KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First Cycle, 15 Credits

A Predictive Analysis of Customer Churn

ANNA BACKMAN AND OLIVIA ESKILS

Stockholm, Sweden 2023

Abstract

Churn refers to the discontinuation of a contract; consequently, customer churn occurs when existing customers stop being customers. Predicting customer churn is a challenging task in customer retention, but with the advancements made in the field of artificial intelligence and machine learning, the feasibility to predict customer churn has increased. Prior studies have demonstrated that machine learning can be utilized to forecast customer churn. The aim of this thesis was to develop and implement a machine learning model to predict customer churn and identify the customer features that have a significant impact on churn. This Study has been conducted in cooperation with the Swedish insurance company Bliwa, who expressed interest in gaining an increased understanding of why customers choose to leave.

Three models, Logistic Regression, Random Forest, and Gradient Boosting, were used and evaluated. Bayesian optimization was used to optimize the models. After obtaining an indication of their predictive performance during evaluation using Cross-Validation, it was concluded that LightGBM provided the best result in terms of PR-AUC, making it the most effective approach for the problem at hand.

Subsequently, a SHAP-analysis was carried out to gain insights into which customer features that have an impact on whether or not a customer churn. The outcome of the SHAP-analysis revealed specific customer features that had a significant influence on churn. This knowledge can be utilized to proactively implement measures aimed at reducing the probability of churn.

Sammanfattning

Att förutsäga kundbortfall är en utmanande uppgift inom kundbehållning, men med de framsteg som gjorts inom artificiell intelligens och maskininlärning har möjligheten att förutsäga kundbortfall ökat. Tidigare studier har visat att maskinlärning kan användas för att prognostisera kundbortfall. Syftet med denna studie var att utveckla och implementera en maskininlärningsmodell för att förutsäga kundbortfall och identifiera kundegenskaper som har en betydande inverkan på varför en kund väljer att lämna eller inte. Denna studie har genomförts i samarbete med det svenska försäkringsbolaget Bliwa, som uttryckte sitt intresse över att få en ökad förståelse för varför kunder väljer att lämna.

Tre modeller, Logistisk Regression, Random Forest och Gradient Boosting användes och utvärderades. Bayesiansk optimering användes för att optimera dessa modeller. Efter att ha utvärderat prediktiv noggrannhet i samband med krossvalidering drogs slutsatsen att LightGBM gav det bästa resultatet i termer av PR-AUC och ansågs därför vara den mest effektiva metoden för det aktuella problemet.

Därefter genomfördes en SHAP-analys för att ge insikter om vilka kundegenskaper som påverkar varför en kund riskerar, eller inte riskerar att lämna. Resultatet av SHAP-analysen visade att vissa kundegenskaper stack ut och verkade ha en betydande påverkan på kundbortfall. Denna kunskap kan användas för att vidta proaktiva åtgärder för att minska sannolikheten för kundbortfall.

Contents

Chapter 1

Introduction

Customer Relation Management (CRM) is about the relationship between organization and customer. During the twentieth century, executives and academics became concerned with the topic of CRM. CRM is a broad subject, but it has four key elements, namely, customer identification, customer development, customer attraction and customer retention. Customer retention is a central concern of CRM, and it is about pleasing customers expectations and demands so that they become loyal to the organization. When these demands are not met, the opposite situation can occur, customers churn. Customer churn is when existing customers stop being customers. In order to manage customer churn, the customers who are at risk of churning should be recognized and then they should be convinced to stay [1]. The cost of acquiring new customers can be 12 times larger than retaining already existing customers [2]. This makes customer retention an important topic for many businesses.

With businesses having access to an ever-increasing amount of data, there is a growing interest in using data-driven operations and business analysis to obtain valuable and relevant insights. A key concern in this area of focus is identifying valuable and tangible metrics that have a business impact, as well as the methodology for extracting and calculating these metrics through various models. One such model which can be used in strategic business development is a churn prediction model, a predictive binary classification model that estimates the likelihood of individual customers discontinuing their services [3]. This makes churn prediction a matter of significance within a company's data driven CRM strategy concerning customer retention. By analyzing this model in terms of feature importance of the outcome of the classification, it is possible to determine the features that have a significant impact on churn [4].

1.1 Purpose

This study has an exploratory approach with two main objectives. Firstly, to shed light on how churn can be suitably modeled within the insurance industry based on related work. Secondly, to investigate the feasibility of implementing such a churn prediction model and identifying which attributes that have a significant impact on churn.

1.2 Research Questions

The research questions that intends to be approached are the following:

1. What is the feasibility of creating and implementing an accurate prediction model of customer churn which determines whether a customer is going to churn and gives insights in why that particular customer is going to churn?

2. Which of the models random forest, logistic regression and gradient boosting yields the best results regarding predictive performance?
3. Which customer features affect customer churn the most?

1.3 Delimitations

The delimitations stated below has been made:

- This work is based on one associated cluster of customer data provided by Bliwa.
- The models that will be evaluated are logistic regression, random forests and gradient boosting.
- Customer costs will not be considered.
- This thesis treats confidential customer information, hence all sensitive information will be anonymized throughout the report.

1.4 Thesis Structure

This thesis is divided into three parts. The first part which consists of chapters 2-3 intends to give the required background knowledge for the problem at hand. It discusses related work, current trends in the insurance industry, what customer churn is and gives insights in how the machine learning framework looks like. The second part, which is chapter 4, presents the implementation of the models and methods used in this project. The last part, which consists of chapters 5-6, presents the yielded results and depicts a discussion of the preceding chapters, consideration concerning future implementation and presents conclusions with the results in center.

Chapter 2

Background

This chapter provides a description of previous work in the field of churn prediction, which has shaped the machine learning framework of this project. Additionally, it highlights current trends that impact the insurance industry, along with the definition of customer churn.

2.1 Related Work

Customer churn is a binary classification problem. Within the topic of binary classification there exists well-defined, conventional methods which have shown to be efficient [3]. Moreover, there exists several similar projects regarding machine learning analysis of customer churn conducted on insurance companies, but more usual on telecom companies.

Lalwani et al. [3] conducted a customer churn prediction, evaluating a variety of machine learning algorithms. These were random forest, decision trees, support vector machine, naive bayes, logistic regression and boosting algorithms & extra tree classifiers such as XGBoost, AdaBoost and CatBoost. They found that XGBoost and AdaBoost performed superior regarding accuracy and AUC score. Random forest obtained a result of 78.04% in accuracy and 82% AUC score. Logistic regression got a result of 80.45% in accuracy and 82% AUC score. XGBoost received a result of 80.8% in accuracy and 84% AUC score.

Peng et al. [4] conducted a research on modeling a prediction of customer churn using GA-XGBoost, optimized using the genetic algorithm. They also integrated the model with a SHAP analysis in order to predict the actual reason behind churn which they meant was more in line with the business.

Adbelrahim et al. [5] used algorithms based on a tree structure in order to predict customer churn. These were decision tree, GBM tree algorithm, XGBoost and random forest. In the comparative analysis, they found XGBoost to perform the best in regard of AUC accuracy. However, using an optimization algorithm for the process of feature selection could further improve the performance.

Vafeiadis et al. [6] performed a comparative study of machine learning models for the purpose of customer churn prediction. They used decision tree, support vector machine, logistic regression and naive Bayes. The result they obtained showed that support vector machine using AdaBoost yielded the best performance. However, using feature selection strategies could improve the performance.

Coussement et al. [7] conducted an analysis of customer churn using random forest, logistic regression and support vector machine. Initially, the performance of support vector machine was almost equal to that of random forest and logistic regression, but after hyperparameter optimization, support vector machine was superior to the others in terms of AUC score and classification accuracy.

Y. Huang et al. [8] used a variety of classifiers in a project of churn prediction. The results showed that random forests performed superior in terms of around 55% in PR-AUC and around 87.5% in AUC score. When dividing the data into a number of the top U customers that were the most likely to churn they obtained a result of 71.55% PR-AUC and 93.26% AUC score. However, using optimization techniques for the process of feature extraction could improve the performance.

2.2 The Insurance Industry

2.2.1 Current Trends

In 2021, inflation started to rise due to monetary monetary and fiscal stimulus during the Covid-19 pandemic, as well as issues with the supply of semiconductors and other inputs. Russia's invasion of Ukraine has continued to underpin this development and meant higher prices for above all energy, grain, fertilizer and some raw materials. In the autumn of 2022, inflation in Sweden was on its highest level in more than 30 years and a similar development was seen in the US and within the euro area as well. Although energy prices have increased the most, inflation excluding energy prices shows that the high inflation includes significantly more goods. This suggests that the inflation will be relatively long-lasting [9].

The Riksbank and other central banks have since the beginning of 2022 tightened monetary policy through an increasing key interest rate to dampen the high inflation and rising inflation expectations. The higher interest rates have led to a slowdown in economic development and the high inflation affects the insurance industry in several different ways, especially if it persists for a long time. For non-life insurance companies, it is about higher costs for repairs that make the handling of damages more expensive, and for life insurance and occupational pension companies about larger future payments of value-guaranteed defined benefit pensions. The extent to which the premiums will rise as a result of the high inflation depends, among other things, on the competitive situation. A slowdown in the economy affects the amount of premiums paid for non-life insurance because demand for insurance typically declines when consumption and investment decline. The growth in premiums paid for non-life insurance is therefore expected to decrease in the coming years when the premiums are adjusted for inflation [9].

2.2.2 Customer Churn

The word churn is derived from turn and change and means the discontinuation of a contract. In other words, churn represents when customers stop being customers. There are different types of churn:

- Deliberate/active - the customer chooses to quit the contract in order to change provider.
- Incidental/rotational - the customer chooses to quit the contract without switching to another provider.
- Non-voluntary/passive - the company chooses to discontinue the contract.

It is hard to predict voluntary churn, which includes deliberate/active and incidental/rotational churn, but it is of big interest to companies to be able to react and take appropriate actions in order to prevent voluntary churn to minimize the potential losses [10].

Chapter 3

Machine Learning Framework

This chapter presents a depiction of the machine learning framework used in this project. It includes a theoretical description of the different methods and algorithms employed throughout the project. In the subsequent chapter, a description is provided on how this framework was applied on the problem at hand.

3.1 Data Preprocessing and Transformation

Data preprocessing and transformation is often needed for the purpose of converting raw data into a convenient format that is suitable for the model that is being used. For a model to be able to fit a relationship between the explanatory variables and the output label, these variables has to be transformed into an appropriate scale of measurement. It can include several different steps depending on the model and the characteristics of the data [11].

3.1.1 Feature Engineering

Feature engineering aims to enhance the comprehension of the data and the task at hand, while performing experimental analysis to investigate the behavior of both the data and the model. Its primary objective is typically to construct an accurate prediction model. The process of feature engineering involves iterating through data selection and model evaluations until achieving a satisfactory outcome [12].

Feature engineering typically involves four main steps. Firstly, brainstorming features is carried out by studying relevant literature to gather inspiration. Then, the most appropriate features are selected and extracted based on the specific problem and the characteristics of the available data. The third step involves selecting the relevant features to be used in training the model. Finally, the model is evaluated using the selected features [12].

3.2 Models

3.2.1 Logistic Regression

Logistic regression is a discriminative classifier where the response variable is the *log* of the expectation of being classified in a certain group of a multi-class or binary response. Logistic regression makes a number of assumptions such as normally distributed responses of the explanatory variable, independence and constant variance. A transformation is applied to the response variable to achieve a probability distribution that is continuous over the output classes which are bounded between 0 and 1. This transformation is called the sigmoid function where z corresponds to the *log* expectation divided by the *logit*. The sigmoid function is given by

$$\sigma(Z) = \frac{1}{1 + exp(-z)}\cdot$$

In the case of binary classification, the logistic regression model can be phrased by a summation over the linear combinations of weighted input features, plus a bias term which is given by

$$p(y^{(i)} = 1 \mid x^{(i)}, w) = \frac{1}{1 + exp(-w^T x^{(i)} - b)}$$

respectively

$$p(y^{(i)} = 0 \mid x^{(i)}, w) = 1 - \frac{1}{1 + exp(-w^T x^{(i)} - b)}\cdot$$

The objective is to determine the set of weights which corresponds to a minimization of the negative *log* likelihood over the training set by using optimization techniques such as stochastic gradient descent or gradient descent. The loss function which is also referred to as the so called cross-entropy measures the difference between the predicted class and the true label. The loss function is the object of minimization in order to minimize this difference between the predicted class and the true label. The loss function is given by

$$L(\theta) = -\frac{1}{m} \sum p_i log(y_i) + (1 - p_i) log(1 - y_i)$$

as per [13].

3.2.2 Random Forest

Random forest is a learning algorithm which is ensemble based. It includes a collection of M de-correlated decision trees and is based on the idea of bootstrap aggregation (bagging). Bagging means that each decision tree is created from a random vector based on the given amount of data, thus probably not all data points occur exactly once [14].

In the case of classification, random forest uses multiple trees to compute majority votes in the concluding leaf nodes when predicting the label. Decision trees are simply a tree-like structure where the node at the top is considered the root and equals the premier predictor variable and is recursively split until the decision node or concluding node is reached. The decision tree algorithm is a greedy algorithm, that is, an algorithm which takes the simplest solution, top down approach which partition the data set into small subsets. To decide which feature to split at each decision node, the entropy, which is the difference between the true label and the predicted class, is computed [13]:

$$Entropy = -plog_2(p) - qlog_2(q)\cdot$$

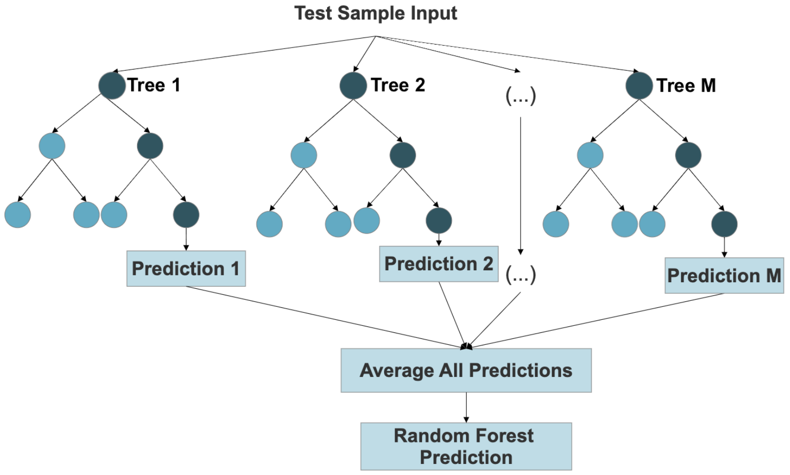

Figure 3.1: Algorithm of Random Forests

3.2.3 Gradient Boosting

Boosting algorithms such as gradient boosting combine, in an iterative way, weak learners into a strong learner. Weak learners are learners that are slightly better than a random learner. Given a training data set, $D = \sum_{i=1}^{N}(x_i, y_i)$, the objective of gradient boosting is to determine an approximation, $\hat{F}(x)$, of the function $F^*(x)$ which systematically describes instances of x to their output values y. This is done by a minimization of the expected value of the loss function, $L(y, F(x))$. The approximation of $F^*(x)$ is built in an additive way as a weighted summation of functions

$$F_m(x) = F_{m-1}(x) + p_m h_m(x),$$

where p_m is the weight generated by the m^{th} function, i.e., $h_m(x)$. These functions are models of the ensemble, for example a ensemble of decision trees. This approximation is constructed in an iterative way. First, an approximation that is constant is acquired as

$$F_0(x) = \arg\min_{\alpha} \sum_{i=1}^{N} L(y_i, \alpha),$$

and the following models are intended to minimize

$$(p_m, h_m(x)) = \arg\min_{p,h} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + ph(x_i)).$$

Instead of instantly solving the optimization problem, each h_m can be regarded as a greedy step in an optimization following gradient descent for F^*. Hence, each model h_m is trained on another data set $D = \sum_{1}^{N}(x_i, r_{mi})$, where r_{mi} are the pseudo-residuals which are calculated by

$$r_{mi} = \left[\frac{\partial L(y_i, F(x))}{\partial F(x)}\right]_{F(x)=F_{m-1}(x)}.$$

Subsequently, the value of p_m is determined through a line search optimization problem. Gradient boosting is an algorithm which can suffer from over-fitting, which means, generalize bad to new, unseen data, if the iterative process is not regularized properly. To control this additive process various regularization hyperparameters can be considered [15].

3.3 Model Optimization

Machine learning models have hyperparameters which are manually set parameters as opposed to model parameters which are internally coefficients and found through training. Hyperparameters often have a known effect on the associated model, but it can be ambiguous how to choose the optimal ones, especially since the models can have a variety of hyperparameters which can operate in non-linear ways. Consequently, hyperparameter optimization is the problem of determining such a set of optimal hyperparameters for the learning algorithm. This optimization process includes defining a search space which can be thought of in terms of an n-dimensional geometric space, where each hyperparameter corresponds to a different dimension and the dimension scale is the value that the hyperparameter takes. Each point in this search space is a vector which represents one specific model configuration with values for each hyperparameter. The goal is hence to find the vector that generates the best model performance [16].

3.3.1 Bayesian Optimization

Bayesian optimization is an efficient method to find the extrema where the nature of the minimized or maximized function is characterized as computationally expensive. Given a black-box function $f : \mathbb{X} \to \mathbb{R}$, the objective of the Bayesian optimization is to find the sampling point

$$x^+ = \arg\min_{x \in \mathbb{X}} f(x)$$

that minimizes f globally and $\mathbb{X}$ represents the search space of x. The aim of Bayesian optimization is to combine the prior distribution of $f(x)$, $p(f)$, with the sample information to determine the posterior of this function. Subsequently, the posterior information is utilized to obtain where $f(x)$ is minimized pursuant to a criterion which is represented by an acquisition function, $a_{p(f)} : \mathbb{X} \to \mathbb{R}$ [17]. The acquisition function is usually an inexpensive function which can be evaluated at a given point which is proportional with how beneficial it is expected to be for the minimization problem to evaluate f at x. The acquisition function is then optimized to obtain the next sample point to be able to maximize the expected utility [18]. The Bayesian optimization algorithm is displayed below.

1. Find $x_{n+1} \in \arg\max a_p(x)$ by numerical optimization.
2. Evaluate the often noisy and expensive function $y_{n+1} \sim f(x_{n+1}) + \mathcal{N}(0, \sigma^2)$ and add the subsequent data point (x_{n+1}, y_{n+1}) to the set $D_n = (x_j, y_j)_{j=1,\ldots,n}$ of observations.
3. Update $p(f \mid D_{n+1})$ and $a_{p(f|D_{n+1})}$.

Often the evaluation of the acquisition function is cheap compared to the evaluation of f such that the optimization effort is insignificant [17].

3.4 Model Evaluation

3.4.1 K-Fold Cross-Validation

K-Fold Cross-Validation is a method often used to evaluate the performance of a machine learning model by testing its ability to generalize to new, unseen data [19]. It involves partitioning the available data into k subsets: $k-1$ training sets and one validation set. The model is repeatedly trained on the $k-1$ training sets and then evaluated on the validation set. This is done k times. K-Fold Cross-Validation can be useful in identifying overfitting, which occurs when a model performs well on the training data, but poorly on new data. By evaluating the model on multiple validation sets, K-Fold Cross-Validation can provide a more accurate estimate of the model's ability to generalize to new data [19]. The most frequent used values of k are 10 or 5 in the literature. This is because these values are argued to yield test rate estimates that neither suffer from an overly high variance nor high bias [20].

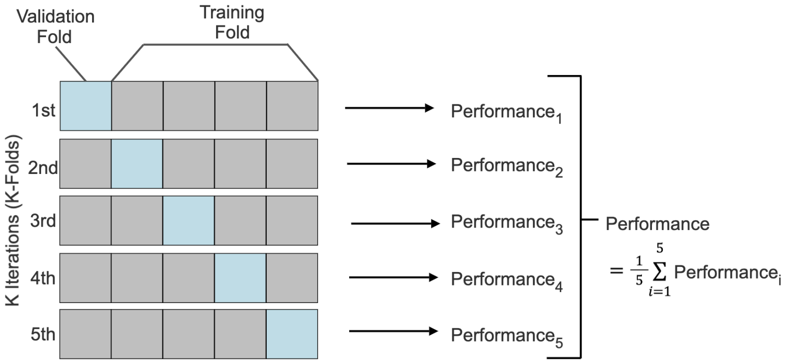

Figure 3.2: Algorithm of K-Fold Cross-Validation

3.4.2 Confusion Matrix

The performance of a model can be assessed by using predefined categories and a confusion matrix, which measures the model's ability to predict the correct class. The figure below demonstrates how the matrix is constructed.

		Predicted Class	
		Predicted True	Predicted False
Target Class	Actual True	True Positive	False Negative
	Actual False	False Positive	True Negative

Figure 3.3: Confusion Matrix

The results can be interpreted and evaluated with the measures accuracy, precision and recall. Accuracy is a ratio of the correctly predicted labels. The accuracy measure is given by

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}.$$

Precision is a measure of how good the model is at predicting the observed true class as true. The precision measure is given by

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}.$$

Recall provides notion of the coverage of the positive class. The recall measure is given by

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

as per [21].

3.4.3 ROC-AUC

ROC-AUC is a performance metric used to evaluate binary classification models [22]. The ROC curve is a plot of the true positive rate against the false positive rate at different classification thresholds. The AUC is the area under the ROC curve and it ranges from 0.5 to 1. A higher AUC indicates better performance of the model in distinguishing between the two classes.

3.4.4 PR-AUC

PR-AUC is another performance metric that evaluates a binary classifier's ability to accurately predict positive instances [23]. It measures the area under the precision-recall curve, which plots the precision against the recall. PR-AUC is particularly useful for evaluating models on imbalanced data sets, where the number of positive instances is much smaller than the number of negative instances [23].

3.5 Shapley Additive Explanations (SHAP)

SHAP is an additive feature attribution method and is based on game theory. SHAP generates reduced inputs w by mapping x to w using $x = h_x(w)$. Based on this reduced input w, it is possible to approximate the initial model $f(x)$ via a linear function of binary variables,

$$f(x) = g(w) = \varphi_0 + \sum_{i=1}^{N} \varphi_i w_i$$

where $w = \{0,1\}^N$, N is the amount of input features, $\varphi_0 = f(h_x(0))$ and φ_i is the value of the feature attribution:

$$\varphi_i = \sum_{A \in F \setminus \{i\}} \frac{\mid A \mid!(N- \mid A \mid! - 1)!}{N!} [f_x(A \cup \{i\} - f_x(A))]$$

$$f_x(A) = f(h_x^{-1}(w)) = E[f(x) \mid x_A]$$

where φ_i is the SHAP value, i.e., the measure of additive feature attributions. Here, F is a non-zero set of input in w and A is a subset of F where the i^{th} feature is excluded from F [24].

Chapter 4

Methodology

This chapter provides a descriptive account of the different steps of the methodology and outlines the actions undertaken in each step.

4.1 Data

4.1.1 Data Collection and Description

The data used in this project was provided by the insurance company Bliwa. The provided data was hashed and all confidential information was excluded in order to avoid the risk of connecting the values of an independent ID to its real identity. Furthermore, in the subsequent chapters of this report, some values in the figures and tables may be missing or anonymized in order to prohibit the disclosure of information regarding the insurance company's customers.

The data set consisted of approximately 300.000 customers and their corresponding personal features and their insurance-related features from the past 5 years. Every customer had 4 distinct personal features and 5 insurance features that are confidential and, as such, cannot be disclosed.

4.1.2 Data Preprocessing

The data was delivered in 7 different files due to its size, which meant that the first step of the data preprocessing involved parsing and merging the data sets into a single data set. This was accomplished using Python and Anaconda, with Jupyter Notebook as the work space. With help of the libraries Pandas and NumPy, the merged data set was divided and separated into two seperate data sets - one containing customers and their features, and another containing insurances and their features - with the customer ID acting as the linking factor between them. These two data sets could in turn be grouped by customer ID to once again create a single data set where each row represented one customer along with their personal and insurance features.

At this point, a definition of churn was formulated in order to categorize each customer as either a churned customer or not. The definition of churn weighted in both the aspect of timeframe and number of churned insurances. However, due to the confidentiality of the treated information, the exact definition cannot be disclosed. This definition of churn resulted in a churn rate of 28.11% for the data set.

Subsequently, the data set was modified to meet the model's requirements, which included numeric values in each column, as well as performing feature engineering to retrieve additional information that could assist the model in predicting the outcome. To transform the data set into the model requirements, Scikit-learn's ordinal encoding was used to convert categorical features into integers for the tree models, while one-hot encoding was used for logistic regression to transform categorical values.

4.1.3 Data Analysis

For data analysis, the libraries Matplotlib and Seaborn were used as tools to visualize the data through histograms, box plots and graphs for various variables to gain more understanding of the data distribution. Statistics were calculated and visualized, including for example minimum, maximum, mean and count for all features. This information could later on be used when performing feature engineering. None of these figures can be displayed since they reveal sensitive information.

4.2 Variable Selection and Feature Engineering

The initial variable selection was made in cooperation with Bliwa, the data provider. Joint discussions where held to identify factors that could impact customer churn and to determine which variables could be extracted from their data systems. This formed the basis for selecting the variables to be used in this project.

Regarding feature engineering, several additional features were created based on the original set of features. This was accomplished by using aggregations, such as minimum, maximum, mean, count and mode, which is the most frequent class in a feature. This was done in order to condense several insurances to one set of features on the customer level and extract further information that the original features did not contain, which could be utilized by the model when making predictions. After performing feature engineering, there were 28 new features, counting up to a total of 32 different personal and insurance related features.

4.3 Machine Learning

To perform a churn prediction, which is typically a classification problem, there exists several models. However, model performance is often affected by data characteristics [3]. Therefore, three different models were tested and evaluated to find the most suitable one for the described purpose. These three models were chosen based on a review of related work where their efficiency and reliability were proven.

4.3.1 Hold-Out Set

In order to obtain a reliable estimate of the performance of final model on new, unseen data, a hold-out set was created. The data was split in 80% / 20% where the hold-out was not touched until the final, optimal model configuration was identified. The remaining 80% was used for iterative experimentation and metrics measured on that data were only seen as indicative.

4.3.2 K-Fold Cross-Validation

To obtain an indication of model performance without using the final hold-out set, we used Sklearn's implementation of k-fold cross-validation with k=5 in combination with modeling. The reason for choosing k=5 was a result of 5 being a conventional number of folds to use, as well as considering our data set not being largely enough for 10 folds and the additional processing time that would take. The chosen evaluation metrics were given as an average value generated by the folds of cross-validation. When performing cross-validation, the ratio of churned customers was stratified across the folds.

4.3.3 Modeling

The modeling was performed by iterating cross-validation folds and training each model type on the 4 training folds, while evaluating metrics on the remaining validation fold. The models used were a dummy baseline that always predicted the majority class, as well as random forests, logistic regression and gradient boosting. The baseline, logistic regression and random forest were implemented using Scikit-learn, while gradient boosting model used LightGBM from Microsoft.

4.3.4 Model Optimization

For model optimization, i.e. finding the optimal hyperparameters, the library Optuna was used were Bayesian optimization is the default method of optimization. The first step in performing hyperparameter optimization was to determine the search space for the hyperparameters of the different models. This required setting up the ranges for the search space and defining how values were to be sampled from them. Moreover, k-fold cross-validation with k=5 was performed again, but this time in combination with the optimization algorithm in order to obtain more robust performance metrics. The optimization algorithm was responsible for identifying the set of hyperparameters that yielded the highest average PR-AUC, which was selected to be the main metric for evaluation in this study. Since the aim of the model was to identify customers with a high risk of churning, which was the minority class, PR-AUC was considered a powerful measure of a model's ability to correctly identify as many minority events as possible [23].

4.3.5 Evaluation and Comparison

To determine which model performed best among logistic regression, random forests and gradient bosting, several performance metrics were used including accuracy, precision, recall, ROC-AUC and PR-AUC, with pre-defined implementations in scikit-learn. As mentioned in the previous section, PR-AUC was considered the main measure and given greatest importance in evaluating and comparing models to determine the optimal model for the problem. The model with the best performance was then evaluated using its optimal set of hyperparameters on the hold-out set to obtain the final scores.

4.4 Attributes Impacting Decision (SHAP-Analysis)

In order to determine which features had the highest impact on the outcome of the prediction, i.e. both positive respectively negative impact on churn, the python library SHAP was used. The resulting attributes that impacted the decision were then visualized using a bar plot, scatter plot, violin plot and force plots to be able to interpret and analyse the results.

Chapter 5

Results

This chapter presents the obtained results, which includes the results regarding model performance and the SHAP-analysis.

5.1 Model Performance

The resulting performances of the different models obtained from the cross-validtion performed during the model optimization are shown in Table 5.1. As seen in the table, LightGBM achieved the best performance among the models.

Model	Accuracy	Precision	Recall	ROC-AUC	PR-AUC
Dummy (baseline)	0.7189	0.0000	0.0000	0.5000	0.2811
Logistic Regression	0.7189	0.0000	0.0000	0.7211	0.4228
Random Forest	0.7372	0.5927	0.2080	0.7773	0.5229
LightGBM	**0.7396**	**0.5990**	**0.2235**	**0.7801**	**0.5303**

Table 5.1: The different models performances

In Table 5.2, the final scores of LightGBM is presented which were achieved when evaluating the model on the unseen, hold-out data set and using its determined optimal set of hyperparameters.

Model	Accuracy	Precision	Recall	ROC-AUC	PR-AUC
LightGBM	0.7385	0.6031	0.2202	0.7832	0.5354

Table 5.2: Final model scores of LightGBM

Table 5.3 shows the model performance of LightGBM when the hold-out data set was divided into quantiles based on the likelihood of customers to churn according to the model. The quantiles are related to the proportion of customers with the highest probability of churn. This means that the zero quantile contains all customers, the 0.50 quantile includes the half of the customers that are most likely to churn, the 0.75 quantile includes the quarter of customers with the highest likelihood to churn and so on.

Quantile	Accuracy	Precision	Recall	ROC-AUC	PR-AUC
0.00	**0.7385**	0.6031	0.2202	**0.7832**	0.5354
0.50	0.5908	0.6031	0.2756	0.5914	0.5601
0.75	0.5713	0.6031	0.4837	0.5957	0.6210
0.90	0.6037	0.6037	**1.0000**	0.5726	0.6952
0.95	0.6520	**0.6520**	**1.0000**	0.6020	**0.7571**

Table 5.3: Model performance of LightGBM when dividing into risk quantiles

5.2 SHAP-analysis

Figure 5.1 shows a bar plot of the features which had the most impact on the model output regarding churn. The features are arranged on the y-axis in the order of importance, with the most important at the top and the least important at the bottom.

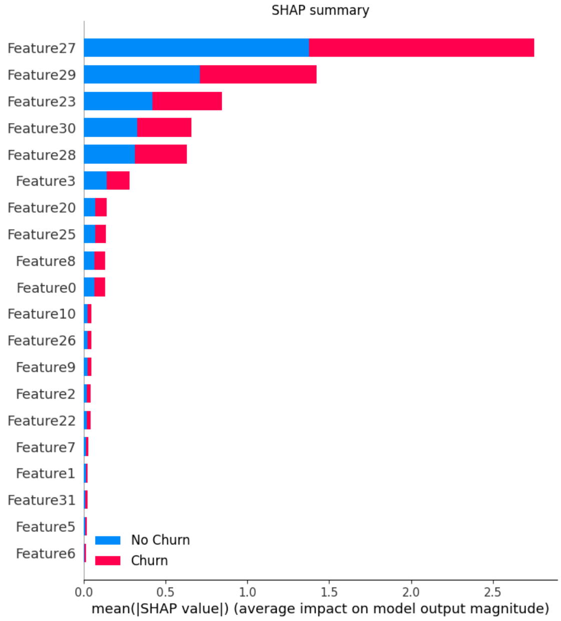

Figure 5.1: Bar plot of SHAP analysis

Figure 5.2 and 5.3 shows a scatter plot and a violin plot respectively, of how the various features influenced the model's assignment of churn probabilities. These plots illustrate the same information but with slightly different visual representation.

The scatter plot can be interpreted as follows: as previously mentioned, the features are displayed on the y-axis in order of importance. The SHAP value, which indicates the change in log-probability on the outcome, is shown on the x-axis. Each point on the plot represents one data point, i.e., one costumer. The color of the point corresponds to the original value of that feature, with red indicating a high feature value and blue indicating a low feature value. For boolean features, only two colors can be used, whereas for numerical features, the entire spectrum is available. While examining the graph, if a large proportion of the blue data points are on the right side of the vertical zero-axis and at the same time a large proportion of the red data points are on the left side of the zero-axis, this means that a low feature value implicates a higher risk of churn and vice versa.

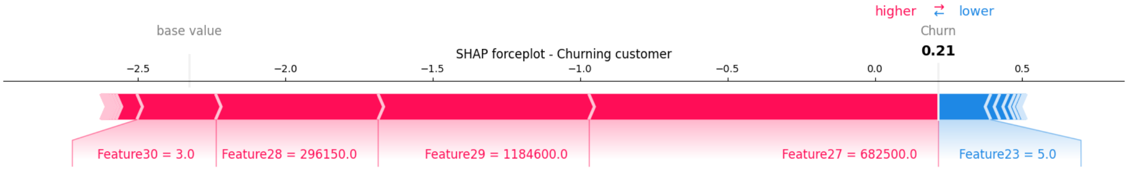

Figure 5.4: Force plot of a churning customer

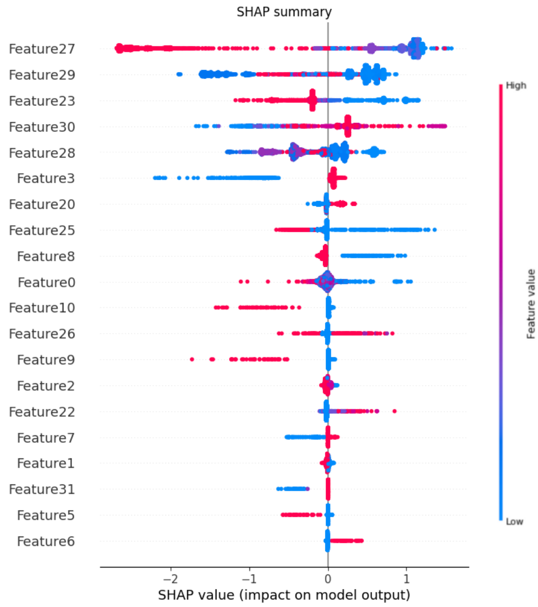

Figure 5.2: Scatter plot of SHAP analysis

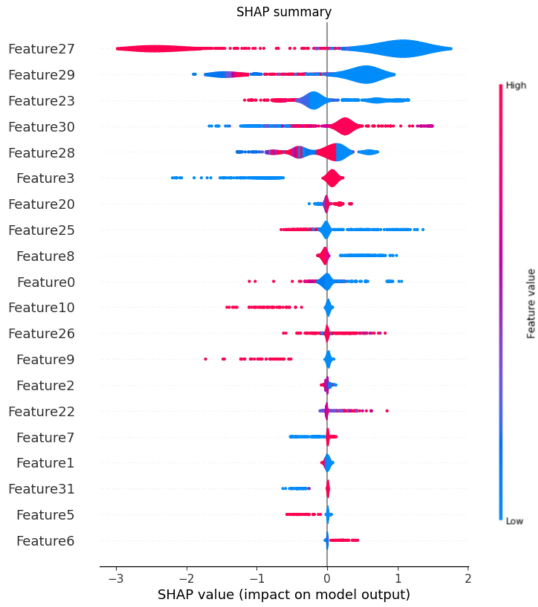

Figure 5.3: Violin plot of SHAP analysis

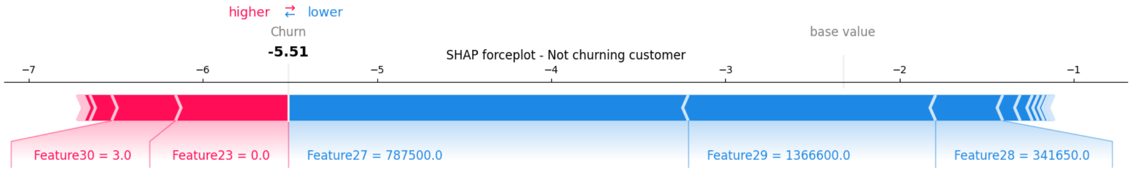

Figure 5.5: Force plot of a non-churning customer

Figure 5.4 and 5.5 show force plots of a correct prediction the model made of a churning customer and a non-churning customer, respectively. The force plot illustrates the impact that different features had on the model's prediction for a specific customer. This provides insights into what contributed to a certain prediction for a specific customer.

Chapter 6

Discussion

This chapter provides a discussion on the obtained results, focusing specifically on their relation to business analysis and the comparison of predictive accuracy. Additionally, considerations regarding future implementations are presented, and a conclusion is drawn concerning the research questions.

6.1 Business Analysis

The introduction highlighted the importance of customer retention for maintaining loyal customers, profitability and avoiding the cost of acquiring new customers. Thus, implementing a customer churn prediction model can provide insights on which customers are at risk of churning, offering an opportunity to take proactive actions. Additionally, current inflation trend and tightened monetary policy underscore the importance of churn prediction. As described in the background, an economic slowdown can lead to decreased demand for insurance, making it crucial to retain existing customers to maintain profitability.

Regarding which customer features that affect customer churn the most, it can be seen in Figure 5.1 that features 3, 23 and 27-30 seem to have the most impact on the model output regarding whether a customer is going to churn or not, according to the SHAP analysis.

When reviewing the scatter plot, Figure 5.2, and the violin plot, Figure 5.3 of the SHAP analysis, it can be observed that a lower value on features 23, 27 and 29 caused the model to assign a higher probability of churn, and vice versa. For features 3 and 30, a higher value caused the model to assign a higher probability of churn, and vice versa. As for feature 28, the result is difficult to interpret.

The feature importance results can guide proactive actions such as targeted marketing campaigns or personalized offers for customers with high churn risk based on these features. Although all features in this report are anonymized, a comment can be made on the relevance of the results regarding feature importance. The features that had most impact on churn respectively non-churn were sufficiently evident and comprehensible to be able to direct proactive actions.

6.2 Predictive Accuracy

In terms of predictive performance, Table 5.2 shows that the model scored 53.54% in terms of PR-AUC score on the final test, which is above the baseline of 28.11%. The model also achieved an accuracy of 73.85%, compared with the baseline of 71.89%, and an AUC score of 78.32%, compared with the baseline of 50%. Furthermore, based on the analysis in Table 5.3 it can be seen that when dividing the data into risk quantiles the model is shown to perform better when focusing only on those with a higher predicted churn probability. This is aligned with the findings of Y. Huang et al. [8], which used a similar method. Identifying high-risk customers is crucial for the company to take proactive actions before they potentially churn. By dividing customers into different risk groups based on quantiles, the company can take corresponding proactive actions depending on risk group. Metrics can therefore be weighted against the cost of action and the potential efficiency of these actions.

Regarding predictive accuracy, these results align with those of related studies. Achieving even higher predictive accuracy may be challenging due to difficulty in capturing through data what causes people to churn in reality. The reasons for churn can be diverse, ranging from competitor marketing campaigns to changes in family relationships, which are hard to predict and capture in terms of data that the model can interpret. However, a very high accuracy is not necessarily required for the model to be useful, as its goals can be to act as a guideline or direction to identify specific groups of customers at high risk of churning. If the goal is to prevent high-risk customers from churning through proactive measures, mislabeling a customer as churn and providing proactive actions would not cause significant issues if they were not actually going to churn.

6.3 Considerations Concerning Future Implementation

There are several considerations to keep in mind for future implementation and improvements. For example, it would be beneficial to try and evaluate additional models and collect a larger data set. In this project, all the data points for the examined population were available, but over time, more data points becomes available as new customers arrive and others churn. This increase in data could potentially improve the accuracy of the model.

A more extensive feature engineering could also improve the performance of the model. This includes trying to extract more customer and insurance features from the data system, which were not used in this project. It also involves having a more extensive discussion with Bliwa and potential experts on the topic of customer churn to understand what makes people churn and what kind of features that could be created to capture this information in an interpretable manner for the model to understand. This extends beyond the data already in their system and could include external factors such as inflation rates and other environmental factors.

6.4 Conclusion

To summarize this project, it can be concluded that it is feasible to create and implement a relatively accurate prediction model of customer churn which determines the likelihood of a customer churning and provides the corresponding probability. Additionally, the model offers insights into why a specific customer is likely to churn or not, in terms of feature importance of the outcome of the model.

Regarding predictive performance, it could be concluded that random forest and LightGBM performed quite equally, but LightGBM scored slightly better and seems to be the best approach to the described problem and therefore also answers the research question of which model yields the best results regarding predictive performance.

In order to address the question regarding the customer behaviour and characteristic that have the greatest impact on customer churn, it can be concluded that lower values on feature 27, 29 and 23, as well as higher values on feature 30 and 3, have the most significant effect customer churn.

Bibliography

[1] C. Huigevoort and R. Dijkman, "Customer churn prediction for an insurance company," *Eindhoven Teknoloji Üniversitesi*, 2015.

[2] G. Torkzadeh, J. C.-J. Chang, and G. W. Hansen, "Identifying issues in customer relationship management at merck-medco," *Decision Support Systems*, vol. 42, no. 2, pp. 1116–1130, 2006.

[3] P. Lalwani, M. K. Mishra, J. S. Chadha, and P. Sethi, "Customer churn prediction system: a machine learning approach," *Computing*, pp. 1–24, 2022.

[4] K. Peng and Y. Peng, "Research on telecom customer churn prediction based on ga-xgboost and shap," *Journal of Computer and Communications*, vol. 10, no. 11, pp. 107–120, 2022.

[5] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," *Journal of Big Data*, vol. 6, no. 1, pp. 1–24, 2019.

[6] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1–9, 2015.

[7] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," *Expert Systems with Applications*, vol. 34, no. 1, pp. 313–327, 2008.

[8] Y. Huang, F. Zhu, M. Yuan, K. Deng, Y. Li, B. Ni, W. Dai, Q. Yang, and J. Zeng, "Telco churn prediction with big data," *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 607–618, 2015.

[9] E. Erlandsson, L. Friman Blomgren, P. Karlsson, and J. Söderberg, "Omvärldstrender 2023 försäkring i en orolig tid," *Svensk Försäkrings Rapportserie*, pp. 3–4, 2022.

[10] V. Lazarov and M. Capota, "Churn prediction," *Business Analytics Course. TUM Computer Science*, vol. 33, p. 34, 2007.

[11] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised leaning," *International Journal of Computer Science*, vol. 1, no. 2, pp. 111–117, 2006.

[12] J. Brownlee, "Discover feature engineering, how to engineer features and how to get good at it," *Machine Learning Mastery*, 2014. `https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/`.

[13] K. Kirasich, T. Smith, and B. Sadler, "Random forest vs logistic regression: binary classification for heterogeneous datasets," *SMU Data Science Review*, vol. 1, no. 3, p. 9, 2018.

[14] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[15] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.

[16] J. Brownlee, "Hyperparameter optimization with random search and grid search," *Machine Learning Mastery*, 2020. `https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/`.

[17] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast bayesian optimization of machine learning hyperparameters on large datasets," *Artificial Intelligence and Statistics*, pp. 528–536, 2017.

[18] J. Wilson, F. Hutter, and M. Deisenroth, "Maximizing acquisition functions for bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[19] D. Berrar, "Cross-validation," *Artificial Intelligence and Statistics*, 2018. `https://doi.org/10.1016/B978-0-12-809633-8.20349-X`.

[20] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of machine learning algorithms with different k values in k-fold cross-validation," *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 61–71, 2021.

[21] A. Luque, A. Carrasco, A. Martín, and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, 2019.

[22] S. Narkhede, "Understanding auc-roc curve," *Towards Data Science*, vol. 26, no. 1, pp. 220–227, 2018.

[23] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, 2019.

[24] D. Wang, S. Thunéll, U. Lindberg, L. Jiang, J. Trygg, and M. Tysklind, "Towards better process management in wastewater treatment plants: Process analytics based on shap values for tree-based machine learning methods," *Journal of Environmental Management*, vol. 301, pp. 113–941, 2022.

TRITA – XXX-XXX 20XX:XX