.1: Stacking Classifier evaluation metrics

From the Table 4.1 can be concluded that Stacking Classifier classifier has highest Accuracy 0.893 in the 4th Fold, highest Precision 0.744 in the 4th Fold, highest Recall 0.606 in the 5th Fold, highest F1 Score 0.615 in the 2nd Fold and highest ROC AUC Score 0.937226597 in the 5th Fold.

Then, a new Evaluation metrics comparison Table 4.2 that contains the averages of each evaluation metric of the 5 selected classification algorithms, including the Stacking classifier was generated.

Averages	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC Score**
Naive Bayes	0.7612	0.3338	0.5992	0.4280	0.744660406
SVM	0.8596	0.5336	0.4942	0.5098	0.823741445
MLP	0.8690	0.5688	0.5178	0.5412	0.877537268
Logistic Regression	0.8804	0.6214	0.5302	0.5656	0.888798712
Random Forest	0.8568	0.5254	0.5242	0.5230	0.859594791
Stacking Classifier	**0.8794**	**0.6178**	**0.5540**	**0.5776**	**0.89395065**

Table 4.2: Evaluation metrics comparison including Stacking Classifier

From the Table 4.2 Evaluation metrics comparison including Stacking Classifier can be concluded that Stacking Classifier has highest average ROC AUC score 0.89395065 and highest average F1 Score 0.5776 among all classification algorithms that were implemented.

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 1

AUC Area Under Curve. 5

CRISP-DM Cross-Industry Standard Process for Data Mining. 3

IQR Interquartile Range. 15

KNN K Nearest Neighbors. 13

MAR Missing At Random. 11

MCAR Missing Completely At Random. 11

MICE Multiple Imputation by Chained Equations. 12

MLP Multi-layer Perceptron. 32

MNAR Missing Not At Random. 11

ROC Receiver Operating Characteristic. 5, 25

ROI Return On Investment. 1

SEO Search Engine Optimization. 2

SMOTE Synthetic Minority Oversampling Technique. 23

SVC Support Vector Classification. 29

SVM Support Vector Machine. 29

Bibliography

[AFJ17] Justice Asare-Frempong and Manoj Jayabalan. Predicting customer response to bank direct telemarketing campaign. In *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*, pages 1–4. IEEE, 2017.

[AOOM18] TR Ayodele, ASO Ogunjuyigbe, O Odigie, and JL Munda. A multi-criteria gis based model for wind farm site selection using interval type-2 fuzzy analytic hierarchy process: The case study of nigeria. *Applied Energy*, 228:1853–1869, 2018.

[Bal15] Loredana Patrutiu Baltes. Content marketing-the fundamental tool of digital marketing. *Bulletin of the Transilvania University of Brasov. Economic Sciences. Series V*, 8(2):111, 2015.

[BE15] T Femina Bahari and M Sudheep Elayidom. An efficient crm-data mining framework for the prediction of customer behaviour. *Procedia computer science*, 46:725–731, 2015.

[BS16] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

[CCK+00] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, Rudiger Wirth, et al. Crisp-dm 1.0: Step-by-step data mining guide. *SPSS inc*, 9:13, 2000.

[DGGB20] Thomas Davenport, Abhijit Guha, Dhruv Grewal, and Timna Bressgott. How artificial intelligence will change the future of marketing. *Journal of the Academy of Marketing Science*, 48(1):24–42, 2020.

[Dir15] Cüneyt Dirican. The impacts of robotics, artificial intelligence on business and economics. *Procedia-Social and Behavioral Sciences*, 195:564–573, 2015.

[DVDHSM06] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[DŽ04] Saso Džeroski and Bernard Ženko. Is combining classifiers with stacking better than selecting the best one? *Machine learning*, 54(3):255–273, 2004.

[Efr13] Bradley Efron. Bayes' theorem in the 21st century. *Science*, 340(6137):1177–1178, 2013.

[Els14] Hany A Elsalamony. Bank direct marketing analysis of data mining techniques. *International Journal of Computer Applications*, 85(7):12–22, 2014.

[FHI89] Michael Frigge, David C Hoaglin, and Boris Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.

[GYD+08] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guangtong Zhou. On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE, 2008.

[HS15] Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

[KS13] Gopalan Kesavaraj and Sreekumar Sukumaran. A study on classification techniques in data mining. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*, pages 1–7. IEEE, 2013.

[KS16] Utku Kose and Selcuk Sert. Intelligent content marketing with artificial intelligence. *Scientific Cooperation for the Future in the Social Sciences*, 2016.

[LD13] Rushi Longadge and Snehalata Dongre. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.

[Nob06] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[Nor05] Leonardo Noriega. Multilayer perceptron tutorial. *School of Computing. Staffordshire University*, 2005.

[NSK+17] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B Khalil, and Deepak S Turaga. Learning feature engineering for classification. In *Ijcai*, pages 2529–2535, 2017.

[R+01] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[RS04] Jennifer Rowley and Frances Slack. Conducting a literature review. *Management research news*, 2004.

[SOA04] Neil C Schwertman, Margaret Ann Owens, and Robiah Adnan. A simple more general boxplot method for identifying outliers. *Computational statistics & data analysis*, 47(1):165–174, 2004.

[SSM+16] Matthew Sadiku, Adebowale E Shadare, Sarhan M Musa, Cajetan M Akujuobi, and Roy Perry. Data visualization. *International Journal of Engineering Research And Advanced Technology (IJERAT)*, 2(12):11–16, 2016.

[SSSK19] Neha Soni, Enakshi Khular Sharma, Narotam Singh, and Amita Kapoor. Impact of artificial intelligence on businesses: from research, innovation, market deployment to future shifts in business models. *arXiv preprint arXiv:1905.02092*, 2019.

[SU07] Andrew I Schein and Lyle H Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.

[Thi18] T Thiraviyam. Artificial intelligence marketing, 2018.

[WH00] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1. Springer-Verlag London, UK, 2000.

[Xi08] Jingke Xi. Outlier detection algorithms in data mining. In *2008 Second International Symposium on Intelligent Information Technology Application*, volume 1, pages 94–97. IEEE, 2008.

[XLQ] Covers XGBoost, Spark NLP LightGBM, and Butch Quinto. Next-generation machine learning with spark.