VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS
MASTER'S STUDY PROGRAMME

CHURN PREDICTION IN TELECOMMUNICATION INDUSTRY USING MACHINE LEARNING

Master's thesis

Author: Tautvydas Latvaitis
VU email address: tautvydas.latvaitis@mif.stud.vu.lt
Supervisor: Lekt., Dr. Rimantas Eidukevičius

Vilnius

2022

Abstract

The digitization and rapid growth of technological advancement has changed the way companies do business. With more and more services to chose from, consumer churning has become one of the biggest threats to all companies. In this paper we analyze a machine learning based churn prediction model for a telecommunication service provider. The main aim is to find the best churn prediction model and implement the model on real data. Three machine learning algorithms are compared: Random Forest classifier, XGBoost classifier and Support Vector Machine classifier. The models are evaluated using the metrics, accuracy, precision, recall, and F-scores. The prediction performance is tested on a data set provided by one of the telecommunication companies in Lithuania. It was found that the data is imbalanced with a majority of non-churners, so we also used two balancing techniques, Random Under-Sampling and Random Over-Sampling, to see if they improve the results of our models. We conclude that machine learning is a very useful process for customer churn prediction and that Random Forest and XGBoost models are better at predicting churn than the Support Vector Machine model.

Keywords: Customer churn, churn prediction, machine learning, Random Forest, XGBoost, Support Vector Machine

Santrauka

Spartus technologijų pažangos augimas bei visuotinis skaitmeninimas pakeitė daugumos įmonių verslo modelius. Atsirandant vis daugiau pasirinkimo paslaugų sektoriuje, klientų praradimas tapo viena didžiausių grėsmių įmonėms, kurios konkuruoja tarpusavyje. Šiame darbe analizuojame mašininiu mokymusi pagrįstus klientų praradimo prognozavimo modelius. Darbo pagrindinis tikslas yra surasti geriausią klientų praradimo prognozavimo modelį telekomunikacijų srityje. Lyginami trys mašininio mokymosi algoritmai: Atsitiktinio miško klasifikatorius, "XGBoost" klasifikatorius ir Atraminių vektorių klasifikatorius. Modeliai vertinami pagal kelias metrikas, teisingumą, tikslumą, jautrumą bei F matus. Prognozės našumas tikrinamas pagal duomenų rinkinį, kurį pateikė viena iš telekomunikacijų įmonių Lietuvoje. Nustatyta, kad duomenys yra nesubalansuoti, nes dauguma jų vis dar esami klientai, todėl taip pat naudojome du balansavimo būdus – atsitiktinę per mažą atranką ir atsitiktinę perteklinę atranką, kad pamatytume, ar jie pagerina mūsų modelių rezultatus. Kyla išvada, jog mašininis mokymasis yra labai naudingas procesas numatant klientų praradimą ir kad Atsitiktinio Miško ir "XGBoost" modeliai geriau prognozuoja klientų praradimą nei Atraminių Vektorių klasifikatorius.

Raktiniai žodžiai: Klientų praradimas, praradimo prognozavimas, mašininis mokymasis, Atsitiktinis Miškas, XGBoost, Atraminiai Vektoriai

Contents

1 Introduction

In any business, in order to have great success and a growing revenue, the customer base plays a crucial role. Knowing this more and more companies become aware that in order to succeed they need to gain their customers' satisfaction and loyalty. Nowadays, the digitization and the advancement of the world has led to new business models and companies around the world have no choice but to adapt, if they want to survive. Customer Relationship Management (CRM) refers to the principles, practices and policies that an organization follows when interacting with its customers. From an organization's perspective, this relationship includes direct interactions with customers, forecasting, and analysis of customer trends and behaviors. Ultimately, CRM serves to improve the overall customer experience [5]. CRM is intensively used in many different fields: banking, retail market, insurance, telecommunications, etc. The main objective of CRM is analyzed in this paper - customer retention. The importance of preservation of customers is evident: of course opinions differ about the ratio of Customer Acquisition Costs (CAC) and Customer Retention Costs (CRC), with figures fluctuating from 2 to 30 times, depending on the sector examined. For the meantime, a consensus has been made that CAC should be higher than CRC [1].

Customer churn occurs when customers stop doing business with a company or service. Churn is also known as customer attrition and is an important metric for many businesses around the world. Companies all around the world identify churning as a big problem and a massive loss, because they have already spent a lot of money and invested numerous hours in attracting the clients in the first place. Usually it is really hard to define the sole reason why a customer has decided to leave as there might be a dozen different motives behind this decision. The telecommunication industry, where our focus will be targeted at, is affected by customer churn to a great extent. It always faces the threat of financial losses from potential churn. It has become so much easier to find alternative service providers in the era of the internet and all of the information being online. To minimize this possible customer loss, firms are starting to invest enormous amounts of money in customer churn prediction, hoping to be able to apply preventative measures even before the customer comes up with an idea of searching for a better deal elsewhere. Nevertheless, all of this is not free and also not all customers are actively thinking of churning, thus telecommunication companies should not put all of their focus on their entire client base. Identifying which customers are more inclined to leave and targeting those precise customers is key to avoiding preventable use of costs and resources. Therefore, an efficient churn prediction model is one of the best ways for a company to take care of their client retention. Such a model not only secures revenue but also provides guidance to the management to target potential churners by reducing the market relevant weaknesses.

As stated above, nowadays more than ever customers are motivated to churn and especially in a subscription-based services such as telecommunications. In this study different machine learning (ML) techniques are used to find the most effective and efficient model for customer churn prediction by: 1) defining customer churn in a mobile telephony service provider; 2) taking a look at the current customer churn prediction models; 3) trying to implement a working model to predict customer churn; 4) testing the model on real data of customer of one of the leading telephony service providers in Lithuania. In this experiment, the prediction performance is tested on a data set provided by one of the telecommunication companies in Lithuania but due to sensitive and classified information the company name will not be mentioned in the study. To answer the question if a good churn prediction model can be found for our data set, we examine several aspects. Of particular importance is to determine an appropriate model for binary churn prediction, investigate which algorithm is most appropriate, and examine data processing issues including balancing of the data set. This is particularly important in the prediction of customer churn, where non-churners tend to dominate. There are a few algorithms to choose from for prediction tasks. Since we are working with a classification problem, we reject almost all algorithms used for regression problems. The choice of algorithms is based on previous studies, we chose Random Forest and XGBoost algorithms but compared them with Support Vector Machine (SVM), which can be used for both classification and regression analysis. We haven't found much comparison between these two ensemble learners and SVM, so wanted to perform this evaluation and see which algorithm does best.

The following chapters will include: In chapter 2, the literature review is presented. This includes some theoretical concepts that are related to machine learning in churn prediction. Also, we take a look at different previously used methodologies for predicting churn, investigating and comparing the work of other authors. Chapter 3 covers the research methodology, with a broader overview of the selected machine learning algorithms and the data used for this study. In chapter 4 the focus is on the results of the study and the performance of the different models is emphasized. We follow this up with a discussion of our findings and with the final conclusions of the study in chapter 5.

2 Literature review

To have success and a growing revenue all businesses rely on their customer base. Customer satisfaction and engagement should be a priority for anyone who wishes to succeed in the era of digitization and advancement of the world. Customer Relationship Management (CRM) can be considered as a strategy of a company, aimed at reducing costs, increasing profitability, and improving customer relationships by offering the right product or service to the right customer [2].

CRM is often associated with the knowledge of the customer [3]. CRM can be performed in different ways, companies might choose various processes or methods, depending on specifics for each industry [2, 4]. Nevertheless the main objective of CRM stays constant - acquiring customers, studying them, discovering the most beneficial ways of serving them and the combining all this knowledge in order to retain them [2]. Companies may achieve this with CRM systems. The systems have a purpose of not only enabling communication with the customer, but also to analyze the stored customer data in order to get to know the customer base better [4]. Ultimately, CRM serves to improve the overall customer experience. It is evident that the preservation of customers is very important: there are some different opinions, but, depending on the sector examined, a consensus has been made that the of Customer Acquisition Costs (CAC) should be higher than Customer Retention Costs (CRC), with the ratio fluctuating from 2 to 30 times [1].

Customer churn is a term used to define customer attrition - the loss of customers by a company. Nowadays more than ever customers are motivated to churn and especially in a subscription-based services such as telecommunications. Customer churn management, the idea of identifying customers who might churn, comes into play here. It has become one of the main strategies for companies to survive within a specific industry [9]. With successful customer churn management potential churners can be targeted in advance with proactive campaigns of retention incentives [10].

Hadden et al. [9] have said, that there are two main types of churners: voluntary and non-voluntary. The company itself removes non-voluntary churners from its customer base, so these churners are easy to predict and bring no real value to prediction models. On the other hand, voluntary churners come to a decision to leave the service provider and are harder to predict. A number of different things might force a customer to churn and the company cannot do anything about it unless customer churn management is implemented. Hung et al. [10] state that churn management is considered a part of CRM and from a business perspective, churn management includes two main tasks. Firstly, trying to predict customers who might churn. Secondly, coming up with the best retention strategies that would benefit the organization the most.

Machine Learning (ML) nowadays plays a huge part in customer churn management by developing several different algorithms to predict churn. Machine learning is the study of computer algorithms that can improve automatically through experience and by the use of data [11]. ML algorithms build models from sample data, which is known as training data, and from it makes predictions or decisions without being programmed to do so, this is known as supervised learning. Supervised learning algorithms use data sets that contain inputs and desired outputs to build mathematical models [12]. This type of learning tries to find patterns in the data set and the identifies outputs of unseen instances [13]. Supervised learning can be then further split into classification and regression. Classification is used when the outputs are categorical, for example churn or not churn. In a regression problem the outputs are continuous values, such as the days until a potential customer might churn [14]. In this study we focus on classification.

A common ML technique is ensemble learning, which combines different learning models (base learners) to combine their outputs into one classifier [15]. A group of weaker machine learning models are implemented together as one stronger model and make very accurate predictions in ensemble learning. This ML technique is quickly becoming a first choice for machine learning models in the world of data science [16].

Different statistical and machine-learning techniques are used to customer churn. Many attempts have been made to compare and benchmark the used techniques for churn prediction.

Khan et al. [17] compares Decision Trees, Logistic Regression and Neural Network models. In the study the authors had found that Neural Network outperformed the other two models.

Pamina et al. [18] analyzed three well known classifiers: K Nearest Neighbout (KNN), Random Forest and XGBoost. The comparison of evaluation metrics such as accuracy score and F score were calculated. The results have shown that the XGboost classifier held the highest accuracy and F score while the KNN model yielded the lowest scores.

Another interesting study was performed by Hanif [19] in 2020. He compared the LogReg and the XGBoost algorithms for predicting customer churn. XGBoost algorithm has been proven to give better prediction compared with LogReg algorithm based on its prediction accuracy, specificity, sensitivity and ROC curve. XGBoost model sensitivity remains high, while LogReg model sensitivity remains very low, indicating that XGBoost can handle imbalanced-classes data better than LogReg does.

Some studies go even further by comparing as much as 8 churn predictions models. In 2018 Sabbeh [20] compared logistic regression, decision trees, Naive Bayes, Support Vector Machine, k Nearest Neighbour, random forest, artificial neural network and linear discriminant analysis models. In this big comparison of several ML classifiers, the Random Forest classifier showed the best results.

Overall, according to most studies Random Forest and XGBoost algorithms seem to be best for a classification problem such as predicting customer churn. So, four our study we chose to use these two ensemble learners, but also compare them to the Support Vector Machine model as we have not found many research done specifically on these three classifiers.

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is now the leading machine learning library for regression, classification, and ranking problems[21].

3 Methodology

3.1 Random Forest classifier

Random Forest is an ensemble machine learning method that is built on decision trees. So just as trees, random forests can be used for classification and decision problems. Even though trees are easy to understand and implement, they are not that effective in all situations, because their nature is inflexible. Randoms forests assist with this problem by reducing errors due to variance and bias. They achieve this by averaging out predictions from a group of decision trees, making them really powerful models. The basic method of Random Forests was first proposed by Ho in 1995 [6]. Ho confirmed that forests of trees with angled hyperplanes can achieve higher accuracy as they grow without suffering from over-training. For that to happen the forests have to be randomly restricted to be sensitive to only selected feature dimensions. Random Forests were properly introduced in a paper by Leo Breiman [7]. The paper described a way of building a forest of unrelated trees using a Decision tree learning like procedure, which was combined with bagging and randomized node optimization. A few ingredients are combined in the paper, which form the basis of the modern practice of random forests [2].

A Random Forest is basically a collection of many decision trees, that averages out the results of all decision trees and lowers the variance. Below, in Figure 1 the basic principle of Random Forest is shown. To explain it simply, all of the decision tress return a label - churn or no churn and then the majority is found and assigned to the final output.

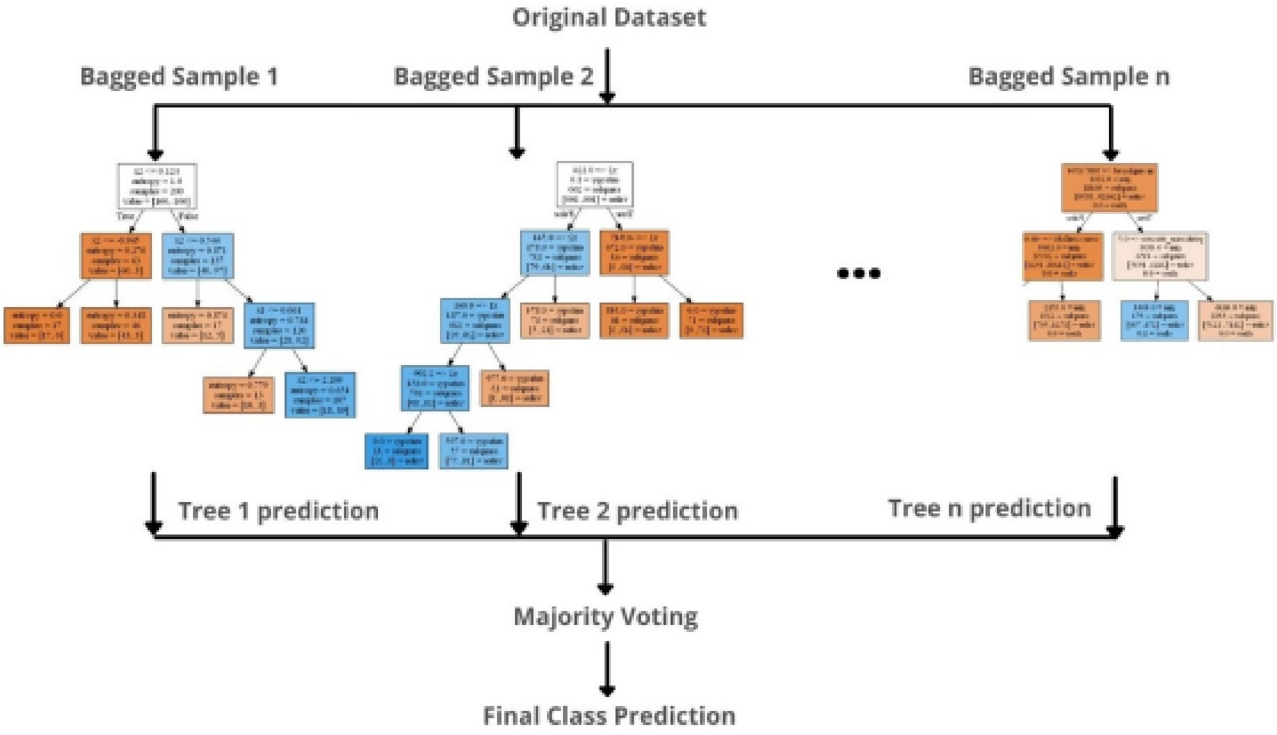


Figure 1: Random Forest

The method that makes it possible to build many different trees from one data set is called bagging. Bagging stands for bootstrap aggregating was introduced by Breiman in 1996 [8] to reduce the variance of a predictor. Bootstrapping is an ensemble method that samples random rows from the original data set with replacement. Ensemble means combining several machine learning models together in order to produce a more accurate prediction than any single model would. When dealing with a data set of m samples, n $\geq$ m samples are taken from this set. A selected row is not stripped from the original set but remains unchanged which means that any row can be randomly selected over and over. A new data set with a subset of the original rows is created, with some of them possibly duplicated. This is done as many times as there are trees created, generally, more trees result in a more accurate prediction.

3.2 Support Vector Machine (SVM) classifier

Support Vector Machine or SVM is one of the most popular supervised learning algorithms used for both classification and regression problems. However, it is primarily used for classification problems in machine learning.

The goal of the SVM algorithm is to create the best line or decision boundary that divides the n-dimensional space into classes so that we can easily assign the new data point to the correct category in the future. This best decision boundary is called a hyperplane. SVM selects the extreme points/vectors that help to create the hyperplane. These extreme cases are called support vectors, and hence the algorithm is also called Support Vector Machine. In Figure 2 we see an example where two different categories are classified using a decision boundary or hyperplane.

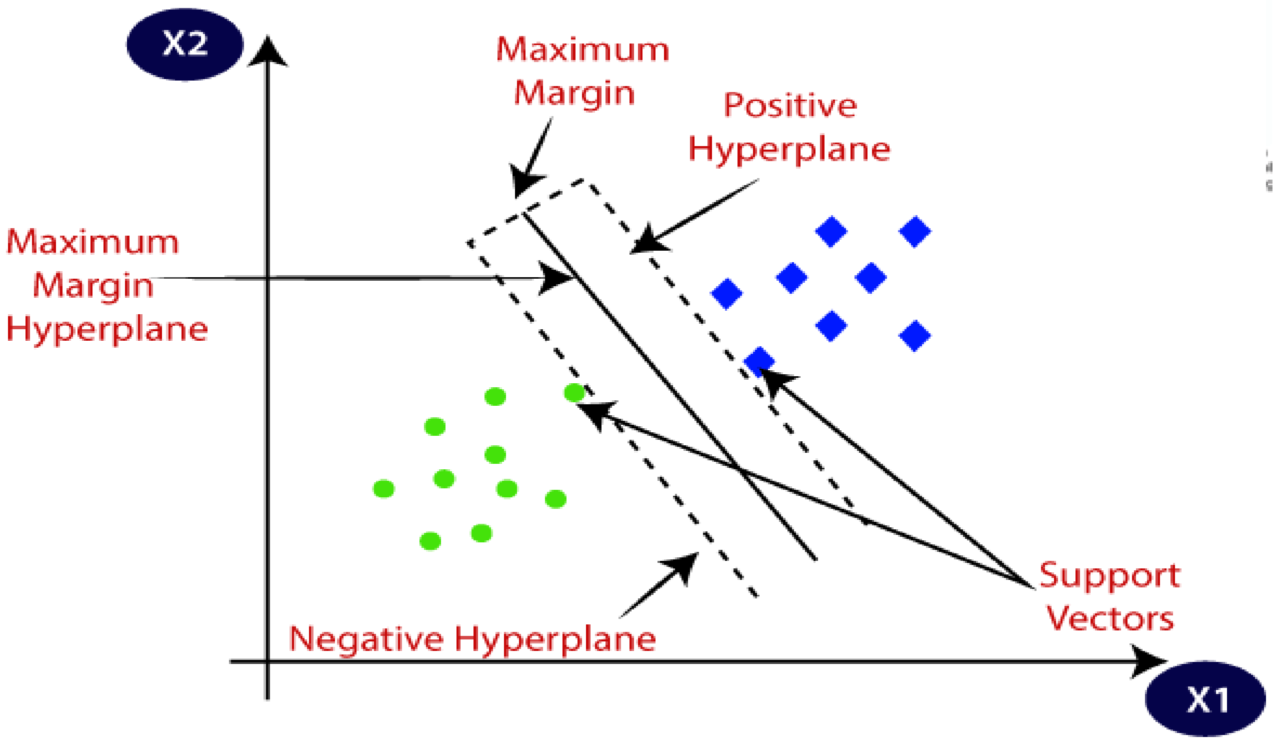


Figure 2: An Example of a Support Vector Machine

SVM can divided into two types:

Linear SVM is used for linearly separable data, which means if a data set can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM is used for non-linearly separated data, which means if a data set cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

3.3 XGBoost classifier

XGBoost or eXtreme Gradient Boosting is a relatively new type of boosting algorithm. In machine learning, boosting is an ensemble meta-algorithm primarily used to reduce bias and also variance in supervised learning, and a family of machine learning algorithms that transform weak learners into strong ones. In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals. XGBoost is one of the most accurate machine learning algorithms, also it is quite fast and is not prone to overfitting. Below we take a look at the features of the model in more detail:

An objective function with training loss and regularization is taken:

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{1}$$

Then the tree parameters have to be found. In order to do that we need to find functions $\mathrm{f_i}$, where each of them shows the structure of a tree. This is very challenging to do for all the trees at once, so a strategy is used: with the addition of a new tree all previous errors are corrected. Then the value of the prediction in step t is written down as $\mathrm{y_i}^{(0)}$ and obtained by:

$$\begin{aligned} \hat{y_i}^{(0)} &= 0 \\ \hat{y_i}^{(1)} = f_1(x_i) &= \hat{y_i}^{(0)} + f_1(x_i) \\ \hat{y_i}^{(2)} = f_1(x_i) + f_2(x_i) &= \hat{y_i}^{(0)} + f_2(x_i) \\ &\dots \\ \hat{y_i}^{(t)} = \sum_{k=1}^{i} f_k(x_i) &= \hat{y_i}^{(t-1)} + f_t(x_i) \end{aligned} \tag{2}$$

Then a tree for each step has to be chosen, a tree that will optimize the objective function:

$$obj^t = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i) = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{t-1} + f_t(x_i) + \Omega(f_t) + const. \tag{3}$$

The next important step is regularization - a regression method, which decreases the possibility of overfitting by punishing the more complex trees. Thus, the complexity of the trees needs to be found.

Firstly, the decision tree formula:

$$f_t(x) = w_q(x), w \in R^T, q : R^d \to \{1, 2, ..., T\}, \tag{4}$$

here w - a vector wit the weights of the leafs, q - a function, which assigns values to each leaf and T - the number of leaves. Then the complexity of the XGBoost model is expressed by:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2. \tag{5}$$

3.4 Model Performance measurements

3.4.1 Confusion Matrix

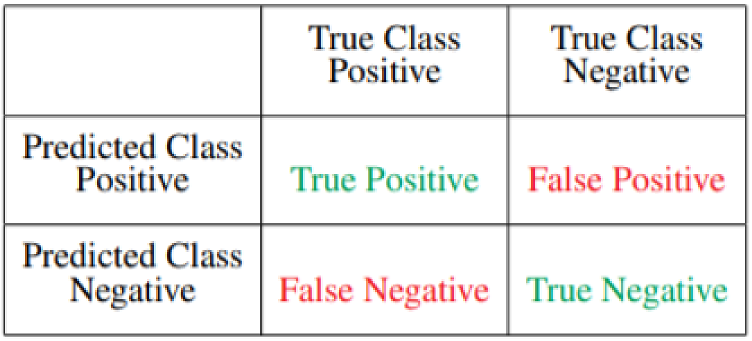
A table that is used to describe the results of a classification model is called a confusion matrix. It contains the number of instances of the predicted classes vs real classes. Figure 3 shows the general concept of a confusion matrix.

True Positive (TP) - predicted positive and actually positive, correct prediction.

False Positive (FP) - predicted positive but actually negative, incorrect prediction.

False Negative (FN) - predicted negative but actually positive, incorrect prediction.

True Negative (TN) - predicted negative and actually negative, correct prediction.



	True Class Positive	True Class Negative
Predicted Class Positive	True Positive	False Positive
Predicted Class Negative	False Negative	True Negative

Figure 3: Confusion Matrix

3.4.2 Accuracy

Accuracy is the simplest, most intuitive and therefore most popular metric of model evaluation. Accuracy can be any value between 0 and 1 with 1 being the aspired result. The formula for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

Unfortunately, accuracy is not always a reliable measure. If the data is strongly imbalanced, accuracy might give a very misleading result.

3.4.3 Precision

Precision is the ratio between True Positive and all Positive predictions. Precision is especially important when false positives need to be avoided at any cost. The formula for precision:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

3.4.4 Recall

Recall measures the True Positive Rate - accurately identified Positives. It gives the ratio of how many positive instances were correctly labelled as true out of all true positives. Recall is often used when false negatives are extremely undesirable.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

3.4.5 F1 and F2 Scores

Both precision and recall are useful, but usually a balance is needed between both of them. The F1 score is used in order to get an averaged evaluation of both metrics. The F1 score is described as a harmonic mean of the Precision and Recall measures where 1 is considered the best and 0 the worst score.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

The F2 score is an example of the Fbeta-measure with a beta value of 2. It has the effect of lowering the importance of precision and increase the importance of recall. If maximizing precision minimizes false positives, and maximizing recall minimizes false negatives, then the F2 score puts more attention on minimizing false negatives than minimizing false positives. The F2 score is calculated as follows:

$$F2Score = \frac{5 * Precision * Recall}{4 * Precision + Recall} \tag{10}$$

3.4.6 k-Fold Cross-Validation

Another popular validation algorithm is k-fold Cross-Validation (CV). Data are split into equal-size k subsets (folds). A single subset is used for error estimation, and the rest are used for fitting the model. Leave-one-out CV can be considered a special case of k-fold CV in which the number of folds is equal to the sample size . In this paper, 10-fold CV is used.

4 Analysis and Results

In this study we used three different machine learning algorithms to see which one predicts customer churn best. The choice of the algorithms was based on previous studies. This section is dedicated to present the analysis and the results for each of the three classifiers used: Random Forest, XGBoost and the Support Vector Machine.

4.1 Data preparation and visualization

In order to perform the analysis historical and live data from one of the Lithuanian telecommunication companies was used. The raw data was extracted via Apache Hadoop using the Structured Query Language and then worked on using the Python programming language. The data set includes information about customers who have joined the company since the start of 2015, including churners and non-churners. There are more than 155000 rows of data and they all have the following attributes:

- churn flag - whether the customer has churned or not.
- msisdn - phone number used as a unique customer ID.
- sex - the gender of the customer.
- age group - to which age group the customer belongs to.
- aging - how long they have been a customer for.
- contract duration - what is the duration of the current contract of the customer.
- months to contract end - how many months left until the end of the contract.
- price plan price - the price of the data plan.
- data gb - the amount of gigabytes on the plan.
- avg data usage - the average gigabyte usage of the customer.
- consumption category - the category of the data usage. Under users use up to 20% of their data plan per month, right users use from 20% to 80% of their data plan per month and over users use more than 80% of their data plan per month.
- lifecycle - the current lifecycle of the customer in the company.
- billing device - whether the customer also pays for a device next to his data plan.
- selfcare - whether the customer uses online self care.
- sim device type - the device type in which the SIM card is placed.
- avg bill - the amount the customer pays on average per month.
- total bill - the amount the customer has paid in total so far.

- voice count - the amount of Mobile Voice SIM cards the customer has.
- mbb count - the amount of Mobile Broad Band SIM cards the customer has.

In Figure 4 we can see the churn rate for the data set used. It is apparent that the data set is unbalanced and that balancing could be applied to evaluate if it helps with getting better results from the machine learning algorithms.

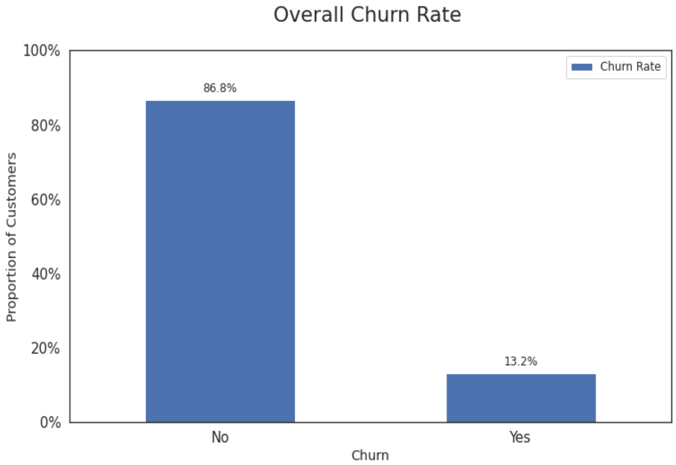


Figure 4: The churn rate of the customers.

Figure 5 shows the values of the numerical columns in the data set and a few observations can be made here. The majority of the clients have been with the company for 3 years or less. The most common contract duration is 24 and 18 months, with others having very low numbers. The price of the plan varies mostly between 5 and 16 euros. Almost every client also has a Mobile Voice SIM card. Below that, in Figure 6 we see several visualizations of customer counts by different categorical variables, which include age group, data bucket size and consumption category.

For the data to be usable in the models, we had to make some changes to the data. Since more than half of the variables were categorical, i.e. non-numeric values, they have been dealt with Label Encoding and one-hot encoding using get dummies method in the pandas library in Python [23]. The encoding creates new columns from categorical variables and the values are represented by ones and zeros, where one indicates present and zero indicates non-present.

Moving forward, we split the data to a training set that is used for training and fitting of the model and a test set used for evaluation. The proportions used were 80% and 20% respectively. Various studies might use different splits but there is no clear evidence that one proportion is better than the other [22].

Since there are still some numerical variables that are not encoded the last step before running our models was performed - normalization of the data set to put the data in the same scale.

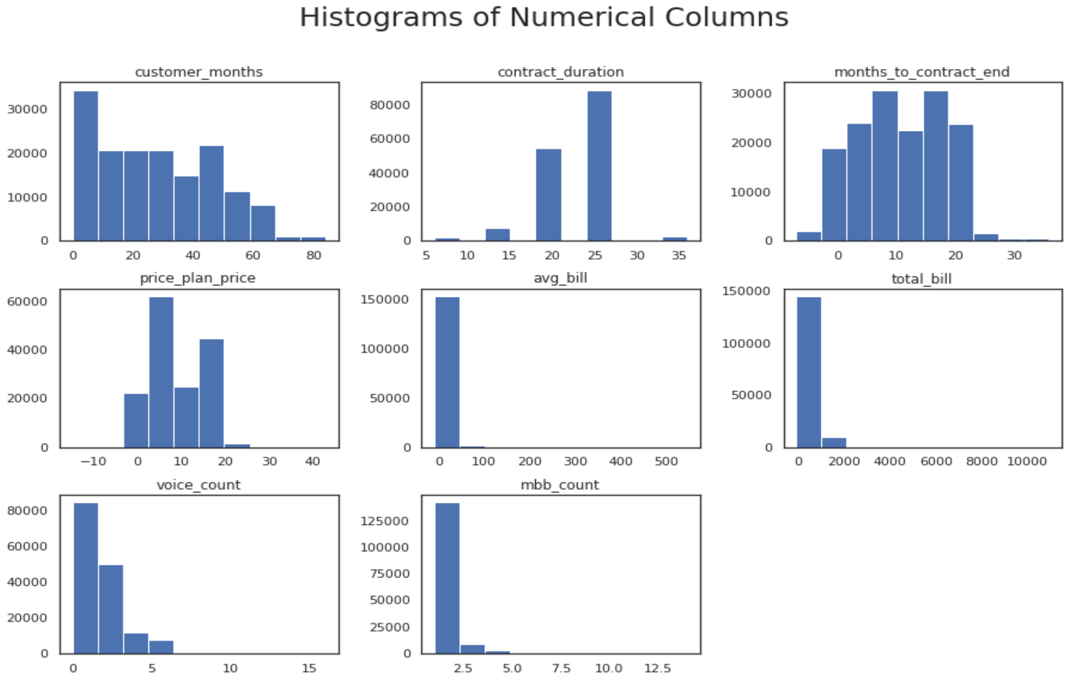


Figure 5: Histograms of numerical columns.

4.2 Identification of the optimal number of trees for Random Forest model

The Random Forest classifier by default uses 100 trees in the forest [24], but in reality we do not know what is the optimal number of decision trees to compose the best random forest. Therefore we have written a loop in Python that iterates 100 times to figure out the optimal number of decision trees for our data set. In Figure 7 we can see that the optimal number of tress for the Random Forest model is 80.

4.3 Evaluation of the models

Evaluating the models is a very important part of the machine learning process. The performance of the classifiers is measured by different indicators in order to select the best algorithm from the ones used. We use accuracy, precision, recall, F1 score and F2 score to evaluate the models. The focus is on the F1 and F2 scores because they show the balance between precision and recall. The test set is used for the evaluation process, that is why it is important to split the data into a training and a test set to actually see if the model performs well on values not used during training.

Below, in Figure 8 we can see how each of the three classifiers, Random Forest, XGBoost and SVM performed. The results show that all three models are capable of correctly classifying the test data with high accuracy. SVM had an accuracy of 96%, while both Random Forest and XGBoost had an accuracy of 98%. As mentioned earlier accuracy might not be the most reliable metric to look at. For us the most important thing is to predict actual churners with high accuracy and minimize False Negatives.

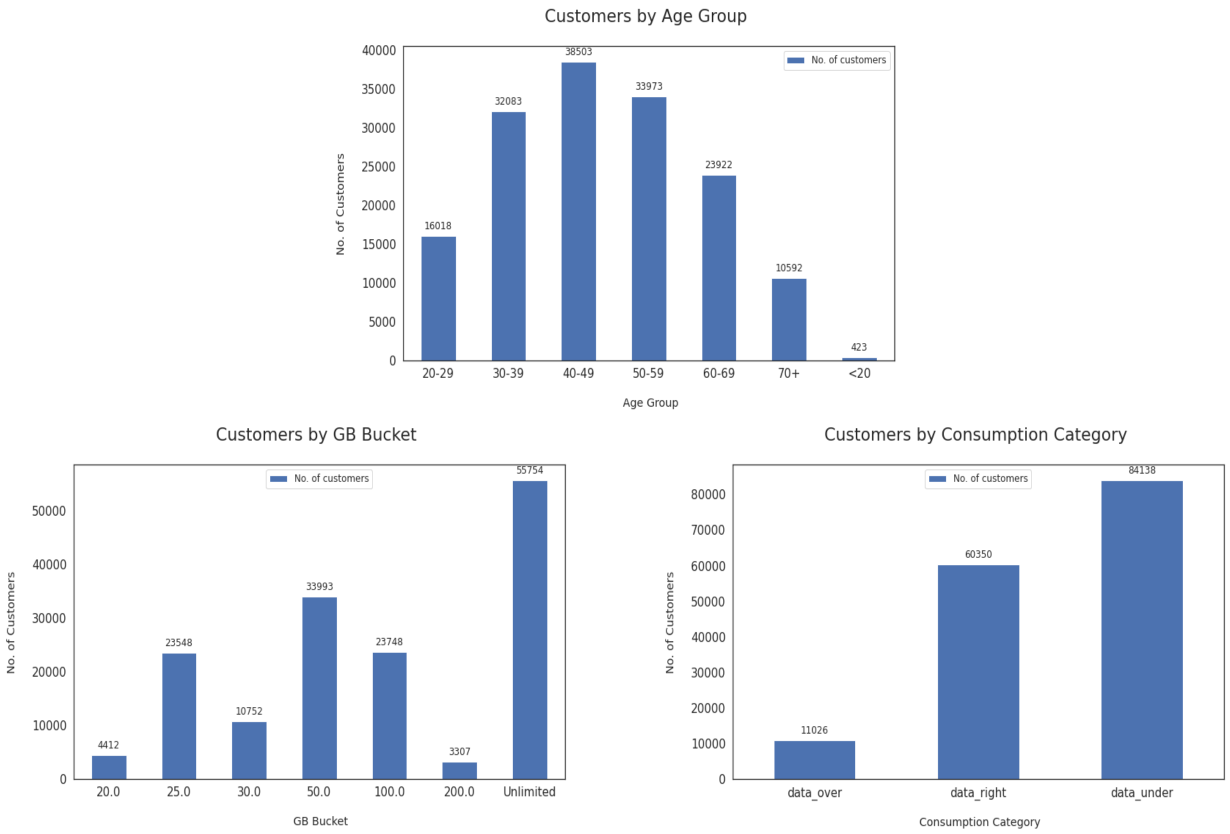


Figure 6: Visualization of some of the categorical variables.

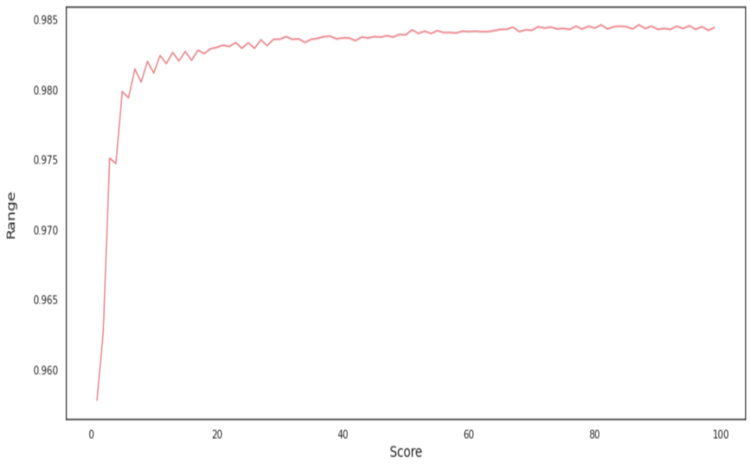
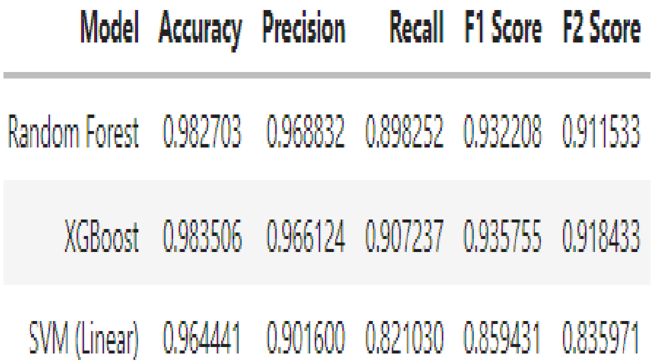


Figure 7: Optimal number of trees for Random Forest Model

By taking look at other metrics we see that Random Forest and XGboost, having very similar results, outperformed the SVM model. For Random Forest classifier the precision is 97%, recall is 90%, F1 score is 93% and the F2 score is 91%. XGBoost classifier has achieved a precision of 97%, recall of 91%, F1 score of 94% and the F2 score is 92%. The XGBoost model very slightly outperformed the Random Forest model. Having obtained such similar results for both Random Forest and XGBoost, we have decided to also try a couple of techniques to try and balance our original data set.



Model	Accuracy	Precision	Recall	F1 Score	F2 Score
Random Forest	0.982703	0.968832	0.898252	0.932208	0.911533
XGBoost	0.983506	0.966124	0.907237	0.935755	0.918433
SVM (Linear)	0.964441	0.901600	0.821030	0.859431	0.835971

Figure 8: Comparison of model metrics

One of the techniques is random under-sampling, which is a fast and easy way to balance the data by randomly selecting a subset of data for the targeted class. Under-sample the majority class by randomly picking samples. For the Random forest model we repeated the process of identifying the optimal number of trees and found that 60 (Figure 9) was the optimal number. After that we ran the two remaining models again, this time on our under-sampled data. We can see the results in Figure 10. For Random Forest classifier the precision was 88%, recall was 96%, F1 score was 92% and the F2 score was 94%. XGBoost classifier has achieved a precision of 89%, recall of 96%, F1 score of 93% and the F2 score is 95%. Under-sampling decreased the precision quite considerably, but increased the recall for both models. We are looking for a higher recall percentage, but the considerable decline in precision also lowered the F1 and F2 scores, so under-sampling might not be the way to go.

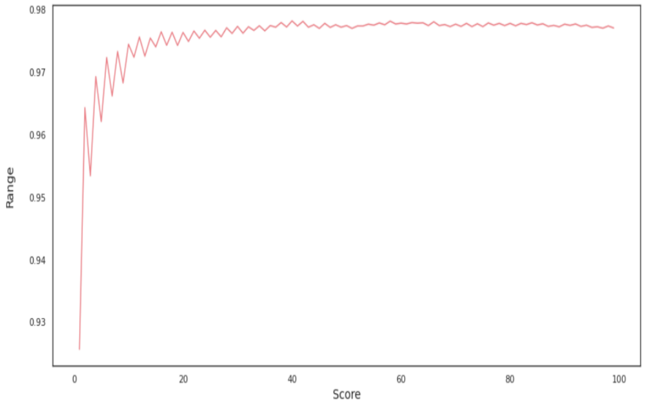
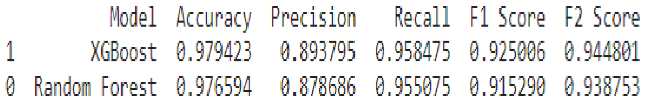


Figure 9: Optimal number of trees for Random Forest Model after Under-Sampling



```
           Model  Accuracy  Precision    Recall  F1 Score  F2 Score
1        XGBoost  0.979423   0.893795  0.958475  0.925006  0.944801
0  Random Forest  0.976594   0.878686  0.955075  0.915290  0.938753
```

Figure 10: Comparison of model metrics after Under-Sampling

Another way to fight imbalance in the data is to generate new samples in the minority class (churners in our case). For this we used the random over-sampling method. For the Random forest model we once again identified the optimal number of trees and found that 75 (Figure 11) was the optimal number. After that we ran the two remaining models again, this time on our over-sampled data. We can see the results in Figure 12. For Random Forest classifier the precision was 96%, recall was 92%, F1 score was 94% and the F2 score was 93%. XGBoost classifier has achieved a precision of 90%, recall of 96%, F1 score of 93% and the F2 score is 95%. Once again we got very similar results for both models and compared to the original and under-sampled data.

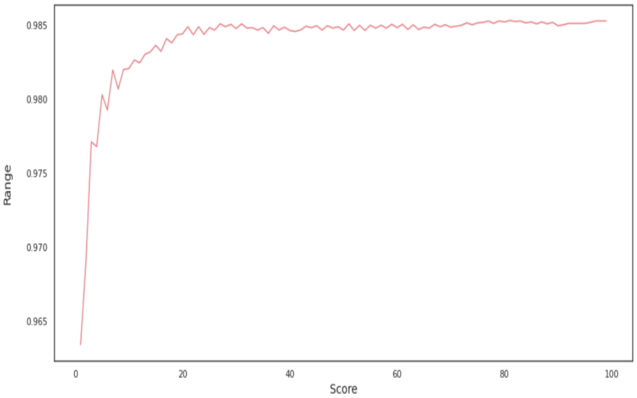
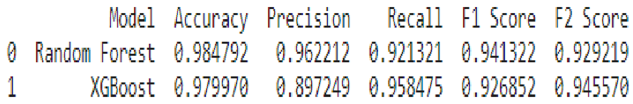


Figure 11: Optimal number of trees for Random Forest Model after Over-Sampling



```
           Model  Accuracy  Precision    Recall  F1 Score  F2 Score
0  Random Forest  0.984792   0.962212  0.921321  0.941322  0.929219
1        XGBoost  0.979970   0.897249  0.958475  0.926852  0.945570
```

Figure 12: Comparison of model metrics after Over-Sampling

In our case under and over sampling did not make a considerable difference to the results, so even though we could use both balancing methods, the original data set is good enough for quite high churn prediction results. The XGBoost classifier very slightly outperformed the Random Forest classifier, but both are good at predicting customer churn. Let's fit the selected model (XGBoost in this case) on the training data set and evaluate the results.

4.4 k-Fold Cross-Validation for XGBoost

Model evaluation is usually done by the technique of 'K-fold cross-validation', which primarily helps us to fix the variance. The variance problem occurs when we get good accuracy when we run the model on one training set and one test set, but the accuracy looks different when the model is run on another test set. To address the variance problem, k-fold cross-validation basically splits the training data set into 10 folds and trains the model on 9 folds before testing it on the test fold. This gives us the flexibility to train our model on all ten combinations of 9 folds, which gives us enough room to resolve the variance. The k-fold Cross Validation displayed that the accuracy while running the model on any test set would be between 92% and 98%.

4.5 Results Visualization on a Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classifier on a set of test data for which the true values are known. As seen in Figure 13, the Confusion Matrix shows that we have 26847 + 3737 correct predictions and 379 + 138 incorrect predictions. We can calculate the accuracy rate using a simple formula: Accuracy rate = number of correct predictions/ total predictions * 100. So we have achieved an accuracy rate of 98%, which is an outstanding result, signalling characteristics of a good model.

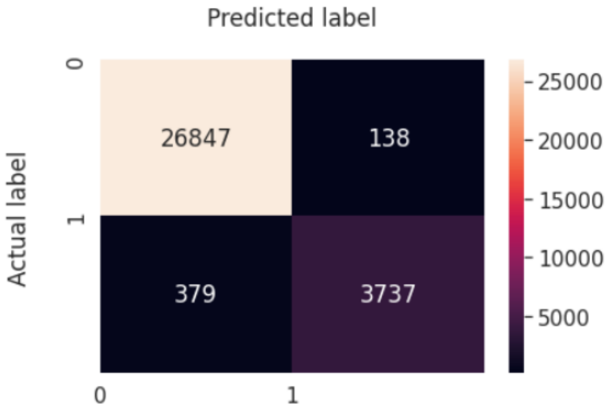


Figure 13: Confusion Matrix

4.6 ROC Graph for the XGBoost classifier

It is good to re-evaluate the model with a ROC graph. The ROC Graph shows us the ability of a model to discriminate between classes based on the AUC mean. The orange line represents the ROC curve of a random classifier, while a good classifier tries to stay as far away from this line as possible. As can be seen in Figure 14 below, the XGBoost model has a much higher AUC value.

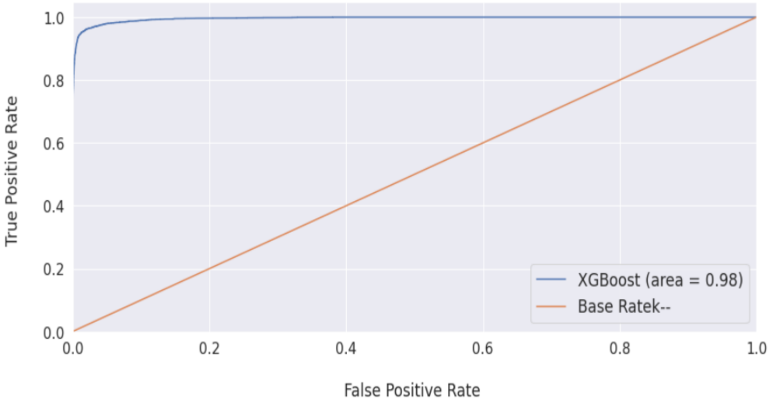
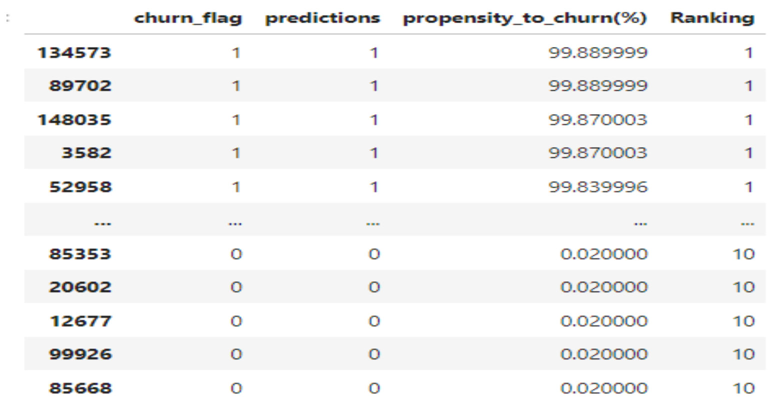


Figure 14: ROC Graph

4.7 Final results with the propensity to churn

Unpredictability and risk are the close companions of any predictive model. Therefore, it is always a good practice in the real world to produce a propensity score in addition to an absolute predicted outcome. Instead of just retrieving a binary estimated target result (0 or 1), each "customer ID" could be given an additional level of Propensity Score indicating the percentage probability of performing the desired action. In Figure 15 we see a snippet from the final results table with the percentage of the propensity to churn for each customer.



	churn_flag	predictions	propensity_to_churn(%)	Ranking
134573	1	1	99.889999	1
89702	1	1	99.889999	1
148035	1	1	99.870003	1
3582	1	1	99.870003	1
52958	1	1	99.839996	1
...	...	...	...	...
85353	0	0	0.020000	10
20602	0	0	0.020000	10
12677	0	0	0.020000	10
99926	0	0	0.020000	10
85668	0	0	0.020000	10

Figure 15: The propensity to churn for each customer

5 Conclusions

In the telecommunication industry customer churn management, in particular, churn prediction has become crucial in the past few years. Companies deal with the problem of recognizing customers with a high probability to churn in the close future. The absence of an accurate model that monitors clients' behavior is one of the main reasons why it is so hard to differentiate churners from loyal customers. Sending all of the customer base incentives is clearly a waste of money, so a need for a churn prediction model is apparent, because this can save a lot of time and money.

The results gathered in the study have shown that customer churning can be predicted with quite a high precision using machine learning. Moreover, the results indicate that ensemble learners, using boosting and bagging, improved the performance of our churn prediction model, relative to the Support Vector Machine model. We cannot only look at the accuracy of the classifiers to assume that the performance is good. Alternative evaluation metrics commonly used in the literature are precision, recall, and F1-F2 scores, all of which indicate the performance of the classifier for a given target class, in our case, the target class is churn. The results show that all three models are pretty good at predicting churn on our data set, nevertheless Random Forest and XGBoost outperformed SVM. SVM the obtained precision of 90 percent, recall of 82 percent, F1-score of 86 percent and F2-score of 84 percent. The Random Forest classifier scores a precision of 97 percent, the recall of 90 percent, which indicates that the model is able to predict actual churners with a high accuracy. Since the F1-score takes both these values in consideration, it is also pretty high. For XGBoost the precision of the classifier is 97 percent, recall is 90 percent, F1-score is 94 percent and F2-score is 92 percent. Based on this the results indicate that the XGBoost classifier is, very slightly, the best performing classifier in the study.

We have studied different aspects related to customer churn prediction using machine learning. By studying these aspects, which are captured in the machine learning process, we contribute to building a complete understanding on how machine learning can be used for churn prediction.

Having gathered the results the company can use the model for a better customer retention initiative, but the prediction is only the first part of customer churn management. For future work it would be interesting to investigate, based on the variables, what measures could be taken related to retention strategies and how the organization should actively act towards customers that are predicted to churn. In addition, it would be of interest what other algorithms could be used and how they could be modified in order to achieve the best result.

References

[1] Jianxun, W. A Study on Customer Acquisition Cost and Customer Retention Cost [U+FF1A] Review and Outlook. *Proceedings of the 9th International Conference on Innovation Management*, p.p. 799-803, 2012.

[2] Gulliver, S. R., Joshi, U. B. and Michell, V. Adapted customer relationship management implementation framework: Facilitating value creation in nursing homes. *Total Quality Management and Business Excellence. **24***, p.p. 9-10, 2013.

[3] Khodakarami, F. and Chan Y. Exploring the role of customer relationship management (CRM) systems in customer knowledge creation. *Information Management, **51***, p.p. 27-42, 2014.

[4] Payne, A. and Frow, P. A Strategic Framework for Customer Relationship Management. *Journal of Marketing, **69**, no. 4*, p.p. 167-176, 2005.

[5] Investopedia, *Customer Relationship Management (CRM)*, 2020-11. https://www.investopedia.com/terms/c/customer$_{r}elation_{m}anagement.asp.$

[6] Ho, Tin Kam. Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1995, p.p. 278–282.

[7] Breiman, L. Random Forests. *Machine Learning. **45** (1)*, 2001, p.p. 5–32.

[8] Breiman, L. Bagging predictors. *Machine Learning. **24***, 1996, p.p. 123–140.

[9] Hadden, J., Tiwari, A., Roy, R., and Ruta, D. Computer assisted customer churn management: State-of-the-art and future trends. *Computers Operations Research, **34**, no. 10*, p.p. 2902-2917, 2007.

[10] Hung, S.-Y., Yen, D. C., and Wang, H.-Y. Applying data mining to telecom churn management. *Expert Systems with Applications , **31**, no. 3*, p.p. 515-524, 2006.

[11] Mitchell, Tom. Machine Learning. New York: McGraw Hill. 1997.

[12] Russell, Stuart J.; Norvig, Peter. Artificial Intelligence: A Modern Approach (Third ed.). Prentice Hall, 2010.

[13] Oral, M., Oral, E. L. and Aydin, A. Supervised vs. unsupervised learning for construction crew productivity prediction. *Automation in Construction, **22***, p.p. 271-276, 2012.

[14] Verdhan, Vaibhav. Learning with Python: Concepts and Practical Implementation Using Python, Apress, 2020.

[15] Zhou, Zhi-Hua. Ensemble Methods - Foundations and Algorithms, Taylor Francis group, LLC, 2012.

[16] Kumar, Alok and Jain, Mayank. Ensemble Learning for AI Developers: Learn Bagging, Stacking, and Boosting Methods with Use Cases, Apress, 2020.

[17] Khan, A. A., Jamwal, S. and Sepehri, M. M. Applying data mining to customer churn prediction in an Internet Service Provider. *International Journal of Computer Applications **9**, No.7*, 2010.

[18] Pamina, J. and Raja, B. and SathyaBama, S. and S, Soundarya and Sruthi, M. S. and S, Kiruthika and V J, Aiswaryadevi and G, Priyanka. An Effective Classifier for Predicting Churn in Telecommunication. *Jour of Adv Research in Dynamical Control Systems, **11***, 2019.

[19] Hanif, I. Implementing Extreme Gradient Boosting (XGBoost) Classifier to Improve Customer Churn Prediction. *Proceedings of the 1st International Conference on Statistics and Analytics, ICSA 2019*. 2020.

[20] Sabbeh, S. F. Machine-Learning Techniques for Customer Retention: A Comparative Study. *International Journal of Advanced Computer Science and Applications, **9**, No. 2*, 2018.

[21] nVidia, *XGBOOST*. https://www.nvidia.com/en-us/glossary/data-science/xgboost/.

[22] Raschka, S., Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. 2018.

[23] Seger, C. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing. KTH, Stockholm, 2018.

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