Machine learning in churn prediction

The impact of data availability on the performance of single and combined classifiers

Student details

Name: Floris H. Zanders
Student number: 2048881

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee

Supervisor: Dr. A. Hendrickson

Second reader: Prof. M. Louwerse

Words: 8797

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
January, 2022

Table of Contents

Preface

In my two years at Tilburg University, I have only been physically on campus for the first two months. After that, the lockdown settled in due to COVID-19 and online studying began. Therefore, it has been a strange but nevertheless educational couple of years. I am grateful that the university swiftly adopted an online method of teaching to enable me to continue with my Master's degree smoothly.

Finalizing my education with this thesis has been a fun, educational, but also stressful process. I would like to thank my supervisor, Andrew Hendrickson, for his calm and useful guidance throughout the past months. Next to that, I would like to thank the other members of my thesis group for the informative sessions.

Finally, I would like to thank my friends and family for supporting me in just the right way. Finding the right balance between working on my thesis and having time off would not have been possible without your understanding of the workload.

Abstract

Losing clients to a competitor is known as customer churn and is a common problem for many businesses. A sector that suffers from this problem is the telecommunications industry, due to its saturated and highly competitive nature. Commercials lure clients into switching between providers, causing customer churn. Attracting new customers in this market costs more effort than retaining customers and therefore it is more interesting to investigate the latter. This logically starts with identifying the customers who are most likely to churn and that is where the use of Machine Learning (ML) techniques can be valuable. A wide range of ML approaches, that use customer data, enable the prediction of customers with the highest tendency to churn.

Where earlier research mostly focused on enhancing the predictive performance of existing and novel approaches, this thesis focuses on the impact that data availability has on the classification performance of existing ML classifiers. It does so by looking into the influence of dimensionality reduction on k-Nearest Neighbors (k-NN) and Support Vector Machine (SVM) classifier performance. Next to that it compares this performance to the same classifiers combined with K-Means clustering to investigate the difference between single and combined classifiers. Lastly, it tests both classifier types in circumstances where training data is available to a lesser extent. The analyses are performed based on the Telco customer churn dataset from IBM, which is a fictional dataset on Californian customers of a telecommunication business.

In this thesis, it is concluded that single classifiers based on Principal Component Analysis outperform the single classifiers based on Factor Analysis of Mixed Data. Next to that, k-NN is negatively correlated and SVM is positively correlated with the decrease in dimensionality. Also, it was found that single and combined classifiers perform nearly identical based on accuracy and the macro averages of F1-score, recall and precision when

the train set size decreases. The only performance drop-off point was found by decreasing the train set size from 80% to 70%.

Keywords

Telecommunications · Data availability · Churn · Machine Learning · K-Nearest Neighbors · Support Vector Machines · K-means clustering · F1-score · Factor Analysis of Mixed Data · Principal Component Analysis

The impact of data availability on the performance of single and combined classifiers in churn prediction

Within the telecommunications business it is common for customers to occasionally switch between providers that offer better deals. This results in one party losing a client and suffering from customer churn. Research states that attracting new customers in a market as saturated as the telecommunications market costs a company much more than retaining customers and that is why it is interesting to investigate how customers can be retained (Adhikary & Gupta, 2021; Amin, et al., 2019; Kaya, et al., 2018). Such an investigation logically starts with identifying the customers who are on the verge of leaving for a competitor and this is exactly where Machine Learning (ML) techniques can prove to be valuable. These techniques make it possible to predict, on the basis of historical customer data, which customers are the most likely to churn (Amin, et al., 2019).

A wide range of ML techniques has already been investigated in previous studies. Some of these techniques only use one classifier to predict churn, such as Naïve Bayes (NB) (Huang, Kechadi, & Buckley, 2012), Decision Tree (DT) (de Caigny, Coussement, & de Bock, 2018), k-Nearest Neighbors (k-NN) (Lee, Wei, Cheng, & Yang, 2012) or Support Vector Machines (SVM) (Vafeiadis, Diamantaras, Sarigiannidis, & Chatzisavvas, 2015). Others use combinations of multiple techniques (hybrids) such as K-Means clustering combined with SVM to eliminate the weaknesses of the individual techniques (Rajamohamed & Manokaran, 2018). In addition, research has also been conducted on more advanced methods based on Neural Networks (NN) and Fuzzy Logic (Sivasankar & Vijaya, 2019), to predict even more accurately which customers have a high tendency to churn.

The studies stated above, all have the goal of trying to achieve a better predictive performance regardless of which technique or combination of techniques is used. To differentiate from this goal, the aim of this study is to determine whether the use of machine

learning techniques in churn prediction remains valuable as data availability decreases. With this approach, this thesis project will contribute to the existing literature by exploring the limits of ML. Investigating the added value of ML algorithms on data with few dimensions and on varying sized data samples, may allow businesses to perform analyses sooner and with smaller datasets. Next to that, customers may benefit from being identified as churner sooner by receiving personalized offers. Analyzing the performance limits of churn predicting models could highlight aspects of this problem, that ultimately indicate if the investment in churn prediction can be worthwhile in situations where data availability is limited.

Apart from the motivation above, this research also dives deeper into the impact of dimensionality reduction techniques on churn predicting models. Next to that, it examines the usability of these same models in situations where data availability is scarce. Lastly, the models in this research are fit to maximize F1-score which is also less common in existing and earlier mentioned literature. In churn prediction, imbalanced datasets are the standard (Ahmed & Maheswari, 2017) and accuracy alone cannot provide a complete evaluation. Therefore, precision and recall, which explain more about the number of churners correctly classified, are also taken into careful consideration during the evaluation of the models.

This research is structured by means of a main research question and three sub-research questions. Given the focus and general approach of the literature introduced in this chapter and the accompanying motivation for this research, the main research question (RQ) has been formulated as follows: *To what extent does data availability impact the performance of machine learning models in churn prediction?*

This question has been broken down into three parts. The first part will revolve around the influence of dimensionality reduction on two single ML classifiers being k-NN and SVM. This will be investigated by using two techniques: Factor Analysis of Mixed Data (FAMD)

and Principal Component Analysis (PCA). The first technique is applied because it was created for reducing the dimensionality of mixed datasets (Pages, 2004). The second technique is applied because it has been used in all Kaggle submissions on this dataset (Kaggle, 2021). With dimensionality reduction, the size of the dataset and the amount of noise in it decreases, therefore it requires less storage space and less time for computation, which is only beneficial (Nguyen & Holmes, 2019). With the motivation given above, the first sub RQ reads: *How do dimensionality reduction techniques impact the performance of single classifier ML models?*

The second part of this research revolves around the performance comparison of single and hybrid classifiers. Being able to eliminate the weaknesses of individual classifiers by finding the right combinations is a powerful tool (Rajamohamed & Manokaran, 2018) and should therefore be investigated further. Since customer churn datasets are often imbalanced (Ahmed & Maheswari, 2017), accuracy alone can be a misleading evaluation metric. Therefore, this research will compare the performances primarily on F1-score and secondarily on recall, precision and accuracy to present a clear overview. The second sub RQ reads: *Do hybrid classifiers outperform single classifiers based on "F1-score" in churn prediction?*

The third part revolves around data reduction. The acquired dataset consists of 7043 records. However, in any other circumstance where data is scarce, it is interesting to see if single and hybrid classifiers start to drop in performance at a certain point. This could potentially indicate the amount of data that is necessary to conduct a meaningful prediction using ML techniques. The final sub RQ is therefore: *At what point does the performance of both single and hybrid classifiers drop off, if the amount of training data is decreased?*

The main findings of this thesis are that single classifiers based on PCA outperformed the single classifiers based on FAMD. Next to that, k-NN is negatively correlated and SVM is positively correlated with the decrease in dimensionality. Also, it was found that the single and hybrid classifiers perform nearly identical to each other based on accuracy and the macro averages of F1-score, recall and precision when decreasing the training set size. The only performance drop-off point was found by decreasing the train set from 80% to 70%.

This thesis has been structured as follows. In the second chapter, the related works will be discussed. The third chapter will contain an overview of the dataset and will explain the methods used in the experiments. In the fourth chapter, the experimental setup for the three experiments is covered. The fifth chapter summarizes the results of these experiments. These results are then discussed in chapter six. To finalize this thesis, chapter seven provides the conclusion.

Related Work

This chapter includes an overview of the importance of churn prediction and both common and advanced ML algorithms that have been used in earlier researches on this subject. Next to that, it elaborates on dimensionality reduction techniques and the overall shortcomings of existing research on churn prediction.

Importance of Churn Prediction

The main motivation for research on churn prediction is that the attraction of new customers in saturated markets, such as telecommunications, costs companies more than the retention of existing customers (Adhikary & Gupta, 2021; Amin, et al., 2017; Kaya, et al., 2018). Being able to very accurately predict all customers who have a high tendency to churn, results in the ability for companies to intervene at the right time to retain these customers (Ahmed & Maheswari, 2017). By retaining more customers, sales figures tend to rise while the marketing costs are reduced (Amin, et al., 2017). These benefits have resulted in customer churn prediction being an important tool in the decision-making process of many companies that operate in saturated markets (Amin, et al., 2017).

Frequently Used Algorithms

An algorithm that has been used to predict customer churn is SVM. The benefit of this algorithm is that it can be used for both regression and classification. However, one of its weaknesses is a large dataset, since this classifier requires a relatively long training time (Rajamohamed & Manokaran, 2018). Vafeiadis et al. (2015) investigated the difference between boosted and non-boosted classification algorithms for customer churn prediction. This study was conducted on a churn dataset that consisted of 5000 records on 18 mostly numerical variables and one target variable. There was no dimension reduction technique used in this research. It was found that especially the boosted variant of SVM with a polynomial kernel reached a very high accuracy of 96% and an F1-score of 80%. With these

scores, the algorithm outperformed classifiers such as NB and DT in this particular research (Vafeiadis, Diamantaras, Sarigiannidis, & Chatzisavvas, 2015). Next to that, Gordini and Veglio (2017) used SVM based on the Area Under the Curve (AUC)-parameter selection technique (SVMauc), to create a churn predicting model in the e-commerce sector. This study was conducted on an Italian online fast-moving goods company dataset that consisted of 40,000 records on roughly 30 numerical variables and one target variable. No dimension reduction was used in this research. It was ultimately found that this type of setup performed good on imbalanced data that is noisy and non-linear. Next to that, it also generalizes well (Gordini & Veglio, 2017).

Previous research on churn prediction also explores k-NN. This algorithm yields high accuracy, is easy to understand and is useful on non-linear data. However, a downside of this algorithm is that its use can become computationally expensive on large datasets, since it stores all data used for training (Keramati, et al., 2014). Bhatnagar and Srivastava (2019) specifically used k-NN to create a rough model to compare it to Linear Regression (LR). This study used a telecommunications dataset of 3336 records on 18 numerical features. Here, k-NN proves to be the favorable approach over LR when it comes to churn prediction by scoring nearly a perfect recall and 2% higher on accuracy. No dimension reduction was used in this research (Bhatnagar & Srivastava, 2019).

Advanced Approaches

An approach that has grown in popularity in recent studies is the hybrid classifier. This is a combination of single classifiers, created to eliminate the weaknesses of the single classifiers used as building blocks (Rajamohamed & Manokaran, 2018). Earlier research tested the algorithms mentioned in the previous sub-section as single classifier and also combined K-Means clustering with DT, NB, k-NN and SVM. This study used no dimension reduction and was conducted on a Taiwanese banking dataset consisting of 30,000 records on 23 mixed type

variables. Here, the data was clustered first and then the algorithms were trained and tested per cluster. For most techniques, creating such a combination of classifiers led to a performance increase based on accuracy of around 10% (from ±80% to ±90%), but only with SVM the difference in performance between hybrid (95%) and single (93%) classifier was rather small (Rajamohamed & Manokaran, 2018).

De Caigny et al. (2018) proposed an approach where DT was combined with LR to eliminate the incapability of handling linear relations that DT has. LR was chosen because it is capable of handling these linear relations, but struggles to handle the interaction effects. This study was conducted on fourteen mixed type datasets of which the industry differs. The smallest telecom dataset used had 47,761 records on 43 features. The greatest dataset used was from the financial industry and had around 600,000 records on 232 features. Fisher's score was used to reduce the dimensionality of every dataset to at most 20 features. The combination of DT and LR (a hybrid approach) significantly outperformed its building blocks in this research based on the AUC (de Caigny, Coussement, & de Bock, 2018).

Next to hybridization of the more standard classifiers, Deep Learning in the form of Neural Networks (NN) have been used as well to find a more efficient system to predict customer churn. However, this approach is more complex and computationally expensive (Zikria, Afzal, Kim, Marin, & Guizani, 2020). Sivasankar & Vijaya (2019) conducted a research on a churn dataset consisting of 100,000 records on 172 mixed features. This study showed that creating a hybrid model by adding probabilistic possibilistic fuzzy C-means clustering to a NN, outperformed Neural Networks by itself and other existing churn prediction methods. No dimension reduction techniques were used (Sivasankar & Vijaya, 2019).

Dimensionality Reduction in Churn Research

Dimensionality reduction is a technique that is used to transform high-dimensional data into lower-dimensional data, while most important properties of the data are being retained

(Nguyen & Holmes, 2019). Although dimensionality reduction is not used in most of the researches stated in this chapter so far, there have been researchers who explored the usefulness of it. For instance, Fathian et al. (2016) used PCA to reduce the dimensionality of a churn dataset which had 40,000 records on 76 features. This research explored hybrid approaches based on boosting and bagging in churn prediction. It was found that a combination of clustering, PCA and boosting resulted in a better performance than using single classifiers (Fathian, Hoseinpoor, & Minaei-Bidgoli, 2016). Next to this, De Bock & van den Poel (2011) used PCA, Independent Component Analysis (ICA) and Sparse Random Projections (SRP) in an evaluation of rotation-based classifiers that predict churn. The biggest dataset used in this research is a telecom dataset of roughly 35,500 records on 529 features. It was found that, based on AUC, rotation forests in combination with ICA outperformed all other classifiers considered in this research (de Bock & van den Poel, 2011).

Shortcomings in Existing Research

In addition to the machine learning techniques used in churn prediction, this thesis mainly focuses on the combination of data availability and machine learning and at what point the performance of ML models drops off. This is a subject that seems to be underexposed in the current research field on churn prediction. This chapter indicated that datasets with different sizes have been used in earlier researches but in most cases, only one dataset with a fixed size was used per research.

Next to that, existing literature on churn prediction does not go further than stating that various evaluation metrics do not perform up to par when classifiers are applied to a small dataset and therefore, a combination should be used (Jain, Khunteta, & Srivastava, 2021). However, in other branches such as image recognition, research has recently been published on the impact of small datasets on Neural Networks, concluding that small datasets result in worse results than big datasets, and that a new tailored methodology should be adopted to

work with these small datasets (Pastor-López, et al., 2021). To widen the perspective of this last research and to add valuable insights to the existing research on churn prediction, the impact of data availability on machine learning classifiers predicting customer churn should be investigated.

Methodology

This methodology chapter introduces the dataset, necessary pre-processing steps and an ethics statement. Then, the methods are explained which are used in the experiments that follow in chapter four. This section ends with two sub-sections respectively covering the evaluation metrics and hyperparameter tuning.

Dataset Description and Pre-Processing

The dataset used to conduct the experiments is the Telco customer churn dataset from IBM, which has been publicly accessible via Kaggle since July of 2019. It contains fictional data on Californian customers of a telecommunications business. The data has been uploaded to Kaggle in a folder, which contains one big dataset and five smaller datasets on different subjects regarding the customers. The subjects are demographics, location, population, services and status. These smaller datasets are all subsets of the big dataset but some of these subsets have additional features that are not included in the big dataset. The data is available in .xlsx format (IBM Cognos Analytics, 2019). To complete the big dataset, a combined total of eleven features were added from the demographics and services subsets. In addition, nine features were removed from the big dataset, because those had the same value for all records or because those were simply a duplicate of another feature. Records were not added nor removed. The changes ultimately leave a dataset with 7043 records and 35 features, of which 34 predictors and one target variable (churn value) as can be seen in Table 1. This dataset will be referred to as the main dataset and complete descriptions of all features are in Appendix A.

Main Dataset Description							
Number	Name	Records	Type	Number	Name	Records	Type
0	Zip Code	7043	Nominal	18	Paperless Billing	7043	Binary
1	Latitude	7043	Continuous	19	Payment Method	7043	Nominal
2	Longitude	7043	Continuous	20	Monthly Charges	7043	Continuous
3	Gender	7043	Binary	21	Churn Value	7043	Binary
4	Senior Citizen	7043	Binary	22	Churn Score	7043	Continuous
5	Partner	7043	Binary	23	Customer Life Time Value	7043	Continuous
6	Dependents	7043	Binary	24	Age	7043	Continuous
7	Tenure Months	7043	Continuous	25	Married	7043	Binary
8	Phone Service	7043	Binary	26	Number of Referrals	7043	Continuous
9	Multiple Lines	7043	Nominal	27	Avg Monthly GB Download	7043	Continuous
10	Internet Service	7043	Nominal	28	Streaming Music	7043	Binary
11	Online Security	7043	Nominal	29	Unlimited Data	7043	Binary
12	Online Backup	7043	Nominal	30	Total Refunds	7043	Continuous
13	Device Protection	7043	Nominal	31	Total Extra Data Charges	7043	Continuous
14	Tech Support	7043	Nominal	32	Total Long Distance Charges	7043	Continuous
15	Streaming TV	7043	Nominal	33	Total Charges	7043	Continuous
16	Streaming Movies	7043	Nominal	34	Total Revenue	7043	Continuous
17	Contract	7043	Nominal				

Table 1: Main Dataset Description

Data source/code/ethics statement

Work on this thesis did not involve collecting data from human participants. The original owner of the data used in this thesis retains ownership of the data during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data. All figures and tables in this thesis have been created by the author of the thesis. The code used in this thesis is publicly available in the following repository: https://github.com/FHZ1997/Thesis_code.

Synthetic Minority Oversampling Technique for Numerical and Categorical Data (SMOTENC)

The chosen customer churn dataset is imbalanced and consists of mixed data types. To counter the data imbalance and take the mixed data into account, oversampling in the form of SMOTENC will be used. This is an oversampling technique that has been created specifically for datasets that contain both numerical and categorical data. By using SMOTENC, the minority class of the target variable (the churners) is enlarged to the size of the majority class, by the generation of synthetic examples as can be seen in Figure 1 (Imbalanced Learn, 2021).

These synthetic examples are not exactly the same as the original ones, which counters the chance of overfitting the models (Thabtah, Hammoud, Kamalov, & Gonsalves, 2020).

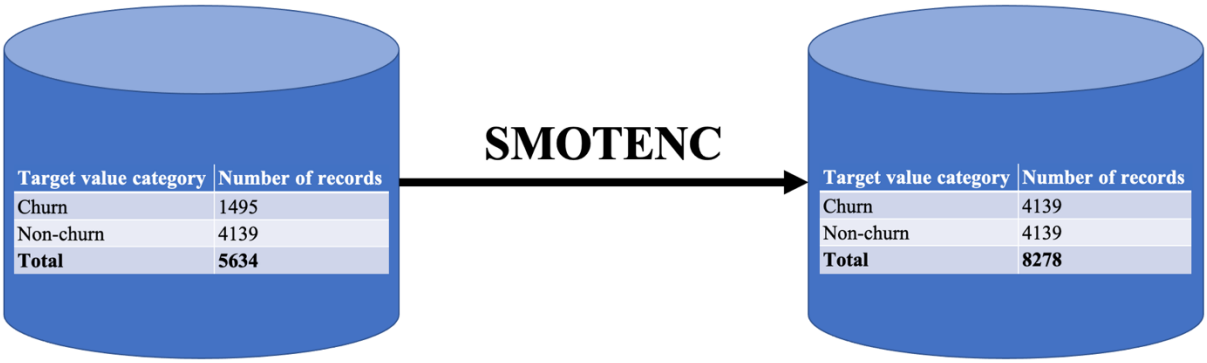


Figure 1: The SMOTENC operation

Factor Analysis of Mixed Data (FAMD)

This dimensionality reduction method enables feature reduction in a dataset where both categorical and continuous data exist (Pages, 2004). Therefore, it is used in this thesis. FAMD can be seen as a combination of PCA, which should only be used for continuous features, and Multiple Correspondence Analysis (MCA) which is only applicable to categorical features (Abdi & Williams, 2010). FAMD works in a similar way by scaling the continuous features to the variance of units, transforming the categorical features into separate tables and then scaling these as in MCA. This way, categorical and continuous variables have a balanced influence in FAMD (Pages, 2004).

Principal Component Analysis (PCA)

The other dimensionality reduction method used in this thesis is PCA. Even though this technique is theoretically not applicable to mixed data (Abdi & Williams, 2010; Nguyen & Holmes, 2019), earlier Kaggle submissions on this dataset all used PCA after creating dummy variables (Kaggle, 2021). With PCA, all categorical variables have to be converted into numerical or binary variables. Processing categorical variables like this is not preferred because qualitative information about the variables is lost, which results in a less meaningful analysis of the variables (Nguyen & Holmes, 2019).

Support Vector Machines

SVM is used in this thesis-project, because previous research shows that this algorithm is often one of the greatest performing standard machine learning techniques used for churn prediction (Gordini & Veglio, 2017; Rajamohamed & Manokaran, 2018). In addition, the previously discussed disadvantage of SVM (see chapter two) is negligible in for this thesis since the dataset used is not large.

Support Vector Machines are supervised machine learning algorithms, which can be used for both classification and regression. This algorithm seeks to classify linear or non-linear data by finding a hyperplane that best separates the data from each class (Vapnik, 1995). Whenever data is linear, SVM classifies the data points by outputting a line that maximizes the distance to the nearest element of each category. If data is non-linear, the output is a plane in a three-dimensional space which aims for the same result as the line. The exact shape of the line or plane depends on the data and the kernel function used (Vapnik, 1995).

K-Nearest Neighbors

This method is used in this thesis-project because it classifies inputs in a different way than Support Vector Machines and has yielded great results in previous research (Bhatnagar & Srivastava, 2019; Rajamohamed & Manokaran, 2018). Next to that, just as with SVM, the algorithm can handle the chosen dataset and the disadvantage of this technique that has been discussed in chapter two is negligible due to the relatively small size of the chosen dataset.

K-Nearest Neighbors is a lazy supervised machine learning algorithm, that can be used for both classification and regression. It is ‘lazy’ in the sense that it simply memorizes the inputs, instead of learning a discriminative function. When this algorithm has to predict the class of a certain observation, it searches its memory for an input which is closest to the observation that has to be predicted based on a certain distance metric. When it finds one, it

assigns the class of the memory observation to the input that needs to be predicted (Keramati, et al., 2014).

K-Means Clustering

This clustering technique is used in this thesis to enhance the performance of the single classifiers SVM and k-NN. By clustering the data, similar datapoints are given the same cluster label. Due to the similarity of the datapoints in the clusters, the classification algorithms used in this thesis can be trained more efficiently and possibly reach higher scores (Rajamohamed & Manokaran, 2018).

The K-Means clustering algorithm is used for unsupervised clustering of data. It does this by dividing data points into a pre-specified number of groups (k). These groups are formed in such a way that the sum of squares within each cluster is minimized (Rajamohamed & Manokaran, 2018). Table 2 explains the general workflow of a K-Means algorithm step-by-step. In experiment two and three of this research, K-Means is used to cluster data that is similar, before the supervised classification algorithms are applied. These experiments consider hybrid classifiers created by combining K-Means with respectively SVM and k-NN.

K-Means algorithm workflow

Step 1	Determine the number of clusters
Step 2	Select k clusters randomly
Step 3	Measure Euclidean distance between points and centroid
Step 4	Assign each point to nearest cluster
Step 5	Calculate mean of formed clusters and create a new centroid
Step 6	Repeat step 3 to 5 until the new centroids do not differ from the old ones or until the maximum number of iterations has been completed

Table 2: The K-Means Algorithm Workflow

Stratified K-Fold Cross Validation

This method will be used to split the data into k parts where k-1 parts will be used for training and the remaining one for validation (Wu, Yau, Ong, & Chong, 2021). This is done to test the performance of the ML models on unseen data. Figure 2 displays an example where k = 10, which is also the value for k that will be used in this thesis. Wu et al. (2021) performed similar research, and there the value of k had been set to 10. By splitting the data as mentioned above, the models can be validated extensively even though the amount of data is limited. To ensure that each fold includes the same percentage of the target class as the original dataset, a stratified version of k-fold cross validation is used. This enables even training and validation among all folds. (Thabtah, Hammoud, Kamalov, & Gonsalves, 2020).

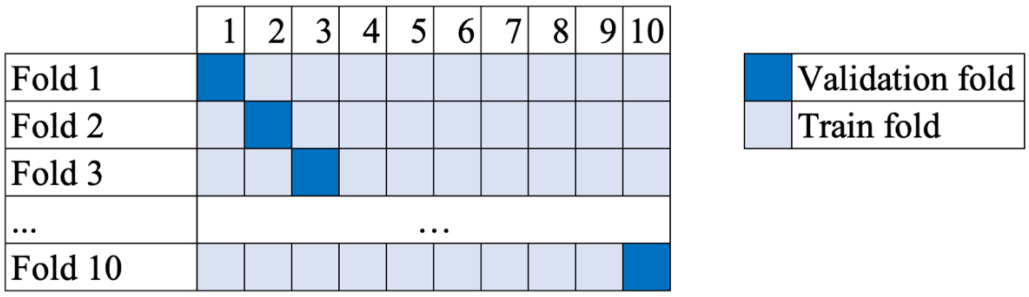


Figure 2: 10-fold cross validation

Evaluation metrics

The performance of all classifiers in each experiment will be measured with the same evaluation metrics. These metrics can be derived from confusion matrices. Although each experiment will have different outcomes for the confusion matrix, the general structure will be the same and can be found in Table 3. In this table, O_{11} stands for the customers who are churners and are predicted likewise. Next to that, O_{12} are actual churners predicted otherwise, O_{21} are actual non-churners predicted as churners and finally O_{22} are non-churners predicted as non-churners.

		Predicted	
		Churn	Non-Churn
Actual	Churn	O_{11}	O_{12}
	Non-Churn	O_{21}	O_{22}

Table 3: Confusion Matrix

The evaluation metrics used in this research are (macro) F1-score, recall, precision and accuracy. The macro variant calculates an average of the class labels while giving each class the same importance (Sklearn developers, 2021). These metrics enable the analysis and comparison of the performances of all classifiers in this research. The aim is to maximize the F1-score, but this should not be drastically at the expense of the other three metrics. Therefore, recall, precision and accuracy are taken into careful consideration in the evaluation of the classifier performances. Based on the confusion matrix, the four metrics mentioned above can be calculated. These calculations are displayed in equations one to four.

$$F_1 score = \frac{O_{11}}{O_{11}+\frac{1}{2}(O_{21}+O_{12})} \quad (1)$$

Equation 1: F1-score

$$Recall = \frac{O_{11}}{O_{11}+O_{12}} \quad (2)$$

Equation 2: Recall

$$Precision = \frac{O_{11}}{O_{11}+O_{21}} \quad (3)$$

Equation 3: Precision

$$Accuracy = \frac{O_{11}+O_{12}}{O_{11}+O_{12}+O_{21}+O_{22}} \quad (4)$$

Equation 4: Accuracy

Hyperparameter tuning

To maximize the performance of the models based on F1-score, the hyperparameters of the classifiers will have to be tuned. This specific type of parameter cannot be derived from the data by the classifier itself and therefore, it has to be done 'manually'. To tune these parameters, GridSearchCV from the 'sklearn' package will be used. It works by putting in the values to be tested per hyperparameter and then GridSearchCV will provide prediction scores considering

all possible value combinations (Sklearn developers, 2021). The cross-validation method used in this approach is the stratified 10-fold cross validation technique explained in this chapter. The parameters to be adjusted for the k-NN classifiers are the number of neighbors (n_neighbors), the distance metric and the different weightings of members in a certain 'neighborhood' (weights). The hyperparameters to be adjusted for SVM classifiers, are the kernel function and the penalty coefficient (C). The hyperparameter values that are considered by GridSearchCV for SVM and k-NN can be found in respectively Table 4 and Table 5.

Hyperparameters SVM

Parameter	Possible Values
C	[100; 10; 1.0; 0.1; 0.01]
Kernel	['Radial basis function', 'Linear', 'Polynomial', 'Sigmoid']

Table 4: Hyperparameter Values SVM

Hyperparameters k-NN

Parameter	Possible Values
N_neighbors	Odd values in range [1, 21]
Distance metric	['Manhattan', 'Euclidean', 'Minkowski']
Weights	['Distance', 'Uniform']

Table 5: Hyperparameter Values k-NN

The number of neighbors and the C value are continuous in nature. So, to limit the computational expense and the required time that is associated with it, the chosen values are a logarithmic scale for C and the odd values between one and twenty-one neighbors.

The 'k' in K-Means is another hyperparameter to be tuned. It will be tuned according to the Silhouette coefficient. This is a value that indicates for every data point how well it fits

into the cluster that it has been assigned to. It does so by comparing the coherence within each cluster to the cluster separation. It produces a value that reaches from -1 to 1, where a high score means that a data point is well matched to its own cluster and badly to other clusters (de Amorim & Henning, 2015).

Experimental Setup

This chapter will cover the details of the baseline model and the three experiments, divided over four sub-sections. These experiments provide the insights required to answer the sub research questions that have been developed in chapter one. Therefore, the findings of these experiments ultimately lead to the answer of the main research question.

Baseline model

Before the start of the first experiment, the data is split into predictors (x) and target variable (y) and a baseline model is created, using the DummyClassifier package of 'sklearn'. This package classifies the majority class for all observations and the outcome will be compared to all model performances in the following experiments.

Impact of dimensionality reduction on single classifiers

The main workflow of this first experiment is visualized in Figure 3. First, a train/test split following an 80%/20% ratio is conducted on the main dataset using 'train_test_split' from 'sklearn.model_selection'. Due to the imbalanced nature of churn dataset, SMOTENC from the 'imblearn' package is then used to oversample the minority class of the train set. Next, the dimensionality of the data will be reduced by either FAMD or PCA. For FAMD the data is imported in R were FAMD will be conducted with the packages 'FactoMineR and 'factoextra'. To determine the impact of reducing the dimensionality, the results of the FAMD are divided into five datasets of which the cumulative variance explained decreases (see Table 6). Then the results are transferred back to Python.

Percentage of total variance explained per number of dimensions

Dimensions	43	23	16	11	7
Variance explained	100%	90%	80%	70%	60%

Table 6: Percentage of total variance explained per amount of dimensions

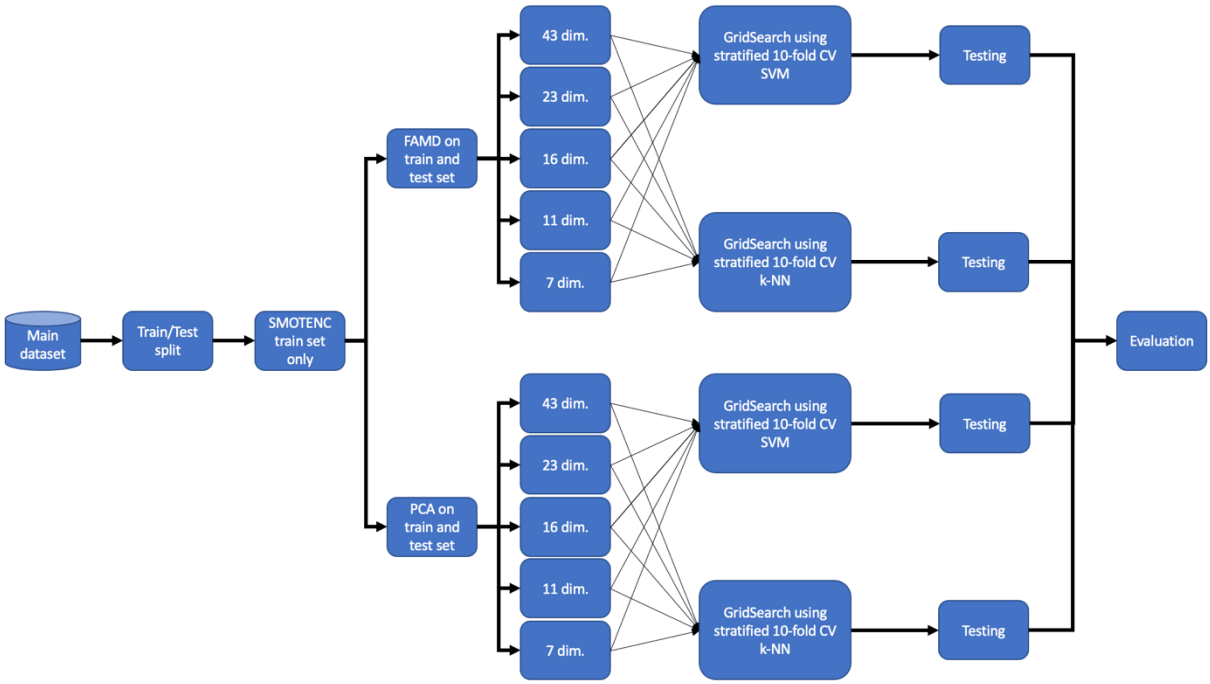


Figure 3: Workflow of the first experiment

For PCA, the data stays in Python and the 'sklearn_decomposition' package will be used. The PCA results are divided in the same way as the FAMD results. In Python, the hyperparameters are tuned per model and per dataset using the Stratified 10-fold cross validation of GridSearch CV, following the approach and values described in the previous chapter. This hyperparameter tuning is done in order to maximize the F1-score without sacrificing much on the other evaluation metrics. By using a cross validation method, the performance of the single SVM and k-NN classifiers is validated on validation sets. When the right hyperparameters for each classifier are found, the performance per dataset is tested on the test sets. The confusion matrices and evaluation metrics associated with each of the ten models created, are compared with the baseline model and stored so that they can be used as benchmarks for other models later in this study. In the other experiments, only FAMD will be used to reduce dimensionality of the dataset since this is the most suitable according to the literature and yields the average performance for SVM and k-NN (Nguyen & Holmes, 2019; Pages, 2004).

Comparison of single classifiers and hybrid classifiers enriched with k-means

The second experiment investigates the difference between single and hybrid classifier models by enriching single SVM and K-NN with K-Means clustering to create hybrids. This experiment is identical to experiment one until the FAMD that reduces the data to 16 dimensions (see Figure 4).

The optimal value for the Silhouette coefficient is then calculated and determines that the data will be divided into two clusters when conducting the K-Means algorithm. The K-Means algorithm is then fitted on the train set which creates the cluster labels for the train set. These labels are then used to predict the cluster labels of the test set using k-means to ensure that no test data leaks. When the clusters labels for both the train and test set are available, they are added as a column in the corresponding dataset. Important to note is that the target variable is not used in the formation of the clusters. Subsequently, the hyperparameters are tuned per model and per dataset using the stratified 10-fold cross validation of GridSearchCV following the approach and values described in the previous chapter. This also validates the performance of the hybrid SVM and k-NN classifiers.

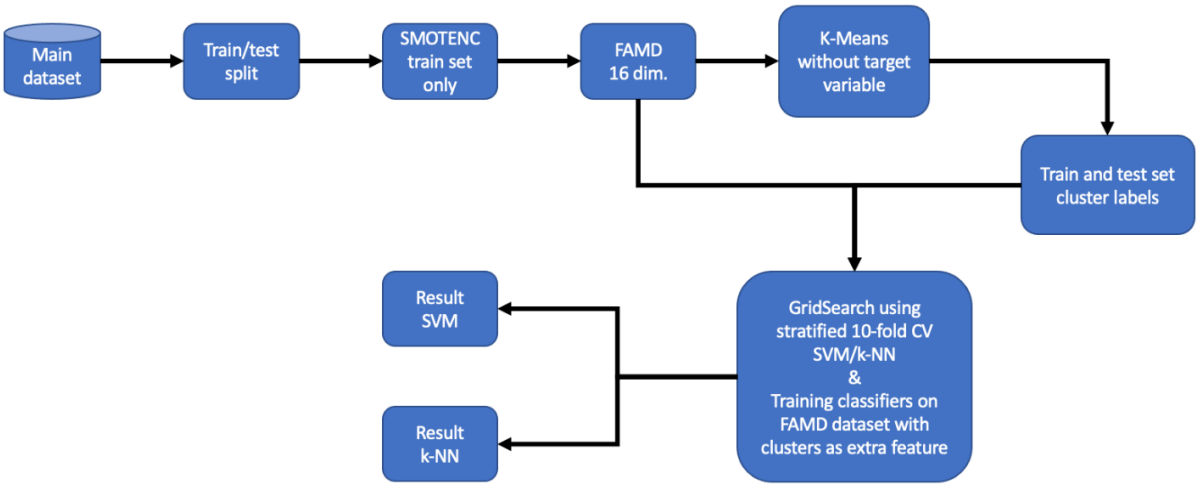


Figure 4: Workflow of the second experiment

After the tuning and validation, the classifiers are trained on the train set (with cluster labels) and set to predict the test set (with cluster labels). The chosen approach ensures that

the two trained models are as complete as possible. The confusion matrices and associated evaluation metrics are then compared with the performance of the models from the first experiment and the baseline, which delivers insights on the difference between single and hybrid classifiers for churn prediction.

Performance Comparison of Single and Hybrid Classifiers in Data Limiting Circumstances

The third and final experiment investigates the influence of data availability on the performance of both single and hybrid classifier ML models. The approach to create the single classifier models in this experiment is nearly identical to the first experiment. The first difference is that the train set (80%) is split into seven more datasets (see Figure 5) to decrease the amount of data in the train set, used to create the single classifiers. The second difference is that the classifiers all have to predict the same test set, that was created with the first 80%/20% split. This is the most logical method for comparison purposes. The differences between the third and the first experiment, also count for the differences between the third and the second experiment. The hybrid classifier models are also created on eight different train sets and are all set to predict the same test set. Just as described above, the 80% train set is the set of which the other subsets are taken from.

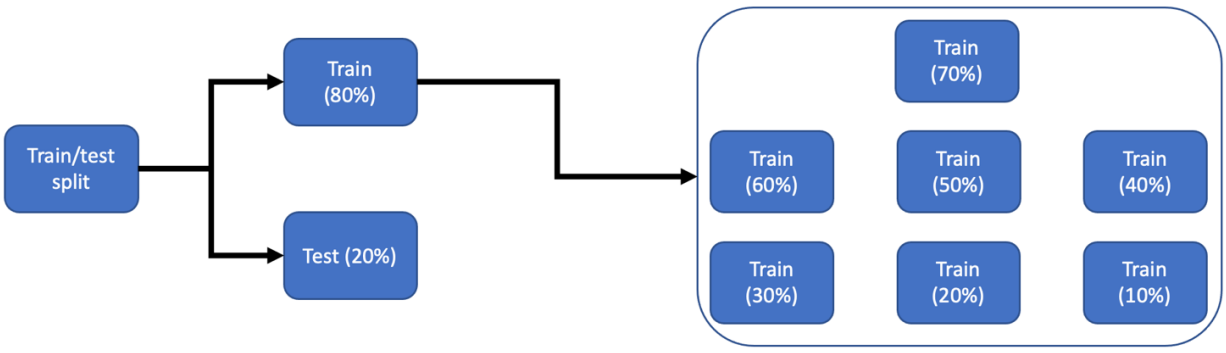


Figure 5: Train/test split operation of the third experiment

When both the single and hybrid classifiers have been created, the performances are compared to each other and to the baseline, based on the previously discussed evaluation

metrics. This final comparison gives insight in the behavior of single and hybrid classifier machine learning models in situations where the data availability decreases. It describes more accurately than available research when machine learning can be applied and when there is simply not enough data to apply it.

Results

This chapter covers the results of the baseline model, hyperparameter tuning and the three experiments, divided over five sub-sections.

Baseline performance

In this first sub-section, the classification performance of the baseline algorithm described in chapter three on the main dataset will be presented. The chosen parameters for the baseline model have resulted in the majority class being predicted for all records in the main dataset and this translates to the evaluation metrics stated in Table 7. The classes of the target variable 'churn value' have been used as column names. One finding to point out is that in this situation, the recall for the non-churners is 1.0 since these were all predicted correctly. But the recall, precision and F1-score for churners is zero since these were all predicted incorrectly. Despite these zero values, the baseline classifier does yield an overall accuracy score of 0.73.

Baseline classifier performance

	Churner	Non-churner	Macro average
F1-score	0.00	0.85	0.42
Recall	0.00	1.00	0.50
Precision	0.00	0.73	0.37
Accuracy		0.73	

Table 7: Baseline Classifier Performance

Hyperparameter tuning results

In this second sub-section, hyperparameter tuning results of the classifiers of all experiments (see chapter four) will be presented. All classifiers in this research have been tuned individually with GridSearchCV and a standard set of possible values (see chapter three) to maximize F1-score. The tuned hyperparameters for the SVM and k-NN classifiers are stated in Table 8 to Table 13. In Tables 8, 10 and 12, 'Rbf' represents Radial Basis

Function. In Tables 9, 11 and 13‘Ma’ represents Manhattan, ‘Eu’ stands for Euclidean, ‘Dis’ stands for Distance and ‘Uni’ stands for Uniform.

Hyperparameters SVM

	Experiment 1				
Dimensions	43	23	16	11	7
C	1.0	10	10	100	100
Kernel	Rbf	Rbf	Rbf	Rbf	Rbf

Table 8: Hyperparameters of SVM models in the first experiment

Hyperparameters k-NN

	Experiment 1				
Dimensions	43	23	16	11	7
Distance metric	Ma.	Eu.	Eu.	Ma.	Ma.
N_neighbors	5	5	7	7	17
Weight	Dis.	Dis.	Dis.	Dis.	Dis.

Table 9: Hyperparameters of k-NN models in the first experiment

Hyperparameters SVM

	Experiment 2
C	10
Kernel	Rbf

Table 10: Hyperparameters of SVM models in the second experiment

Hyperparameters k-NN

	Experiment 2
Distance metric	Eu.
N_neighbors	7
Weight	Dis.

Table 11: Hyperparameters of k-NN models in the second experiment

Hyperparameters SVM

	Experiment 3							
Train set size	80%	70%	60%	50%	40%	30%	20%	10%
C	10	10	10	10	10	10	10	1.0
Kernel	Rbf	Rbf	Rbf	Rbf	Rbf	Rbf	Rbf	Rbf

Table 12: Hyperparameters of SVM models in the third experiment

Hyperparameters k-NN

	Experiment 3							
Train set size	80%	70%	60%	50%	40%	30%	20%	10%
Distance metric	Eu.	Eu.	Eu.	Eu.	Ma.	Ma.	Eu.	Ma.
N_neighbors	7	9	7	11	15	7	5	9
Weight	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Dis.	Uni.

Table 13: Hyperparameters of k-NN models in the third experiment

In the second and third experiment, K-Means is used to cluster the dimensionality reduced data by fitting on the train set, and predicting the cluster labels of the test set. Tuning the number of clusters was done by finding the highest value of the Silhouette coefficient. The results are displayed per number of clusters in Table 14. The coefficient score gradually decreases whenever the number of clusters increases.

Silhouette coefficient

Number of clusters	Score
2	0.29
3	0.23
4	0.23
5	0.18
6	0.18
7	0.13
8	0.13

Table 14: Silhouette coefficient per number of clusters

Single classifier performances

In this third sub-section, classification performance of respectively the SVM and k-NN algorithm of experiment one, described in chapter four, on the FAMD and PCA datasets with varying dimensions will be presented.

The SVM and k-NN classifiers are used to predict the five FAMD and five PCA test-sets with varying dimensions. Figure 6 consists of two graphs that compare classification performances based on the macro average of the F1-score. The graph on the left indicates that

the single classifier SVM performs gradually worse when the number of dimensions decrease. The graph on the right indicates that the single classifier k-NN performs gradually better when the number of dimensions decrease. Both graphs show that using PCA to reduce dimensionality overall yields higher macro average F1-scores compared to when FAMD is used. For comparison purposes though, 16 FAMD dimensions will be used in the other experiments. This way, all following models will be made with the same number of dimensions. FAMD is chosen instead of PCA, because the literature indicates that this is the correct method to use in the case of mixed data (Nguyen & Holmes, 2019; Pages, 2004).

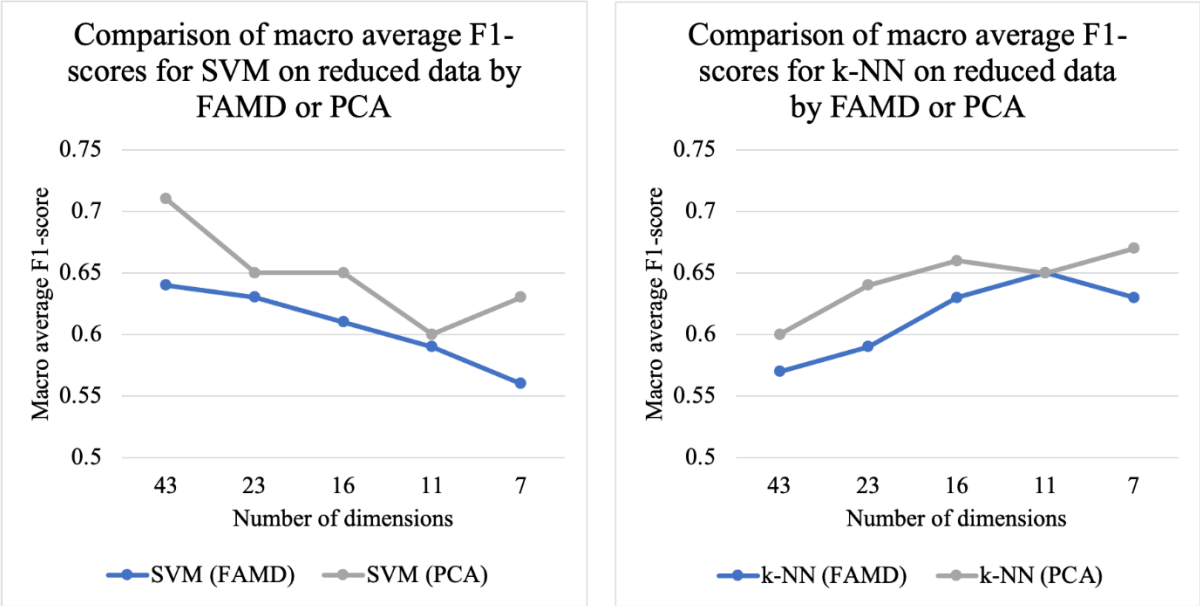


Figure 6: Macro average F1-score per number of dimensions and per classifier

A comparison of both single classifier models and the baseline can be seen in Table 15. Here and in Table 16, 'Ch' stands for Churn, 'N-Ch' stands for non-churn and 'M. Avg.' stands for macro average. When comparing the results of the models with 16 FAMD dimensions to the baseline, both classifiers perform worse on accuracy but significantly better on the macro averages. Based on the churner evaluation metrics, the classifiers cannot perform worse than the baseline. When comparing the classifiers to each other, it can be seen

that k-NN scores 10% better on churn recall while the rest of the metrics are roughly identical.

Results of SVM and k-NN single classifiers

	SVM 16 dimensions			k-NN 16 dimensions			Baseline 16 dimensions		
	Ch	**N-Ch**	**M. Avg.**	**C**	**N-Ch**	**M. Avg.**	**Ch**	**N-Ch**	**M. Avg.**
F1-score	0.50	0.71	0.61	0.54	0.71	0.63	0.00	0.85	0.42
Recall	0.69	0.61	0.65	0.79	0.59	0.69	0.00	1.00	0.50
Precision	0.39	0.84	0.62	0.41	0.89	0.65	0.00	0.73	0.37
Accuracy	0.63			0.65			0.73		

Table 15: Comparison of 16 FAMD dimension SVM and k-NN models and the baseline

Hybrid classifier performances

In this fourth sub-section, classification performance of respectively the SVM and k-NN algorithm, enriched with K-Means of experiment two described in chapter four on the dimensionality reduced dataset will be presented.

Results of SVM and k-NN hybrid classifiers

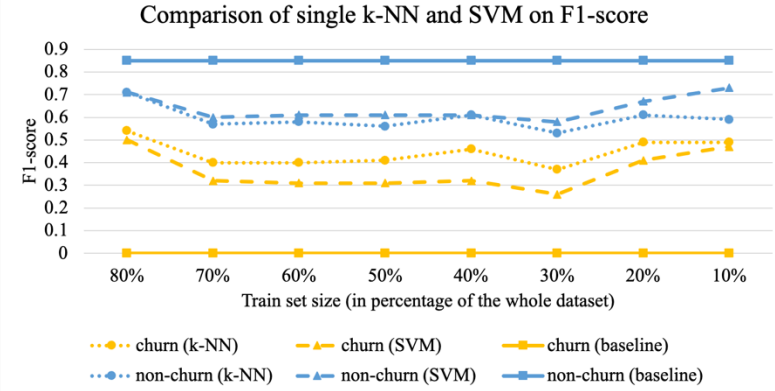
	Hybrid SVM			Hybrid k-NN			Baseline		
	Ch	**N-Ch**	**M. Avg.**	**Ch**	**N-Ch**	**M. Avg.**	**Ch**	**N-Ch**	**M. Avg.**
F1-score	0.50	0.71	0.60	0.56	0.71	0.64	0.00	0.85	0.42
Recall	0.68	0.61	0.65	0.84	0.59	0.71	0.00	1.00	0.50
Precision	0.39	0.84	0.62	0.42	0.91	0.67	0.00	0.73	0.37
Accuracy	0.63			0.65			0.73		

Table 16: Comparison of 16 FAMD dimension hybrid SVM and k-NN models and the baseline

K-NN seems to outperform SVM on the churn recall, non-churn precision and all macro averages when both are transformed to a hybrid classifier (see Table 16). These classifiers outperform the baseline method on the evaluation metrics related to churners and on the macro averages of the evaluation metrics. When compared to the single classifier versions created in experiment 1, the accuracy of SVM increases with 9%, while the other results of both single and hybrid k-NN and SVM yield roughly identical results.

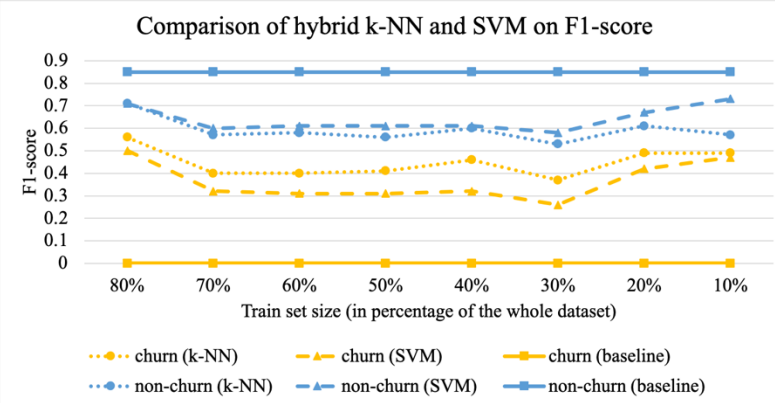
Single and Hybrid Classifier Performance with Varying Data Availability

In this final sub-section, classification performance of respectively the single and hybrid classifier k-NN and SVM algorithm on 80% to 10% of the main train dataset as described in experiment three (chapter four) will be presented. Figures 7 to 14 compare the classification performances of the single and hybrid models respectively on F1-score, recall, precision and accuracy. These figures indicate that both the single and hybrid classifier models perform nearly identical on all evaluation metrics. However, in most cases k-NN performs slightly better than SVM based on macro evaluation metrics.



Percentage	Macro average F1 k-nn	Macro average F1 svm	Macro average F1 baseline
80%	0.63	0.61	0.42
70%	0.49	0.46	
60%	0.49	0.46	
50%	0.48	0.46	
40%	0.53	0.46	
30%	0.45	0.43	
20%	0.55	0.54	
10%	0.54	0.60	

Figure 7: Comparison of single k-NN and SVM on F1-score with varying sized training samples

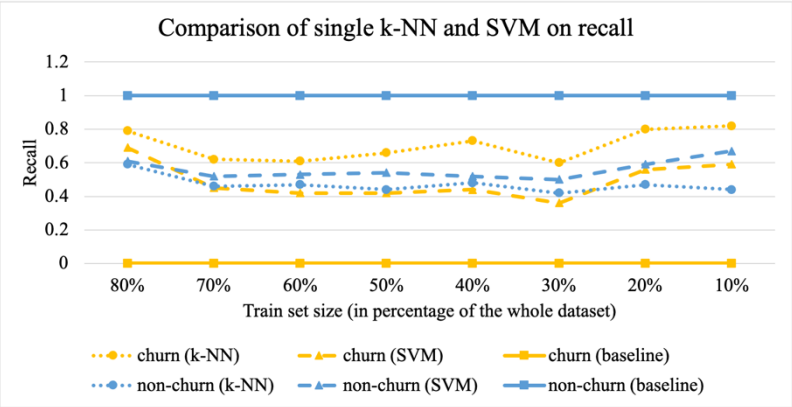


Percentage	Macro average F1 k-nn	Macro average F1 svm	Macro average F1 baseline
80%	0.64	0.60	0.42
70%	0.49	0.46	
60%	0.49	0.46	
50%	0.48	0.46	
40%	0.53	0.46	
30%	0.45	0.42	
20%	0.55	0.54	
10%	0.53	0.60	

Figure 8:Comparison of hybrid k-NN and SVM on F1-score with varying sized training samples

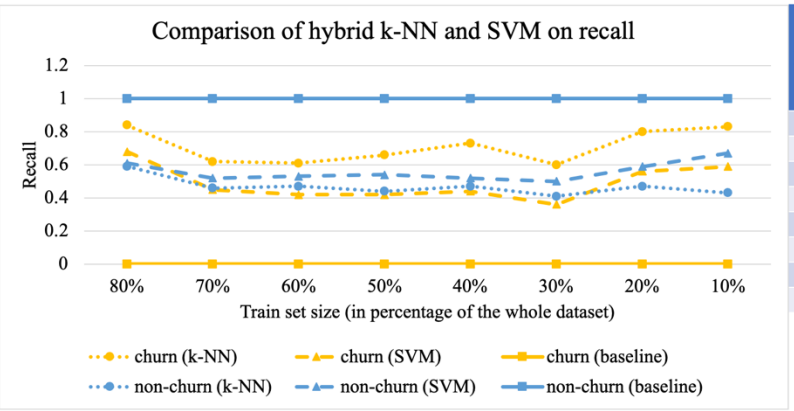
Both single and hybrid k-NN perform worse than the baseline when it comes to non-churn F1-score, but SVM slightly outperforms k-NN (see Figure 7 and 8). With regards to churn F1-score, both single and hybrid classifier models perform better than the baseline and

k-NN outperforms SVM. The overall trend seems to be that performance of both single and hybrid classifiers decreases until a train set of 30% and then increase again until a train set of 10%.



Percentage	Macro average recall k-nn	Macro average recall svm	Macro average recall baseline
80%	0.69	0.65	0.5
70%	0.54	0.49	
60%	0.54	0.47	
50%	0.55	0.48	
40%	0.6	0.48	
30%	0.51	0.43	
20%	0.64	0.57	
10%	0.63	0.63	

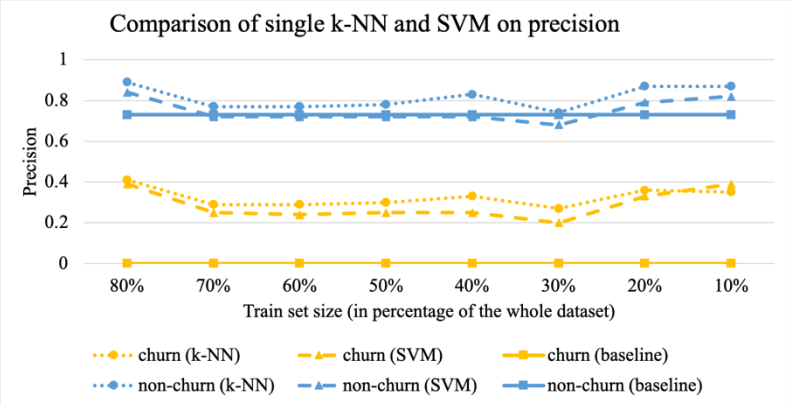
Figure 9:Comparison of single k-NN and SVM on recall with varying sized training samples



Percentage	Macro average recall k-nn	Macro average recall svm	Macro average recall baseline
80%	0.71	0.65	0.5
70%	0.54	0.48	
60%	0.54	0.48	
50%	0.55	0.48	
40%	0.6	0.48	
30%	0.51	0.43	
20%	0.64	0.57	
10%	0.63	0.63	

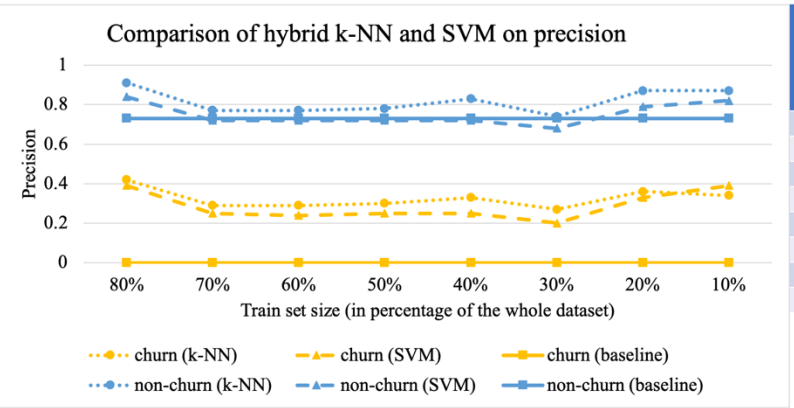
Figure 10: Comparison of hybrid k-NN and SVM on recall with varying sized training samples

Both single and hybrid classifiers perform better than the baseline when it comes to churn recall and k-NN outperforms SVM (see Figure 9 and 10). With regards to non-churn recall, both single and hybrid classifier models perform worse than the baseline and SVM outperforms k-NN. The overall trend seems to be that performance of both single and hybrid classifiers stays roughly equal until a train set of 30% and then slightly increases until a train set of 10%.



Percentage	Macro average precision k-nn	Macro average precision svm	Macro average precision baseline
80%	0.65	0.62	0.37
70%	0.53	0.49	
60%	0.53	0.48	
50%	0.54	0.48	
40%	0.58	0.49	
30%	0.51	0.44	
20%	0.61	0.57	
10%	0.61	0.6	

Figure 11: Comparison of single k-NN and SVM on precision with varying sized training samples



Percentage	Macro average precision k-nn	Macro average precision svm	Macro average precision baseline
80%	0.67	0.62	0.37
70%	0.53	0.49	
60%	0.53	0.48	
50%	0.54	0.48	
40%	0.58	0.49	
30%	0.51	0.44	
20%	0.61	0.56	
10%	0.61	0.6	

Figure 12: Comparison of hybrid k-NN and SVM on precision with varying sized training samples

Both single and hybrid classifiers perform better than the baseline when it comes to churn precision and k-NN outperforms SVM (see Figure 11 and 12). With regards to non-churn recall, single and hybrid k-NN perform better than the baseline and SVM is nearly identical to the baseline. The overall trend seems to be that performance of both single and hybrid classifiers stays roughly equal with a slight decrease at 30%.

Both single and hybrid classifiers perform worse than the baseline when it comes to accuracy (see Figure 13 and 14). Both single classifiers seem to follow the same trend with a performance decrease in the beginning, but an increase at 20% and 10%.

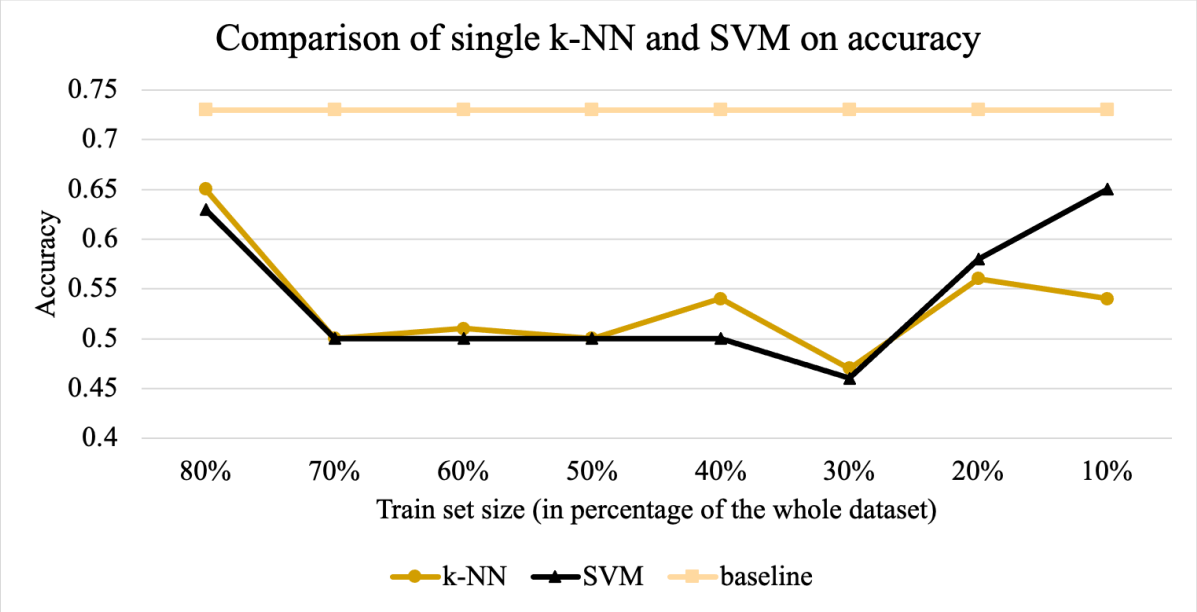


Figure 13: Comparison of single k-NN and SVM on accuracy with varying sized training samples

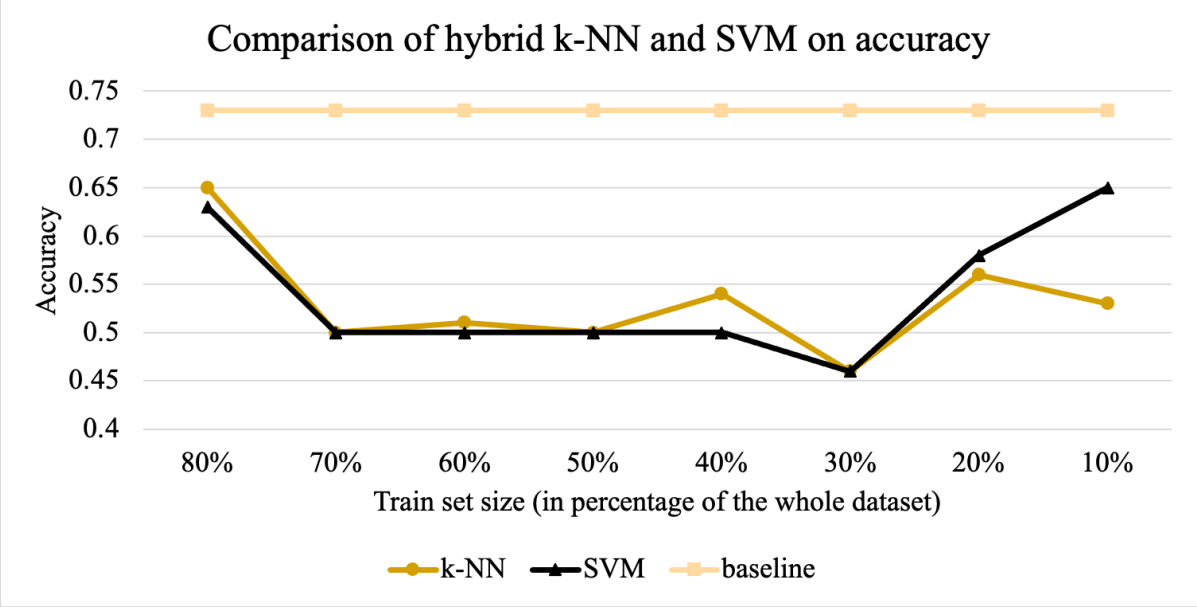


Figure 14: Comparison of hybrid k-NN and SVM on accuracy with varying sized training samples

Discussion

In this chapter, the research findings will be discussed in a structured manner. The overall goal of this research will be re-stated and the results of the previous chapter will be elaborated on per research question.

The goal of this thesis is to determine whether the use of ML techniques in churn prediction remain valuable when the data availability decreases. Investigating the added value of ML algorithms in these circumstances has potential practical benefits, such as the ability for businesses to perform analyses sooner, the opportunity for potential churners to receive personal offers and the ability for businesses to determine if churn prediction with ML can be worthwhile in their situation. To find out whether ML algorithms are valuable in circumstances where data is the limiting factor, the following main research question has been developed: *To what extent does data availability impact the performance of machine learning models in churn prediction?*

This question has been broken down into three parts, with the first part focusing on the impact of dimensionality reduction techniques on the predictive performance of single SVM and k-NN classifiers. This has been performed by means of an experiment involving FAMD and PCA to create multiple datasets with a varying number of dimensions.

This first experiment led to the finding that the predictive performance of k-NN is negatively correlated and that the predictive performance of SVM is positively correlated with the number of dimensions. Therefore, only k-NN's performance is in line with the literature that indicates that reducing dimensionality can lead to a reduction of noise in a dataset that allows the discovery of new hidden patterns in the dataset, which improve prediction scores (Nguyen & Holmes, 2019).

An interesting finding from this first experiment was that the single classifier models both reached higher scores on datasets that were reduced in dimensionality by PCA instead of FAMD. However, according to the literature, PCA is not meant for mixed datasets but only for continuous datasets (Abdi & Williams, 2010; Nguyen & Holmes, 2019). Next to that, an odd finding was that the total number of dimensions after FAMD and PCA (43) was higher than the number of variables in the dataset (34). As described in chapter three, FAMD and PCA work by compressing the size of the dataset by computing principal components (PC). This operation should decrease the size of the dataset, but the number of variables extend from 34 to 43. The possible cause for this could be the amount of noise in the dataset which makes the compression more difficult (Nguyen & Holmes, 2019).

In addition, it is striking that k-NN performed slightly better than SVM in this experiment, based on the considered evaluation metrics. This is not in line with the current literature, because it repeatedly shows that SVM outperforms k-NN when it comes to churn prediction (Rajamohamed & Manokaran, 2018; Vafeiadis, Diamantaras, Sarigiannidis, & Chatzisavvas, 2015). A possible explanation for this could be that the hyperplane in the SVM is not able to separate the data as well as in other researches, which could be caused by limitations of this dataset in the sense of not having enough predictive features that correlate heavily with the target variable. This ultimately leads to lower classification performance. To possibly increase this performance, finding predictive variables that correlate better with the target variable can be valuable, or using alternative ML classifiers that do not classify data by separating it with a plane or a line can be used.

The second part of this research addressed the question whether hybrid classifiers outperform single classifiers in churn prediction. This was investigated by means of an experiment in which K-Means clustering was used beforehand to split the dataset into clusters.

The results indicate that the hybrid classifiers perform nearly identical to the single classifiers in terms of almost all evaluation metrics considered in this thesis. This finding is not in line with the literature, where several studies indicate that hybrid classifiers outperform single classifiers on equivalent evaluation metrics (de Caigny, Coussement, & de Bock, 2018; Rajamohamed & Manokaran, 2018). A possible explanation for this could be that this thesis uses a different approach when it comes to creating hybrid classifiers. In the current literature for example, Rajamohamed and Manokaran (2018) use an approach in which unsupervised clustering procedures are applied to the entire dataset. After that, the dataset is split into clusters and then, per cluster, train and test splits are made to train and test the classifiers. However, the disadvantage of this method is that information leaks through clustering before the train/test split takes place. To prevent this from happening, a train and test split have been made in this thesis before applying K-Means to the dataset. The cluster labels of the test set are therefore predicted based on the labels of the train set, instead of being fitted as in the method that the literature describes (Rajamohamed & Manokaran, 2018). As a result, the test data remains truly unseen and therefore no information leaks that could possibly affect the classification performance.

The third and final part of this thesis was concerned with investigating the influence of data reduction on the classification performance of single and hybrid classifiers. The classifiers were trained on 80% to 10% of the main train dataset and all tested on the same 20% of the main dataset. The results of this experiment indicate that the classification performance of the single and hybrid classifiers is nearly identical for all train sets. The performance of all classifiers considered drops with the first train set decrease of 10%, then either stays equal or decreases slightly until a 30% train set. From that point, all performances increase.

It is interesting to note that the performance of all classifiers does not decrease constantly or has a clear drop-off point in experiment three. In fact, the performance even starts to increase from a train set of 20% and 10%. While common sense may lead to the idea that the performance of ML classifiers decreases with less training material, the following comparison of studies shows otherwise. For example, the aforementioned research into boosted and non-boosted SVM shows that an accuracy of 96% with a churn dataset can be achieved with only 5000 records (Vafeiadis, Diamantaras, Sarigiannidis, & Chatzisavvas, 2015). However, another study with a different approach, namely SVM based on the AUC parameter selection technique (SVMauc), shows that with a churn dataset of 40,000 records, an accuracy of 89% is the maximum achievable (Gordini & Veglio, 2017). It can be derived from this, that the size of the dataset does not have the sole influence on the performance of ML classifiers but for instance how a certain ML classifier is approached can also be a factor as shown above by Vafeiadis et al. (2015). In addition, another factor can be the composition of the dataset, which has been covered in recent research by applying one ML method to many different datasets (de Caigny, Coussement, & de Bock, 2018).

Based on the insights explained above, it can be concluded that the main question of this study has been partly answered. This thesis shows that single classifiers perform better when the dimensionality of the dataset is reduced by PCA than by FAMD. Next to that, it shows that hybrid classifier models do not outperform single classifier models when it comes to churn prediction in this particular research. Finally, more specific insights have been developed regarding the application of ML classifiers, when the number of records decrease. In this case, there was no clear performance drop-off point noticeable other than the one going from a train set size of 80% to a 70% train subset. Future research is recommended to build on this research by applying the same methodology to a real dataset. Since the dataset used in this thesis is a fictional customer churn dataset that is based on certain possible

factors (Kaggle, 2021), it is interesting to see what kind of results this methodology would achieve when the values could take on more different values in a real-world scenario.

Conclusion

The aim of this thesis has been to determine whether the use of ML techniques in churn prediction remains valuable when the data availability decreases. To do this, the performance of single and hybrid ML classifiers has been compared in circumstances where the dimensionality and the size of the dataset varied. In the first sub-question, it has been questioned how much, and what sort of impact dimensionality reduction has on the performance of SVM and k-NN. It was ultimately found that classifiers based on PCA outperformed single classifiers based on FAMD. Next to that, k-NN was negatively correlated with the decrease in dimensionality while SVM was positively correlated with the decrease in dimensionality. In the second sub-question, is has been questioned if enriching the aforementioned single classifiers with K-Means before classification, improved the performance. It was found that these hybrid classifiers perform nearly identical based on accuracy but k-NN performed slightly better than SVM on the macro averages of F1-score, recall and precision. In the third and final sub-question, the performance of single and hybrid k-NN and SVM on smaller datasets was questioned. Here it was found that there was no clear performance drop-off point noticeable other than one when going from a train set size of 80% to 70%. Performance even increased when there was only 20% or 10% of original training data available.

Future research is recommended to build on this research by applying the same structure and experiments of this thesis to other datasets and specifically real datasets. Since the dataset used in this thesis is a fictional customer churn dataset that is based on certain possible factors (Kaggle, 2021), it would be interesting to see what kind of results this methodology would achieve when the records could take on more different values in a real-world scenario or when there are more features that highly correlate to the target variable in order to investigate the true potential of ML classifiers. Another interesting idea for further research

would be to compare the outcomes of PCA and FAMD on different types of datasets to discover more about the inner workings of both dimensionality reduction methods. Finally, it is recommended to research the impact of data availability on the effectiveness of neural networks, since these are the current state-of-the-art classifiers when it comes to churn prediction.

Bibliography

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wires computational statistics*, 433-459.

Adhikary, D. D., & Gupta, D. (2021). Applying over 100 classifiers for churn prediction in telecom companies. *Multimedia Tools and Applications*, 35123–35144.

Ahmed, A. A., & Maheswari, D. (2017). Churn prediction on huge telecom data using hybrid firefly based classification. *Egyptian Informatics Journal*, 215-220.

Amin, A., Al-Obeidat, F., Shah, B., Adnan, A., Loo, J., & Anwar, S. (2019). Customer churn prediction in telecommunication industry using data certainty. *Journal of Business Research*, 290-301.

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., Hussain, A., & Huang, K. (2017). Customer churn prediction in the telecommunication sector using a rough set approach. *Neurocomputing*, 242-254.

Amin, A., Shah, B., Khattak, A. M., Moreira, F. J., Ali, G., Rocha, A., & Anwar, S. (2019). Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods. *International Journal of Information Management*, 304-319.

Bhatnagar, A., & Srivastava, S. (2019). *A Robust Model for Churn Prediction using Supervised Machine Learning.* IEEE.

de Amorim, R. C., & Henning, C. (2015). Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, 126-145.

de Bock, K. W., & van den Poel, D. (2011). An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. *Expert Systems with Applications*, 12293-12301.

de Caigny, A., Coussement, K., & de Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 760-772.

Devriendt, F., Berrevoets, J., & Verbeke, W. (2021). Why you should stop predicting customer churn and start using uplift models. *Information Sciences*, 497-515.

Fathian, M., Hoseinpoor, Y., & Minaei-Bidgoli, B. (2016). Offering a hybrid approach of data mining to predict the customer churn based on bagging and boosting methods. *Kybernetes*, 732-743.

Gordini, N., & Veglio, V. (2017). Customers churn prediction and marketing retention strategies. An application of support vector machines based on the AUC parameter-selection technique in B2B e-commerce industry. *Industrial Marketing Management*, 100-107.

Harzevili, N. S., & Alizadeh, S. H. (2018). Mixture of latent multinomial naive Bayes classifier. *Applied Soft Computing*, 516-527.

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

IBM Cognos Analytics. (2019). *datasets: Kaggle*. From Kaggle web site: https://www.kaggle.com/ylchang/telco-customer-churn-1113

Imbalanced Learn. (2021). *Oversampling, SMOTENC: Imbalanced learn*. From Imbalanced Learn Web site: https://imbalanced-learn.org/dev/references/generated/imblearn.over_sampling.SMOTENC.html

Jain, H., Khunteta, A., & Srivastava, S. (2021). Telecom churn prediction and used techniques, datasets and performance measures: a review. *Telecommunication Systems*, 613–630.

Kaggle. (2021, September 27). *telco customer churn code*. From Kaggle web site: https://www.kaggle.com/ylchang/telco-customer-churn-1113/code

Kaya, E., Dong, X., Suhara, Y., Balcisoy, S., Bozkaya, B., & Pentland, A. (2018). Behavioral attributes and financial churn prediction. *EPJ Data Science*, 7-41.

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

Lee, Y.-H., Wei, C.-P., Cheng, T.-H., & Yang, C.-T. (2012). Nearest-neighbor-based approach to time-series classification. *Decision Support Systems*, 207-217.

Nguyen, L. H., & Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *Computational Biology*.

Nie, G., Rowe, W., Zhang, L., Tian, Y., & Shi, Y. (2011). Credit card churn forecasting by logistic regression and decision tree. *Expert Systems with Applications*, 15273-15285.

Ozmen, E. P., & Ozcan, T. (2021). A novel deep learning model based on convolutional neural networks for employee churn prediction. *Wiley*.

Pages, J. (2004). Analyse factorielle de donnees mixtes. *Revue Statistique Appliquee. LII (4)*, 93-111. From R documentation web site: https://www.rdocumentation.org/packages/FactoMineR/versions/2.4/topics/FAMD

Pastor-López, I., Sanz, B., Tellaeche, A., Psaila, G., Gaviria de la Puerta, J., & Bringas, P. G. (2021). Quality assessment methodology based on machine learning with small datasets: Industrial castings defects. *Neurocomputing*, 622-628.

Rajamohamed, R., & Manokaran, J. (2018). Improved credit card churn prediction based on rough clustering and supervised learning techniques. *Cluster Computing*, 65-77.

Sivasankar, E., & Vijaya, J. (2019). Hybrid PPFCM-ANN model: an efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network. *Neural Computing and Applications*, 7181–7200.

Sklearn developers. (2021, January 14). *F1-score module*. From scikit-learn website: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html?highlight=macro%20average

Sklearn developers. (2021). *modules: sklearn*. From sklearn website: https://scikit-learn.org/stable/modules/grid_search.html#grid-search

Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences*, 429-441.

Vafeiadis, T., Diamantaras, K., Sarigiannidis, G., & Chatzisavvas, K. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 1-9.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* New York: Springer New York.

Vijaya, J., & Sivasankar, E. (2019). An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing. *Cluster Computing*, 10757-10768.

Wu, S., Yau, W.-C., Ong, T.-S., & Chong, S.-C. (2021). Integrated Churn Prediction and Customer Segmentation Framework for Telco Business. *IEEE Access*, 62118-62136.

Zikria, Y. B., Afzal, M. K., Kim, S. W., Marin, A., & Guizani, M. (2020). Deep learning for intelligent IoT: Opportunities, challenges and solutions. *Computer Communications*, 50-53.

Appendices

Appendix A

Number	Name	Description	Data type
0	Zip Code	The zip code of a client	Numerical
1	Latitude	The latitude of a client's residence	Numerical
2	Longitude	The longitude of a client's residence	Numerical
3	Gender	The client's gender (male or female)	Categorical
4	Senior Citizen	Whether the client is a senior citizen or not	Categorical
5	Partner	Whether the client has a partner or not	Categorical
6	Dependents	Whether the client lives with dependents or not	Categorical
7	Tenure Months	Total amount of months that the clients has been with the company	Numerical
8	Phone Service	Indicates subscription to phone service	Categorical
9	Multiple Lines	Indicates subscription to multiple lines	Categorical
10	Internet Service	Indicates subscription to internet service	Categorical
11	Online Security	Indicates subscription to online security	Categorical
12	Online Backup	Indicates subscription to online backup	Categorical
13	Device Protection	Indicates subscription to internet device protection	Categorical
14	Tech Support	Indicates subscription to tech support	Categorical
15	Streaming TV	Whether the client uses their internet for streaming TV	Categorical
16	Streaming Movies	Whether the client uses their internet for streaming movies	Categorical
17	Contract	The contract type	Categorical
18	Paperless Billing	Indicates if the client uses paperless billing	Categorical
19	Payment Method	Indicates the method of payment	Categorical
20	Monthly Charges	Total monthly charges for all services used	Numerical
21	Churn Score	A value from 0-100 which indicates how likely a customer is to churn	Numerical
22	CLTV	Customer LifeTime Value, the higher the value, the more valuable the customer	Numerical
23	Age	Age of the customer	Numerical
24	Married	Indicates if the client is married or not	Categorical
25	Number of Referrals	Indicates the number of referrals	Numerical
26	Avg Monthly GB Download	Average monthly GB used for downloading	Numerical
27	Streaming Music	Whether the client uses their internet for streaming music	Categorical
28	Unlimited Data	Indicates subscription to unlimited data	Categorical
29	Total Refunds	Indicates client's total refunds per quarter	Numerical
30	Total Extra Data Charges	Indicates client's total extra data charges	Numerical
31	Total Long Distance Charges	Indicates client's total long distance charges	Numerical
32	Total Charges	Indicates client's total charges combined	Numerical
33	Total Revenue	Indicates revenue made per client	Numerical
34	Churn Value	Indicates whether the client churned or not	Categorical

Table 17: Main dataset description