St. Petersburg University

Graduate School of Management

Master in Business Analytics and Big Data

# CREATION OF A CHURN MODEL FOR THE COMPANY AND NEW PROCESSES FOR CHURN REDUCTION

Consulting project for TELE2

Master's Thesis by the 2$^{nd}$ year students

Concentration – Management:

Alexander Marinskiy
Maksim Solonin

Research advisor: Dmitrii V. Kudryavtsev, Associate Professor, Information Technologies in Management Department

St. Petersburg

2020

## STATEMENT ABOUT THE INDEPENDENT CHARACTER OF THE MASTER THESIS

We, Alexander Marinskiy and Maksim Solonin the 2nd year master students, program «Management», state that our Master Thesis does not contain any elements of plagiarism.

All direct borrowings from printed and electronic sources, as well as from master theses, PhD and doctorate theses, which were defended earlier, have appropriate references.

We are aware that according to paragraph 51 of Charter of the Federal State Institution of Higher Education Saint-Petersburg State University «a student can be expelled from St. Petersburg University for submitting of the Master Thesis, course or graduation qualification work developed by other person (people)».

10.05.2020

# ABSTRACT

| | |
|---|---|
| Master Student's Name | Alexander Marinskiy<br>Maksim Solonin |
| Master Thesis Title | Creation of a Churn model for the company and new processes for Churn reduction |
| Educational Program | Master in Business Analytics and Big Data - MiBA |
| Main field of study | Management |
| Year | 2020 |
| Academic Advisor's Name | Dmitrii V. Kudryavtsev |
| Description of the goal, tasks and main results | Creating customer churn models is one of the most pressing data science challenges for the telecom industry in Russia nowadays. Like companies in many other industries around the world, telecom operators in Russia have faced shrinking consumer demand and fierce competition in 2020. Identifying users who are going to change their mobile operator in time allows the company to start activities aimed at retaining such customers in advance and thus keep revenue. The goal of the current project is to create a churn model for TELE2 company and suggest processes for churn reduction. As a result of the project, a machine learning model was created that allows the company to identify the most vulnerable clients, who are about to abandon the services of the operator. What is more, several economic implications were proposed in this paper, which can be used to estimate the costs of retention campaigns and potential profit gains. Finally, the flowchart of the process for churn reduction was created, which highlights the key areas of improvement for the company and then, particular steps to improve the process were suggested. |
| Keywords | churn prediction, user behavior, telecommunications industry, machine learning, deep learning |

# АННОТАЦИЯ

| | |
|---|---|
| Автор | Александр Марьинский<br>Максим Солонин |
| Название ВКР | Создание модели оттока в компании и формирование процессов по его снижению |
| Образовательная программа | «Бизнес-аналитика и большие данные (Master in Business Analytics and Big Data - MiBA)»<br>(шифр ВМ.5783.*) |
| Направление подготовки | 38.04.02 «Менеджмент» |
| Год | 2020 |
| Научный руководитель | Кудрявцев Дмитрий Вячеславович |
| Описание цели, задач и основных результатов | Создание модели оттока клиентов на сегодняшний день является одной из наиболее актуальных задач в области дата саенс для телекоммуникационной отрасли в России. Как и компании во многих других отраслях по всему миру, мобильные операторы в России столкнулись с сокращением потребительского спроса и ужесточающейся конкуренцией в 2020 году. Своевременное выявление пользователей, которые собираются сменить мобильного оператора, позволяет компании заранее начать деятельность, направленную на удержание таких клиентов, и, таким образом, сохранить выручку. Целью данного проекта является создание модели оттока клиентов для компании TELE2 и предложение процессов по его снижению. В результате проекта была создана модель машинного обучения, которая позволит компании выявлять наиболее уязвимых клиентов, которые собираются отказаться от услуг оператора. Более того, в работе было предложено несколько экономических последствий, которые можно использовать для оценки затрат на кампании по удержанию и потенциального увеличения прибыли. Наконец, была создана блок-схема процесса сокращения оттока, в которой выделены ключевые области улучшения для компании, а затем были предложены конкретные шаги по улучшению процесса. |
| Ключевые слова | предсказание оттока пользователей, поведение пользователей, телекоммуникационная индустрия, машинное обучение, глубокое обучение |

# TABLE OF CONTENTS

# INTRODUCTION

The world is developing at very high speed and competition is arising everywhere, especially in technological spheres, where knowledge spillovers are very fast. Telecommunication services is one of the industries with fierce competition and where the bargaining power of customers is high due to their low switching costs between mobile operators. Russian market, which is of our interest, is one of the most technologically advanced and has vast potential for development.

To increase revenue and profit, mobile operators need to increase their user base. There are two ways to achieve this. Companies can attract new users or they can reduce the outflow of existing users. It is true for both the international and Russian telecom market that retaining one user is cheaper than attracting one new user. Several studies have demonstrated that attraction of a new customer is usually around six times more costly than retaining an existing one. (Colgate & Danaher, 2000) However, user retention measures are also quite expensive and the company cannot carry out user retention measures for all customers.

Apart from the aforementioned facts, the relevance of the paper is justified by the volume of the market. The Russian mobile telecom market is saturated and approaches 200%, meaning that almost every person has 2 SIM-cards. Therefore, as the time goes, it is getting harder to attract new customers and these new customers are often the most price-sensitive ones from other companies.

Creating customer churn models is one of the most pressing data science challenges for the industry. Identifying users who are going to change their mobile operator in time allows the company to start activities aimed at retaining such customers in advance. This paper focuses on creation of a churn prediction model and giving recommendations for churn reduction using the data provided by Russian mobile operator TELE2.

TELE2 is one of the most popular telecom operators in Russia. It was established in 2003 and now operates in 65 regions of Russian Federation, providing services in high-speed internet connection of 3G/4G standard. The main track of a company's development is creation of products with the best available price-quality ratio. It has significant success among clients and has become the fastest growing operator by average year revenue from services. To sum up, the current approach allowed the company to not only attract the price-sensitive customer segment, but also capture the most growing segment of digital clients, who use the data (internet access).

There are two important points in building customer retention processes in the company. Firstly, retention of the customer does not happen instantly, which means that company needs to start retention measures before the user decides to change mobile operator. The second important point is that some users bring more profit to the company than others, which means TELE2 needs to differentiate customer retention measures and spend more resources on retention of customers, which are important for the company. The object of research in this paper is the actual churn of the customers in telecom mobile operators. The subjects of this paper are the created customer churn prediction models.

The remainder of the paper proceeds as follows. Firstly, churn definition will be stated according to both the industry and TELE2 data. Then, most popular international and Russian practices for churn prediction will be reviewed. After that, most common problems with modelling will be regarded and solutions will be discussed. In the empirical part, preliminary data analysis will be performed, results of different models will be presented and compared. Final section gives conclusions, ideas for further research development and discusses research limitations.

The goal of the project is to create a churn model for the company and suggest processes for churn reduction. Considering the project objectives, they are the following:

1. Define churn.
2. Choose type of the model (regression, binary classification, multiclass classification)
3. Determine how much time is required for customer retention measures and based on this determine the prediction horizon for the model.
4. Choose a quality metric.
5. Build a model on historical data.
6. Suggest an experiment design for testing models on real users.

# CHAPTER 1. CUSTOMER CHURN ROLE IN TELECOMMUNICATION MARKET AND TOOLS TO HANDLE IT

Before moving to the analysis of TELE2 dataset, creating a churn model and formulating any recommendations for churn reduction, it is needed to analyse the state of the art in the field. This section of the article defines churn both in terms of the industry and our data, analyses most popular approaches to the problem in international practices and specificities of the Russian telecom market, and, finally, reviews general shortcomings in the modelling process and provides options to mitigate them.

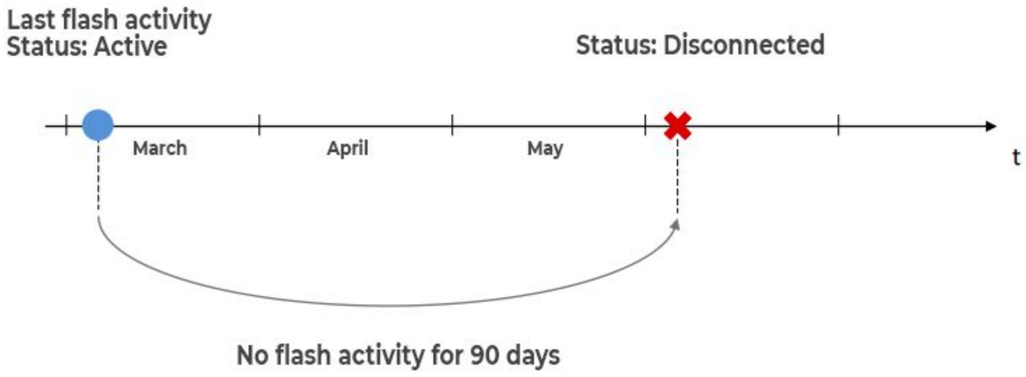
## 1.1. Definition of churn

Churn is a complicated process, which involves analysing all customer characteristics and behaviour, account information, call details, satisfaction, while also gathering information about competitors' offerings.

Berson et al. (2000), defined churn as the movement of existing customers from one service provider to another. Regarding the nature of churn and people's behaviour, it is intuitively clear that churn is a low-chance event in many industries, including telecommunications. Zhu et al. (2017) claim that average monthly churn rate in telecom is approximately 2%. Therefore, it can be stated that churn prediction is a binary classification task with sample imbalance in most cases. Another domain of churn, which is closely related to its definition is time definition, because despite clear markers of churn – switch to another operator, time after which the client considered to be churned is important.

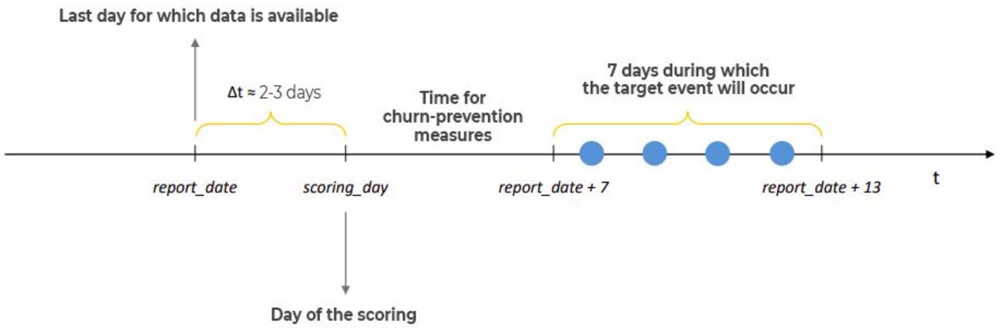
According to TELE2, the target event (churn) is defined as an absence of client's flash activity for 90 days (3 months). Thus, our goal is to identify key factors of churn and create a model, which will predict the probability of a client to perform his last flash activity and assist in decision-making about clients' retention.

The company needs 5 days to conduct customer retention measures. It is clear that such measures should be carried out before the client ceased to use the services of a mobile operator. Thus, the churn model goal is to determine the likelihood that the user will make his last flash activity in the next 7-14 days.



*Pic 1. Churn Definition*

Thus, the chronology of customer retention measures will be following. Suppose on March 3 we are scoring all clients (scoring_day). At this moment, we have data available as on March 1 (report_day). Then our model should predict the probability that the client will make the last flash activity in the period from March 8 to March 14. Now the company has time from March 3 to March 8 to try to contact the user, offer him/her a discount, special offer and so on.



*Pic 2. Churn Timeline*

## 1.2. Telecommunications market overview

This section provides an overview of the telecom market in the current situation and demonstrates an impact of coronavirus on the operations of telecom operators and customers' churn. Then, this section shows the Russian telecom market statistics and Tele2 company in particular. Finally, this section gives the crucial steps, which should be taken in order to cope with churn and retain as many customers as possible.

Global telecommunications sector is a rapidly developing industry with many innovations and new technologies and infrastructure introduced. According to BuddeComm Intelligence Report (Wansink, 2020), there are approximately 7.7 billion active mobile subscriptions in 2020, which is a huge growth since 2015's 3.3 billion. Such growth is mainly justified by introduction of new technologies such as 4G LTE, therefore, continuing focus on quality and speed of network (5G) is expected. Considering the revenue from the mobile telecommunications industry, it is also increasing every year and in 2019 it was slightly less than 1.1 trillion dollars.

As any other industry, telecommunications industry will be heavily influenced by the current coronavirus pandemic. According to Omdia agency's prediction, the revenue of mobile telecommunications industry in 2020 will be 4.1% less worldwide and 9.1% in Europe. The same trend is true for net profits. The main and most obvious reasons for such behaviour is the reduction of clients' costs and disappearance of roaming income due to absence of travel and tourism. Based on these facts, it can be claimed that in global terms the most vulnerable group of mobile operators is in Europe, which is the main tourist destination and, thus, was receiving a lot of income from roaming and SIM-cards selling. However, compared to other industries mobile operators will suffer less than the others. Distant communication is at its peak and operators are gaining more and more clients, while their marketing spending was reduced. At the same time, market shares are rather stable and operators can focus on network development (5G, etc.).

Regarding churn as the main topic of our project, it is also interesting to see how it was influenced by coronavirus. According to Jefferies consulting agency, it turned out that attrition has drastically dropped in sampled countries: France (63%), Italy(59%) and Belgium (55%). The potential effect of pandemic on churn in the long-term is the increase of interval from clients desire to leave to his actual change of the operator. It can significantly change the dynamic of attrition and, consequently, approach to its prevention. In Russia this type of attrition is called portation (transfer of number from one operator to another) and will reduce by 80%. However, the ordinary churn will increase by 7-8%, but only for the 2 quarters (AC&M Consulting, 2019). It is quite reasonable, because change of operator in these circumstances is not quite beneficial, while reduction of costs by cancelling the subscription is possible.

Now, we will provide more details about the Russian telecommunications market. The table below based on the data from AC&M Consulting demonstrates continuing growth of customers base for key Russian mobile operators. In 2019 the penetration of mobile network services in Russia increased from 174 to 178%. It is clear that Tele2 is the most rapidly increasing in size. It is responsible for 77% of growth, while MTS has a share of 19% of net growth.

| Key mobile operators | 2018 | 2019 |
|---|---|---|
| **MTS** | **78 010 000** | **79 100 000** |
| **VimpelCom** | **55 252 584** | **54 648 951** |
| **Megafon** | **75 200 000** | **75 200 000** |
| **Tele2** | **44 050 000** | **48 350 000** |

*Table 1: Customer base of TOP-4 Russian mobile operators (AC&M Consulting, 2020)*

According to the vice-president of MTS Inessa Galaktionova, yearly churn for mobile operators is on average 40%, while the package deals churn, such as mobile + TV is 2.5 times smaller. The bar + line chart below demonstrates us the dynamics of churn and attraction of new customers by selling SIM-cards. It can be seen that churn rate diminishes along with SIM-card sales (however, the latter is still rather stable and increases sometimes).

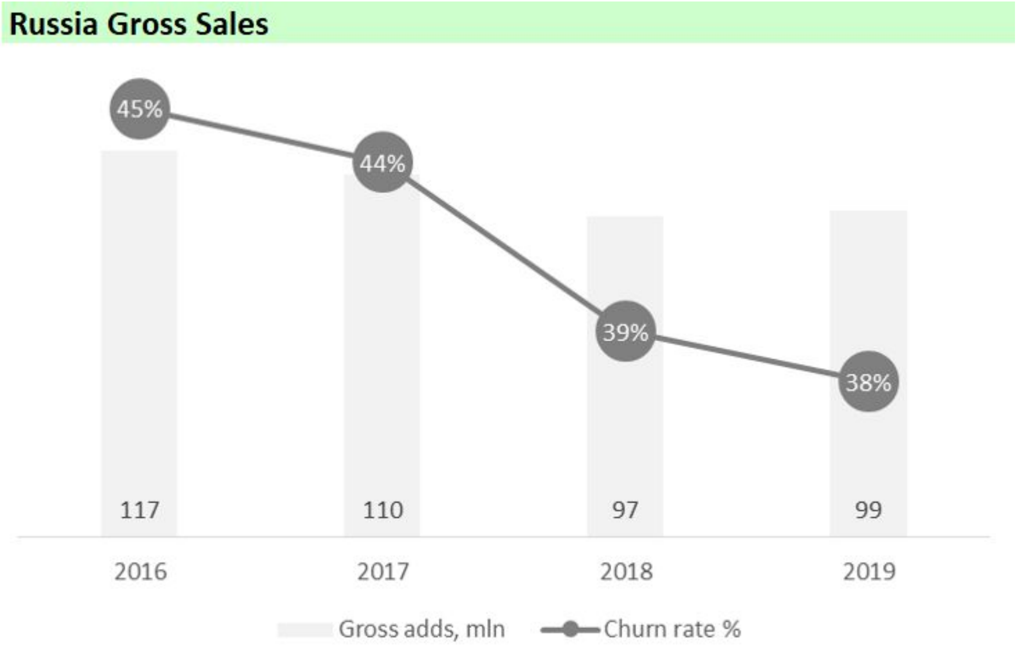


*Figure 3: SIM-card Sales in Russia (AC&M Consulting, 2019)*

Analysing key Russian mobile operators, we have come to the conclusion that MTS still maintains its leading position in revenue and clients base. Its revenue is 472.6 billion RUR (320 billion from mobile network service), while net profit is 99 billion RUR. According to table above, MTS has the largest customer base of 79.1 million people. Average revenue per user (ARPU) is equal to 344 RUR. Reuters states that MTS plans to decrease churn rate three times by means of fintech services (MTS-Bank) and paid video content development. (Tsydenova, 2019) Company is aimed at reaching the level of 10 million users for these branches of services. What is more, Russian leading mobile operator plans to reach ARPU almost twice as large from today in 2023, which will be equal to 600 RUR.

Another mobile operator from Russian BIG3 is Megafon placed second. It earned 343.4 billion RUR (274.9 billion – mobile network service) in 2019 with net profit of 13.5 billion. It has 75.2 millions of clients, which has not changed since 2018. ARPU of Megafon is smaller than same figure of MTS and is equal to 307 RUR.

Final member of telecom BIG3 is VimpelCom (Beeline brand). This mobile operator has a revenue of 289.9 billion RUR (225.5 billion – mobile network service) for 2019 and net profit equal to 17.18 billion RUR. Beeline is currently on the downturn in terms of number of clients. It experienced a decrease from 55.2 million users to 54.6 million. It is the only case of decrease among 4 key Russian mobile operators. However, it demonstrates 2nd largest ARPU among all 4 companies – 335 RUR.

Finally, we will describe the most promising mobile operator in Russia that demonstrates biggest growing trends – Tele2. In 2019 Tele2 had revenue of 163 billion RUR, which is the smallest value of all the 4 mentioned mobile operators. Tele2's net profit was equal to 6.6 billion RUR, which is approximately 2.5 times (145%) bigger than net profit of 2018. The customer base increased significantly from 44 million people to 48.35 million, which is larger than the growth of all the BIG3 operators. ARPU increased by 8.3% and is equal to 305 RUR. At the same time yearly churn rate decreased by 2.6 p.p. and now equal to 34.8%, which is lower than average 38%. Despite stagnating telecom market, Tele2 with appropriate churn reduction strategies, price-quality ratio goals and innovations-oriented approach can continue to increase its market share. Another recent operation, which will help in increasing the market share is the acquisition of Tele2 by Rostelecom. The company will keep its own brand and consolidated retail business will be managed by Tele2 because its retail sector is larger than that of Rostelecom.

To sum up, it is needed to identify what are the key factors of success in churn rate reduction by the means of analysis. McKinsey published a good report on churn reduction and highlighted four best practices in churn reduction task. (Jain & Surana, 2017)

The first recommended practice is to establish a general view of the client and link it to the result. It can be claimed that this advice is about using the 3Vs of our century from Harvard Business Review (Davenport, 2012) – volume, velocity and variety, meaning that by timely joining data of different types and from different sources, telecom mobile operator will be able to know the client better than he does and can follow him for an entire customer journey and predict his decisions and act accordingly. This data can include customer socio-demographic information, web logs, complaints, offers and promotions usage, calls aggregated data, etc.

Second advice is quite a basic one and is used by almost all the companies, thanks to the data science popularity. It suggests using cutting-edge analytical techniques. It is clear that this factor implication depends on the people hired and the company's policy.

Third decisive factor is the division of the customer base into scores of microsegments. It suggests giving much focus to personalisation of treatment and identification of clients with highest probability to leave. This is a standard approach for advanced algorithms, so it is closely related to the second mentioned practice. McKinsey provides an example of developing a library of over 50 offers and then setting up a mechanism for launching and estimating campaigns. As a result, churn rate was reduced by 15% over the next 3 years.

Final idea is about the introduction of agile test-and-learn processes. It is clear that, nowadays, everything is about the speed of applying your knowledge and guesses. Therefore, when a lot of data was gathered really fast, an advanced tool was applied and the needed microsegment was identified (first 3 practices mentioned), it is needed to test and understand hypotheses about different offers, promotions, etc. Thus, dividing tasks into short iterations with adjustments using agile methods is the key to success in churn management (particularly reduction).

The next subsection will be focused on the second and third practices of advanced tools and segmentation. It will review the main methods, which are used in the field to create churn models.

## 1.3. Literature review

### 1.3.1. Predictive modelling approaches and results

Precise and reliable prediction of customer churn is crucial in the development of appropriate retention strategies. The data analysis techniques and tools that help in gaining valuable insights and hidden patterns play an important role in the decision-making process. Data mining tools nowadays are used for all the tasks, where data is involved and its volume is large enough to have advantage. The most common basic approach to data mining tasks in any industry is a cross-industry standard process for data mining (CRISP-DM).

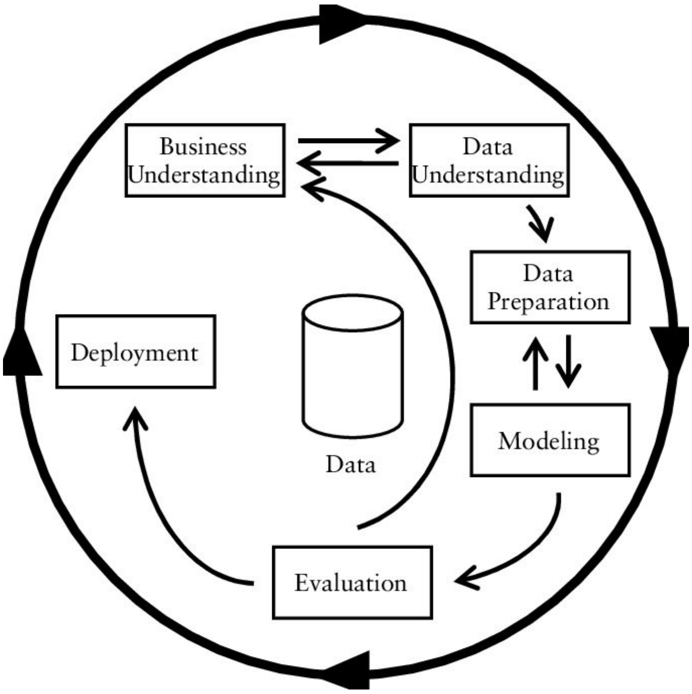


*Figure 4: CRISP-DM Framework*

The relatively slow growth of telecom is currently experienced, which is absolutely natural and inevitable for the market, developed by saturation. Russian mobile market, which is of our interest, almost reached its maximum in terms of potential customer base and stopped in extensive growth. The global market is still developing due to a slight increase in the subscriber base, mainly in developing countries with technological breakthroughs and large population – India and China. Hence, the most straightforward way for the company to stay in the market and improve its performance is to retain as many customers as they can. There are many different types of churn, mostly split by reasoning, meaning that customers attrite for different reasons. Churn can be active, rotational or nonvoluntary. People may change their job, where their company paid for their mobile services, they can move to another city, where the current mobile operator is not covering or does not have beneficial propositions for customers. Another reason is the most common one – dissatisfaction with provided services. The signal might be bad in some areas, expenditures turned out to be larger than expected, etc. And the customer can not only notify about his decision by calling to the company, but also just stop using services without breaking the contract. Therefore, machine learning tools are applied to solve the churn prediction task.

Regarding approaches to customer churn prediction, supervised machine learning (ML) techniques are the most popular ones. Supervised ML is a branch of modelling, where models are learning from labeled data. ML includes a wide range of algorithms such as decision trees, k-nearest neighbours, logistic regression, neural networks, support vector machines (SVM) and etc. In case of customer churn, it is obvious that it is a binary classification task with no churn / churn division. Such formulation of task suggests that most telecom providers use models, which predict customers with the highest probability to attrite. Analysing attrition of customers, it is needed to understand that some clients are very prone to churn, for instance, because they have prepaid contracts and are mostly anonymous. Therefore, churn prediction models are utilized in contractual settings and more focused on customers in the postpaid segment. Much more information for this type of clients is available to the company, such as socio-demographic parameters (clients profile), call behaviour statistics and patterns (average length of calls, number of calls to the helpdesk, etc.). What is more, the company provides incentives to integrate with social networks, apps and gather even more data about the client. Huge amounts of different types of data about clients are used as factors in the models for churn prediction.

As it was already mentioned, one of the most widespread reasons for changing the mobile operator is dissatisfaction with the services provided by the company. Rengarajan and Kavipriya (2012) assessed this level of customer satisfaction. According to their results, the greatest impact in division of customers is made by income and tenure. The authors also discovered that network quality has the biggest influence on satisfaction followed by personalised offers, general quality of service, chosen tariff and customer care services availability.

Another research, which analysed factors influencing customers satisfaction was conducted using socio-demographic parameters of client (gender, income, age), usage patterns, satisfaction level, etc. (Kamath, 2011) Kamath stated that the most crucial factor influencing satisfaction and, consequently, churn is network coverage. It is quite reasonable, because a wide network coverage area guarantees that customers will have access to the services. Another conclusion from this paper is the difference between pre-paid and post-paid clients' satisfaction. It claims that pre-paid customers are more satisfied with the service. Despite this result, it is clear that such clients have a shorter lifetime. For both reasons of high satisfaction and short lifetime, there is no reason to include them in the churn prediction model.

Russian reality has almost the same patterns in terms of clients satisfaction and its influence on churn. Astakhov and Rudakova (2018) concluded that the customer dissatisfaction is mostly affected by the result of using the Internet service, which is usually associated with access speed and volume of data available.

Factor choice for analysis in the model is very important, however, for predictive models the main goal is still to predict the probability of churn and recommend a group of customers for retention measures. Therefore, in this section it is more reasonable to focus on ML algorithms, which were used to predict churn, while also it is needed to spot the differences in factors analysed.

Most of the articles have the same structure and conduct a comparative analysis of different machine learning techniques and then, some of them propose a complex model, which demonstrates better results in a particular situation. For instance, Kim and Yoon (2004) compared linear regression, decision tree and artificial neural networks (ANNs) applying it to the data related to customer complaints. They had an artificially balanced dataset, where churn/no churn classes are equally presented. It is done to escape from the sample imbalance problem, which we will discuss later. The results demonstrated that ANN is a good algorithm for predicting potential churn clients. Another paper (Omar et al., 2014) used multilayer perceptron (MLP) in order to analyse the customer churn. The article focused on analysing billing information (fees, rates), intensity of usage (number of calls, SMS, their length, etc.), plans. It turned out that monthly fees had the largest influence on churn followed by number of minutes used. Therefore, we can state that it could be the best pressure lever (incentive) in the retention campaign.

Keramati et al. (2014) applied a bunch of ML techniques for churn prediction including ANN, KNN, SVM and decision trees. Among them, SVM and ANN performed the best with 0.92 and 0.90 recall respectively. The authors also created a hybrid model based on the aforementioned techniques' nuances, which demonstrated better results, i.e. recall and precision were above 0.95.

Huang et al. (2012) gathered a large set of factors for modelling including call details, customer profile, billing data, complaints information, etc. The authors tested 7 different methods and compared their performance. These techniques were decision tree (DT), MLP, SVM, logistic regression (LR), naïve Bayes (NB), linear classifications (LC) and evolutionary data mining algorithm (genetic classification) (DMEL). The authors arranged all these techniques according to their computational complexity. The lowest complexity is claimed for

NB and DT, followed by LC, LR and SVM. DMEL demonstrated poor predictive power, while also being very computationally expensive. MLP was also quite expensive but demonstrated decent results, however, not recommended for application on large datasets. One of the most important conclusions of this paper is that choice of modelling should be justified by objectives of the decision-makers, meaning that DT and SVM, for instance, should be used if interested in the true churn rate and false churn rate, while LR might be used if the goal is to identify the churn probability.

Qureshi et al. (2013) applied similar models to the churn prediction task, including logistic regression, ANN, K-Means, DT, Exhaustive CHAID (variation of DT), etc. Using a large dataset of 106,000 customers and numerous factors, the authors came to the conclusion that exhaustive CHAID was the best model according to F-measure.

Lu et al. (2012) studied telecommunication company data and proposed to utilize boosting in order to improve the churn prediction model. The novelty of this paper is that the authors use boosting to separate customers into two different clusters based on weights assigned by the algorithm.  Then logistic regression was applied to each cluster and it showed that boosting really enhanced the churn prediction compared to single logistic regression without cluster division.

Verbeke et al. (2012) and Zhu et al. (2017) developed and mentioned respectively maximum profit criterion, which was successfully used to evaluate and deploy customer churn prediction model. This criterion allowed to identify the optimal model more precisely compared to traditional performance measures. The experiment in the first paper demonstrated that optimization of customers fraction in retention campaign and application of maximum profit criterion allow to be more cost-efficient. The authors of the second article also found an interesting result that ensemble techniques turn out to be the most successful and default random forest and bagging algorithm without applying any tools to solve class imbalance problem get the best results according to the maximum profit criterion.

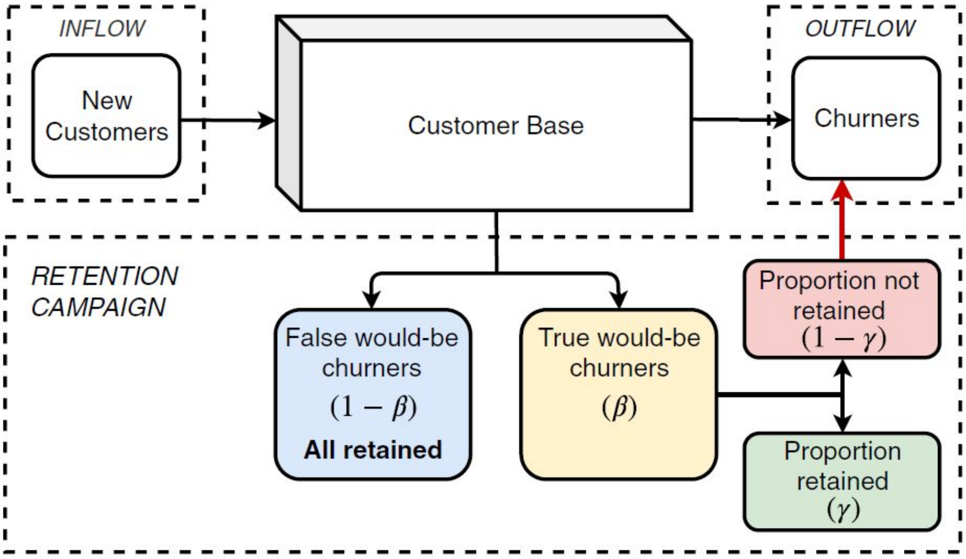
Umayaparvathi and Iyakutti (2012) analysed data of a Singaporean mobile operator and applied decision trees and neural networks in order to predict the churn of customers. The results of the paper state that the decision tree was better in terms of accuracy and was much easier to build. However, no retention policies were formulated in this article, we will consider this part of churn management processes later.

### 1.3.2. Business implication of the models

Creating churn prediction models in order to divide customers into different segments is crucial for churn management and choice of model method plays a significant role. Nevertheless, prediction itself does not give any result in terms of business process for churn reduction. Choosing a specific method with the best quality of prediction will only improve our retention campaign, however, it does not say anything about different options of retention campaigns. We suppose that discount offers as part of retention campaigns should be personalised and formulated according to customer profile characteristics. We cannot develop them on our data due to absence of heuristics (names of the variables). Considering the comparison of already developed retention campaigns, the most obvious way is to rank them according to profits gained. A couple of articles mentioned above used maximum profit criterion originally introduced by Neslin et al. (2006). Apart from comparison of different ML methods with this criterion and identifying the most cost-efficient, Neslin introduced the following profit formula:

$$\Pi = N\alpha[\ \beta\gamma(\ b - c_{contact} - c_{incentive}) + \beta(1-\gamma)(-\ c_{contact}) + (1-\beta)\ (-c_{contact} -\ c_{incentive})\ ] - A,\ where$$

- N*α - the number of customers targeted in the campaign (induce costs and bring benefits);
- A - fixed administrative costs, which reduce profits irrespective of model performance
- $\beta\gamma(\ b - c_{contact} - c_{incentive})$ - net profit achieved from the retention campaign by receiving the benefits from retained would-be churners predicted by our churn model ( $\beta$ - proportion of true would-be churners, $\gamma$ - retention rate, c - costs for contacting customer and proposing an offer (incentive to stay))
- $\beta(1-\gamma)(-\ c_{contact})$ costs related to correctly identified would-be churners who were not retained.
- $(1-\beta)\ (-c_{contact} -\ c_{incentive})$ - costs from targeting non-churners with the campaign



$$\Pi = N\alpha[\underbrace{\beta\gamma(b - c_{\text{contact}} - c_{\text{incentive}})}_{\text{Retained churners}} + \underbrace{\beta(1-\gamma)\,(-c_{\text{contact}})}_{\text{Non-retained churners}} + \underbrace{(1-\beta)\,(-c_{\text{contact}} - c_{\text{incentive}})}_{\text{False would-be churners}}] - A$$

*Figure 5: Visual representation of retention campaign profit formula*

*(Devriendt et al., 2019).*

According to Verbeke et al. (2012) retention rate and CLV (customer lifetime value = benefit) are assumed to be constant and independent of the number of clients targeted in the campaign. At the same time, Devriendt et al. (2019) suggest that these assumptions are too unrealistic in business environment and maximum profit should be modified to maximum profit uplift (MPU) with variating parameters. To define the MPU, the authors introduced liftup measure that is a function of targeted customers share. This measure emphasizes the uplift values at a certain cutoff as to the overall baseline uplift achieved when targeting all customers. According to the article's conclusions the liftup is shown to be directly related to profit, therefore, MPU measure could be an appropriate option to apply for ranking retention campaigns.

Ahmed and Maheswari (2018) analysed churn classifiers in telecom applying uplift modelling and stated that utilisation of customer uplift modelling methods resulted in almost 50% cost savings. An important detail is that the authors created a model that divides churn customers into three different segments - active, passive and rotational and concentrated their research only on active ones. It allowed them to reduce costs significantly and apply retention campaigns much more efficiently.

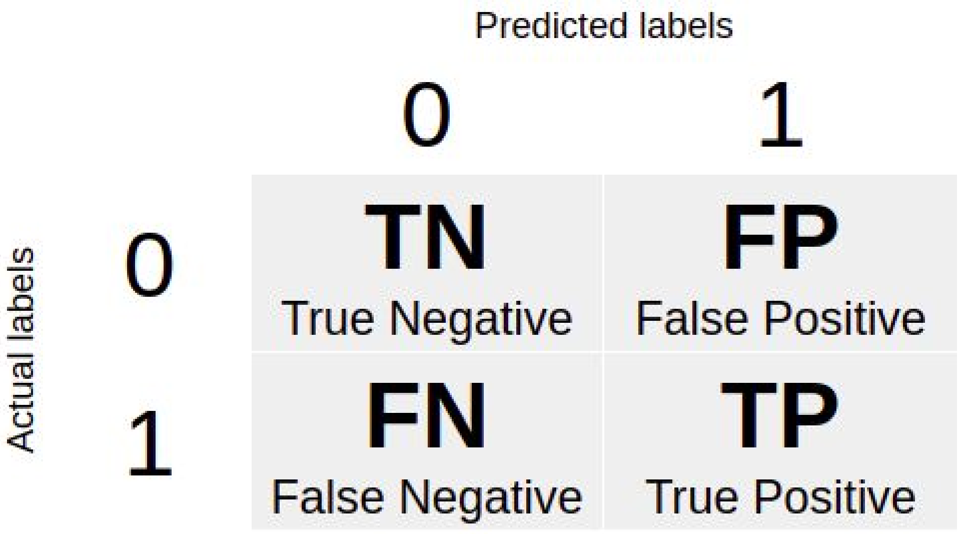
# CHAPTER 2. CHURN PREDICTION MODELLING AND IMPLICATION OF RESULTS

## 2.1. Metric of quality

In the empirical part of the work, we operate several quality metrics. In this section, we briefly describe the essence of these metrics, and also justify the choice of a quality metric for choosing the best option between models.

**Confusion matrix**

It is reasonable to start a conversation about metrics with the confusion matrix.



|  |  | Predicted labels |  |
|---|---|---|---|
|  |  | 0 | 1 |
| Actual labels | 0 | TN<br>True Negative | FP<br>False Positive |
|  | 1 | FN<br>False Negative | TP<br>True Positive |

*Figure 6: Confusion matrix.*

Confusion matrix is a matrix with two dimensions: actual labels and predicted labels. Typically, actual labels are in the rows of the matrix, and predicted labels are in the columns. In the cells of the matrix are:

1. True Negative (TN). These are elements of class “0” that were correctly predicted as “0”.
2. False Positive (FP). These are elements of class “0” that were incorrectly predicted as “1”.

3. False Negative (FN). These are elements of class “1” that were incorrectly predicted as “0”.

4. True Positive (TP). These are elements of class “1” that were correctly predicted as “1”.

**Accuracy, Precision, Recall, F1 score**

Now, based on the confusion matrix, we can introduce the following three metrics for the quality of machine learning models: accuracy, precision and recall.

**Accuracy** is the percentage of dataset elements that have been correctly classified. Accuracy can be calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

100% accuracy means that for all elements of the dataset, the model predicted the class label correctly, and 0% accuracy means that for no element of the dataset the class was predicted correctly.

For balanced datasets, accuracy is a good quality metric. However, in the case of unbalanced datasets, the influence of the majority class on this metric may be excessive. For example, consider a dataset in which 99% of the data belongs to the class “0”, and 1% of the data belongs to the class “1”. A model that predicts class “0” for all data will receive a 99% accuracy. At the same time, the value of such a model is close to none, since it does not distinguish between classes “1” and “0”. Therefore, we need to introduce two different metrics: precision and recall.

**Precision** is a percentage of elements of the dataset predicted positive that is classified correctly. It can be calculated using following formula:

$$Precision = \frac{TP}{TP + FP}$$

**Recall** is a percentage of elements with a positive label in the dataset that is classified correctly. It can be calculated using following formula:

$$Recall = \frac{TP}{TP + FN}$$

In our case precision answers the question: “Among all the elements that our model predicted as “churn”, what percentage actually churned?”. In turn, recall answers the question: “Among all churned customers, what percentage of our model was able to predict as “churn”?”

Unlike accuracy, precision and recall metrics allow us to evaluate the quality of the model on imbalanced datasets. However, these metrics have two significant problems.

Firstly, they depend on threshold value. Almost all models of binary classification allow determining the probability of an element belonging to class “1” or “0”. If the probability is greater than the threshold value, then the element labeled as class “1”, and if it is less, then it is labeled as class “0”. By default, the threshold value is set to 0.5. In other words, if the probability that an element belongs to the class “1” is greater than or equal to 50%, then we label the element as “1”, otherwise we label it “0”. However, a threshold value of 50% does not always correspond to the business goal of the model. The fact is that the cost of misclassification can significantly differ for classes “1” and “0”.

The second problem with using precision and recall metrics is that there are two of them. In a situation where one model has better precision metric and another has better recall we don’t understand which one should be chosen. In order to solve this problem, F1-score is sometimes used.

F1 score is a harmonic mean of precision and recall. It can be calculated using following formula:

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

The indisputable advantage of using F1 score over Precision and Recall is that now we have one metric instead of two and we can clearly understand that one model is better than another. However, an important problem with using the F1 metric is that it gives the same importance to the Precision and Recall metrics, which may not correspond to the business logic. Hand & Christen (2018) point out that “F-measure has a major conceptual weakness: the relative importance assigned to precision and recall should be an aspect of the problem and the researcher or user, but not of the particular linkage method being used.”

### AUC ROC and AUC PR

In order to overcome above mentioned drawbacks two different metrics are often used. These metrics are AUC ROC and AUC PR.

AUC ROC means area under ROC curve. ROC curve is receiver operating characteristic curve. It is a curve that plots true positive rate (TPR) against false positive rate (FPR) for all possible threshold values.

AUC PR means area under Precision-Recall curve. This curve Precision against Recall for all possible threshold values.

Both AUC ROC and AUC PR allow to compare the quality of the different models with a single metric without knowing the exact threshold value that is going to be used. However, for imbalance datasets AUC PR is usually preferred. Branco et al. (2016) recommend to use Precision-Recall curves instead of ROC curves for "for highly skewed domains where ROC curves may provide an excessively optimistic view of the performance."

Thus, we examined various metrics for comparing classifiers with each other: accuracy, precision, recall, F1 score, AUC ROC and AUC PR. Based on the results of the review, it was decided to use the AUC PR metric to compare classifiers. There are three key reasons for this decision. Firstly, to compare the classifiers, we need one metric, not several. Secondly, in creating a customer churn model we will deal with an unbalanced dataset (the "Non churn" class will be much larger than the "Churn" class). Thirdly, we need a metric independent of the threshold value, since the optimal threshold value will be determined later in accordance with business logic.

## 2.2. Exploratory data analysis

### Initial overview

The company provided us with a dataset with 280,000 records. Each record in the dataset is describing Tele2 customers. For the code of analysis see Appendix 1.

Dataset has columns "client_id", "report_date", "label" and 46 columns describing the characteristics of clients. These characteristics are named like "feat_1", "feat_2" and so on. The company refused to provide a description of the features. This is a common practice among companies that work with sensitive user data for example banks, telecom operators, online services. However, Tele2 assured us that none of the features are categorical, which is an important point for building the models later on.

Each entry in the dataset has a class label. Label “1” means that the client churned, label “0” means that the client remained with the company.

Each entry also has a “client_id” field which has the unique id of the client. All of the values in this field are unique, therefore we don't have records for one customer for different report dates.

Four reporting dates are presented in the dataset: October 1, 2018, November 1, 2018, December 1, 2018 and January 1, 2019. For each date, there are exactly 70,000 records. Out of these 70,000 records for each report date exactly 10,000 clients churned and exactly 60,000 clients did not churn. Therefore as predicted we are dealing with an imbalanced dataset and need to apply different techniques to eliminate sample imbalance.

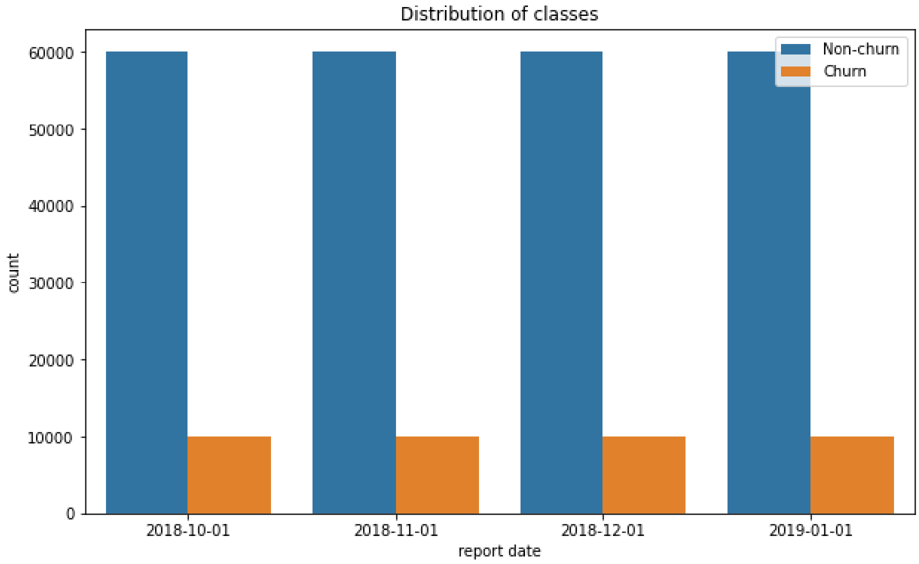


*Figure 7: Distribution of classes in initial dataset.*

## Missing values

In our dataset features 1-31 and 46 do not have missing values. Features 32-45 have some missing values. Moreover, features 32-38 all have exactly 40123 missing values. It’s important to point out that these features are missing simultaneously. This signals us that these features are somehow connected.

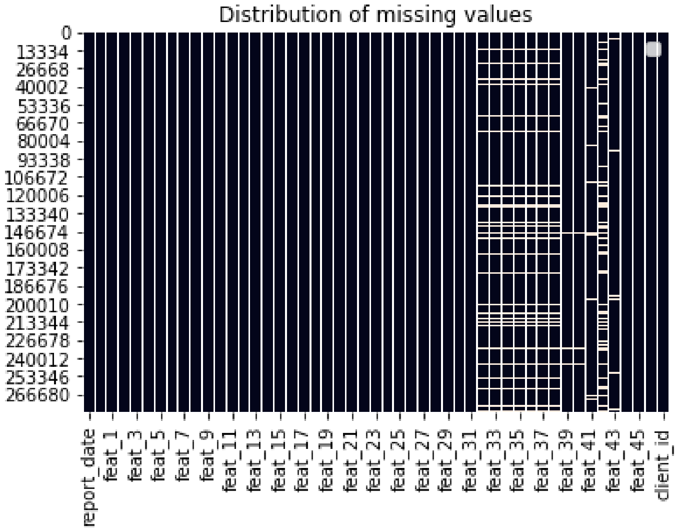


*Figure 8: Distribution of missing values in initial dataset.*

**Correlation**

Now let's check for correlation between features. It is important to check for it because some of the models are sensitive to feature collinearity. On the heatmap we can see that some of the features have strong positive correlation (e.g. 17 and 10, 23 and 9, 26 and 11), while others have strong negative correlation (e.g. 42 and 35, 42 and 36, 35 and 32).

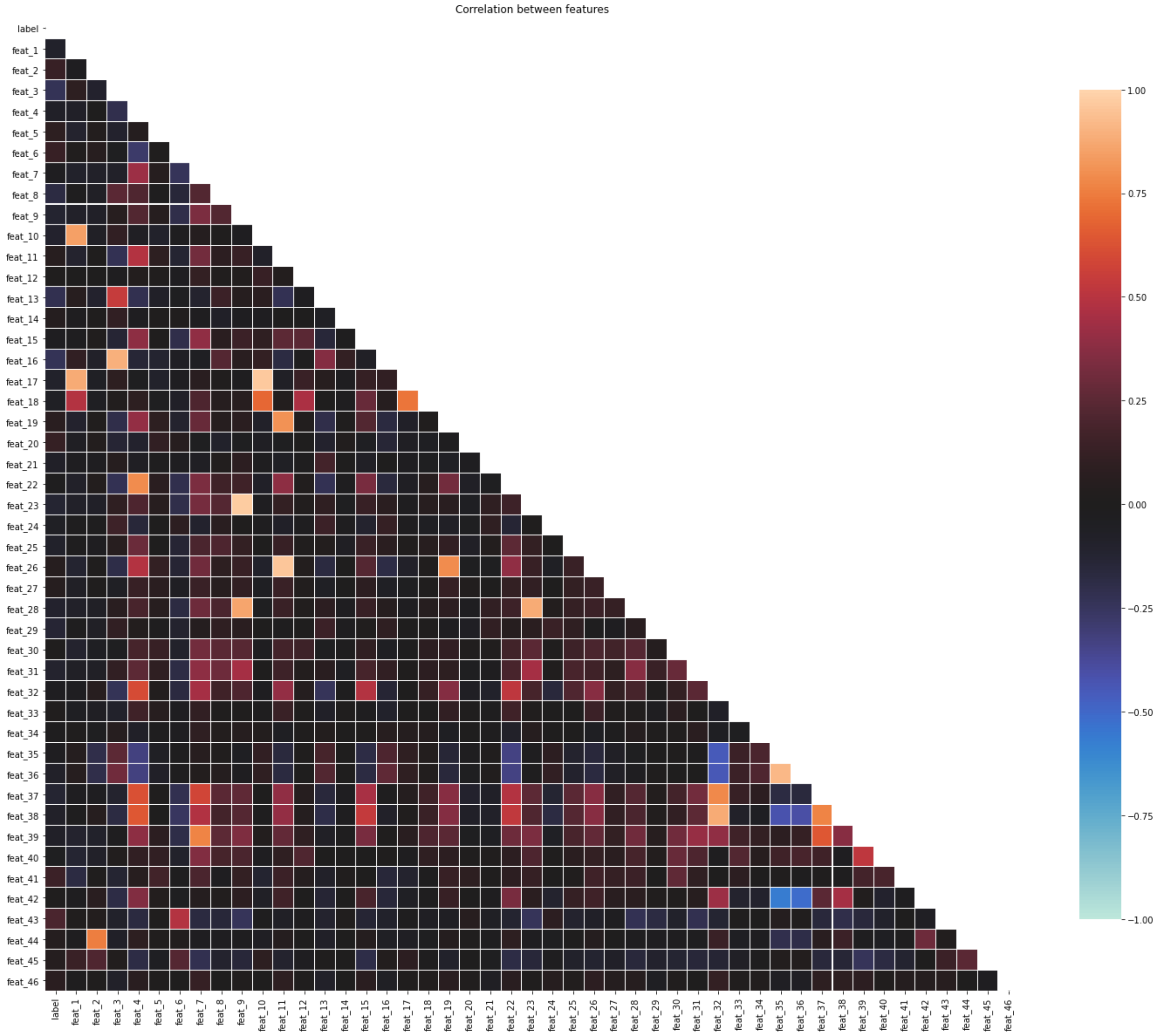


*Figure 9: Heatmap of correlation between features.*

## 2.3. Feature engineering & Holdout dataset

### Feature engineering

Since we don’t have the interpretation of features our opportunities for feature engineering are extremely limited. However, there is one thing we can do. 14 features in our dataset have missing values. Before training models we will replace missing values by some number. For example, it can be zero, the minimum or maximum value in a column, or the median value. However, the absence of a value itself may turn out to be a significant feature for the model. Therefore, it is reasonable to save this information. We will do this by creating boolean columns “feat_XX_isnull” for each column that has missing values.

### Holdout dataset

In the later stages of model creation we are going to need holdout dataset to ensure the quality of the model. Typically, the holdout set is approximately 20% of the available data. In our case, it makes sense to raise this figure to 25% and use as a holdout set all the data for the latest available date, i.e. for January 2019. This way we can simulate the real work of the model. It will be trained and validated on historical data and then launched on data that came later. For this section reproducible code see Appendix 2.

## 2.4. Dimensionality Reduction

Some machine learning models are susceptible to collinearity of features. In addition, the oversampling methods SMOTE and ADASYN use KNN. KNN is known for being very susceptible to “Curse of Dimensionality”. The problem is that with an increase in the number of dimensions, the Euclidean distance ceases to be adequate to find the nearest neighbors. Code is available in Appendix 3.

Therefore, we will use PCA to reduce the dimensionality of the input data.

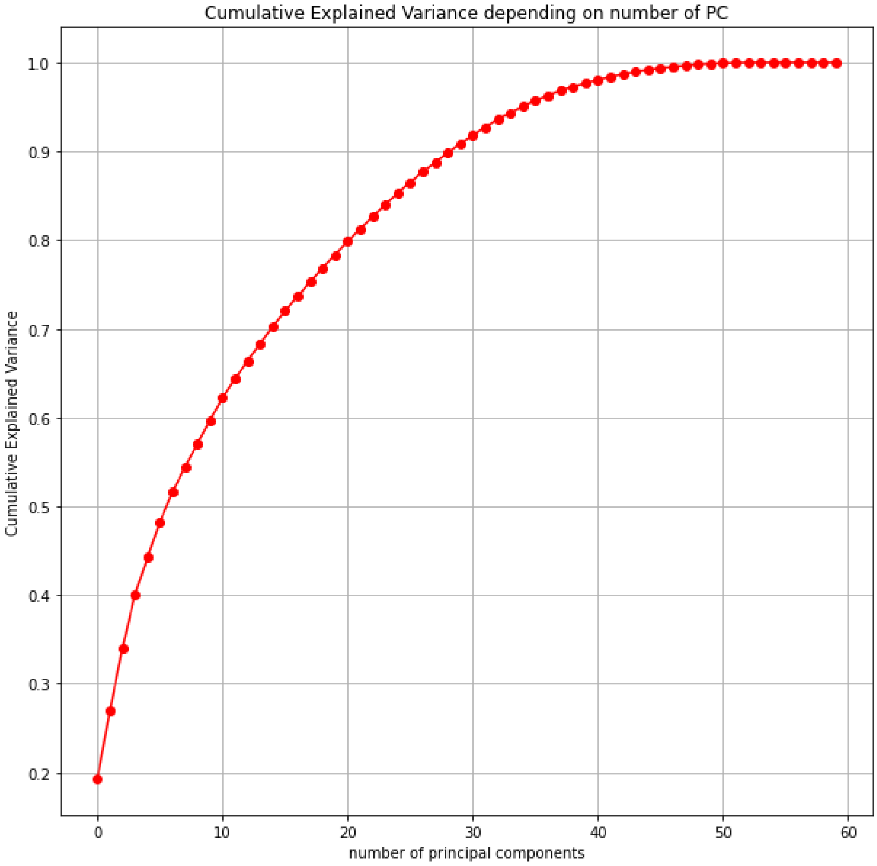


*Figure 10: Cumulative Explained Variance depending on number of PCs.*

PCA analysis shows that 35 principal components explain 95% of the total variance. Therefore we can effectively reduce the number of dimensions in our dataset from 60 to 35.

## 2.5. Handling sample imbalance & Data generation

Tasks associated with modeling the outflow of users often face the problem of an unbalanced dataset. The problem is that the churn class is usually noticeably smaller than the non-churn class. This is true for our dataset. For the code on sample imbalance problem see Appendix 4.

Consider this situation. Suppose we have a dataset in which 99% of the data belongs to class "0" and 1% of the data belongs to class "1". In this case, if the model predicts the class "0" for all data, it will already achieve an accuracy of 99%. At the same time, we can say with confidence that such a model is useless for business. It cannot predict which users may stop using the services of the company. This means that the company cannot use it to carry out measures against the outflow.

There are three main strategies to deal with unbalanced datasets: undersampling, reweighting and oversampling.

**Undersampling** strategy means that not all data from the majority class will be used when training models. If we talk about the example that was given at the beginning of the paragraph, then in order to balance the dataset, only 2% of the initial data will be used. 1% of class “0” and 1% of class “1”. This strategy is usually only mentioned by researchers, but not used in practice. The reason for this is that we lose a large share of information about the majority class with this approach.

The second approach to the problem of an unbalanced dataset is called **reweighting**. This approach means that the majority and minority classes are given different weights. During training, the model is fined more for an error in a minority class than for an error in a majority in a class. The advantage of this approach compared to undersampling is that we do not lose information about the majority class.

Finally, a third approach to solving the problem of an unbalanced dataset is called **oversampling**. In this approach, additional elements of the minority class are added to the dataset. The simplest kind of oversampling is random oversampling. In random oversampling we add extra elements of the minority class using random sampling with the replacement of the currently available elements of minority class.

In addition to random oversampling, there are two other types of oversampling that are often used in the scientific literature. They are called Synthetic Minority Oversampling Technique (SMOTE) (Chawla et. al. 2002) and Adaptive Synthetic (ADASYN) (Haibo et al., 2008).

While random oversampling creates duplicates of existing elements of the minority class, SMOTE and ADASYN generate new data similar to existing data by interpolation. However, they use slightly different approaches. ADASYN focuses on generating new elements next to elements that were incorrectly classified using KNN. SMOTE, in turn, does not distinguish between elements that are difficult or easy to classify using KNN.

Thus, oversampling not only saves all the information about the majority class, but also allows you to generate new elements of the minority class by interpolation. This approach improves the predictive properties of the models.

In the research we used six different datasets that were generated using three different oversampling methods with and without dimensionality reduction using PCA. As part of the study, we will examine how the characteristics of models built on these six datasets differ.

## 2.6. Models used and results

### 2.6.1. K-nearest neighbours

Regarding the analysis methods, one of the simplest and most intuitive ones is K-nearest neighbours (KNN). It was introduced by E. Fix and J.L. Hodges in US Air Force School of Aviation Medicine report (Fix & Hodges, 1951). The logic behind it is that similar objects tend to belong to the same class.

KNN assigns class label to an object based on its neighbours, meaning that it chooses the class of the majority of closest neighbours and assigns it to the analysed object. In binary classification (e.g. churn prediction) it is reasonable to take odd number of neighbours to avoid the situations with equal number of neighbours from different classes.

This method is quite simple in application and can be used as a good baseline model in machine learning tasks. What is more, KNN is quite flexible in terms of kernel and metric and can detect similarities of complex objects. Finally, the method is easy to interpret and, thus, can be easily explained.

However, KNN has several drawbacks. First one is its high computational complexity, the method calculates distances between each pair of observations. Speed is also impacted by the flexibility, highly customizable parameters (kernel, number of neighbours) are hyper-tuned and it takes a lot of time. Final disadvantage is KNN's instability, i.e. a slight change in one of the features can significantly change the neighbourhood of an object and their classes.

Considering the results of KNN model, six different variations were tested. All the models were used on three different datasets based on method of fighting sample imbalance (random oversampling, SMOTE, ADASYN), while also had two distinct models with different features (applying principal component analysis (PCA) and not). The results of the models are presented on the precision-recall curve with AUC-PR metric stated. PCA features datasets demonstrated lower results for all the KNN models. The best results are provided by model trained on SMOTE with AUC-PR equal to 0.4191, followed by random oversampled data model with metric value of 0.3979 and, finally, the lowest result is shown by ADASYN dataset with AUC-PR equal to 0.3874. (See Figure 11-16)

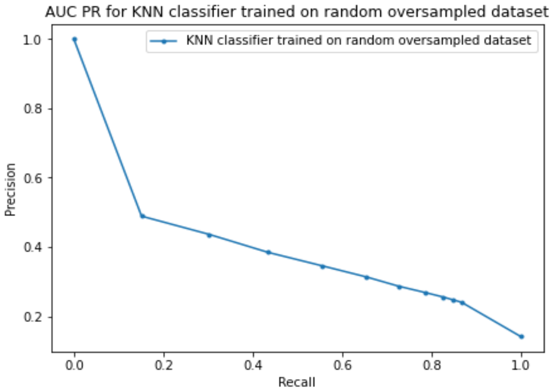


*Figure 11:. AUC PR = 0.3979 for K-nearest neighbours classifier trained on Random oversampling dataset.*

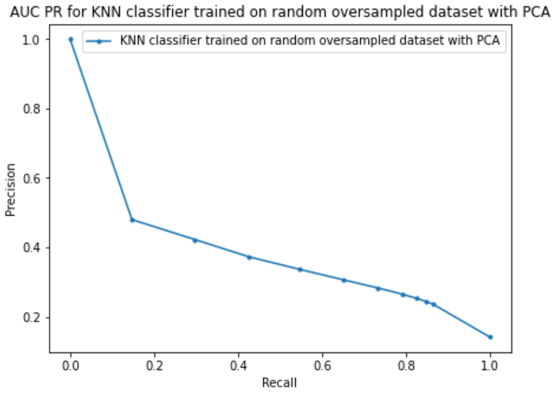


*Figure 12: AUC PR = 0.3884 for K-nearest neighbours classifier trained on Random oversampling dataset with PCA.*

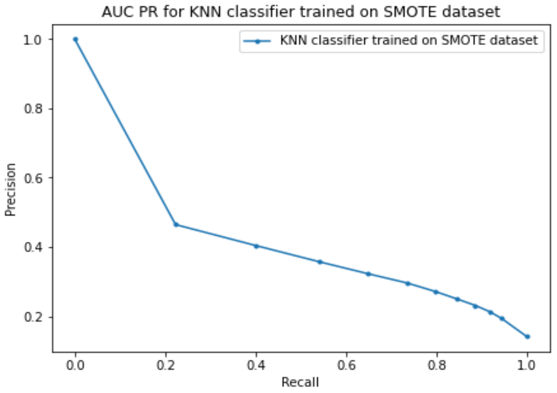


*Figure 13: AUC PR = 0.4191 for K-nearest neighbours classifier trained on SMOTE dataset.*

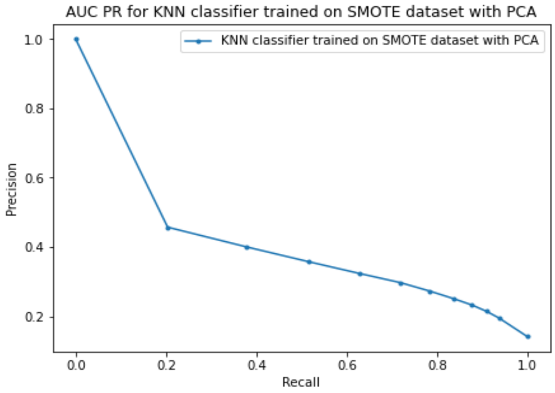


*Figure 14: AUC PR = 0.4075 for K-nearest neighbours classifier trained on SMOTE dataset with PCA.*

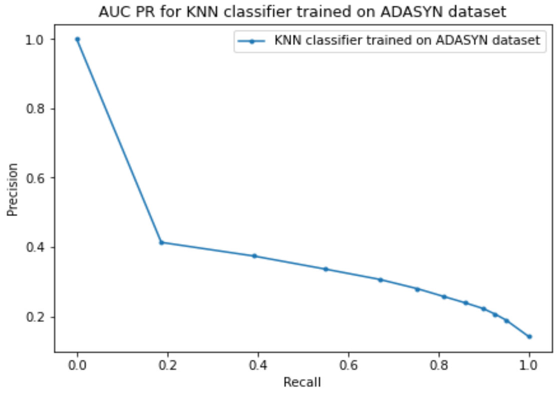


*Figure 15: AUC PR = 0.3874 for K-nearest neighbours classifier trained on ADASYN dataset.*

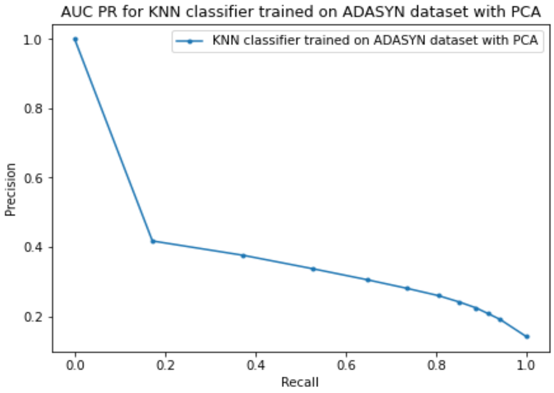


*Figure 16: AUC PR = 0.3810 for K-nearest neighbours classifier trained on ADASYN dataset with PCA.*

## 2.6.2. Logistic Regression

Logistic Regression is one of the most basic models for classification. It is used with categorical dependent variable. It tends to be very vulnerable to class imbalance and mostly predict majority class, while ignoring minority class and treat it as noise. Therefore, application of aforementioned techniques for coping with sample imbalance is very important. Logistic regression does not predict the class, it actually returns the probability of affiliation to a particular class (churn/active in our case). The main idea of logistic regression is that the space of initial values can be divided by a linear boundary into two regions corresponding to the classes. This boundary is set depending on the available input data and the training algorithm.

Analysing the results of logistic regression models, the same procedure of comparing 6 different models on 3 datasets and two types of features is applied. The improvement of results is obvious compared to the KNN models, meaning that the worst AUC-PR score for logistic regression is equal to 0.4378 and is better than the top result of KNN models. The logic of features choice continues and models trained on non-PCA data outperform the PCA. The best AUC-PR score for logistic regression model is equal to 0.4736 and reached on SMOTE dataset without PCA features. Considering the precision-recall curves shapes, they are clearly better than KNN and do not have such a sudden drop of precision with increase of recall. (See Figure 17-22)

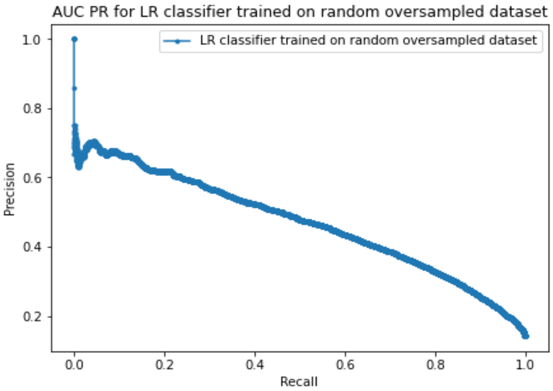


*Figure 17: AUC PR = 0.4691 for Logistic Regression classifier trained on Random oversampling dataset.*

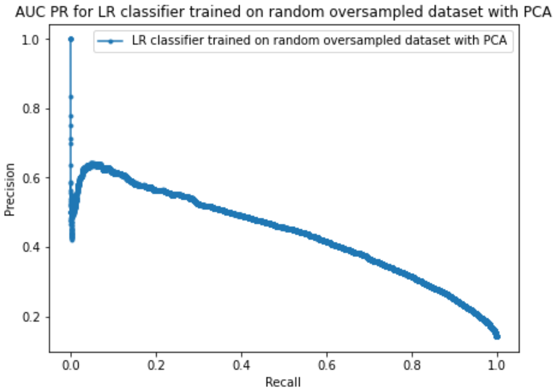


*Figure 18: AUC PR = 0.4392 for Logistic Regression classifier trained on Random oversampling dataset with PCA.*

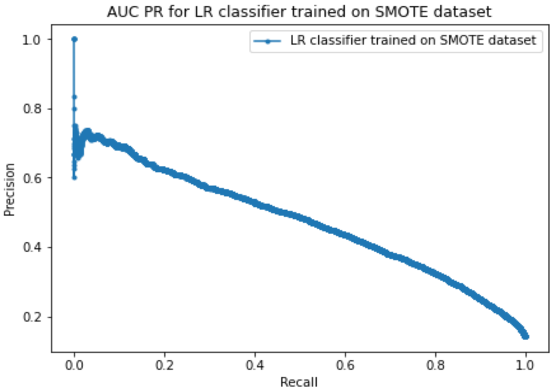


*Figure 19: AUC PR = 0.4736 for LogisticRegression classifier trained on SMOTE dataset.*

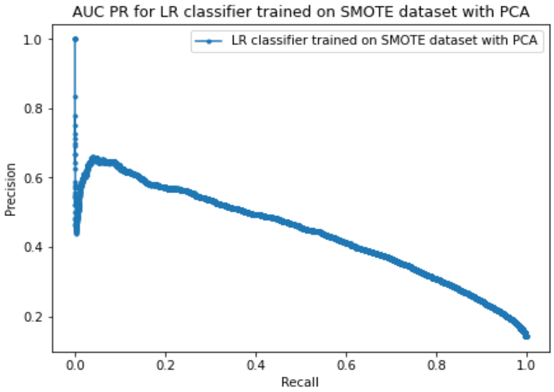


*Figure 20: AUC PR = 0.4414 for LogisticRegression classifier trained on SMOTE dataset with PCA.*

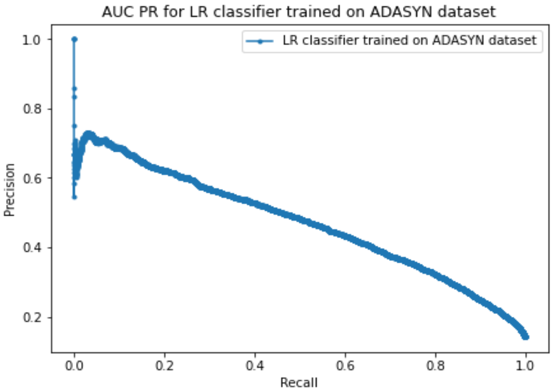


*Figure 21: AUC PR = 0.4706 for LogisticRegression classifier trained on ADASYN dataset.*

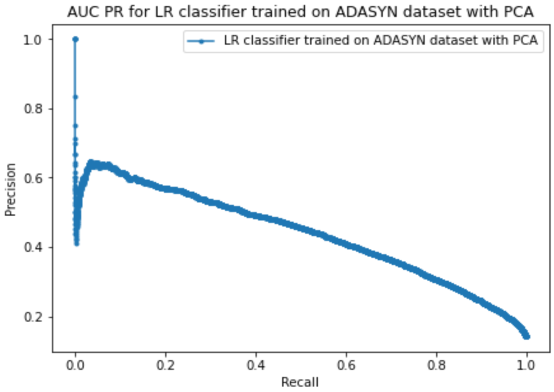


*Figure 22: AUC PR = 0.4378 for LogisticRegression classifier trained on ADASYN dataset with PCA.*

### 2.6.3. Random Forest

Random forest classifier is a powerful algorithm, which is based on bagging (bootstrap aggregating) and is an ensemble method that consists of decision trees. First algorithm of random decision forests was introduced by Tin Kam Ho (1995). Later on an extension of this algorithm was developed by Leo Breiman and Adele Cutler (2001, 2004).

The idea of the model is to randomly take multiple subsamples of the data with the size equal to the initial dataframe (with replacement) and use each of them to train a different decision tree. After that all the decision trees predict the class label and the final label is chosen with the majority voting, so that most of the errors from bad decision trees will be mitigated. This method is less prone to overfitting, can handle missing values automatically and is quite robust to outliers. What is more, it is good with high-dimensional data, which coincides with our case. However, it has high computational complexity compared to simple methods, requires more time to train and lacks interpretability.

Regarding the results of random forest, they are in general better than those of previously mentioned models. The variations of random forest model are done in the same manner and divided by sample imbalance approach and features. In this model, PCA also fails compared to non-PCA features data. Precision-recall curves shapes are better and than that of KNN and LR methods, they are much more smooth than of KNN and maintain even better results than logistic regressions. (See Figure 23-28) The best result in this method is not demonstrated by a model trained on a SMOTE dataset as in two previous algorithms, but by model trained on random oversampled data. Best AUC-PR score for random forest is equal to 0.5419.

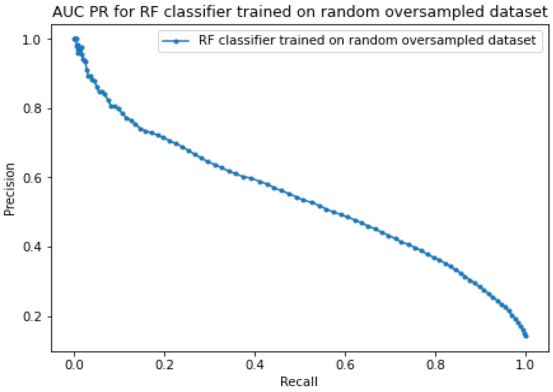


*Figure 23: AUC PR = 0.5419 forRandom Forest classifier trained on Random oversampling dataset.*

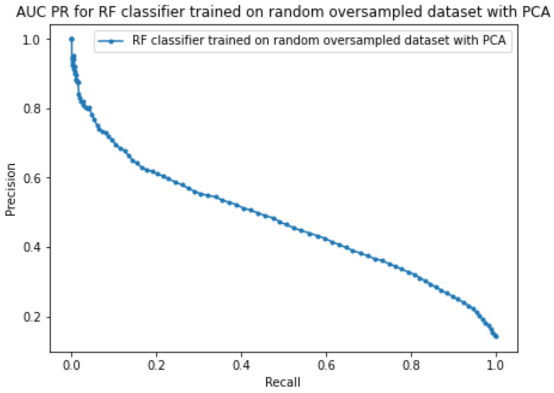


*Figure 24: AUC PR = 0.4753 for Random Forest classifier trained on Random oversampling dataset with PCA.*

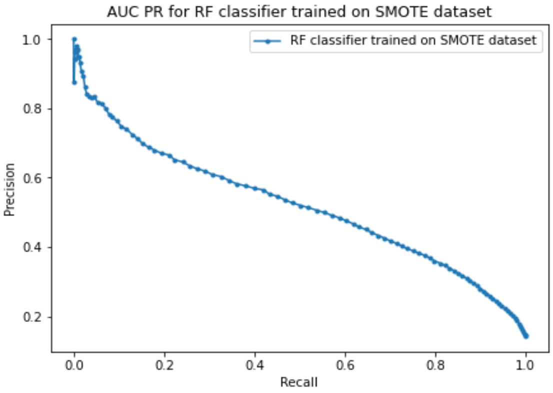


*Figure 25: AUC PR = 0.5205 for Random Forest classifier trained on SMOTE dataset.*

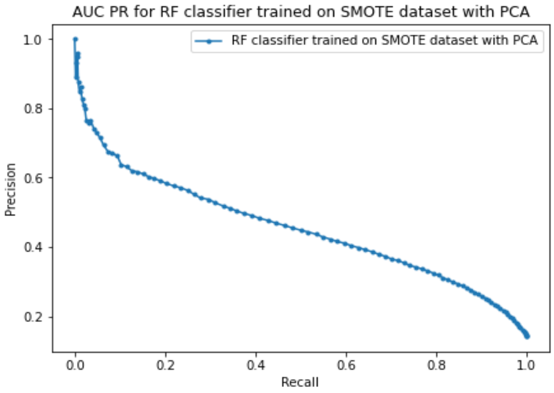


*Figure26: AUC PR = 0.4558 for Random Forest classifier trained on SMOTE dataset with PCA.*

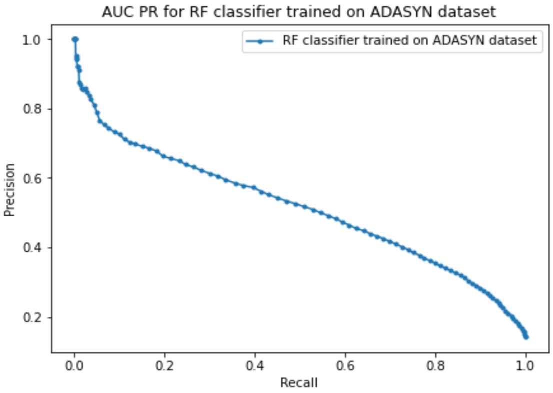


*Figure27: AUC PR = 0.5144 for Random Forest classifier trained on ADASYN dataset.*

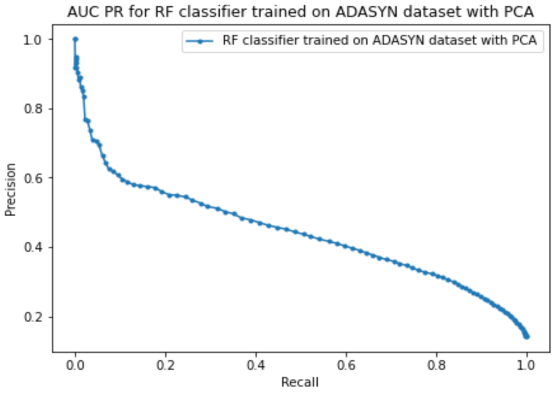


*Figure 28: AUC PR = 0.4427 for Random Forest classifier trained on ADASYN dataset with PCA.*

### 2.6.4. XGBoost

XGBoost is a machine learning algorithm that was first introduced at the SIGKDD Conference in 2016 by Tianqi Chen and Carlos Guestrin. Nowadays it is one of the most commonly used machine learning algorithms. It is an extended, more regularized version of a gradient boosting algorithm. The idea behind it is kind of the opposite to bagging approach used in random forest. While bagging trains multiple simple models simultaneously and uses majority voting to provide class labels, gradient boosting creates an ensemble model, which consists of a sequence of models. These models are trained one after one learning on the results of the previous one and consecutively updating weights in order to minimize the weighted error function.

Results of XGBoost are better than all of the three models mentioned before. The best AUC-PR score for this technique is equal to 0.5399. It is reached on the random oversampled dataset without PCA features. Considering the precision-recall curves, they behave the way similar to the random forest, but still the area under the curve is larger for XGBoost. (See Figure 29 - 34)

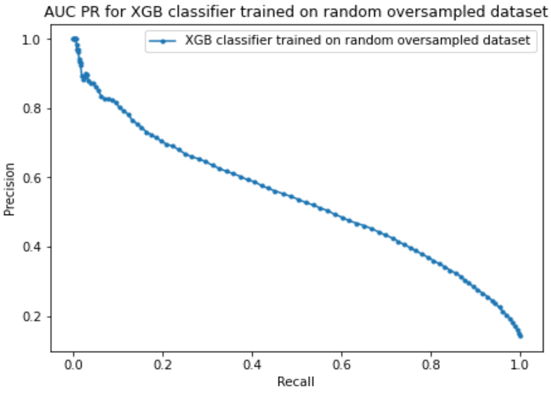


*Figure 29: AUC PR = 0.5399 for XGBoost classifier trained on Random oversampling dataset.*

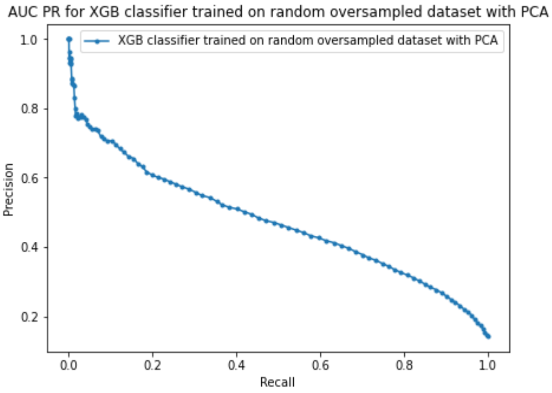
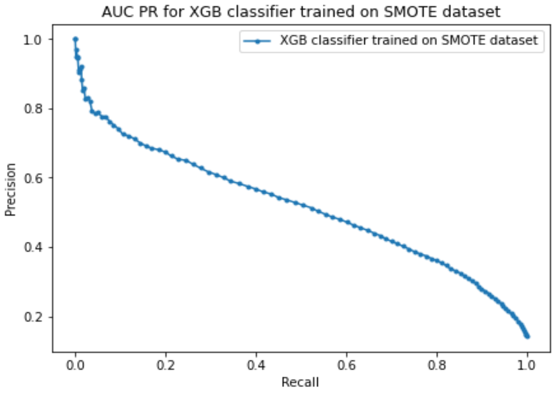


*Figure 30: AUC PR = 0.4737 for XGBoost classifier trained on Random oversampling dataset with PCA.*



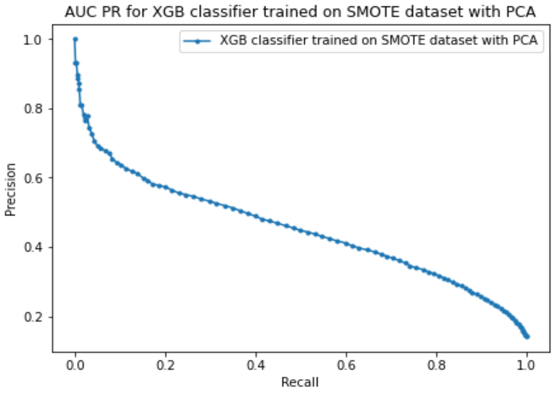
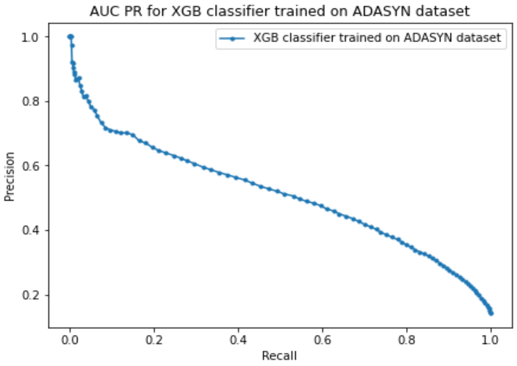


*Figure 31: AUC PR = 0.5167 for XGBoost classifier trained on SMOTE dataset.*

*Figure 32: AUC PR = 0.4525 for XGBoost classifier trained on SMOTE dataset with PCA.*



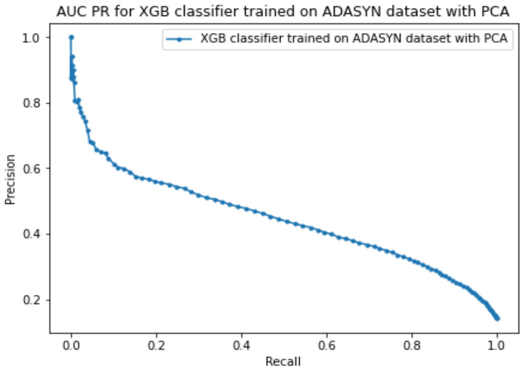


*Figure 33: AUC PR = 0.5110 for XGBoost classifier trained on ADASYN dataset.*

*Figure 34: AUC PR = 0.4448 for XGBoost classifier trained on ADASYN dataset with PCA.*

### 2.6.5. ANN

Neural networks are computational networks in which the decision-making process is inspired by the decision-making process in the nerve cells (neurons) of the human or animal nervous system. Of course, this statement is a simplification, but it makes it easier to visualize how ANNs work.

Graupe Daniel (2013) points out that “artificial neural networks are very suitable to solve problems that are complex, ill-defined, highly non-linear, of many and different variables, and/or stochastic”.

The reason for this visualized by Andrew Ng (2017) in the following picture:

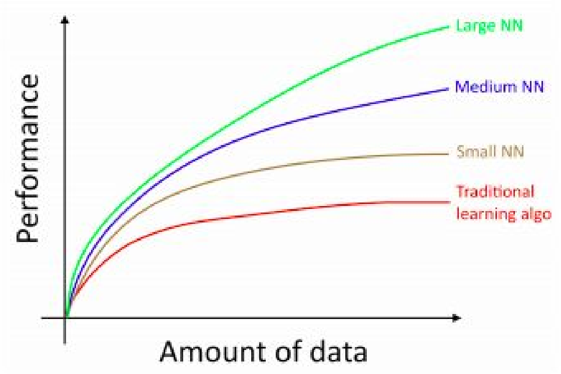


*Figure 35: Performance of machine learning algorithms depending on computational scale and amount of data available.*

He points out that even small neural networks tend to outperform classical machine learning algorithms. And with the increase of the amount of data available and the complexity of ANN the gap in performance  is only increasing.

The architecture of the neural network we used in the research is as follows. Since we are dealing with tabular data, the Sequential model is used. The model has 2 hidden layers. Each of the hidden layers has 128 neurons with a ReLU activation function. Dropout regularization with probability 0.5 applied to both layers. As a loss function used binary cross entropy. As an optimizer - Adam. The training history of the model is available in Appendix 6.

As with previous classifiers, six different neural networks were trained. Results of artificial neural networks are better than all of the models mentioned before. The best AUC-PR score for this algorithm is equal to 0.5488. It is achieved on the random oversampled dataset. (See Figure 36 - 41)

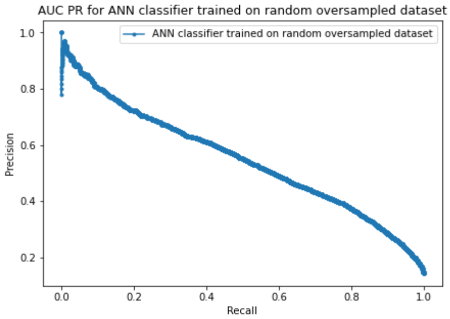


*Figure 36: AUC PR = 0.5488 for ANN classifier trained on Random oversampling dataset.*

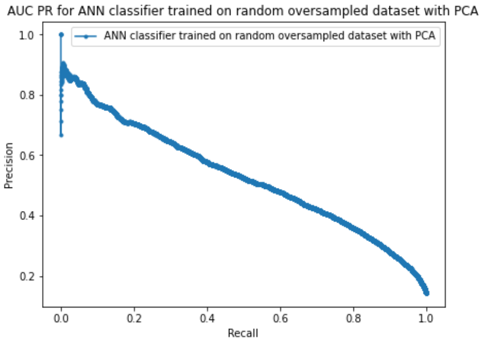


*Figure 37: AUC PR = 0.5299 for ANN classifier trained on Random oversampling dataset with PCA.*

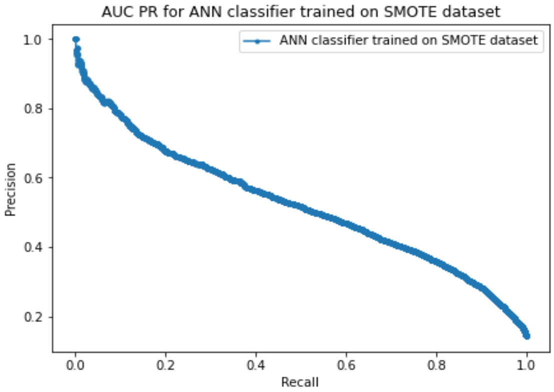


*Figure 38: AUC PR = 0.5227 for ANN classifier trained on SMOTE dataset.*

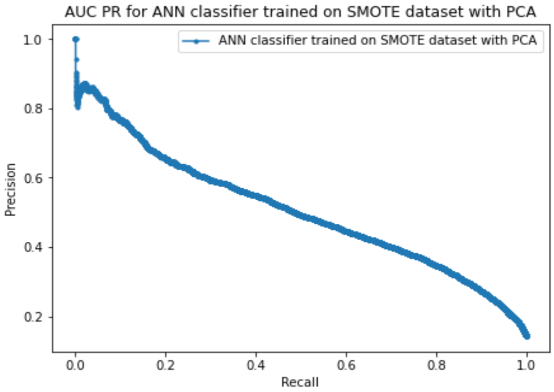


*Figure 39: AUC PR = 0.5057 for ANN classifier trained on SMOTE dataset with PCA.*

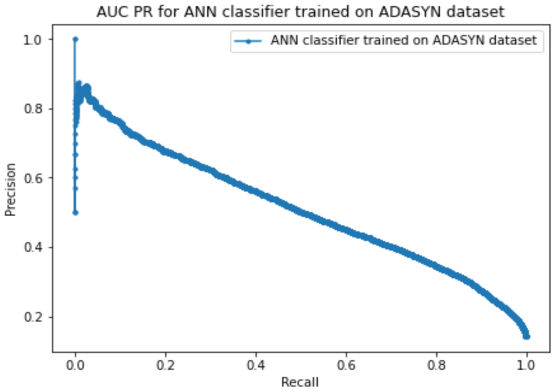


*Figure 40: AUC PR = 0.5110 for ANN classifier trained on ADASYN dataset.*

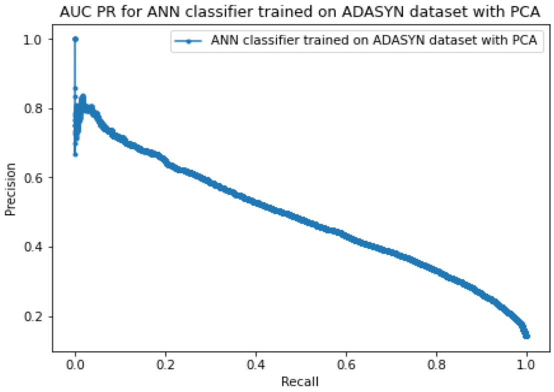


*Figure 41:  AUC PR = 0.4861 for ANN classifier trained on ADASYN dataset with PCA.*

### 2.6.6. Models results discussion

The following table compares AUC PR values on a testing set of five different machine learning algorithms trained on six different datasets. The machine learning algorithms considered are: K-nearest neighbors, Logistic Regression, Random Forest, XGBoost and Artificial neural network. The datasets considered are the following: Random Oversampling Dataset (ROS), Random Oversampling Dataset with PCA (ROS + PCA), SMOTE Dataset (SMOTE), SMOTE Dataset with PCA (SMOTE + PCA), ADASYN Dataset (ADASYN), ADASYN Dataset with PCA (ADASYN + PCA). The best AUC-PR value in each of the models is highlighted in bold, while the best overall result is also underlined. The code for all the training of the models can be seen in Appendix 5.

| | **K-nearest neighbors** | **Logistic Regression** | **Random Forest** | **XGBoost** | **Artificial neural network** |
|---|---|---|---|---|---|
| **ROS** | 0.3979 | 0.4691 | **0.5419** | **0.5399** | **<u>0.5488</u>** |
| **ROS + PCA** | 0.3884 | 0.4392 | 0.4754 | 0.4737 | 0.5299 |
| **SMOTE** | **0.4192** | **0.4736** | 0.5205 | 0.5167 | 0.5228 |
| **SMOTE + PCA** | 0.4075 | 0.4414 | 0.4558 | 0.4525 | 0.5057 |
| **ADASYN** | 0.3874 | 0.4706 | 0.5144 | 0.5110 | 0.5098 |
| **ADASYN + PCA** | 0.3810 | 0.4378 | 0.4427 | 0.4448 | 0.4861 |

*Table 2: AUC PR value for classifiers trained on different datasets.*

It can be seen that in the churn prediction problem for TELE2, the artificial neural network trained on the dataset created using Random oversampling turned out to have the best result on the test dataset. The AUC PR value for this classifier turned out to be 0.5488.

Since the company also asked to maximize F1 score, we find the threshold value for which F1 score is maximum. To do this, we calculated F1 score for each threshold value from 0 to 1 in increments of 0.01. The results of this calculation are presented in the graph:

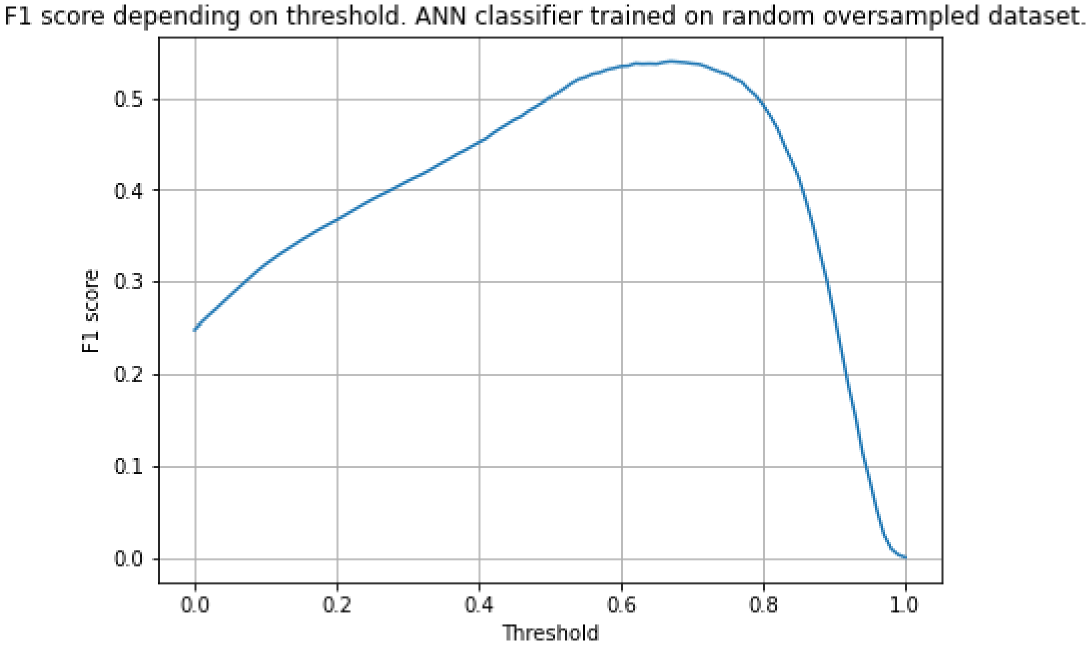


*Figure 42: F1 score depending on threshold value for ANN classifier trained on random oversampling dataset.*

Maximum F1 score is 0.5398. It is achieved with threshold 0.67.

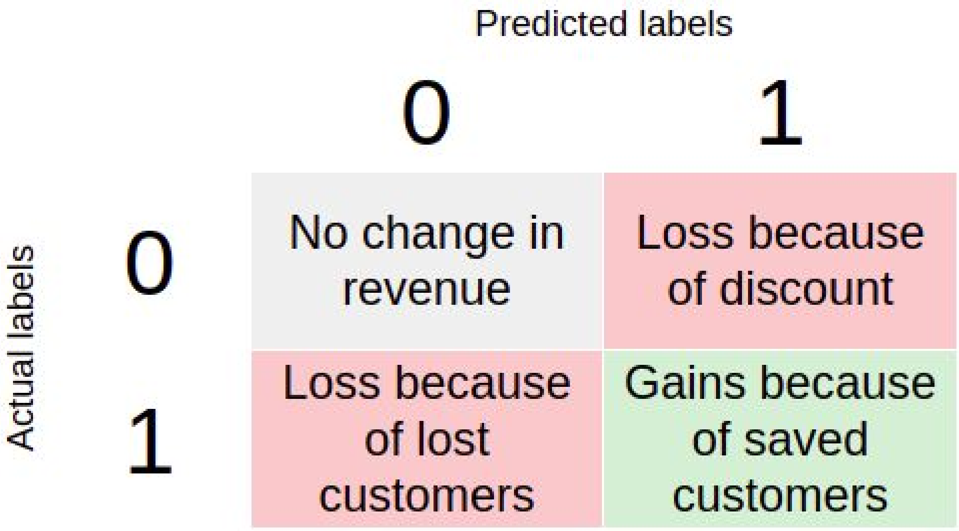
An interesting moment in the results is that the dimensionality reduction using PCA did not lead to an improvement in quality for any of the classifiers. At the same time, there are studies where the use of PCA in conjunction with oversampling led to a noticeable improvement in the results of models on unbalanced datasets. For example, Mehdi Naseriparsa and Mohammad Mansour Riahi Kashani (2014) were able to achieve a quality improvement in the Lung Cancer prediction problem in this way. The reason for this may be that the dimension of the original dataset was not very large.

To sum up, an artificial neural network trained on the dataset created using random oversampling showed the best result in our case. It is this model that will be used in the remainder of the project when building the economic model.

## 2.7. Managerial implications

In order to improve the retention of clients, it is needed to establish a process for churn reduction. It assumes the usage of a wide range of management and IT tools. Firstly, it is required to conduct a detailed analysis and estimation of an existing retention system. Secondly, churn impact on customer base and financial figures should be estimated. Thirdly, the customer base needs to be segmented and churn influence should be measured for each segment. After that the distinct causes of churn should be spotted for each segment. Then each of the segment should be defined in terms of profitability for the company (e.g. ARPU) and proposals should be adapted according to that information (more offers for highly valuable clients and limited offers for clients with low value). Finally, using the microsegment analysis it is needed to develop an individual approach and offer for each client in order to maximize the revenue.

After creating multiple customer churn prediction models and identifying the best one, it is possible to predict quite well, whether the customer is going to leave the company or not. However, such a result is only an intermediate step and analysis should be continued in terms of retention campaigns. It means that the company proceeds to the final step of the business analytics cycle - prescriptive analytics. Decision-makers from the company are mainly interested not in churn prediction model predictive power and its technical realisation, but in general result and financial impact. Therefore, it is important to understand how churn prediction model influences the company operations in money terms, not only the quantity of people retained. Before suggesting any implications, it is needed to understand how each of the possible model outcomes impacts company’s revenue. The following table describes all the 4 possible options, where 0 corresponds to non-churn customers and 1 - to churn customers:



| | | Predicted labels | |
|---|---|---|---|
| | | 0 | 1 |
| Actual labels | 0 | No change in revenue | Loss because of discount |
| | 1 | Loss because of lost customers | Gains because of saved customers |

*Figure 43: Classifier prediction table*

It is clear that when our model predicts a client as no-churn and he turns out to be non-churn, the company does not experience any change in revenue, because customers stay and pay the same amount of money. At the same time, if the customer is not leaving, but is predicted as would-be-churn, he will receive an offer, which will not change his decision (he was going to stay in any case), but will incur additional costs for the company, because the customer will accept the offer. This loss can be measured as discount loss. Another not beneficial option is quite the opposite, when the algorithm did not detect the real would-be-churn customer and the company did not take action to retain the client. Consequently, the company lost a customer and his CLV. Final option is the actual goal of the model - precise prediction of would-be-churners. If the algorithm succeeds in prediction, the company will timely contact the client and provide him an offer in order to keep him using the services. It will bring additional revenue, because, otherwise, the customer would have left and the company would have experienced losses.

The most important point of model results application is identifying the ideal retention campaign in terms of profits. As long as success of the campaign usually varies significantly with the change of number of customers included, it is reasonable to maximize the profit of the campaign with respect to the share of customers included in the targeted group. According to Devriendt et al. (2019) the profit equation (also mentioned in the literature review) is the following (see section 1.3.2):

$$\Pi = N\alpha[\beta\gamma(b - c_{contact} - c_{incentive}) + \beta(1-\gamma)(-c_{contact}) + (1-\beta)(-c_{contact} - c_{incentive})] - A$$

It is then maximised with respect to $\alpha$ - proportion of customers targeted in the retention campaign. Therefore, such a procedure will help to identify the most profitable option and target the optimal number of clients.

Apart from the estimation of the campaigns, it is also needed to choose the best possible classifier. It may seem that it is quite an easy task, which uses only technical objective metrics, such as computational complexity, precision, recall, area under curve and F1-score. However, choice of classifier should also be justified from business perspective. Thus, it is needed to estimate the application of each particular classifier using cost-based approach. This idea is presented by Ahmed & Maheswari (2018), who showed a general view of the cost matrix for customer churn model. (See Table 3)

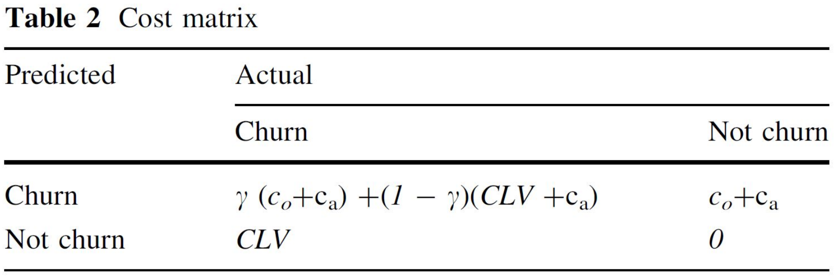


**Table 2** Cost matrix

| Predicted | Actual | |
|---|---|---|
| | Churn | Not churn |
| Churn | $\gamma\ (c_o+\mathrm{c_a})\ +(1\ -\ \gamma)(CLV\ +\mathrm{c_a})$ | $c_o+\mathrm{c_a}$ |
| Not churn | *CLV* | *0* |

*Table 3: Cost matrix of churn economic model*

In this table $\gamma$ is the proportion of customers retained, CLV stands for the customer lifetime value, $c_0$ corresponds to the cost of the company proposition and $c_a$ is the cost of contacting the customer.

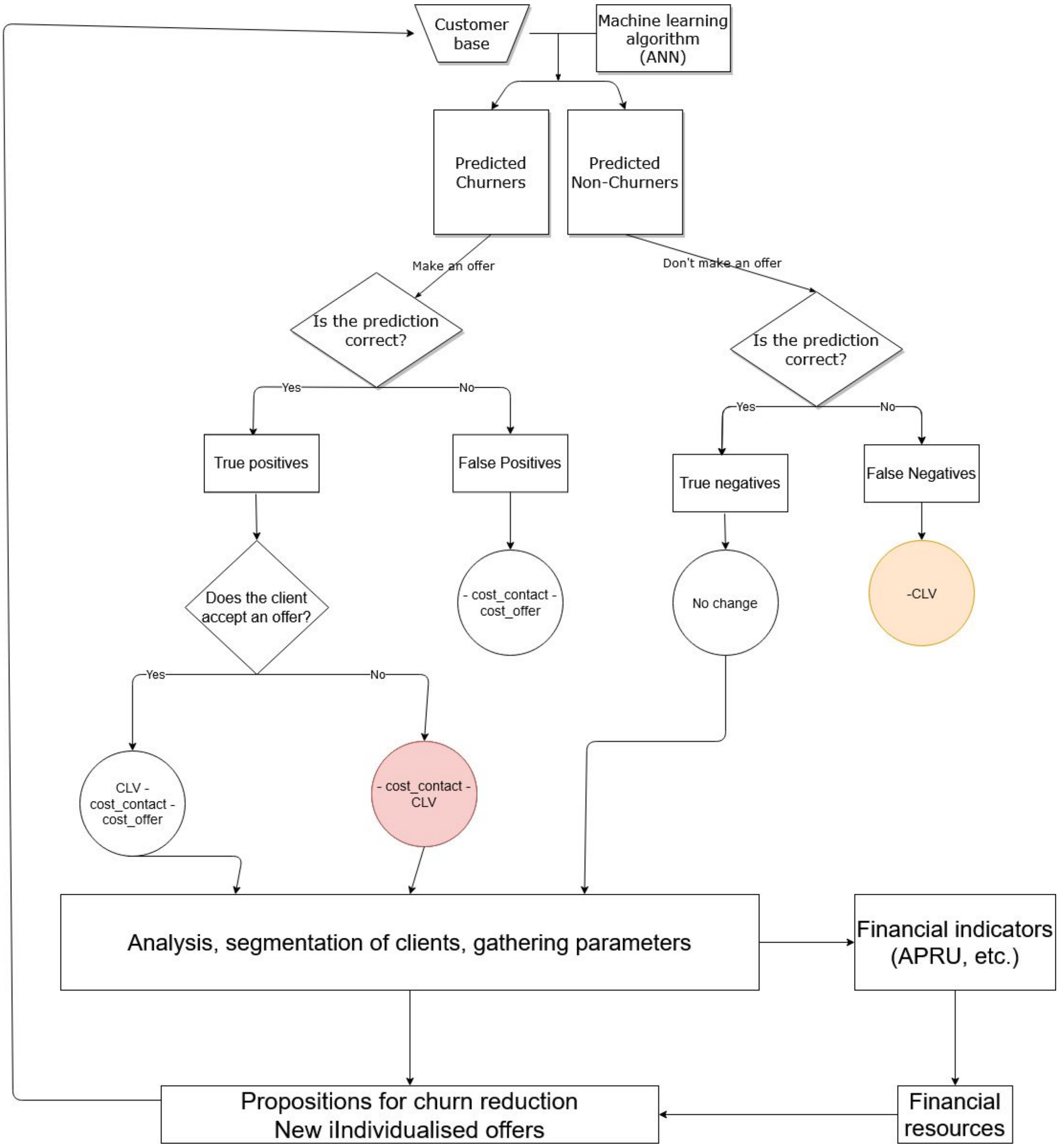
The cost of using a churn prediction classification model is identified with the following equation:

$$Cost_i = y_i(c_i C_{TP_i} + (1-c_i)C_{FN_i}) + (1-y_i)(c_i C_{FP_i} + (1-c_i)C_{TN_i}),\ where$$

$c_i$ - predicted label, $y_i$ - actual class label (0 - not churn, 1 - churn). All the costs are taken from the table above, where TP (True positive) is equal to churn-churn cell.

This equation helps to estimate the goodness of the classifiers and spot hidden insights, because machine algorithms do not make difference between true and false predictions, while their financial impact is explicitly shown by the formula above. Considering the profit equation, it will definitely help to sort the retention campaigns and make a decision on narrowing or widening the target group of customers.

In order to ease the process of customers retention using the model results, a churn prediction process diagram was prepared based on ANN algorithm and aforementioned cost matrix.



Customer base
Machine learning algorithm (ANN)
Predicted Churners
Predicted Non-Churners
Make an offer
Don't make an offer
Is the prediction correct?
Is the prediction correct?
Yes
No
Yes
No
True positives
False Positives
True negatives
False Negatives
Does the client accept an offer?
- cost_contact - cost_offer
No change
-CLV
Yes
No
CLV - cost_contact - cost_offer
- cost_contact - CLV
Analysis, segmentation of clients, gathering parameters
Financial indicators (APRU, etc.)
Propositions for churn reduction
New iIndividualised offers
Financial resources

Figure 44: Churn prediction process diagram

This diagram shows the main steps of model usage and highlights the areas of improvement for the company. The first part of the diagram until the last rectangular-shaped bricks describes the model prediction process. After that the retention campaign part commences. The orange colour is responsible for clients who were not identified as churn, but turned out to be churn and abandoned the company’s services. It is the problem of the classifier and it means that the neural network should be trained better. The circle highlighted in red is the most important from a business perspective. The model correctly predicted the customer as a churn one, however, the company did not manage to persuade the client to stay with their services. Therefore, it is possible that the offer was not good enough and the company should reevaluate its retention campaign policy. It may be a good idea to launch a survey for such clients and ask why they are leaving and what would persuade them to stay. The final steps are similar to the logic described in the beginning of this section. The clients, who should be analysed from the business perspective rather than from the technical one are churners and correctly identified non-churners, because they can bring additional insights on the strengths of the company's value proposition. Based on the gathered parameters and analysis, it is possible to segmentize the customers and estimate their impact on the company's operations from a financial perspective (analyse their financial parameters and financial resources of the company). Having gathered such information, individual offers for customers and general proposition for churn management and reduction can be formulated, and, consequently, future share of retained churn customers will increase.

# CONCLUSION

With an extremely fast pace of development, the world is experiencing the growth of digital data and technologies, and, consequently, industries apply more and more innovations increasing their degree of digitalisation. The farther the technologies evolve, the more opportunities and approaches appear, which enable companies to identify and solve difficult tasks.

Telecommunication services industry is one of the most technologically developed industries, especially in Russia, and because it is so developed, it faces many challenges, which need to be handled. As long as the telecommunication market is quite saturated, the companies in the industry have a fierce fight for customers and retaining them is a key factor of success for the company nowadays. Customer churn is one of the major problems for the industry and retaining clients is much cheaper than attracting a new one. Customers, who abandon the services of the company can not only impact the growth of business, but also change the revenues drastically.

The paper shows that customer churn prevention is an important business task and requires creation of customer churn prediction models, which will efficiently predict churn clients using machine learning algorithms. Successful prediction of customer churn mainly depends on the availability of the clients' historical data and its quality. Regarding these parameters, companies have enormous amounts of data about each client, but not always convert their data into useful insights. The most important question, which management tries to solve in churn domain, is what methods should be used to attract new customers and retain the old ones.

This paper provided a definition of churn, identified time frame for application of customer retention measures, demonstrated a general telecommunication market overview and Russian market in particular, taking into account the impact of recent COVID pandemic. Russian market analysis was done comparing top-4 mobile telecom operators (MTS, Megafon, VimpelCom and Tele2). Tele2 turned out to be the most promising and fastest developing operator. What is more, academic literature review was made and most popular and powerful methods for customer churn prediction were identified.

In this paper the goal of creating customer churn prediction model was reached and the resulting model can be used to accomplish the task of clients' retention. In this thesis the dataset provided by Tele2 mobile operator is used. Churn prediction task's solution is difficult because there is a strong imbalance between classes in the original dataset (between classes “Churn” and “Non-churn”). In addition, the features in the dataset are anonymized, which creates additional complexity at the feature engineering stage.

To solve the problem of class imbalance, three different approaches to oversampling were used: random oversampling, SMOTE and ADASYN. Our study did not reveal significant differences between different approaches to oversampling. Apparently this is due to the features of the dataset and on other data the results could be different.

Five different classifiers were used to solve the prediction problem, which can help in churn reduction: K-Nearest Neighbours, Logistic Regression, Random Forest, XGBoost and Artificial Neural Network. The best result in the outflow prediction problem was shown by the ANN classifier. Area under the curve (AUC) and F1-score were chosen as quality metrics for the model. ANN results turned out to be the best and are equal to 0.5398 and 0.5488 respectively.

The results received allow the company to identify the most vulnerable clients, who are about to abandon the services of the operator. What is more, several economic implications were proposed in this paper, which can be used to estimate the costs of retention campaigns and potential profit gains. Therefore, the company can use the model and economic proposition to identify the target group of clients and adapt the retention campaign. Finally, the processes for churn reduction were suggested and presented in the paper on different levels - considering microsegmentation and developing individual propositions and creating a flowchart with machine learning algorithm links to the retention campaign.

It is very important to have an efficient predictive model and take steps to increase the customer lifetime, because the costs incurred from his attraction, connection and maintenance should be recouped. If the company focuses only on the attraction of new customers, in the current state of saturation and aforementioned costs structure it risks to fall in long-term debt.

Considering the future development of the paper, the role of economic implication should be more emphasized and assumption of constant retention rate should be removed. Thus, future studies should continue testing ANN models, which proved to be successful, while also applying uplift modelling.

# REFERENCES

1. Ahmed, A. A., & Maheswari, D. (2019). An enhanced ensemble classifier for telecom churn prediction using cost based uplift modelling. *International Journal of Information Technology*, *11*(2), 381-391.

2. Astakhov, I.V. & Rudakova E.A. (2018). Study of the value churn rate of subscribers a large telecom network in a highly competitive market of the metropolis. *Radiolocation, navigation, network.* 89-92.

3. Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR), 49(2), 1-50.

4. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

5. Breiman, L., Cutler, A., (2004). Random Forest. <http://www.stat.berkeley.edu/breiman/RandomForests/cc_home.htm> (accessed 10.05.20).

6. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

7. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

8. Colgate, M. R., & Danaher, P. J. (2000). Implementing a customer relationship strategy: The asymmetric impact of poor versus excellent execution. *Journal of the Academy of marketing Science*, *28*(3), 375-387.

9. Daniel, G. (2013). Principles of artificial neural networks (Vol. 7). World Scientific.

10. Davenport, T. H., & Patil, D. J. (2012, October). Data Scientist: The Sexiest Job of the 21st Century. *Harvard Business Review*. Retrieved from https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

11. Devriendt, F., Berrevoets, J., & Verbeke, W. (2019). Why you should stop predicting customer churn and start using uplift models. *Information Sciences*.

12. Fix, E., Hodges, J.L. (1951). Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.

13. Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. Statistics and Computing, 28(3), 539-547.

14. He, Haibo, Yang Bai, Edwardo A. Garcia, and Shutao Li. (2008) "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," In IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 1322-1328.

15. Ho, T. K. (1995, August). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282). IEEE.

16. Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, *39*(1), 1414-1425.

17. Jain, P., & Surana, K. (2017, December 12). McKinsey Report: Reducing churn in telecom through advanced analytics. Retrieved from https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/reducing-churn-in-telecom-through-advanced-analytics#

18. Kamath, D. (2011) A Critical Evaluation of Customer Satisfaction of Cellular Phone Services in Pune, Doctoral Thesis, Symbiosis International University, Pune.

19. Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozaffari, M., & Abbasi, U. (2014). Improved churn prediction in telecommunication industry using data mining techniques. *Applied Soft Computing*, *24*, 994-1012.

20. Kim, H. S., & Yoon, C. H. (2004). Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market. *Telecommunications policy*, *28*(9-10), 751-765.

21. Lu, N., Lin, H., Lu, J., & Zhang, G. (2012). A customer churn prediction model in telecom industry using boosting. *IEEE Transactions on Industrial Informatics*, *10*(2), 1659-1665.

22. Naseriparsa, M., & Kashani, M. M. R. (2014). Combination of PCA with SMOTE resampling to boost the prediction rate in lung cancer dataset. arXiv preprint arXiv:1403.1949.

23. Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., & Mason, C. H. (2006). Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of marketing research*, *43*(2), 204-211.

24. Ng, A. (2017). Machine learning yearning. Retrieved from: http://www.mlyearning.org/(96).

25. Qureshi, S. A., Rehman, A. S., Qamar, A. M., Kamal, A., & Rehman, A. (2013, September). Telecommunication subscribers' churn prediction model using machine learning. In *Eighth International Conference on Digital Information Management (ICDIM 2013)* (pp. 131-136). IEEE.

26. Rengarajan, P., & Kavipriya, T. (2012). A study on level of customer's awareness in value added services on mobile phone subscribers-With special reference to Tiruppur district, Tamil Nadu. ZENITH International Journal of Business Economics & Management Research, 2(12), 162-175.

27. Tsydenova, N. (2019, November 25). MTS hochet vtroe snizit' ottok klientov k 2023 g za schet razvitija jekosistemy. Retrieved from https://ru.reuters.com/article/businessNews/idRUKBN1XZ1RD-ORUBS

28. Umayaparvathi, V., & Iyakutti, K. (2012). Applications of data mining techniques in telecom churn prediction. *International Journal of Computer Applications*, *42*(20), 5-9.

29. Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, *218*(1), 211-229.

30. Wansink K. (2020). *Key Global Telecom Industry Statistics*. BuddeComm Intelligence Report.

31. Zhu, B., Baesens, B., & vanden Broucke, S. K. (2017). An empirical comparison of techniques for the class imbalance problem in churn prediction. *Information sciences*, *408*, 84-99.

32. AC&M Consulting: Cellular Data 2019. (2020, June 4). Retrieved from http://www.acm-consulting.com/news-and-data/data-downloads/cat_view/7-cellular/37-cellular-2019.html

# APPENDICES

## Appendix 1. Code for Exploratory data analysis

```
#%%

from collections import Counter

import keras
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import xgboost as xgb

from imblearn.over_sampling import SMOTE, ADASYN, RandomOverSampler
from keras.callbacks import EarlyStopping
from keras.layers import Dense, Dropout
from keras.models import Sequential
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import auc, confusion_matrix, f1_score, precision_recall_curve
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler

#%% md

## Code for exploratory data analysis

#%% md

### Initial overview

#%%

data = pd.read_csv('data/raw/churn_data.zip')
data.head(n=10)

#%%

print ("# Rows:", data.shape[0])
print ("# Columns:", data.shape[1])

#%%

print('Distribution of report dates:')
print(data['report_date'].value_counts().sort_index())

#%%

print('Share of target class for different report dates:')
data.groupby(['report_date'])['label'].mean()

#%%

print('Number of target class for different report dates:')
data.groupby(['report_date'])['label'].sum()

#%%

plt.figure(figsize=(10, 6))
ax = sns.countplot(x="report_date", hue="label", data=data, order =
sorted(list(data['report_date'].unique())))
ax.set(xlabel='report date', ylabel='count')
plt.title("Distribution of classes")
plt.legend(loc='upper right', labels=['Non-churn', 'Churn'])
plt.show()

#%%

pd.concat([pd.Series(data.dtypes, name='data_type'),
          data.describe().transpose(),
          pd.Series(data.nunique(), name='unique_values'),
          pd.Series(data.isnull().sum(), name='missing_values'),
          ], axis=1)

#%% md

### Missing values

#%%

data.sort_values(by='report_date')
sns.heatmap(data.isnull(), cbar=False)
plt.title("Distribution of missing values")
plt.show()

#%% md

### Correlation

#%%

correlations = data.corr()
mask = np.triu(np.ones_like(correlations, dtype=np.bool))

fig, ax = plt.subplots(figsize=(25,25))
sns.heatmap(correlations, mask=mask, vmax=1.0, vmin=-1.0, center=0, fmt='.2f',
           square=True, linewidths=0.01, annot=False, cbar_kws={"shrink": .70})
```

```
plt.title("Correlation between features")
plt.show();
```

## Appendix 2. Code for Feature engineering & Holdout dataset

```
#%% md

## Feature engineering

#%%

missing_value_columns = data.columns[data.isnull().any()].to_list()

#%%

for column in missing_value_columns:
    data[column + '_isnull'] = data[column].isnull().astype(int)

#%%

print("Generated ", len(missing_value_columns), " new columns.")

#%% md

## Holdout set

#%%

holdout = data[data['report_date'] == '2019-01-01']
holdout = holdout.drop(['client_id', 'report_date'], axis=1)
holdout.to_csv('data/raw/holdout.zip', compression='zip', index=False)
print("The size of holdout set is:", holdout.shape)

#%%

train_test = data[data['report_date'] != '2019-01-01']
train_test = train_test.drop(['client_id', 'report_date'], axis=1)

train, test, = train_test_split(train_test, test_size=0.25, random_state=42)
train.to_csv('data/raw/train.zip', compression='zip', index=False)
print("The size of training set is:", train.shape)
test.to_csv('data/raw/test.zip', compression='zip', index=False)
print("The size of testing set is:", train.shape)

#%%

del data
del holdout
del train
del test
del train_test
```

## Appendix 3. Code for Dimensionality Reduction

```
#%% md

## Dimensionality Reduction

#%%

X = pd.read_csv('data/raw/train.zip')
X = X.fillna(0)
y = X.pop('label')

#%%

scaler = StandardScaler()
X = scaler.fit_transform(X)

#%%

pca=PCA(random_state=42)
X_pca = pca.fit_transform(X)
plt.figure(figsize=(10,10))
plt.plot(np.cumsum(pca.explained_variance_ratio_), 'ro-')
plt.title("Cumulative Explained Variance depending on number of PC")
plt.ylabel('Cumulative Explained Variance')
plt.xlabel('number of principal components')
plt.grid()

#%%

pca = PCA(n_components=35)
X_pca = pca.fit_transform(X)

#%%

X_pca.shape
```

Appendix 4. Code for Handling sample imbalance & Data generation

```
#%% md

## Over-sampling & Data Generation

#%% md

### Random over sampling

#%%

X_ros, y_ros = RandomOverSampler(random_state=42).fit_resample(X, y)
np.savez_compressed('data/preprocessed/ros.npz', X_ros=X_ros, y_ros=y_ros)

X_ros_pca, y_ros_pca = RandomOverSampler(random_state=42).fit_resample(X_pca, y)
np.savez_compressed('data/preprocessed/ros_pca.npz', X_ros_pca=X_ros_pca,
y_ros_pca=y_ros_pca)

print("RandomOverSampler dataset label counts: ", sorted(Counter(y_ros).items()))

#%% md

### SMOTE over sampling

#%%

X_smote, y_smote = SMOTE(random_state=42).fit_resample(X, y)
np.savez_compressed('data/preprocessed/smote.npz', X_smote=X_smote,
y_smote=y_smote)

X_smote_pca, y_smote_pca = SMOTE(random_state=42).fit_resample(X_pca, y)
np.savez_compressed('data/preprocessed/smote_pca.npz', X_smote_pca=X_smote_pca,
y_smote_pca=y_smote_pca)

print("SMOTE dataset label counts: ", sorted(Counter(y_smote).items()))

#%% md

### ADASYN over sampling

#%%

X_adasyn, y_adasyn = ADASYN(random_state=42).fit_resample(X, y)
np.savez_compressed('data/preprocessed/adasyn.npz', X_adasyn=X_adasyn,
y_adasyn=y_adasyn)

X_adasyn_pca, y_adasyn_pca = ADASYN(random_state=42).fit_resample(X_pca, y)
np.savez_compressed('data/preprocessed/adasyn_pca.npz', X_adasyn_pca=X_adasyn_pca,
y_adasyn_pca=y_adasyn_pca)

print("ADASYN dataset label counts: ", sorted(Counter(y_adasyn).items()))
```

## Appendix 5. Code for training models

```
#%% md

## Models used and results

#%%

ros = np.load('data/preprocessed/ros.npz')
X_ros = ros.f.X_ros
y_ros = ros.f.y_ros

ros_pca = np.load('data/preprocessed/ros_pca.npz')
X_ros_pca = ros_pca.f.X_ros_pca
y_ros_pca = ros_pca.f.y_ros_pca

smote = np.load('data/preprocessed/smote.npz')
X_smote = smote.f.X_smote
y_smote = smote.f.y_smote

smote_pca = np.load('data/preprocessed/smote_pca.npz')
X_smote_pca = smote_pca.f.X_smote_pca
y_smote_pca = smote_pca.f.y_smote_pca

adasyn = np.load('data/preprocessed/adasyn.npz')
X_adasyn = adasyn.f.X_adasyn
y_adasyn = adasyn.f.y_adasyn

adasyn_pca = np.load('data/preprocessed/adasyn_pca.npz')
X_adasyn_pca = adasyn_pca.f.X_adasyn_pca
y_adasyn_pca = adasyn_pca.f.y_adasyn_pca

#%%

X_test = pd.read_csv('data/raw/test.zip')
X_test = X_test.fillna(0)
y_test = X_test.pop('label')
X_test = scaler.transform(X_test)
X_test_pca = pca.transform(X_test)
```

```
#%%

def make_pr_curve(y_test, y_proba, name):
    precision, recall, _ = precision_recall_curve(y_test, y_proba)
    auc_pr = auc(recall, precision)

    print("AUC PR for", name + ":", auc_pr)
    plt.figure(figsize=(7, 5))
    plt.plot(recall, precision, marker='.', label=name)
    plt.title("AUC PR for " + name)
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.legend()
    plt.show()

#%%

# Helper functions to find maximum F1 score depending on thresholds

def get_classes(y_proba, threshold):
    return [1 if i >= threshold else 0 for i in y_proba]

def get_f1_scores(y_test, y_proba, thresholds):
    return [f1_score(y_test, get_classes(y_proba, i), average='binary') for i in thresholds]

def analyze_f1_scores(model_name, y_test, y_proba):
    thresholds = np.linspace(0.0, 1.0, num=101)
    f1_scores = get_f1_scores(y_test, y_proba, thresholds)
    max_idx = np.argmax(f1_scores)
    print('Maximum F1 score is', f1_scores[max_idx], '. It is achieved with threshold',
thresholds[max_idx])

    plt.figure(figsize=(7, 5))
    plt.plot(thresholds, f1_scores)
    plt.title("F1 score depending on threshold. " + model_name)
    plt.ylabel('F1 score')
    plt.xlabel('Threshold')
    plt.grid()
    plt.show()
```

```
#%%

def evaluate_classifier(clf, X_train, y_train, X_test, y_test, name_clf, name_dataset):
   clf.fit(X_train, y_train)
   print()
   print('__________________________________________________')
   print()
   print(name_clf + " trained on " + name_dataset)
   print('__________________________________________________')
   print()
   y_proba = clf.predict_proba(X_test)[:, 1]
   analyze_f1_scores(name_clf + " trained on " + name_dataset + ". ", y_test, y_proba)
   make_pr_curve(y_test, y_proba, name_clf + " trained on " + name_dataset)

#%%

datasets_X_train = [X_ros,  X_ros_pca,  X_smote, X_smote_pca, X_adasyn, X_adasyn_pca]
datasets_y_train = [y_ros,  y_ros_pca,  y_smote, y_smote_pca, y_adasyn, y_adasyn_pca]
datasets_X_test  = [X_test, X_test_pca, X_test,  X_test_pca,  X_test, X_test_pca]
datasest_name    = ['random oversampled dataset', 'random oversampled dataset with PCA',
               'SMOTE dataset', 'SMOTE dataset with PCA',
               'ADASYN dataset', 'ADASYN dataset with PCA']

#%% md

### KNN

#%%

clf_knn = KNeighborsClassifier(n_jobs=-1, n_neighbors=10)

for i in range(6):
   evaluate_classifier(clf_knn, datasets_X_train[i], datasets_y_train[i], datasets_X_test[i],
y_test, 'KNN classifier', datasest_name[i])
```

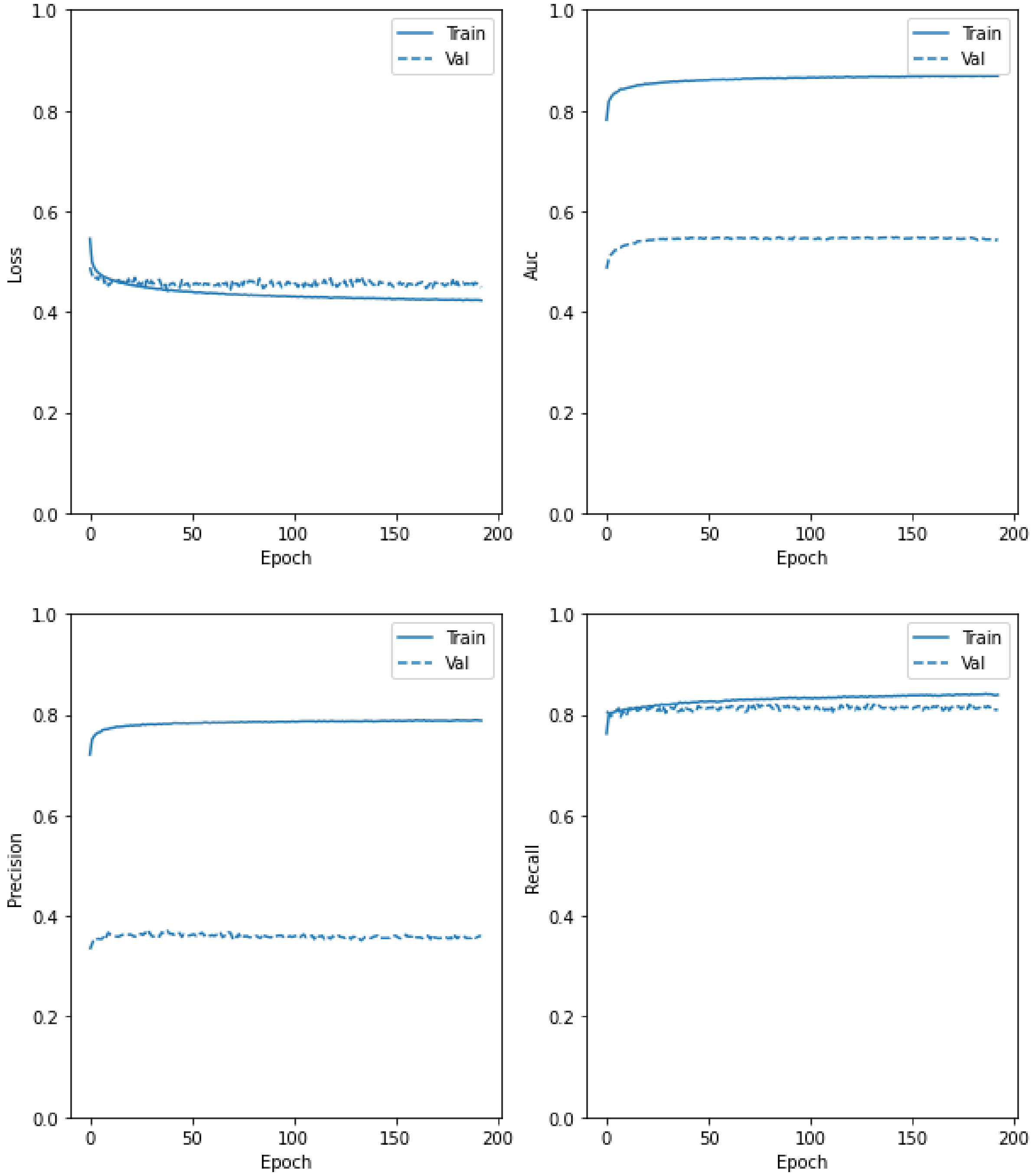
```
#%% md

### Logistic Regression

#%%

clf_lr = LogisticRegression(max_iter=2000)

#%%

for i in range(6):
    evaluate_classifier(clf_lr, datasets_X_train[i], datasets_y_train[i], datasets_X_test[i], y_test, 'LR classifier', datasest_name[i])


#%% md

### Random Forest

#%%

clf_rf = RandomForestClassifier(n_estimators = 100)

#%%

for i in range(6):
    evaluate_classifier(clf_rf, datasets_X_train[i], datasets_y_train[i], datasets_X_test[i], y_test, 'RF classifier', datasest_name[i])


#%% md

### XGBoost

#%%

clf_xgb = xgb.XGBRegressor(n_estimators = 100)

#%%

for i in range(6):
    evaluate_classifier(clf_rf, datasets_X_train[i], datasets_y_train[i], datasets_X_test[i], y_test, 'XGB classifier', datasest_name[i])
```

```
#%%

### ANN

#%%

EPOCHS = 1000
BATCH_SIZE = 2048

#%%

colors = plt.rcParams['axes.prop_cycle'].by_key()['color']

def plot_metrics(history, d_name):
    print("ANN trained on " + d_name)
    plt.figure(figsize=(10, 12))
    metrics =  ['loss', 'auc', 'precision', 'recall']
    for n, metric in enumerate(metrics):
        name = metric.replace("_"," ").capitalize()
        plt.subplot(2,2,n+1)
        plt.plot(history.epoch,  history.history[metric], color=colors[0], label='Train')
        plt.plot(history.epoch, history.history['val_'+metric],
             color=colors[0], linestyle="--", label='Val')
        plt.xlabel('Epoch')
        plt.ylabel(name)
        plt.ylim([0,1])
        plt.legend()
    plt.show()

#%%

def make_model(metrics, input_dim):
    model = Sequential()

    model.add(Dense(128, input_dim=input_dim, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='sigmoid'))

    model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=metrics)

    return model
```

```
#%%

def evaluate_keras_classifier(clfs, X_train, y_train, X_test, y_test, name_clf, name_dataset):
    metrics = [
        keras.metrics.TruePositives(name='tp'),
        keras.metrics.FalsePositives(name='fp'),
        keras.metrics.TrueNegatives(name='tn'),
        keras.metrics.FalseNegatives(name='fn'),
        keras.metrics.BinaryAccuracy(name='accuracy'),
        keras.metrics.Precision(name='precision'),
        keras.metrics.Recall(name='recall'),
        keras.metrics.AUC(name='auc', curve='PR'),
    ]

    clf = make_model(metrics, X_train.shape[1])
    callbacks = [EarlyStopping(monitor='val_auc', patience=100, verbose=0, restore_best_weights=True, mode='max')]

    history = clf.fit(X_train, y_train,
                      validation_data=(X_test, y_test),
                      callbacks=callbacks,
                      epochs=EPOCHS,
                      batch_size=BATCH_SIZE,
                      verbose=0)

    print()
    print('________________________________________')
    print()
    print(name_clf + " trained on " + name_dataset)
    print('________________________________________')
    print()
    y_proba = clf.predict(X_test)
    y_proba = [i[0] for i in y_proba]

    analyze_f1_scores(name_clf + " trained on " + name_dataset + ". ", y_test, y_proba)
    make_pr_curve(y_test, y_proba, name_clf + " trained on " + name_dataset)

    clfs.append(clf)

    plot_metrics(history, name_dataset)
```

```
#%%

clfs = []
for i in range(6):
    evaluate_keras_classifier(clfs, datasets_X_train[i], datasets_y_train[i], datasets_X_test[i], y_test, 'ANN classifier', datasest_name[i])

#%%

final_clf = clfs[0]

#%%

final_clf.save('clf.h5')
```

## Appendix 6. ANN training history



*Training history for ANN classifier trained on random oversampled dataset*