ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΔΙΟΙΚΗΣΗ, ΑΝΑΛΥΤΙΚΗ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΕΠΙΧΕΙΡΗΣΕΩΝ»

Master of Science in
Business Administration, Analytics and Information Systems

ECON409. Διπλωματική Εργασία (Master's Thesis)

«Data Analysis Techniques for Customer Churn Prediction in Banks»

Κοντογιαννίδης Ευθύμιος
ΑΜ: 7341122200013

Επιβλέπων διδάσκοντας
ΚΥΡΙΑΚΟΣ ΔΡΙΒΑΣ

ΑΘΗΝΑ
2024

Abstract

This thesis has accomplished complete research sets of machine learning approaches in predicting customer churn in the banking industry. Our aim is to obtain to obtain higher predictive evaluation rates and to attain interpretive results for the banking industry to maintain their consumers. The study on data preparation includes handling of categorical values, normalization, and handling of data imbalance. The thesis evaluates the performance of some supervised learning techniques including decision trees, random forest, k nearest neighbors, logistic regression, and support vector machines. There are some evaluation indicators that are utilized in the testing process such as k fold cross validation, confusion matrices, precision-based metrics, the f measure, and receiver-operating curves. In this, great emphasis is laid on the pre-processing of the data to the point where quality and relevance get brought out, with depth in the analysis of encoding and scaling, exploring the correlations among the variables. This section describes how this study employed the Stratified Cross-Validation, Pruning Techniques, and Synthetic Minority Over-sampling Technique to improve model training and evaluation. Model selection is carried out through cross-validation and pruning, showing the improvement iterations. An analysis of consistency and reliable findings across the runs. In conclusion, the thesis reviews the limitations faced, which mainly concern hyperparameter tuning and the underlying constraints of the available data. The thesis also offers some directions on future work, which generally involve log transformation, customer division, and additional feature engineering to improve the current models.

TABLE OF CONTENTS

Abstract ii

Figures v

Tables vi

List of Abbreviations vii

Chapter 1: Introduction 1

1.1 Purpose 1

1.2 Scope 2

1.3 Thesis Structure 2

Chapter 2: Theoretical Background 4

2.1 Data Preparation 4

2.1.1 Handling Categorical Values 4

2.1.2 Normalization 5

2.1.3 Imbalance 5

2.2 Supervised learning 6

2.2.1 Decision tree 6

2.2.2 Radom Forest 7

2.2.3 K-nearest neighbor 8

2.2.4 Logistic regression 9

2.2.5 Support vector machine 10

2.2.6 Bias-Variance Tradeoff 11

2.3 Model Evaluations 12

2.3.1 Cross-validation 12

2.3.2 Confusion matrix 12

2.3.3 Accuracy 13

2.3.4 F score 14

2.3.5 Receiver Operator Curve 14

Chapter 3: Research Methodology 16

3.1 Preliminary Data Analysis 16

3.2 Handling Categorical Values 17

3.3 Normalization....17
3.4 Model Selection....18
3.5 Stratified Cross-Validation....18
3.6 Pruning....18
3.7 SMOTE....19
3.8 Evaluation Metrics....19
Chapter 4: Data Preprocess....21
4.1 Preliminary Data Analysis....21
4.1.1 Discrete Variable....22
4.1.2 Continuous Variables....28
4.2 Data Preparation....31
4.3 Encoding....32
4.4 Scaling....32
4.5 Correlation....33
Chapter 5: Model Selection and Preparation....34
5.1 Model's Preparation....34
5.2 Stratified Cross-Validation....34
5.3 Enhanced Pruning....36
5.4 SMOTE....37
Chapter 6: Assessing Model Performance Across Classes-Evaluation metrics....41
6.1 Original Dataset....42
6.2 Minority Class Doubled....43
6.3 Minority Class Tripled....44
6.4 Minority Class Quadrupled....46
Chapter 7: Conclusions....48
6.1 Research Overview....48
6.2 Limitations....49
6.2.1 Hyperparameter Tuning....49
6.2.2 Data Limitations....50
6.3 Future Work....50
6.3.1 Log Transformation....50
6.3.2 Segmentation....50
6.3.3 Feature Engineering....51
References....52

Figures

Fig. 1. Example of Label Encoding [18] 4
Fig. 2. Example of One-Hot Encoding [18] 5
Fig. 3. Decision Tree Example [43] 7
Fig. 4. Ensemble Learning Architecture [74] 8
Fig. 5. Sigmoid Curve [77] 10
Fig. 6. Two Linear Decision Function for SVM [36] 11
Fig. 7. The Bias-Variance Tradeoff [64] 11
Fig. 8. Confusion Matrix Example [60] 13
Fig. 9. The ROC of Different Methods in Simulation Experiments [21] 15
Fig. 10. Churn Values 22
Fig. 11. Active Member Values 23
Fig. 12. Credit Card Values 24
Fig. 13. Gender Values 25
Fig. 14. Country Values 26
Fig. 15. Number of Products Values 27
Fig. 16. Tenure Values 28
Fig. 17. Age Values 29
Fig. 18. Credit Score Values 29
Fig. 19. Estimated Salary Values 30
Fig. 20. Balance Values 31
Fig. 21. Pearson Correlation 33
Fig. 22. Roc Curves for Every Run 47
Fig. 23. Log Transformation Example for Age 50
Fig. 24. Quantile-Quantile Example for Balance 51

Tables

Table 1. Variables Description 21
Table 2. First Model Evaluation 35
Table 3. Second Model Evaluation 36
Table 4. Third Model Evaluation 37
Table 5. Oversampling 41
Table 6. First Run Accuracies 42
Table 7. First Run Evaluations 43
Table 8. Second Run Accuracies 43
Table 9. Second Run Evaluations 44
Table 10. Third Run Accuracies 45
Table 11. Third Run Evaluations 45
Table 12. Fourth Run Accuracies 46
Table 13. Fourth Run Evaluation 46

List of Abbreviations

ML	Machine Learning
SMOTE	Synthetic Minority Over-sampling Technique
DT	Decision Tree
RF	Random Forest
KNN	K-nearest neighbor
LR	Logistic Regression
SVM	Support vector machine
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
ROC	Receiver Operator Curve
CRM	Customer Relationship Management

Chapter 1: Introduction

Throughout their history, banks have been focused on internal procedures and policies, treating clients as the commercial target and prioritizing their strategy on their goods and services. This appears to have changed quite dramatically within the industry. A source of such change is that the rapid entry of new, highly competitive financial products and services is being accelerated by improving capacity and declining costs of computers. The competition has now taken a soar with the drastic launch of new financial technology [30]. Now, customers are able to really spread their money across many banks in order to earn the maximum returns on it and safeguard their assets. Nowadays, maintaining customers and concentrating on the services opportunities for them are an important element of the survival of any bank. Many existing customers have migrated to non-traditional banking, due to the extensive dissemination of Internet finance, characterized by diverse, unique customer requirements. The term "customer churn" means a counter-move on the part of a client, which temporarily (for a definite time) discontinues its current banking interaction to another financial establishment, due to subjective or objective reasons [34]. Besides causing some loss in revenue, the other adverse effect in terms of operation to the business is the client's loss. Churn management refers to the concept of identifying those customers willing to move their assets to a competitor service provider [31]. Customer relationship management is starting to include churn management. It would be of interest for firms to take it into account, in the effort to build a lasting clientele relationship and optimize the value of the clientele [3].

1.1 Purpose

This thesis concentrates on primarily predicting the intent to quit using a company's services. This allows banks to determine which customers are about to quit, so before churn occurs, they can swoop in with targeted interventions to keep these customers' base in their portfolio. Thus, this is accomplished by learning customers' behaviors and detecting signs of dissatisfaction and disengagement at an early stage. [31]. The ability to predict when customers churn is a critical strategy that gives a bank a decisive competitive advantage. By identifying the most probable defections, institutions could prepare tailor-made or pinpointed retention strategies with respect to particular problems or issues that customers raise. This will thus help not in losing important customers alone but also with the enhancement of the overall customer satisfaction and loyalty [69]. This dissertation will focus on not only identifying who will churn but also on why the same is happening using a combination of statistical techniques and machine learning models.

In this way, an analytical approach is expected to produce actionable insights through the possibility of offering implications directly appropriate to real-world scenarios [9]. In summary, the purpose of this thesis is to improve the ability of banks to anticipate and reduce customer churn to ensure continuous business expansion and customer retention. This is done through an extensive examination of consumer information, incorporating advanced analytics to identify significant churn indicators and patterns. The concluding results of this study are expected to provide banks with a basis for forecasting, which is fully integrated into their consumer relationship management system to develop a proactive and client-oriented business culture [76].

1.2 Scope

In this thesis, machine learning methods are utilized to forecasting customer churn in banks. The main goal is to carry out predictive analytics to identify patterns and signs indicating the likelihood that customers will terminate their own subscriptions. It encompasses data mining, algorithms, and validation. The principal aim for this work is to develop ML models that can make effective predictions of customer churn. It means that it requires supervised learning, data pre-processing methods, and evaluation metrics during model development. machine learning literature for their effectiveness in classification tasks includes algorithms such as DT, RF, KNN, LR, and SVM [38]. While the customer churn predictive models can be applied to informing marketing strategies, but that is not an inquiry line the thesis is taking with respect to considering in the application of these insights within the marketing frameworks like Customer Relationship Management (CRM) or Direct Marketing interventions. The following analytical techniques are designed to improve the predictive performance of churn models and are not meant to be used directly as guides for the execution of marketing strategies. CRM systems may indirectly benefit from the findings because a more profound understanding of customer behavior can be derived [46]. This thesis is only limited to the technical scope of the ML methods for churn prediction in banking and it has an objective to add academic and practical input into the area of machine learning and its application to big datasets within the banking industry for predicting customer churn.

1.3 Thesis Structure

Chapter 2 is literature review of the methodological work in data preparation and KNN, LR, and SVM. Moreover, the chapter covers Bias-Variance Tradeoff and the different methods applied to test the models, including K-fold cross-validation, the confusion matrix, accuracy, the F score, and Receiver Operator Curve, all within the theoretical framework provided to the methodologies. Chapter 3 provides a presentation of the research design and methods of data analysis applied in this study, including preliminary data analysis, handling of categorical values, normalization techniques, and pruning with the application of "synthetic minority over-sampling technique" (SMOTE) for the purpose of better model performance. The purpose of the chapter furthers on how the evaluation metrics are used to determine the extent to which the developed model applies. Chapter 4 explains how the data should first be handled and prepared like dealing with discrete and continuous variables, handling categorical values, scaling, and correlation analysis. These preprocessing are the initial steps before using the data for modeling purposes. Chapter 5 explains the model selection and its preparation in the context of predicting customer churn. The first part explains the model preparation, cross-validation, and pruning, for making tree like models more bias, and SMOTE for balancing the data. The models for churn prediction tested in chapter 6 under a number of runs, through different means of evaluation, show the performance and robustness of the individual models. It aims at proving the efficacy of the developed models in realistic settings. The thesis concludes with a summary of the results, the limitations faced in this study, and an overview of directions for future research. This chapter attempts to summarize the contributions of the thesis to the literature and its implications for banking sector strategies in customer retention.

Chapter 2: Theoretical Background

2.1 Data Preparation

One important step in the data analysis process is preprocessing the data, given that the task success is directly impacted [38]. The evaluation methods of the raw data to produce high-quality data are referred to as data preparation. These techniques primarily include data collection, data combination, data normalization, data processing, data removal, and data transformation. [6565]. Furthermore, specific preprocessing methods, like data selection or elimination, may be applied repeatedly until the optimal outcomes for data analysis are achieved [19].

2.1.1 Handling Categorical Values

Because the majority of machine learning algorithms are made to operate with numerical inputs, categorical data requires specific handling [32]. One of the most common ways is Label encoding, which assigns each unique category in a feature to a single integer and one-hot encoding transforms every categorical values into a new binary column, designed to handle scenarios where no ordinal relationship exists among the categories.

Categorical Feature	Label Encoding
United States	1
United States	1
France	2
Germany	3
United Kingdom	4
France	2

Fig. 1. Example of Label Encoding [18]

United States	France	Germany	United Kingdom
1	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	1	0	0

Fig. 2. Example of One-Hot Encoding [18]

2.1.2 Normalization

Normalization in machine learning speeds up the learning algorithms and returns better outcome. This is because the process makes all the features contribute equally towards the result, and it avoids any model from skewing or being biased towards some features [28]. Two of the most common techniques are Min-Max Scaling and Robust Scaling. Min-Max Scaling is the simplest scaling method for making features rescaled into a fixed range by subtracting the minimum value of the feature and then dividing by the range of the feature. Below is the Min-Max scaling formula:

$$\frac{X - \min(X)}{\max(X) - \min(X)}$$

Robust scaling uses the median and interquartile range in scaling features to reduce the effect of outliers. The formula for Robust scaling is:

$$\frac{X - Q1(X)}{Q3(X) - Q1(X)}$$

2.1.3 Imbalance

When the classes of a dataset are not equally represented, for churn prediction in particular, this usually means that the number of churning customers is much less than those who do not churn.

This implies that, most often, the class will be predicted more than the minority class, thus affecting the model sensitivity to pick up the minority class effectively. The resampling methods modify the dataset to show a more balanced distribution. These could be either oversampling the minority class or undersampling the majority class. Among the most successful methods of oversampling is "synthetic minority over-sampling technique" (SMOTE), which simulates between existing instances to create new ones of the minority class [45]. Using metrics that give a better sense of model performance when the dataset is imbalanced, such as the F-score, ROC-AUC, which tend to focus more on the performance regarding the minority class [71].

2.2 Supervised learning

When an algorithm is trained on a labeled dataset, one in which every input data point has a corresponding output label, it is referred to as supervised learning. Using the patterns found in the training data, supervised learning aims to learn a function from inputs to outputs, based on the patterns found in this training data, and by that the algorithm can forecast or decide on previously unknown data [4343].

2.2.1 Decision tree

The decision tree is a tool that helps in analytical decisions by displaying potential outcomes in a structure like a tree. This tree-like graph shows relationships between different events. Decision-tree analysis is used to build a predictive model based on input variables. These variables are represented by each leaf node on the decision tree. Every leaf node contains inputs from the dependent variables [56], while at various levels, those nodes symbolizing different criteria [12]. Decision trees are systematic models that incorporate a series of simple evaluations, where each evaluation compares a numerical attribute to a predetermined threshold or a categorical characteristic to a predetermined range of values. [5858]. In modeling's tasks like customer churn analysis, where customers are classified as either churn or no-churn. Decision-tree models are typically built from the top down, following a divide-and-conquer strategy, where the construction process begins with the selection of the root node and then proceeds to divide the data into subsets based on specific attributes or conditions, forming nodes at each stage of the tree. The choice of the root node involves comparing the information gain of possible root nodes and selecting the one with the most information. Afterward, the same information-based approach is applied when selecting branches emanating from the root node. This process continues until all instances belong to the same class, resulting in 100% accuracy [4].

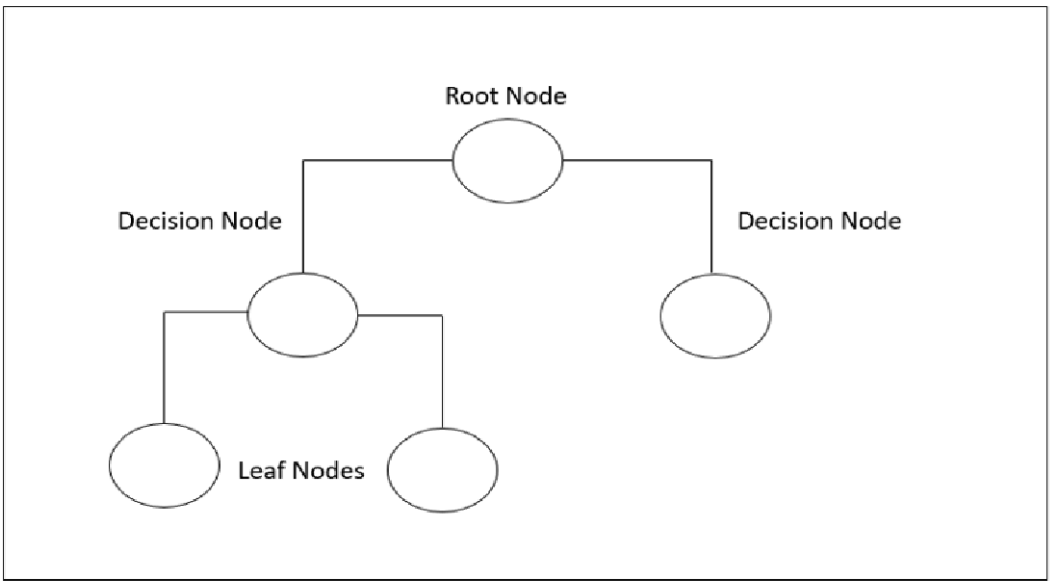

Fig. 3. Decision Tree Example [43]

Tree pruning, if deemed necessary, is the next stage and involves removing branches that may introduce errors. Pruning a tree can enhance the performance of a classifier and simplify the model for easier understanding and knowledge extraction. It's crucial to ensure that the pruning process doesn't remove essential predictive elements of the classifier [73].

2.2.2 Radom Forest

An approach to machine learning known as ensemble learning involves training a number of models, also called "weak learners," to answer a single problem. On the wide scale of principles, ensemble learning is based on the assumption that multiple weak learners combine into a strong one, thus from there, achieving an overall better model with respect to its accuracy [4949]. Every single tree (weak learners) is created from a sample derive from replacement of the training set. In addition, when constructing the tree, during the node splitting, rather than selecting the optimal split from all the features, the best split is determined by randomly selecting a subset of the features. This introduces randomness that makes the model more robust and avoids it being prone to overfitting [8].

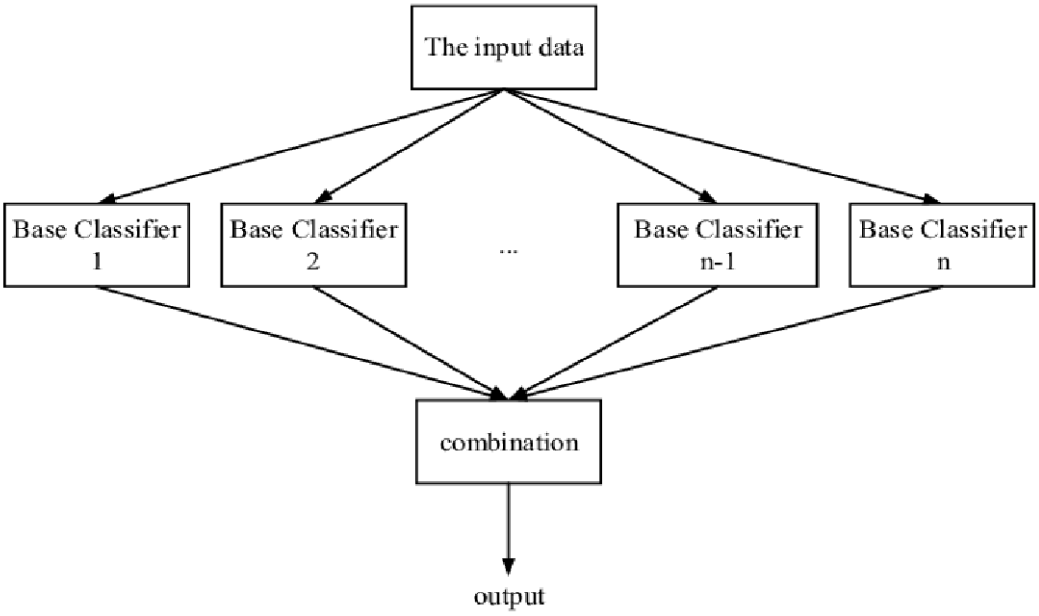

Fig. 4. Ensemble Learning Architecture [74]

Finally, pruning is a method of minimizing the final size of model by eliminating sub-sections that have minimal power in calculating the target variables. This is another way to mitigate overfitting and promote the simplicity of the model. The stability of random forests is very high when it comes to the number of the tree estimators. In particular, this indicates that random forests asymptotically approach the true mean of the distribution. In conclusion, this implies that the actual accuracy of random forests is not too vulnerable to the number of trees within the model. Therefore, restricting to low maximal depth may reduce overfitting as it will retain high or increase predictive [35].

2.2.3 K-nearest neighbor

The K-nearest neighbor algorithm, also known as KNN, is a well-known non-parametric method in ML, primarily used for solving classification problems. The algorithm processes a dataset containing objects with known labels and also handles unknown objects, to determine their classification—much like processing training and test samples. KNN calculates the distances between an object and a new query object. Furthermore, the algorithm evaluates at least one neighbor of the query, previously labeled, to determine the most likely class based on existing data. KNN is characterized as lazy because the function that specifies the class is delayed until the computational interface is needed for evaluation. [5]. The distance metric refers to the level at which two data points in the feature space are alike or different. In the K-Nearest Neighbors algorithm, the distance metric helps in understanding how two instances can

be considered ‘close” or “far” from each other. Hence, the distance measure selection has an impact on the KNN algorithm's outcomes and usually depends on the type of data and problem being solved. Meanwhile, the most frequently used types of distance in K-nearest neighbor are as follows: Euclidean Distance, Manhattan Distance, Minkowski Distance, Cosine Similarity and Chi-square [55]. The value of k is a hyperparameter that needs to be specified. The selection of k can significantly impact the performance of the algorithm. A smaller value of k makes the algorithm more sensitive to noise, while a larger value of k may lead to the smoothing of decision boundaries [37]. Weighting the neighbors based on their distance. Closer neighbors may have a higher influence on the decision than those that are farther away.

2.2.4 Logistic regression

In nearly every field, logistic regression has emerged as the conventional approach for modeling a binary outcome. Logistic regression can handle nearly all the tasks achievable with linear regression, but it is tailored for situations involving binary outcomes, and it offers a high degree of flexibility for various extensions and adaptations [50]. In binary classification establishing a connection between one or more independent variables—which could be qualitative or quantitative—and a categorical dependent variable. This problem also has multivariate counterparts. When the relationship between the dependent and independent variables follows the functional form of a logistic distribution, it is commonly known as logistic regression. The mathematical representation of the logistic regression model typically involves connecting the probability of an event, denoted as E, occurring given a set of explanatory variables represented by vector x. This relationship is established using the logistic cumulative distribution function (logistic Cumulative Distribution Function) as follows:

$$p(x) = \Pr(E|x) = \frac{1}{[1 + \exp\{-(\alpha + \beta' x)\}]}$$

In this equation, the probability p(x) is determined by the values of the parameters α and β, along with the explanatory variables vector x [59]. The most common method for estimating these coefficients in logistic regression is the maximum likelihood estimator. The goal of the maximum likelihood estimator is to identify the values of the parameters that maximize the likelihood function, which calculates the likelihood that the supplied data will be observed under the logistic regression model. These parameter estimates are often found using

optimization algorithms, such as gradient descent, to iteratively refine the estimates until they converge to the maximum likelihood values [1]. Using the logistic function, sometimes referred to as the sigmoid function, one can convert any real number into a value between 0 and 1, which indicates the likelihood that the dependent variable would fall into a specific class [20].

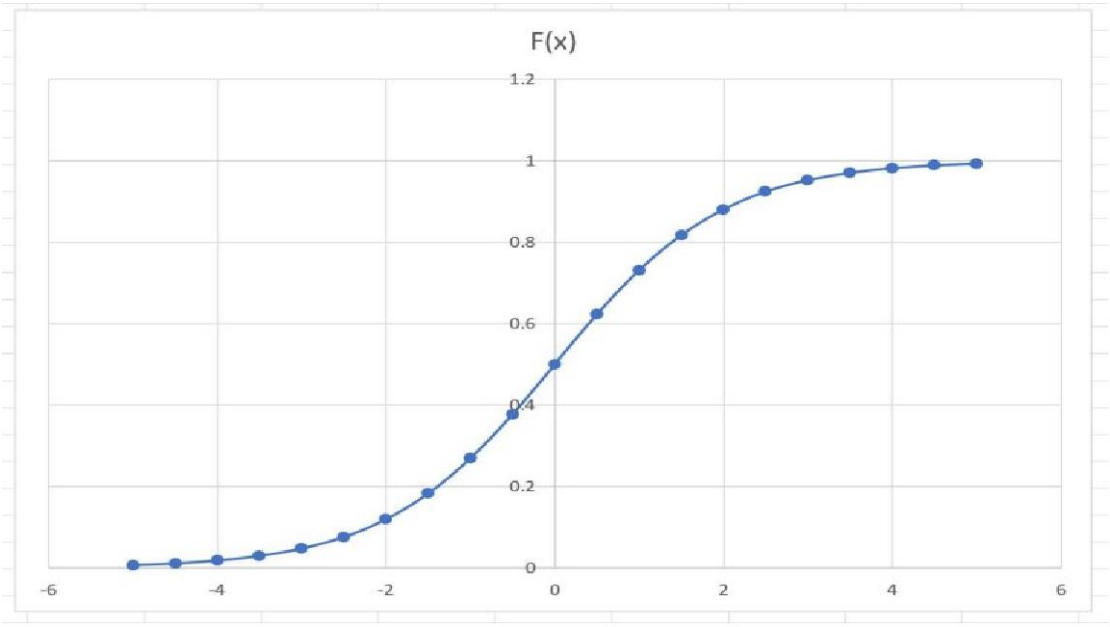

Fig. 5. Sigmoid Curve [77]

2.2.5 Support vector machine

Similar to linear classifiers, support vector machines estimate a linear decision function, but they have the unique feature of potentially requiring a prior transformation of the data into a higher-dimensional feature space. This transformation is defined by selecting a set of functions referred to as kernels [25]. Using labeled training data, the SVM aims to find a hyperplane that optimizes the margin between two classes. The selected hyperplane is utilized as the decision border, and it is designed to reduce future misclassification when predicting the category of fresh instances. The working principle is maximizing the margin that is finding the hyperplane that maximizes the distance to the closest data point of any class. This principle is to enable the SVM to separate the classes using a hyperplane such that future classification will be optimal on unseen data [72]. Kernels project input data into high-dimensional feature spaces where linear separation is feasible, allowing SVM to handle nonlinear classification problems. Also, the transformation is never explicitly computed, as it utilizes the kernel function to determine inner products of unique in the feature space [61].

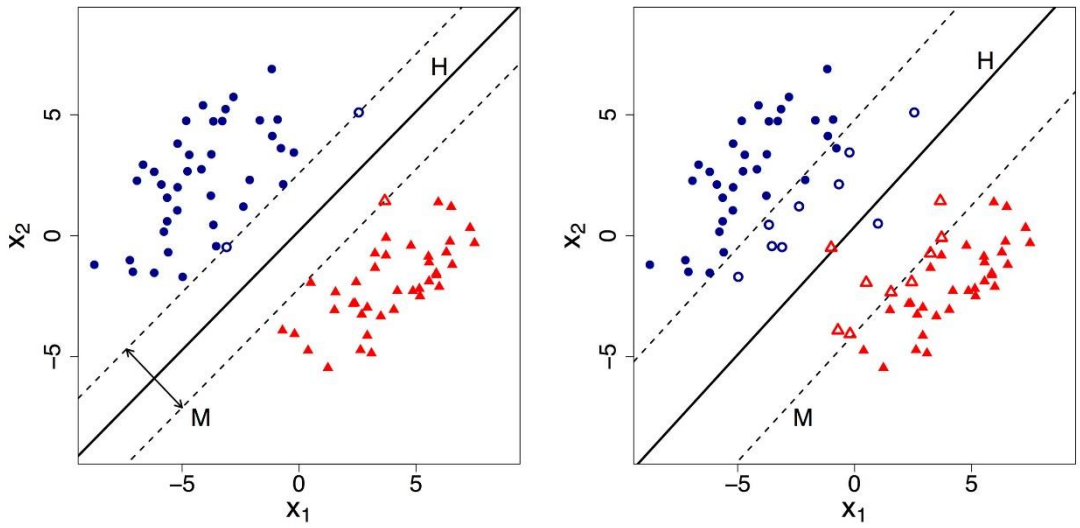

Fig. 6. Two Linear Decision Function for SVM [36]

2.2.6 Bias-Variance Tradeoff

Bias is the inaccurate result of approximating a real-world problem with a simplified model. When a model has a large bias, it is making strong assumptions about the shape of the underlying data distribution, which can cause underfitting. To put it another way, the model is too basic to adequately represent the complexity of the data. Conversely, variance quantifies how inconsistently the model predicts an input. A high variance suggests that the training data may cause overfitting because the model is sensitive to even minute variations. An overfitted model begins to detect noise or random fluctuations in the training set rather than the underlying patterns [10].

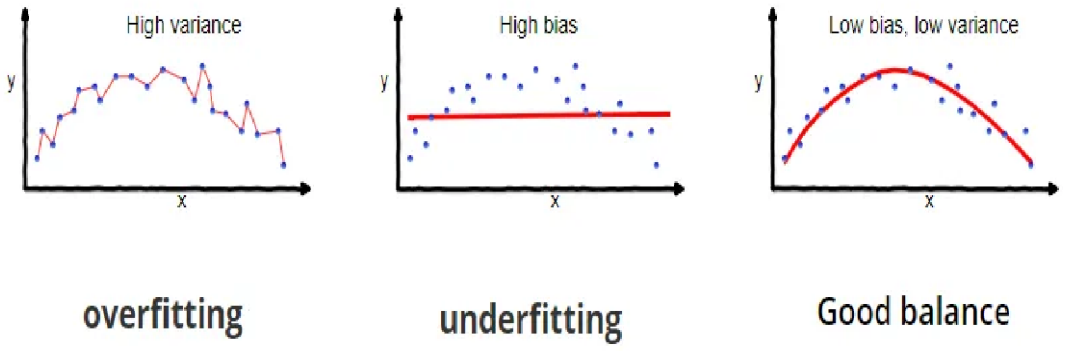

Fig. 7. The Bias-Variance Tradeoff [64]

2.3 Model Evaluations

2.3.1 Cross-validation

The initial sample is split into k equal-sized sections for K-fold cross-validation. The remaining $k-1$ subsets are utilized as training data, while one subset is kept as validation data to test the model. Then, the process is repeated k times, with each of the k subsets used exactly once as the validation data [20]. Generalization is the capacity of a model to predict well on new data that was not used during model training. It is, therefore, a measure of how much the model learns concepts of feature-response relationship inherent in the training data expresses in the external data. In other words, a well-generalized model is one that is able to make accurate predictions across all cases and not just a small similar example given during input training [51]. K-fold cross-validation is mostly used to determine the degree to which the findings of a statistical analysis generalize to a separate set of data. The technique is particularly pertinent in cases where the objective is to forecast a model's performance on a new, unseen data set [57]. One of the benefits of K-fold cross-validation is that it helps reduce generalization errors or issues of underfitting and overfitting. Overfitting occurs when the model is too complex and tries to fit not only the trend in data but also the noise. Conversely, underfitting happens when the model is too basic to identify the underlying pattern in the data. K-fold cross-validation enables one to strike a balance between bias and variance [68]. Being useful in model selection, K-fold cross-validation is majorly used to compare diverse models with each other when wanting to generalize responsive selections or in selecting hyperparameters within the same algorithm. The models' performance averages over the trials to give an overall result.

2.3.2 Confusion matrix

In machine learning, a table called the confusion matrix is used for evaluating and visualizing on how well models perform in supervised classification scenariosIt is a square matrix where the columns represent the cases' expected categories and the rows represent the instances' actual categories. The table that is 2 × 2 consist records true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in the context of binary classification [47]. The outcome of this table contains four components [14]:

- True Positives are the instances in which the model predicts the positive class accurately. Stated differently, these represent the occurrences that were both positively impacted and predicted by the model.

- True Negatives are instances in which the negative class is accurately predicted by the model. These are the cases where the model anticipated a negative outcome even if the actual outcome was negative.
- False Positives occur when the model forecasts the positive class incorrectly. Although the model expected these cases to be positive, they were in fact negative.
- False Negatives occur where the negative class is predicted by the model inaccurately. These are examples where the model projected bad outcomes even though the events were positive.

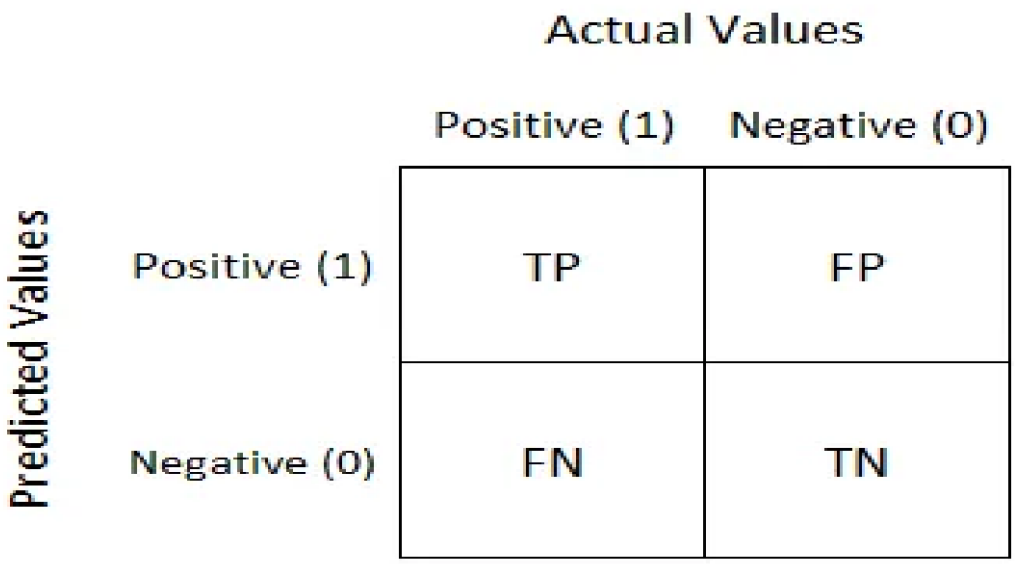

Fig. 8. Confusion Matrix Example [60]

2.3.3 Accuracy

One of the most important performance metrics for models connected to categorization in the context of machine learning and statistics is accuracy. It represents the proportion of true results, including both true positives and true negatives, in the total number of cases examined. Accuracy is defined as the ratio of correctly predicted observations to the total observations [40]. It is expressed as follows:

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

2.3.4 F score

The ratio of real positive forecasts to all positive predictions made is known as precision and it is described by the following equation [26]:

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

Recall is a metric that indicates the percentage of real positives that the model properly identifies; it is sometimes referred to as sensitivity or the true positive rate and it is shown by the following equation:

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

Ideally, the outcome from both Precision and Recall would be close to 1. A low precision implies a relatively high proportion of false positives, which shows that the predictions are strongly conservative. When the recall values are low, it states that a sizable proportion of false negatives are not being properly classified as faulty [7070]. Precision is commonly used in combination with recall, which assesses a model's ability to identify all pertinent examples within a dataset. In many domains, there is a fundamental tension between precision and recall, as boosting precision usually lowers recall and vice versa [15]. This trade-off is captured in performance measures such as the F1 score, which is the balance between these two metrics shown by the following equation:

$$F - score = \frac{(\beta^2 + 1) * precision * recall}{\beta^2 * precision + recall}$$

The F-score is evenly balanced when β = 1 and it known as F1-score. When β > 1 the measure emphasizes on precision, and when β < 1 it emphasizes on recall [41].

2.3.5 Receiver Operator Curve

A visual representation of a binary classifier's diagnostic capability when its discrimination threshold is changed is called the Receiver Operator Curve. It is created when the combinations of recall, also referred to as true positive rate or sensitivity, and false positive rate, sometimes referred to as 1-specificity, for a set of experiments are plotted. The area under the curve (AUC) is typically reported as a number between 0 and 1, where 1 is the optimum value. It is not possible to calculate the confusion matrix from the AUC or vice versa because the AUC

represents the outcome of several tests in which the meta-parameters are changed [14]. The scalar value, that AUC provides, evaluates the model's overall capacity to distinguish between the positive and negative classes across all possible threshold values [52].

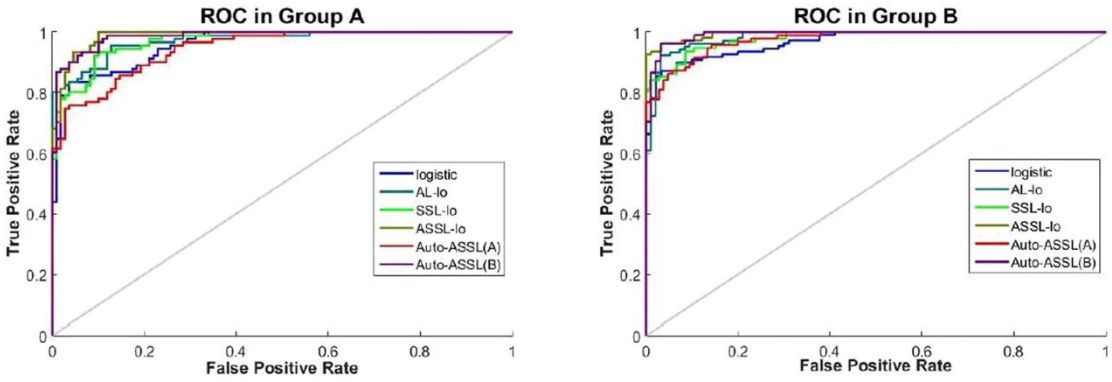

Fig. 9. The ROC of Different Methods in Simulation Experiments [21]

Chapter 3: Research Methodology

This chapter outlines the techniques used to examine the phenomena of bank customer churn. The different sections of the chapter are devoted to the measures associated with each step of the research. The first such step is the preliminary data analysis, which is the measure for data evaluation and readies it for further processing. Different methods from pruning to handling categorical variables, normalizing data distribution to stratified cross-validation are used. Moreover, this chapter focuses on the methods utilized for refining the machine learning models, using the subset of the dataset and SMOTE. The effectiveness of these measures would be measured using a specific set of metrics which are used to analyze this research's predictive models' performance. This chapter would also highlight the Python libraries and tools that already existed to facilitate the use of these measures. Therefore, this methodical plan would ensure that the research enabling is deferred from the data science measures and one which would provide meaningful and useful data to appreciate the banking-industry-mediated phenomenon of churn.

3.1 Preliminary Data Analysis

Exploratory data analysis is an examination of data sets summarizing characteristics of interest. This stage of analysis is very vital since it brings forth an overview of the structures and essentially summarizes insights residing in the data before any activity of modeling. It prepares data for further predictive analysis and informs the strategy that should be applied in modeling [6]. Descriptive analysis is a comprehensive approach to data science, which enables us to grasp the distribution, behavior, and importance of every variable on the dataset level. An analysis of the distribution allows choosing the best ways to record the range, central tendency, and spread of data points based on each attribute. Outliers have a high impact on the output of a predictive model. As emerging on every feature individually, it is possible to detect outliers, which denote either data entry errors or genuine anomalies. An observation is considered an outlier if it drastically deviates from other observations and raises suspicion to be generated by another mechanism. Relationships between the features show potential redundancy in the dataset [33]. Both statistical and visual approaches are covered in the work in order to investigate correlation and select the features to input into a model in avoiding multicollinearity [11].

3.2 Handling Categorical Values

Among different techniques that could handle categorical variables, there are two simple techniques Label Encoding and One-Hot Encoding. Label encoding is converting each value in a categorical column into a unique integer. It is straightforward technique but may introduce a notion of ordinality where none exists. Conversely, One-hot encoding creates a new categorical column for each category value and gives those columns a binary value of either 1 or 0. Each integer represents the presence (1) or absence (0) of the attribute. This method eliminates the issue of ordinality and is particularly useful for non-ordinal categories where no relationship exists between categories [32]. The issues arise when perfuming one hot encoding on a binary variable that creates two independent variables in a dataset are highly correlated, it often indicates that they convey similar information. This scenario is known as multicollinearity. If one variable can be almost accurately predicted from the other, it means they are covering similar aspects of the information, leading to redundancy. Multicollinearity makes interpretation more challenging. It's hard to tell how each variable impacts the dependent variable (like churn likelihood). With multicollinearity, small changes can hugely shift coefficients. This means coefficients become unreliable for explaining variable relationships. Multicollinearity sometimes causes overfitting good training data performance, but poor unseen data results [29]. F. Dormann et al. (2013) in their work the present a threshold of highly correlated variables to ($|r| >= 0.7$). The threshold is commonly used to identify highly correlated variables and will be used for this analysis. However, the specific threshold can indeed vary depending on the context of the study and the nature of the data being analyzed.

3.3 Normalization

The decision of using Min-Max scaling and other scaling techniques such as Standardization or Robust Scaling should also be depends on the distribution of data and if the dataset contains outliers, and all scaling techniques have their advantages and disadvantages for different types of variables and machine learning models. While Min-Max scaling is perfect to keep the original distribution of data, Standardization fit the data to the Gaussian Distribution and Robust Scaling is exceptional to reduce the effect of outliers [42]. In “A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain” normalization techniques designed to take into account both the relationships among features within a dataset and considering the magnitude of the features without regard to their sign. The authors utilized two different scaling techniques, the Min-Max Scaler and the Standard Scaler, to preprocess the data before applying the classifier [22].

3.4 Model Selection

Some of the most common ML models to use in classification problems are KNN, DT, RF, SVM [38] and LR which it is widely used as binary classifier [20].

3.5 Stratified Cross-Validation

As Author stated "when we randomly divide a labeled dataset into training and test sets, we violate the assumption of statistical independence". In the worst-case situation, the test set might contain no instances of a minority class at all after the train - test split in cross validation. As a result, it is advised to split the dataset into several categories. To put it another way, stratification is a method to preserve the original class proportion in the resultant subsets. In this case, stratification simply means that a dataset is split at random the training and the test set so that each class is accurately represented in the ensuing subsets [63]. When there is a significant overlap between the predicted values and the true observed values in the training data, overfitting occurs. When a ML model learns both the structured and noisy components of the training data to a disturbing effect on the model's capability to predict new data samples. Overfitting means that a particular ML model has overly high adaptability regarding the noise density. Conversely, when a model has few predictors, an underfitted event happens. This problem also arises when the training data collection is too small or not representative of the population data. An underfitted model is one that fits the training data poorly, which makes it less likely to predict new data points making it week to unseen data points [48].

3.6 Pruning

Pruning is technique that reduces the complexity of a model, prevents overfitting and enhancing the model's capacity for generalization by removing redundant branches or subtrees in decision tree-based models. Pruning on the DT model can lead to faster inference times and better scalability. Pruning can improve its ability to generalize making less likely to memorize noise or irrelevant details from the training data and is more likely to capture the underlying patterns that are applicable to unseen instances [24].

3.7 SMOTE

In his article Chawla et al. stated “We propose an over-sampling approach in which the minority class is over-sampled by creating “synthetic” examples rather than by over-sampling with replacement”. An approach to provide extra training-data is also used for increasing minority class representation but not to directly manipulate samples in the data set. It is an operation in “feature space” that changes raw data with visual transformations in feature space, such as rotation and skew. More precisely, by creating artificial examples along the line segments connecting members of the minority class with their closest neighbors, it oversamples the minority class..

This involves the following steps:

- In the feature space, look for each sample from the minority class's k-nearest neighbors.
- Randomly pick some, in regard to the level of oversampling, of those neighbors.
- For each such neighbor, develop the vector difference between the neighbor’s feature vector and the sample from the minority class.
- Select a scalar, chosen at random between 0 and 1 and multiply the difference with this scalar and then; from which the point is such that it connects the chosen neighbor.
- Include that point to the original feature vector, and that produces in essence, a new synthetic sample.

The new synthetic instances are close to the original samples, resulting in a more robust and generalized decision region for the minority class. In order to assure variation in the synthetic samples and to imitate a natural diversity within the minority class, random picking of points along the paths connecting two points in a multidimensional space is employed [45].

3.8 Evaluation Metrics

The model's overall correctness is measured by its accuracy. However, it can be misleading due to the crucial issue related to the imbalanced datasets. Firstly, such an issue arises where the accuracy can show the values close to the large class. In this regard, as Davide Chicco and Giuseppe Jurman claim in their article, accuracy is rarely the best choice for this setting and should not be the primary metric, and if it is used, it should be complemented with other measures which will attach more importance to the small class. A well-fitted model has almost the same level of accuracy on the training and testing sets, meaning it can generalize new data relatively well. On the other hand, an overfitted model has a high level of accuracy on the training set but a significantly lower level of accuracy on the testing set. The mentioned

difference suggests the overfitted model used the training data too close, following and extracting diverse noise and anomalies, which can perform ineffectively when applying to the unseen one [48]. In the study by Amgad Muneer et al. (2022), the use of SMOTE and its variants was designed to work on increase the recall of the minority class. By balancing the class distribution, the models were able to detect a higher proportion of actual churners, hence resulting in increased recall and increased F1 score. Although recall improved, this approach can also increase the number of majority class instances misclassified as the minority class, leading to more false positives and consequently a decrease in precision [27]. Such a trade-off is typical for cases in which the heightened detection for more cases of an event leads to more false positives. However, the presented F1-score improvement means that the trade-off was overall appropriate, meaning that the ratio between identifying churners and not sacrificing too much accuracy became better. The use of balanced techniques, in most cases, allowed for proper increase of F1-scores, meaning that the presented techniques stood the challenges of the true nature of churn datasets [74]. Another study which examined the performance of a classifier over a number of imbalanced datasets reported that AUC remains relatively stable, while other metrics such as accuracy and precision can change significantly with variations in class distribution. This stability was attributed to AUC focusing on the ranking of the prediction probabilities, rather than their absolute classification. The present study demonstrated that AUC had good utility for the class distributions and is not overly sensitive to changes in class proportion [7]. In exploring different sampling techniques in tackling class imbalance in datasets using SMOTE found that even after considerable changes in the class ratios, the differences in AUC changes were marginal for different resampling strategies. It was observed through the research that AUC, via such a summary of the model's ability to discriminate between classes at various threshold levels, gives a consistent measure of model efficacy regardless of how the class balance is manipulated [45].

Chapter 4: Data Preprocess

4.1 Preliminary Data Analysis

The dataset refers to the ABC Multinational Bank as an open source from Kaggle repository. The dataset contains 10000 rows and 12 columns, that include account and general information (such as geographic location, gender, etc.) about its clients. The main objective is to use this data to predict client attrition, which is essential to every bank's sustainable future. In the Table 1 will be given a data description as well as the data types that those features hold.

Table 1. Variables Description

Variable	Description	Data types
customer_id	A unique identifier for each customer	Unique
credit_score	The credit score of the customer	Continuous - Integer
country	The country of residence of the customer	Categorical - Text
gender	The gender of the customer	Categorical - Text
age	The age of the customer	Continuous - Integer
tenure	The number of years the customer has been with the bank	Continuous – Integer (from 0 to 10)
balance	The bank account balance of the customer	Continuous - Float
products_number	The number of products the customer has with the bank	Categorical - Integer (from 1 to 4)
credit_card	Indicates whether the customer has a credit card (1) or not (0)	Binary - Integer (0, 1)
active_member	Indicates whether the customer is an active member (1) or not (0)	Binary - Integer (0, 1)
estimated_salary	The estimated salary of the customer	Continuous - Float
churn	Indicates whether the customer has exited (1) or stayed (0) with the bank	Binary - Integer (0, 1)

Since the data set is relatively small and contains only ten inputs, techniques that include dimensionality reduction would not be used. It is worth to mention that customer_id is simply a label assigned to distinguish one customer from another. Therefore, it will not be included for this analysis.

4.1.1 Discrete Variable

a) Churn

The target variable in the prediction models for this thesis will be churn. The main objective is to forecast a client's likelihood of leaving the bank based on a variety of factors, including product consumption trends, customer behavior, and demographics.

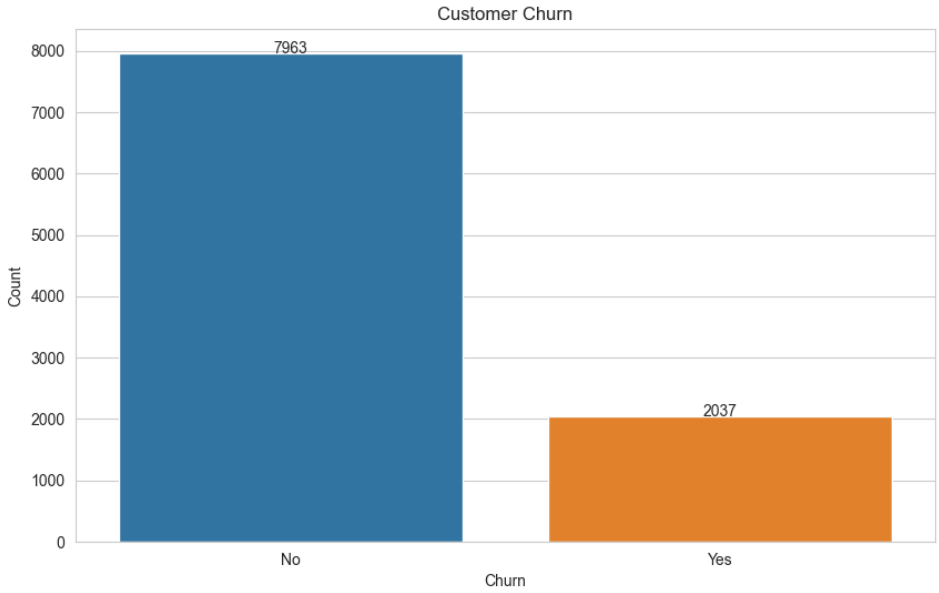

Fig. 10. Churn Values

In Fig. 10, "No" values show a substantial majority of customers, nearly four out of every five, that have remained with the bank. This suggests that the bank has a high rate of customer retention. High retention rates are often an indication of competitive offerings, satisfied consumers, and effective customer service protocols. In general, the bank has been successful in maintaining a steady clientele base. However, in the context of data mining, class imbalance has become common issue in real world scenarios [23]. Because there is more data for the machine learning model to learn from the majority class (in this case, the consumers who stayed), the models may become biased in favor of this group. This can result in poor performance when it comes to effectively anticipating the minority class (the customers who churned), which normally is more important to predict in churn analysis. Performance metrics, such as accuracy, can be misleading in imbalanced datasets. For example, take a model predicting each customer to stay (majority class). Though it would yield a great accuracy, it would justly fail in the prediction of churned customers, which forms an important class to predict in churn analysis [16].

b) Active Member

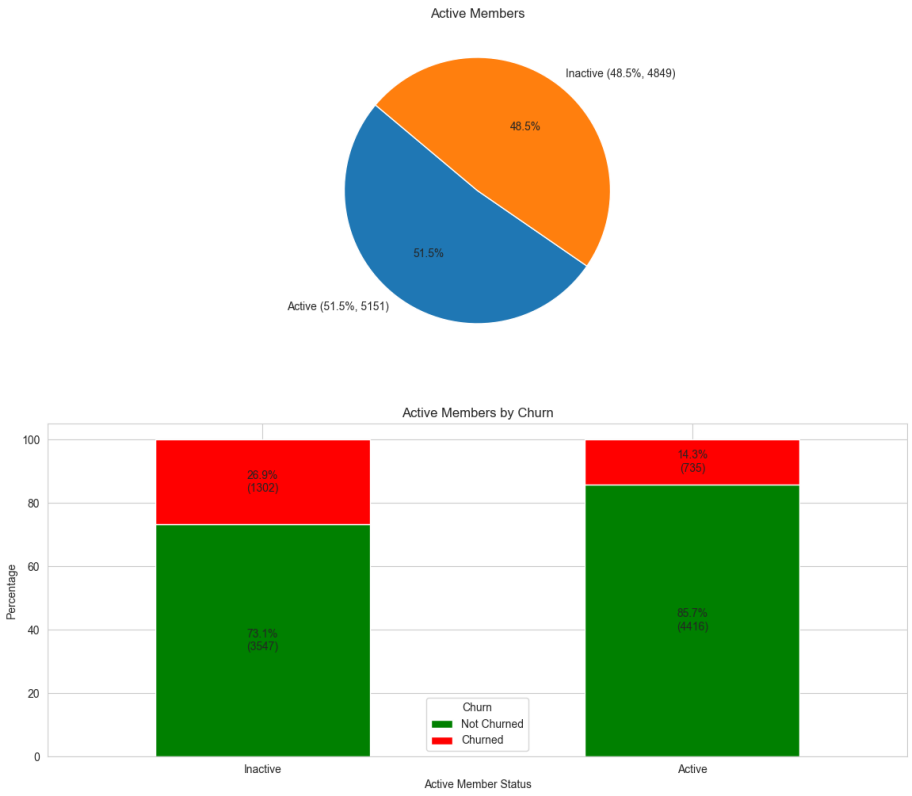

Fig. 11. Active Member Values

Active Member indicates whether the customer is currently considered active by the bank's standards. If the bank can distinguish between simply "active" and "satisfied" customers, they might address the reasons that active customers still choose to churn. According to the results, 44.6% of the members are inactive and 55.4% of them are active. Almost the half of the clients actively participate in or use the bank's services, indicating a moderately engaged customer base, according to this balance. The second graph shows that the churn rates for active and inactive members differ. First, one may see that 1302 members out of 3547 inactive members had churned, which is quite a large share of the inactive group. This implies that a higher chance of leaving the bank is closely correlated with inactivity. However, just 735 of the 4,416 active members have churned (Fig. 11). The notably reduced attrition rate amongst active members emphasizes how crucial it is to keep up consumer involvement as a retention tactic. Active membership by churn gives insight about nodes in tree-based models when looking to evaluate the impurity reduction brought by a feature. Although it does not compute impurity or reduce it

like a DT model does, it may assist in comprehending how well a feature may be able to classify data, which is a fundamental concept behind the impurity measures [39].

c) Credit Card

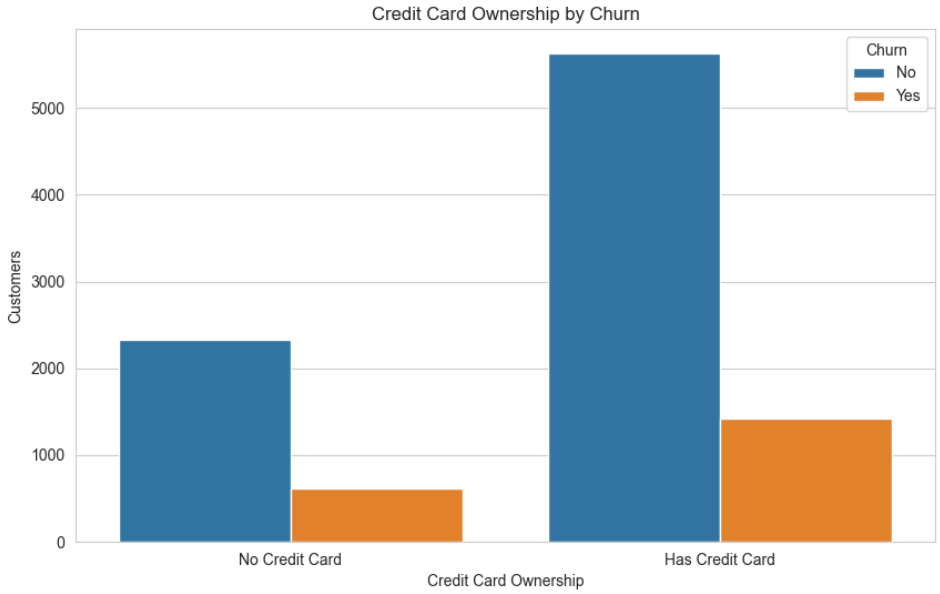

Fig. 12. Credit Card Values

Having a credit card might increase engagement with the bank's services but also introduces factors like credit card fees or rewards that could influence satisfaction. The total values for customers who has credit card are 7055 and for those who do not are 2945. For credit card owners 1,424 of these customers have churned, meaning they've closed or transferred their accounts away from the bank, while the remaining 5,631 customers have chosen to stay with the bank, continuing their banking relationship. The group of customers who do not own credit card 613 have churned, while 2,332 stay with the bank. The churn rates for these groups are 20.18% and 20.81% correspondingly, meaning that the likelihood of a customer churning is not significantly affected by the presence of a credit card. The conclusion that other factors can have a greater impact on customer attrition is supported by the close churn rates between credit card-holding and non-card-holding consumers (Fig. 12).

d) Gender

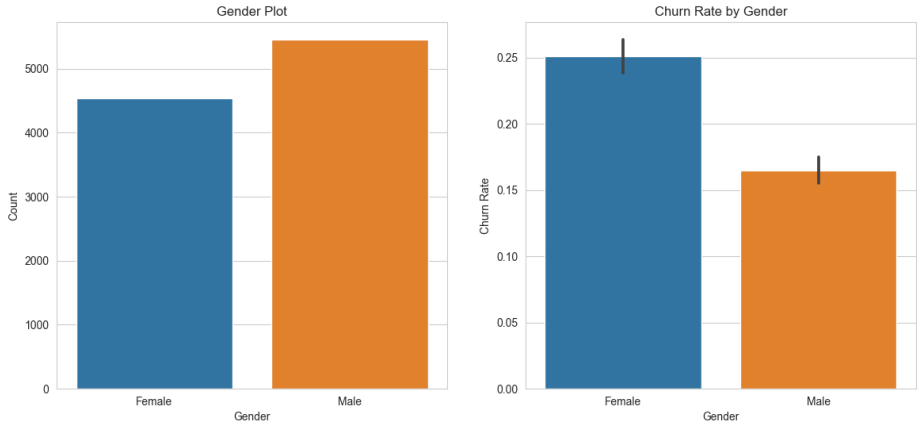

Fig. 13. Gender Values

Understanding how products and services align with gender-specific preferences or needs might help in designing more effective customer retention strategies. The results given from Fig.13 showing that males customers represent 54.57% of the bank accounts, while female accounts represent the 45.43%. The male accounts are slightly prevalent than females accounts. Considering the churn rates, for females, the churn rate is roughly 25.07%, whereas for males, it's roughly 16.46%. Given that females are more likely than males to churn, this difference may increase the possibility that gender may be a factor in churn prediction.

e) Country

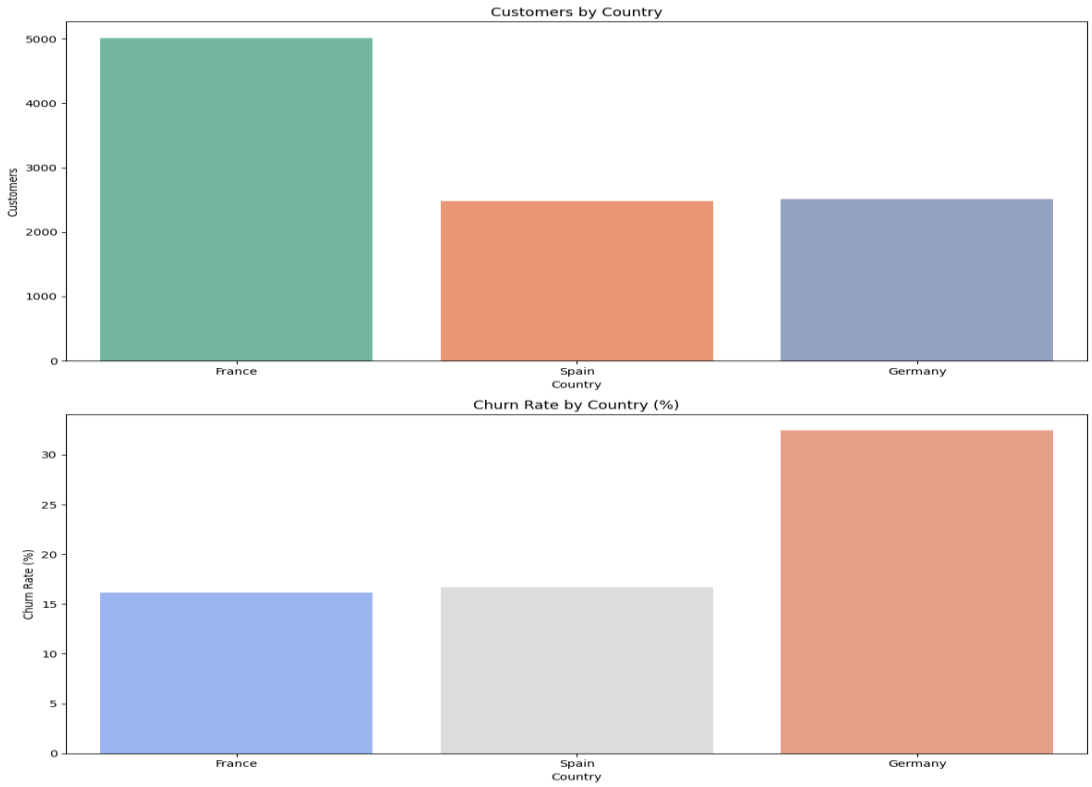

Fig. 14. Country Values

The bank's customers include three countries France, Spain, and Germany. From that order of number, evidently, France takes the lead by the highest total of customers, which is 5014, then Spain follows with 2477, while Germany has a total of 2509. The market penetration in France is quite significant, which means that the banks services are more aligned with the preferences or financial needs of the French population. The second plot highlights the percentage churn rate of those countries. Germany ranks at the top, at the highest of about 32.44%, then an average for Spain at 16.67%, and finally France at the least with a percentage of 16.15% (Fig. 14).

f) Number of Products

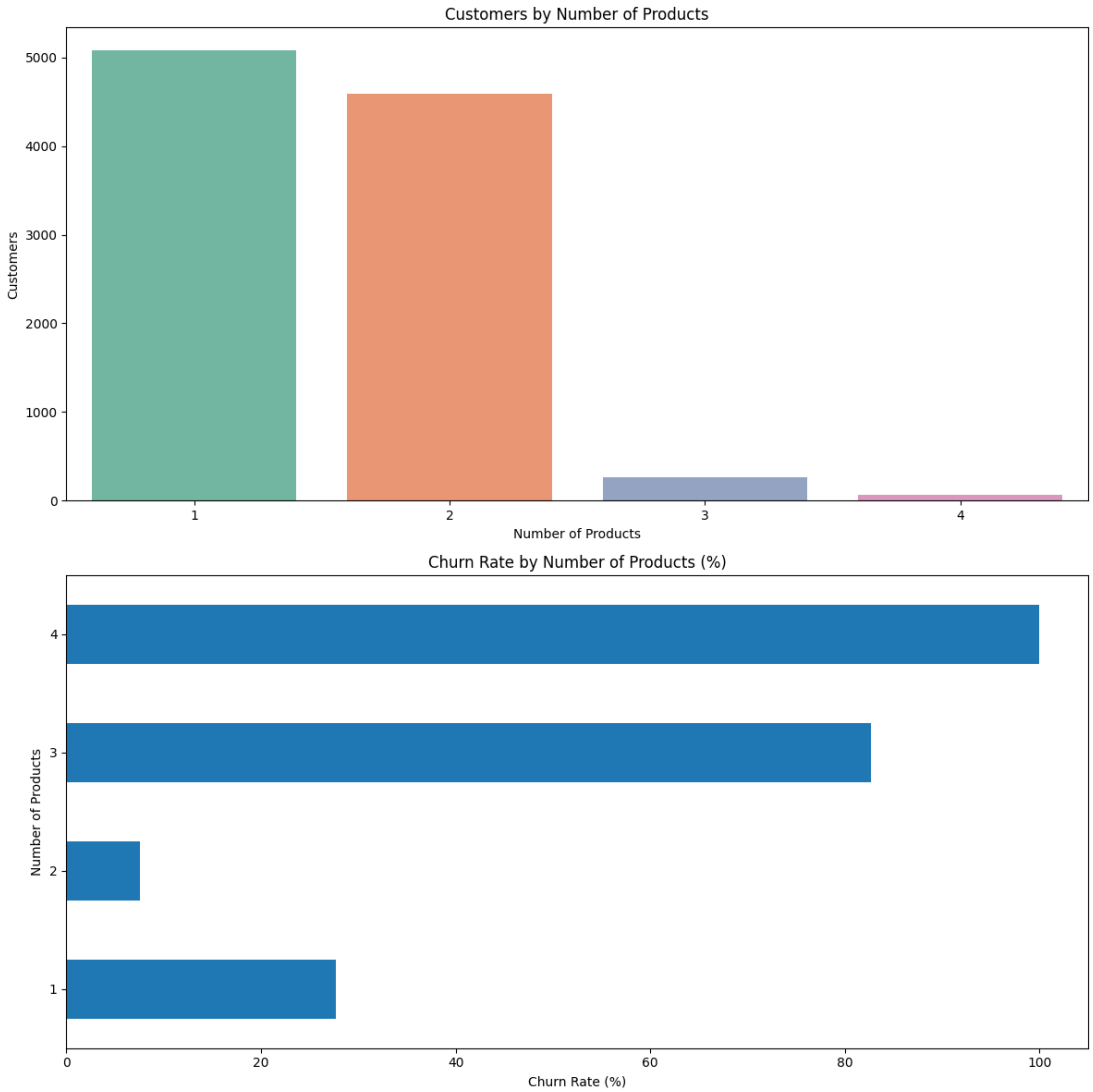

Fig. 15. Number of Products Values

Products Number is an indicator of the extent of the relationship between the customer and the bank. In Fig.15, a significant portion of the bank's clientele of total 5,084 customers, hold a single banking product. This group contain almost the half costumers from this dataset with churn rate of approximately 27.71%. The second largest group, comprising 4,590 customers, engages with two of the bank's products. Costumers with two products have a substantially lower churn rate of 7.58%, indicating the highest-level engagement with the bank's services. A much smaller segment of 266 customers engages with three products. This group's churn rate escalates to 82.71%. Even the group of customers who holds the third products is significantly smaller than the others a large proportion of them chooses to churn. The smallest group, with only 60 customers holding four products, exhibits a churn rate of 100%. This shows that all customers with four products associating with the banks have churned. The quantity of products

owned and customer retention have a nonlinear connection. First, clients who use a second product are far more likely to stay with the bank. This may be explained by the happiness or perceived value that comes from using a wider range of services, potentially improving the banking experience or offering financial incentives that exceed the advantages of engaging with a single product.

4.1.2 Continuous Variables

a) Tenure

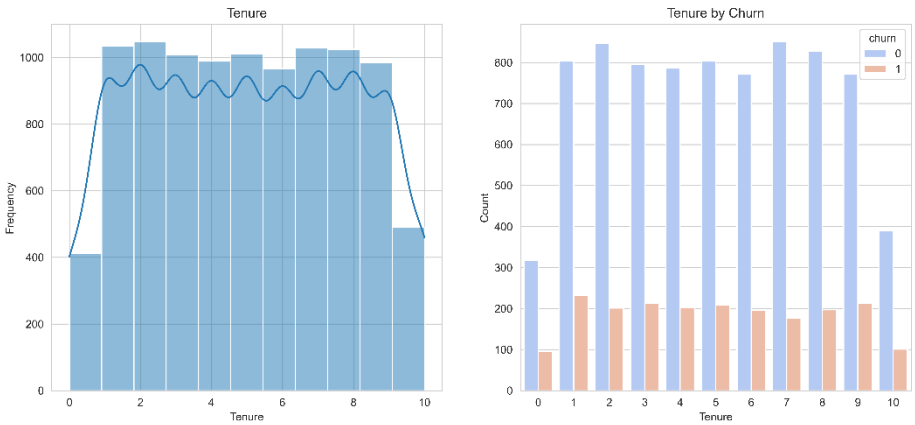

Fig. 16. Tenure Values

Longer tenure usually correlate with lower churn as customers with a long relationship are likely to have high satisfaction levels. In Fig. 16 shows a fairly constant spread across different tenure values with peaks around 0-1 years and 9-10 years. The second plot, comparing tenure with churn, indicates most of the tenure lengths are not much different. On the other end, the 0-1 and the 9-10 groups of tenure customers seem to display slightly diverse patterns compared to other groups. Even though the tenure can take values both 0 and 10, treating tenure as a continuous variable would be reasonable, and it would enable it to capture some trends or patterns. Otherwise, treating it as a discrete variable would make it lose information [20].

b) Age

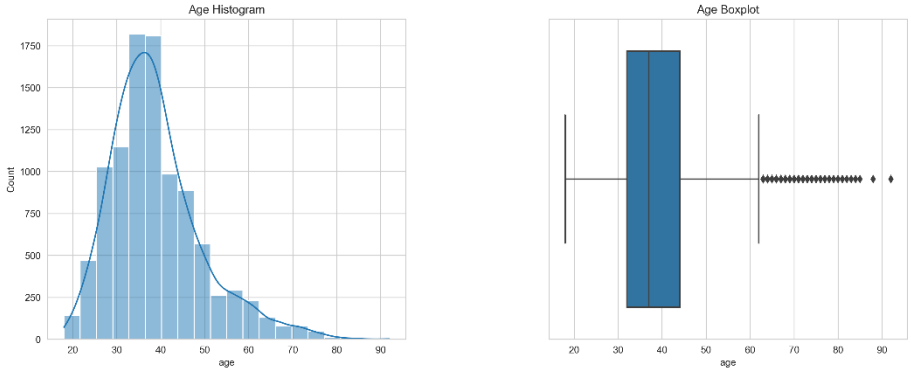

Fig. 17. Age Values

A customer's age is a demographic factor, which influences their banking demands and habits. The age has values from 18 to 92 with mean around 38 and standard deviation approximately 10,5. With median value 37 (very close to the mean), the first plot show that age distribution among the customers is relatively symmetrical with slightly skewed to the right. The presence of outliers towards the right end of the range is indicated by the skewness value, which is approximately 1.01 (>1). The outliers are revealed more clearly by the box plot at the right end of the adjacent values where the age values are greater than 61 [54] (Fig. 17).

c) Credit Score

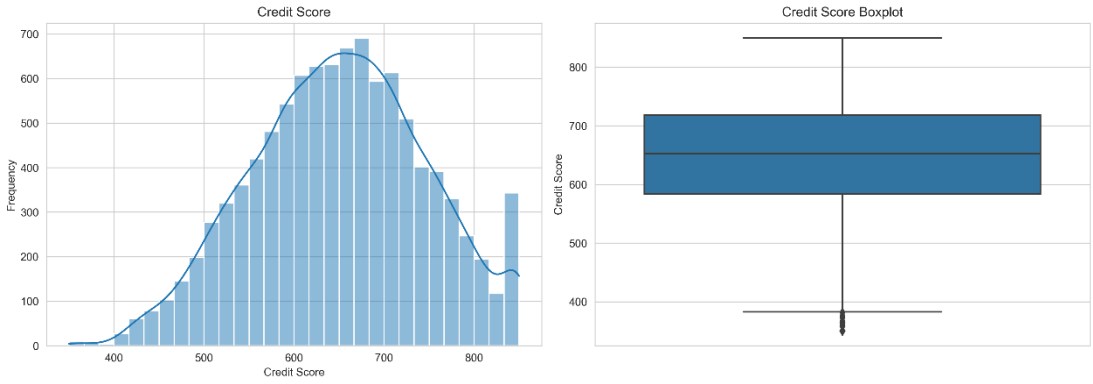

Fig. 18. Credit Score Values

A numerical expression based that represents the creditworthiness of an individual. It could influence a customer's eligibility for better offers or rates and might indirectly affect their

satisfaction. The Credit Score has a range of 350 to 850, with a standard deviation of roughly 96.7 and a mean value of roughly 650.5. The distribution of Credit Score among customers is reasonably symmetrical, because the median value of 652 close to the mean. The histogram visually supports the symmetrical distribution with a very slight skew to the left, due to the values of skewness that are around -0.07, closer to zero. Given the maximum score is 850, which is within the calculated upper bound, which is visually illustrated by the box blot, there are no extreme outliers on the higher end. However, the presence of a minimum score 350 below the lower outlier threshold indicates the presence of outliers outside the margins of the adjacent values (Fig. 18).

d) Estimated Salary

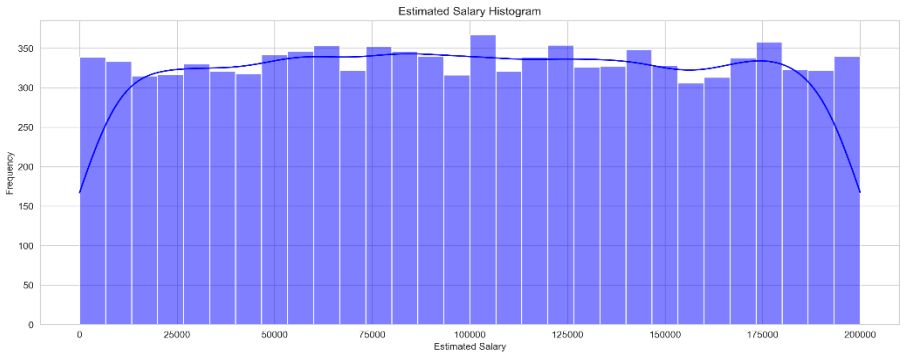

Fig. 19. Estimated Salary Values

The estimated salary could be inferred or based on direct reporting and indicates economic power. It has the mean and median really close to each other around 100,090 and 100,194, respectively. The range of the values starts from 11.58 to 199,992.48, which indicates the wide array of income levels for customers. This high standard deviation of 57,510.49 can also explain the wide spread of the values. Hence, it is important to have an understanding of the significance of variance in the salaries (Fig. 19). The very close to zero value of 0.0021 for the skewness of the distribution only buttresses the earlier observation of a symmetric distribution about the mean. With those results, the probabilities of almost a similar number of salary values falling almost equally on either side of the mean value and having high or low values are not skewed. With a kurtosis value of -1.1815, the distribution is platykurtic. This indicates a small number of outliers with a flatter peak when compared to a normal distribution and there is a much broader, more even spread of values with less clear peaks [2].

e) Balance

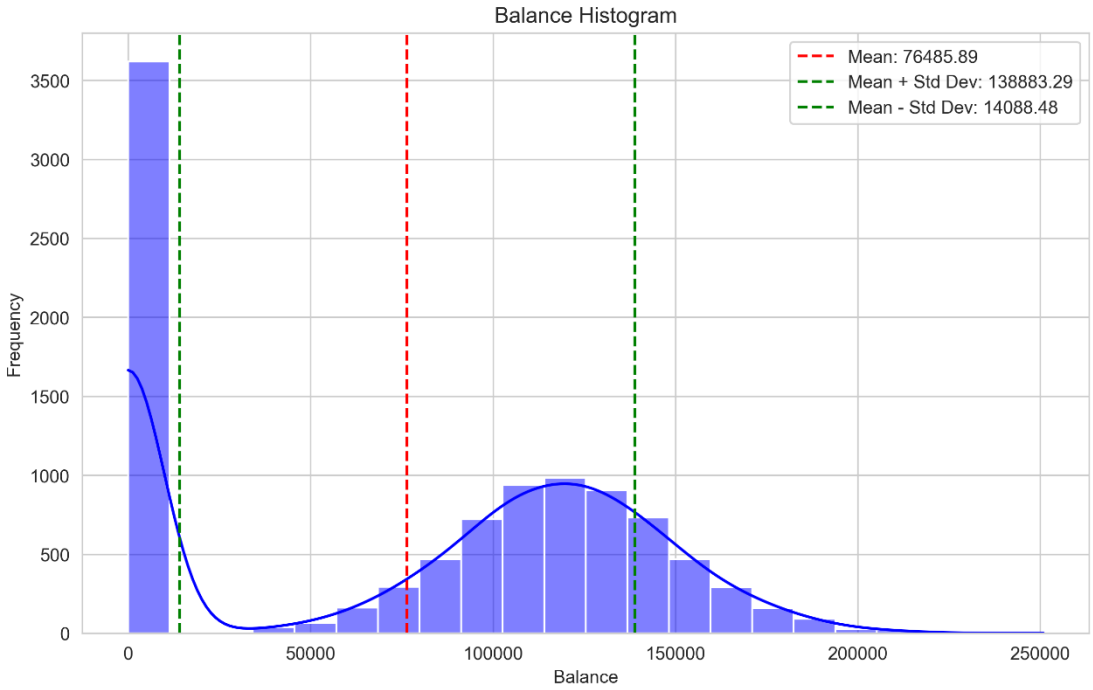

Fig. 20. Balance Values

The balance indicates the depth of the customer's financial relationship with the bank. Fig. 20 illustrates a good share of clients has a balance of zero, meaning that not every customer consistently maintains a positive balance. The range of balances is $0 to about $250,898 and with large standard deviation of $62,397.41 indicates that balance amounts are widely dispersed over a wide range. The median balance is higher than the mean ($97,198.54 vs. $76,485.89), pointing to a right-skewed distribution where the mean is elevated by outliers, suggesting that while most clients hold lower balances, a few have very high balances [77]. Having many zero balances and some very high values, the Robust Scaler can more effectively standardize balance amounts by utilizing the median and IQR, effectively reducing the impact of both zeroes.

4.2 Data Preparation

After first being gathered from ABC Bank, the data is delivered as a CSV file. The data is of great quality as it has received some processing and data cleaning before being uploaded to Kaggle as an open source. The data include float and integer numbers as well as object types. Classification models, such as LR, DT, RF, SVM, KNN that will be used for this analysis, are designed to handle both types of numeric data. It is common in datasets, that represent groups of people, features like age, gender, and country to be frequently repeat. In EDA analysis mentioned, that customer_id is a unique identifier for each customer and doesn't contain any

inherent predictive information regarding their behavior. However, it can be used as a catalyst for identifying duplicates, due to the fact that customer_id was typically designed to be unique for each customer. After the step is done it will be removed from this analysis.

4.3 Encoding

Label encoding may not be the best method when working with categorical variables that lack a natural order due to the false numerical order it creates [32]. On the other hand, if one hot encoding is performing on Gender values, where it has 2 distinct values, the features it creates (Males - Females) will be negative correlated (-1). If one variable can be almost accurately predicted from the other, it means they are covering similar aspects of the information, leading to redundancy [29]. The most efficient approach is to utilize the outcome of the EDA analysis to apply label encoding. Since there are slightly more males in this dataset than females, and because females have greater churn rates than males, males will be ranked with the number one, and females with number zero. Unlike gender, however, performing one-hot encoding on country with three discrete values (France, Germany, Spain) transforms it into a binary matrix, preventing the model from assuming a natural ordering. For algorithms that rely on numerical input because it treats each country as an independent feature, ensuring accurate representation and interpretation of categorical data without introducing ordinal bias.

4.4 Scaling

From EDA the continues features balance have values of zero and non-zero balances. While non-zero form a segment of balance that follows normal distribution, zero values could be considered as outlies. Age has outliers at right ends particularly to the older customers. For Estimated salary the peak is less pronounced, than in the distribution resulting in a broader more even spread with fewer outliers and for Credit Score there are some outliers on the lower end. Robust scaling uses the median and the interquartile range (IQR) for scaling, both of which are less sensitive to outliers than the mean and standard deviation. This means that extreme values have less influence on the scaling transformation, leading to a more uniform distribution of scaled features. While minimizing the impact of outliers, robust scaling preserves the relative rankings and distances of the original data points [28]. Although it is stated in the preliminary data analysis that tenure has 10 discrete values, scaling tenure relatively brings it in line with other continuous variables. Scaling ensures that all features have an equal contribution to the decision-making process, preventing features with large numerical ranges from overwhelming

those with smaller numerical ranges. Despite its discrete character and limited range, it seems that tenure benefits from this equal-footing rule in the sense that its variations are more meaningful within the models. The numerical range of values in estimated salary is much larger than tenure. If the distribution would be as it is now Min-Max scaling would work where a specific range of input features from 0 to 1 can be given.

4.5 Correlation

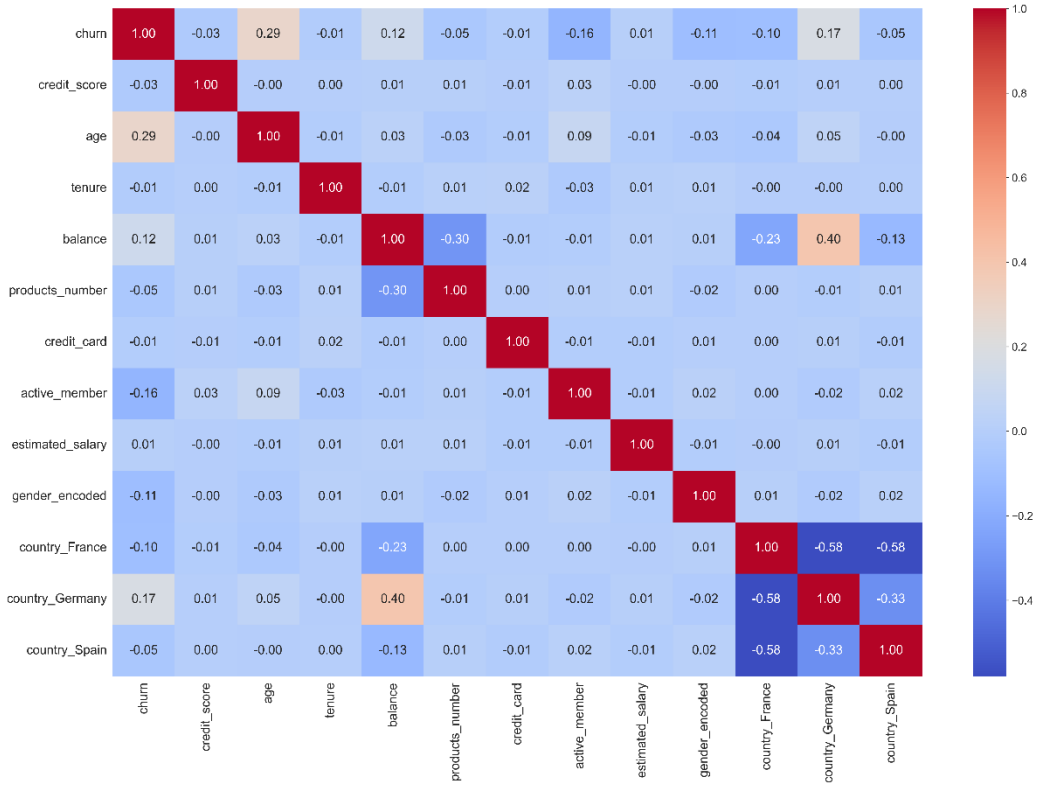

Fig. 21. Pearson Correlation

Given that the correlation, from fig. 21, of each feature is not accessed a standard threshold the extreme values of correlation threshold (|r|>= 0.7), the information in the correlation matrix table above indicates that there is no significant concern regarding [11]. Each variable might contribute unique insights without redundancy since variables exhibit low interdependencies. Since variables are not tightly coupled, they provide unique insights into the model, especially for understanding the outcome variable (churn) without the need for additional feature selection methods [22].

Chapter 5: Model Selection and Preparation

The machine learning models are evaluated to address the classification problem are

- DT
- KNN
- RF
- LR
- SVM

Each model is incorporated into a pipeline with the scaling preprocessing step to ensure a consistent approach to data preparation. The Logistic Regression model's max_iter parameter is specifically increased to ensure convergence given the complexity of the dataset.

5.1 Model's Preparation

The step involves selecting continuous features from the dataset for scaling, based on their range and distribution. The features for scaling include 'credit_score', 'age', 'tenure', 'balance', and 'estimated_salary'. The methodology accounts for how the original data values are distributed and their relative sizes to each other, ensuring that the normalization process appropriately scales the data without distorting its original structure [22]. To scale these features, a Robust Scaler and a Mix-Max Scaler from the scikit-learn library is utilized within a preprocessing pipeline. The Robust Scaler is chosen for its ability to handle outliers on 'credit_score', 'age', 'balance', while Mix-Max Scaler has been applied on 'tenure', 'estimated_salary' so that the value's range are between zero and one given that most of the discrete features are binaries. By choosing Robust Scaling for features likely affected by outliers and Min-Max Scaling for others, the preprocessing is tailored to the data's nature, promoting better model training and validation practices. Those scalers are applied only to the specified features, while the remainder of the discrete features in the dataset are left unscaled [42].

5.2 Stratified Cross-Validation

A five-split stratified K-fold cross-validation approach is used to accurately evaluate each model's performance. Stratified sampling ensures that the percentage of samples of each target class in each fold is about equal to that of the entire set. This not only helps in evaluating the

model more fairly but also in training more robust models as each fold resembles the complete dataset in terms of class distribution. This approach ensures that each fold used in the cross-validation process is a valuable representative of the whole dataset, especially in terms of the distribution of the target classes. Cross-validation scores are calculated based on accuracy, providing a straightforward metric to compare the models' performance [63]. The use of a Pipeline in Scikit-Learn is critical in preventing data leakage during the preprocessing steps. When data from the test or validation set is unintentionally used to inform the model training process, it is known as data leakage and can result in unduly optimistic performance predictions. In the pipeline, preprocessing (like scaling) and the model training happen within each fold of the cross-validation. This means that scaling is applied separately to each training fold, and then the same scaling parameters are used on the validation fold. This prevents information from the validation data from being used to scale the training data, thereby avoiding leakage [62]

Table 2. First Model Evaluation

ML models	*Mean Accuracy*
DT	0.79
KNN	0.84
RF	0.87
LR	0.81
SVM	0.86

In Table 2, even after the Stratified K-Fold cross-validation, the high scores from the Mean Accuracy are proof that imbalanced cause the models to favor the majority class. This is because accuracy measures the proportion of total correct predictions (including true negatives, which are high in imbalanced datasets), but it does not specifically reflect how well the model predicts the minority class [16]. Following those findings, the next phase is to measure accuracy by assigning distinct scores to the training and testing stages of cross-validation. This is done to determent whether ML models have learnt the training data set so well that they perform poorly on test data set. [48].

Table 3. Second Model Evaluation

ML models	*Accuracy of train set*	*Accuracy of test set*
DT	1.000	0.789
KNN	0.877	0.839
RF	1.000	0.862
LR	0.811	0.810
SVM	0.863	0.858

The results of using cross-validation by separating train and test folds helped in selecting a model that not only fits the training data well but also performs consistently on unseen data, with LR and SVM showing the most balanced performance. KNN demonstrates good performance and a decent compromise between fitting the training data and generalizing effectively to unseen data. The relatively small gap between the training and testing scores suggests that the model is not severely overfitting, which is a positive sign of its robustness. This indicates that KNN is effectively capturing the underlying patterns in the data without tailoring too specifically to the noise in the training set. On the other hand, both DT and RF have perfect training accuracy showing signs of overfitting (Table 3). One way to address overfitting for these models is to apply pruning to enhance their generalization capabilities on unseen data. Pruning deploys by removing parts of the tree that do not provide significant meaning in predicting the target variable, thus making the model simpler and more robust [24]. The following code shows the formulation of the process:

5.3 Enhanced Pruning

```
models_pruned = {
   "Pruned Decision Tree": DecisionTreeClassifier(max_depth=10, min_samples_split=50,
min_samples_leaf=25, random_state=42),
   "Pruned Random Forest": RandomForestClassifier(max_depth=5, min_samples_split=50,
min_samples_leaf=25, max_samples=0.8, random_state=42)
}
```

Restricting the depth of the DT and RF to 10 limiting the maximum depth it prevents the models from becoming overly deep and complex. When setting parameter min_samples_split=50 it determines the minimum number of samples a node (or tree in case of RF) must have before it can split. This helps prevent the model from learning overly specific patterns. Setting min_samples_leaf=25 defines the minimum number of samples that a leaf node (or tree) must have and further ensures that the leaf nodes (or trees) are not too specific, improving the generalizability of the model. Additionally, pruned Random Forest with an extra parameter max_samples=0.8 implies that every tree in the forest is created using a random selection of 80% of the samples available. Randomly choosing the data with which to build the trees adds

another layer of randomness to the model but is intended to make the model more robust. Finally, setting a seed value random_state= 42 to ensure that the results are reproducible.

Table 4. Third Model Evaluation

ML models	*Accuracy of train set*	*Accuracy of test set*
Pruned DT	0.8627	0.8608
Pruned RF	0.8628	0.8565

Pruning the DT results in the removal of branches with low importance, which could be due to low information gain. The results in a simplified tree that has a better chance of generalizing to unseen data. In a random forest, pruning each individual tree prevent the ensemble from becoming overly complex (Table 4). Even though random forests are much less sensitive to overfitting than single DT because they are an ensemble of them, pruning may help effectively reduce such a vulnerability by controlling how deep each of its trees may expand [35]. Generally, this method seems to be doing a very good job of its intention, which is minimizing overfitting, with consistent performance from training to testing. That suggests the models are quite able to generalize and will have good performance on new, unseen data without having "learned the noise" from our training dataset.

5.4 SMOTE

Inspired by pseudo-code from paper “SMOTE: Synthetic Minority Over-sampling Technique” in 2002 (page 329) a code is implemented to follow similar logic and create synthetic samples for the minority class. To ensure the integrity of the results the code will be interpreted piece by piece.

```
class SMOTETransformer(BaseEstimator, TransformerMixin):
    def __init__(self, imbalance_ratio=0.0, k=5):
        self.imbalance_ratio = imbalance_ratio
        self.k = k
```

The following line declares a new class called SMOTETransformer from BaseEstimator and TransformerMixin. These are base classes allow the transformer to integrate seamlessly with scikit-learn pipelines and functionalities. Then imbalance_ratio determines how much the

minority class should be oversampled relative to its original size, and k is the number of nearest neighbors to use when generating synthetic examples. By default, imbalance_ratio is set to 0.0 so that no synthetic samples will be created during the first evaluations.

```
def fit(self, X, y=None):
    return self
def transform(self, X, y=None):
    if y is not None:
        minority_sample = X[y == 1]
    else:
        minority_sample = X
```

Scikit-learn's transformer and estimator design, a fit method is used to learn something from the data. For SMOTE, the actual transformation doesn't involve fitting on the data, but fit must still be defined to maintain compatibility with scikit-learn's functionality (like pipelines). This method simply returns the instance (self). This is a placeholder that allows the transformer to be used in scikit-learn pipelines and similar constructs that expect a fit () method. If the target array y is provided. If y is provided, it selects the minority class samples (y == 1). If not provided, it assumes all of X are the samples to be oversampled. To identify and separate the minority class samples which are to be augmented via synthetic sample generation.

```
    T, num_attrs = minority_sample.shape
    num_synthetic_samples = int(len(minority_sample) * self.imbalance_ratio)
    synthetic = np.zeros((num_synthetic_samples, num_attrs))
    new_index = 0
    nbrs = NearestNeighbors(n_neighbors=self.k + 1).fit(minority_sample)
```

This line retrieves the number of minority samples (T) and the number of features (num_attrs). Calculates the number of synthetic samples to generate based on the imbalance ratio. A numpy array synthetic is to store the synthetic samples. new_index is used to track the insertion point in the synthetic array and fits a nearest neighbors' model to the minority samples. self.k + 1 neighbors are considered because the nearest (the first one) will be the point itself.

```
    def populate(N: int, i: int, nnarray: np.array):
        nonlocal new_index
        while N != 0:
            nn = randrange(1, self.k + 1)  # Pick one of k neighbors
            for attr in range(num_attrs):
```

```
                dif = minority_sample.iloc[nnarray[nn]][attr] - minority_sample.iloc[i][attr]
                gap = uniform(0, 1)
                synthetic[new_index][attr] = minority_sample.iloc[i][attr] + gap * dif
            new_index += 1
            N -= 1
```

A nested function that generates synthetic samples. It iteratively interpolates between a minority sample and one of its k-nearest neighbors to create a new sample. It loops ntil N synthetic samples have been created. Randomly selects one of the k neighbors and interpolates between the attributes.

```
        for i in range(T):
            nnarray = nbrs.kneighbors(minority_sample.iloc[[i]], return_distance=False)[0]
            populate(num_synthetic_samples // T, i, nnarray)
        synthetic_df = pd.DataFrame(synthetic, columns=X.columns)
        synthetic_df['churn'] = 1
```

For each minority sample, retrieves its nearest neighbors, and calls populate to generate the corresponding number of synthetic samples, distributed roughly evenly among all minority samples and then converts the synthetic samples array into a DataFrame and assigns the label for the minority class. Here, 'churn' is assumed to be the label field, indicating the samples are of the minority class.

```
        binary_cols = ['credit_card', 'active_member', 'gender_encoded', 'country_France',
'country_Germany', 'country_Spain', 'products_number']
        synthetic_df[binary_cols] = synthetic_df[binary_cols].round()
        return synthetic_df
```

Before return the DataFrame that containing the synthetic samples, it rounds discrete features to ensure the meaningful transformation remain after interpolation.

```
    def fit_resample(self, X, y):
        synthetic_df = self.transform(X, y)
        X_resampled = synthetic_df.drop('churn', axis=1)
        y_resampled = synthetic_df['churn']
        return X_resampled, y_resampled
```

This method combines transform with the resampling process, returning the resampled dataset with features (X_resampled) and target values (y_resampled). This allows it to be used directly in a pipeline.

```
smote_transformer = SMOTETransformer(imbalance_ratio=0.0)
synthetic_df = smote_transformer.transform(X, y)
```

The final snippet creates an instance of SMOTETransformer, setting the imbalance_ratio to 0.0. This ratio typically dictates how many synthetic samples to generate relative to the number of samples in the minority class. By adjusting the ration to 1.0, 2.0 and 3.0, the minority class will by increased by 100%, 200% and 300% respectively, ignoring the default settings. After that, the smote_transformer generate a DataFrame (synthetic_df) containing synthetic samples and applies SMOTE to the features (X) and the target (y) to create these samples for review.

Chapter 6: Assessing Model Performance Across Classes- Evaluation metrics

In the section evaluation metrics such accuracy mean, F1 – Score and AUC-ROC used on machine learning models to determine how they perform under different levels of class imbalance and how they benefit from the application of SMOTE. The default parameters have been used for each model so the results are rounded with two decimals. The main objective is to enhance our understanding of each model's capability to handle imbalanced datasets and to find the most effective strategy for improving their performance through artificial balancing of the class distribution [45]. In this analysis, four distinct datasets are evaluated to determine the impact of class balance on model performance as seen from the Table 5.

Table 5. Oversampling

Churn	*Original Class*	*Minority Class Doubled (100%)*	*Minority Class Tripled (200%)*	*Minority Class Quadrupled (300%)*
0	7963	7963	7963	7963
1	2037	4074	6111	8148

The final models that will be used for evaluation are:

- Pruned DT
- KNN
- Pruned RF
- LR
- SVM

Like before, StratifiedKFold is applied to ensure that each fold retains the same proportion of class labels as the original dataset (Sebastian Raschka, 2020). The dataset is split into training and testing subsets for each fold, while SMOTE is applied to only the training data to avoid

information leakage and to ensure the model learns to generalize from a balanced representation of classes [45]. A ColumnTransformer is set up to apply different scaling strategies to specified groups of features. The preprocessing steps and the actual model are encapsulated within a Pipeline to ensure that all preprocessing steps are fitted only on training data and consistently applied to both training and test data [62]. And finally, each run will calculate mean training and testing accuracies across all folds and computes whether the model is overfitting or underfitting after the input of the synthetic data to the training set [48].

6.1 Original Dataset

The summary of model performances of the classes distribution, where the minority class (churned) has only 2,037 instances compared to 7,963 for the majority class (Not Churned).

Table 6. First Run Accuracies

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Train Accuracy	0.86	0.88	0.86	0.81	0.86
Test Accuracy	0.85	0.84	0.85	0.81	0.86
Overfitting-Underfitting Indicator	0.01	0.04	0.01	0.00	0.00

Since no synthetic data has been added to the training set the results are the same from Model Selection and Evaluation chapter with none of the models showing significant signs of overfitting (Table 6).

Table 7. First Run Evaluations

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Mean Accuracy	0.85	0.84	0.85	0.81	0.86
F1-score	0.53	0.52	0.47	0.31	0.53
AUC	0.84	0.78	0.85	0.77	0.82

The Pruned RF has the greatest score of AUC, showing strong classification ability, but struggles with the F1-score, which could be due to a focus on the majority class. Logistic Regression appears to be the weakest model concerning handling class imbalance, as both F1-score and AUC have the lowest values compare to the other models. In terms of performance, both SVM and Pruned DT have the highest scores, making them potentially more reliable for consistent performance (Table 7).

6.2 Minority Class Doubled

The following run includes of model performances of the classes distribution, where SMOTE is applied to increase the minority class representation by 100%, resulting in a new class distribution where 'Churned' has 4074 instances compared to 'Not Churn' with the same number of instances.

Table 8. Second Run Accuracies

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Mean Train Accuracy	0.80	0.85	0.80	0.74	0.82
Mean Test Accuracy	0.83	0.81	0.85	0.79	0.86
Overfitting-Underfitting Indicator	-0.03	0.04	-0.05	-0.05	-0.04

After the synthetic data have been added all models expect KNN appear to have better generalization as training accuracy decreased. Due to its immediate neighborhood of data points, which may be negatively impacted by the properties of the synthetic data, KNN's performance may have been affected (Table 8).

Table 9. Second Run Evaluations

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Mean Accuracy	0.83	0.81	0.85	0.79	0.86
F1-score	0.59	0.55	0.58	0.47	0.61
AUC	0.83	0.79	0.84	0.77	0.85

After SMOTE, SVM shows the best balance between precision and recall, along with high class separation skills, making it the top performer. As starting to receive new data from the minority class, Pruned RF is managing balanced datasets, because it exhibits notable gains and keeps up high performance across all measures. While Pruned DT and KNN demonstrate some progress, their AUC and F1-score results suggest that they may not be fully capable of capturing subtler differences in the data or complex class relationships. Even yet, Logistic Regression exhibits the least amount of adaptation to the balanced dataset (Table 9).

6.3 Minority Class Tripled

The following run includes model performances of the classes distribution, where SMOTE is applied to increase the minority class representation by 200%, resulting in a new class distribution where 'Churned' has 6111 instances compared to 'Not Churn' with the same number of instances.

Table 10. Third Run Accuracies

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Train Accuracy	0.77	0.85	0.79	0.72	0.82
Test Accuracy	0.80	0.79	0.81	0.76	0.84
Overfitting-Underfitting Indicator	-0.03	0.06	-0.02	-0.04	-0.02

Even if the accuracies score for both training and test are slightly decreased compere to the previous result, the generalization has been improved. KNN's performance, however is starting to show signs of overfitting, another proof that it is affected by SMOTE technique due to the fact that they follow the same logic (Table 10).

Table 11. Third Run Evaluations

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Mean Accuracy	0.81	0.79	0.82	0.76	0.84
F1-score	0.58	0.56	0.59	0.50	0.62
AUC	0.83	0.80	0.84	0.77	0.85

The model that performs the best in this scenario is again the SVM, which demonstrates adaptability to the class balance while maintaining high performance in terms of accuracy, F1-score, and AUC. Because of its resilience, it is ideal for use in situations where balanced class representation and forecast accuracy are critical. Pruned RF is a good option where robust classification is more crucial over a balanced dataset because of its strong performance, particularly in terms of AUC. While they exhibit some adaptation to the balanced data, Pruned DT, K-Nearest Neighbors, and Logistic Regression continue to be less effective than SVM and Random Forest (Table 11).

6.4 Minority Class Quadrupled

In the fourth and final run of the model, where SMOTE is applied to increase the minority class representation by 300% made the dataset almost balanced, resulting in a new class distribution where 'Churned' has 8148 instances slight excessing the majority class of 'Not Churn' which remains with the same number of instances.

Table 12. Fourth Run Accuracies

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Mean Train Accuracy	0.76	0.86	0.78	0.72	0.82
Mean Test Accuracy	0.77	0.78	0.77	0.71	0.81
Overfitting-Underfitting Indicator	-0.01	0.08	0.01	0.01	0.01

The interpretation holds the same logic as now it seems, that the models are starting to have positive values between the differences of train and test, meaning that they are starting to adapt on the patterns (Table 12).

Table 13. Fourth Run Evaluation

	Pruned DT	*KNN*	*Pruned RF*	*LR*	*SVM*
Mean Accuracy	0.78	0.77	0.77	0.71	0.81
F1-score	0.55	0.55	0.56	0.50	0.60
AUC	0.82	0.80	0.84	0.77	0.85

Now that the classes are slightly balanced, with churns surpassing the non-churns with some observations, the metrics are starting to deteriorate. However, the evaluations seem to yield better results compared to the "First Run" where the classes were imbalanced. Measurements here show that Support Vector Machine (SVM) performs better than the others. Because of its robustness, it is ideal for deployment in situations with a balanced class distribution, guaranteeing accurate and fair predictions. While pruned DT and pruned RF exhibit some robustness, they are unable to match SVM's overall performance. In every run, KNN and LR seem less appropriate handle this task than the others (Table 13).

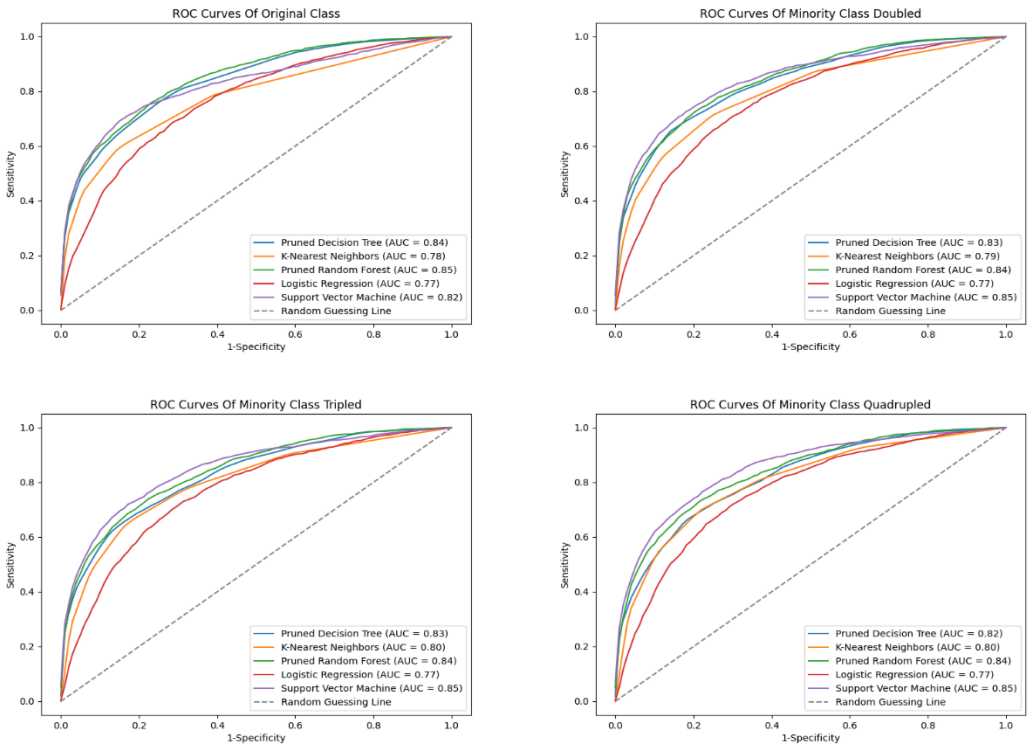

Fig. 22. Roc Curves for Every Run

The optimal model for all runs is SVM which is yields the dominant curve, which is also the curve with the biggest area, if one ROC curve dominates all others. The AUC metric, particularly in the context of ROC (Receiver Operating Characteristic) AUC, is less sensitive to class imbalance compared to other metrics like accuracy or precision [66]. Generally, as the SMOTE level increases (from the original dataset to 300% increase), the AUC scores for most models do not change significantly (Fig. 22).

Chapter 7: Conclusions

This chapter presents a synthesis of the findings of the study and wider implications, given a focus on applying machine learning methods to predict customer churn in a banking context. The following sections, which discuss the proposed methodology, suggest the limitations met during the study's implementation and provide ways for future research development in that direction. The prediction of customer churn with the different machine learning models applied in the project, therefore, demonstrated that it is indeed feasible to reach a relatively good level of performance and accuracy in such prediction with the set of features given. Of the tested models, only the Support Vector Machine (SVM) was robust to all tested scenarios and, in fact, more robust than all models compared, taking into account changes in the class distribution adjusted by SMOTE. The SVM still keeps very high accuracy and exhibits superiority in F1-scores and AUC metrics, pointing to its ability in balancing precision vs. recall effectively and very good capacity for class separation. The use of SMOTE for addressing class imbalance proved instrumental in enhancing the predictive accuracy of models, especially for the less represented classes. Such an approach helped balance out the representation of instances of the minority class and could, therefore, provide subtler insights into the reasons for customer churn.

6.1 Research Overview

This analysis was focused on conducting an extensive assessment of multiple machine learning models that would allow for checking their performance under different conditions of class imbalance and for assessing the extent to which the SMOTE can provide valuable performance improvement. Our intention was to gather more information on the ability of these models to work with imbalanced data and determine the best approach to enhancing their performance that would involve artificial balancing of the class distribution. Our beginning involved evaluating models on a dataset with a substantial class imbalance. Specifically, we examined models such as the Pruned DT, KNN, Pruned RF, LR, and SVM performance and obtained baseline metrics for accuracy, F1-score, and the Area Under the Curve. This first assessment allowed us to acknowledge the fundamental limitations associated with the skewed class distribution. Followed by incrementally increasing the representation of the minority class, it was used with SMOTE at four different levels: original 100%, 200%, and 300%, in order to generate synthetic samples of the minority class, thus balancing the dataset artificially. The same set of models was rerun with each level of adjustment to observe the resulting difference in model performance metrics. This approach helped us test how improving class balance would affect the model, its accuracy, precision, and recall, and, naturally, the general power to separate classes of

outcomes. This was only possible due to the comparison of results between various scenarios, with pinpoint accuracy on which level of SMOTE optimization brought out the most benefit for models. The implementation of a machine learning model for customer churn prediction has shown that a high level of performance and accuracy can be reached with the provided dataset. The Support Vector Machine was the most stable and performed well in all settings and was significantly better even compared with its performance increased by SMOTE adjustments of different class distribution. While some minor losses in accuracy were present, both F1-scores and AUC metrics have shown better discriminative ability of SVM classes. In the original dataset, the highest accuracy was 0.86, which suggests that SVM was quite effective even with the imbalanced data. As SMOTE is introduced, the f1-score improves, showing better balance between precision and recall due to the balanced dataset. The best F1 score of 0.62 was realized with 200% increases in the minority class. The AUC score fluctuated but remained high, at 0.85, meaning that SVM has high capability in distinguishing between classes even under balanced conditions. Notably, the dataset with a 200% increase in the minority class outperformed other alternatives in all aspects. In practical terms, this means that such a dataset achieves the most reasonable balance between the precision and recall of the identified outcomes. The slight decrease in accuracy in the 300% increase scenario shows that too large of an increase in the number of the minority class may have diminishing returns.

6.2 Limitations

6.2.1 Hyperparameter Tuning

Hyperparameter tuning is the process of optimal configuration selection for hyperparameters, which are not learned when the data are trained. Several parameters control the model's training, while hyperparameters are set before the training process starts. It is necessary to remember that a machine learning practitioner should set hyperparameters, as they are not learned from the data. Hyperparameter tuning aims is to find the optimal mixing of these settings that allows the model to perform at its best, which is usually determined by its accuracy, precision, or any other reliable indicator on unseen data. An effective hyperparameter tuning can be essential for ensuring that a model generalizes well in practice, i.e., on real-world data different from the used training dataset. [74]. There was a drawback of the computational cost that hyperparameter search methods are computationally expensive and time-consuming. This is caused by the pros of exhaustive testing of multiple combinations of parameters, thus, the higher is the search space of potential variants the more computational effort is demanded. These expenses also increase with the complexity of the model and the amount of the dataset.

6.2.2 Data Limitations

The unavailability of data sources is one of the limitations encountered in the study. Conformably, data sources limitation frequently occurs due to privacy challenges, proprietary data policies, or simply an absence of data collected on certain variables of interest. The unavailability of more detailed and relevant data sources, thus, may limit more extensive research and introduce bias, as the model ends up being trained for incomplete views of customer behavior [53].

6.3 Future Work

6.3.1 Log Transformation

The log transformation is a proven technique to normalize the observed data obtained during an experiment to better meet the assumptions of some statistical analyses. One of these claims is that the variability of the value of the response factor must be constant. When the degree of variability, measured by the standard deviation on the differential scale, can be roughly equated to a mean measured on the same scale, a log transformation can be used to standardize these degrees. More technically, the variance is stabilized. Log transformation can also provide a normal distribution of real observations. Alternatively, an even more frequently realized normal distribution of sample means is achievable, which is useful for many stages of inference [17].

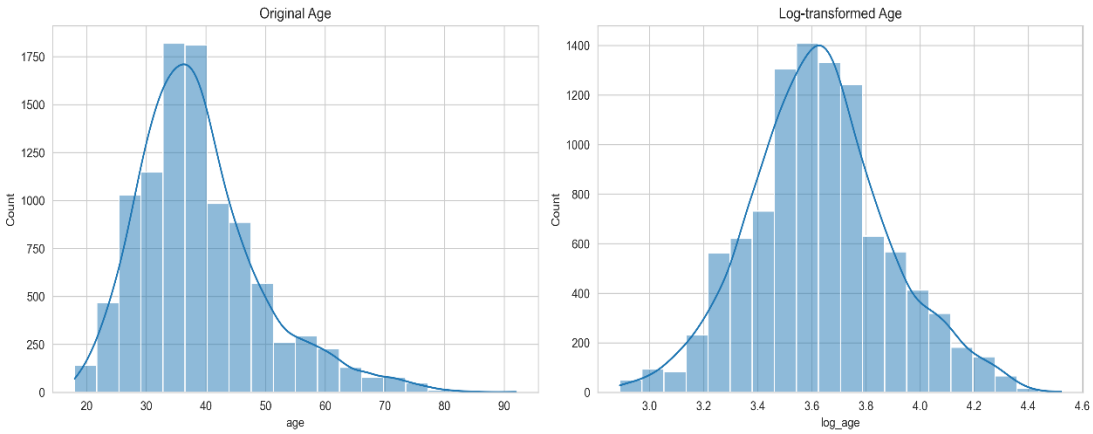

Fig. 23. Log Transformation Example for Age

6.3.2 Segmentation

Segmentation is the process of treating different subsets of the data separately [13]. This could entail treating accounts with zero balances differently from those with non-zero balances, considering the balance feature (Fig. 24).

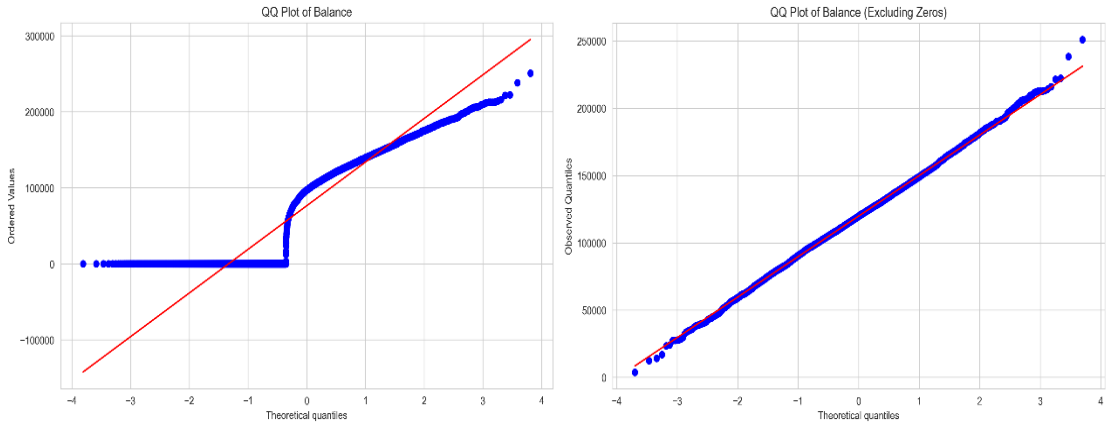

Fig. 24. Quantile-Quantile Example for Balance

Quantile-Quantile (QQ) plots are graphical aids used in the determination of whether a given data set could sensibly have arisen from another data set with a distribution, presumably in theory. Except for the case, all the data points lie in a straight line on the plot, usually at an angle of 45 degrees, in this case, the data is modeled by a normal distribution accurately. Discrepancies from this line reveal normality problems, and the form of the discrepancies might indicate skewness or the presence of outliers [67]. One way is creating a binary feature indicating whether the balance is zero (0) and non-zero (1), but that will lead to loss of information. However, filtering out zero balances and focusing only on non-zero balances is a form of data segmentation. The practice of splitting the data into subsets according to particular criteria to allow for more focused analysis.

6.3.3 Feature Engineering

Feature creation is a process during which new features are created from existing data attributes to enrich the predictive ability of ML models. The procedure intends to discover more insights from the existing data that could not be obtained from the unique features only. Polynomial features and interactions between variables, which allow to capture more complex relationships in data; they make models more flexible [20].

References

1. A.Albert and J. A. Anderson (1984). “On the existence of maximum likelihood estimates in logistic regression models”. DOI: https://doi.org/10.1093/biomet/71.1.1
2. A.Spanos (2019). “Probability theory and statistical inference: Empirical modeling with observational data”. ISBN: 0511010974.
3. Abbas Keramati, Hajar Ghaneei & Seyed Mohammad Mirmohammadi (2016). “Developing a prediction model for customer churn from electronic banking services using data mining”. DOI: 10.1186/s40854-016-0029-6
4. Adam D. Leache, Bruce Rannala (2010). “The Accuracy of Species Tree Estimation under Simulation: A Comparison of Methods”. DOI: https://doi.org/10.1093/sysbio/syq073
5. Ahmed Hamed, Mohamed Tahoun, and Hamed Nassar (2022). “KNN^{HI}: Resilient KNN algorithm for heterogeneous incomplete data classification and K identification using rough set theory”. DOI: https://doi.org/10.1177/01655515211069539
6. Amgad Muneer et al. (2022). “Predicting customers churning in banking industry: A machine learning approach”. DOI: 10.11591/ijeecs.v26.i1.pp539-549
7. Andrew P. Bradley (1997). “The use of the area under the ROC curve in the evaluation of machine learning algorithms”. DOI: https://doi.org/10.1016/S0031-3203(96)00142-2
8. Andy Liaw & Matthew Wiener (2022). “Classification and Regression by randomForest”. Link: https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf
9. Aurélie Lemmens & Christophe Croux (2006). “Bagging and Boosting Classification Trees to Predict Churn”. DOI: https://doi.org/10.1509/jmkr.43.2.276
10. Briscoe, Erica, Feldman, Jacob (2006). “Conceptual complexity and the bias/variance tradeoff”. DOI: https://doi.org/10.1016/j.cognition.2010.10.004
11. Carsten F. Dormann et al. (2013). “Collinearity: a review of methods to deal with it and a simulation study evaluating their performance”. DOI: 10.1111/j.1600-0587.2012.07348.x
12. Chih-Fong Tsai & Yu-Hsin Lu (2010). “Data Mining Techniques in Customer Churn Prediction”. Link: https://www.researchgate.net/publication/240976667_Data_Mining_Techniques_in_Customer_Churn_Prediction . DOI: http://dx.doi.org/10.2174/1874479611003010028
13. Christos Bialas, Andreas Revanoglou & Vicky Manthou (2019). “Improving hospital pharmacy inventory management using data segmentation”. DOI: https://doi.org/10.1093/ajhp/zxz264
14. David Bowes, Tracy Hall & David Gray (2012). “Comparing the performance of fault prediction models which report multiple performance measures: recomputing the confusion matrix”. DOI: https://doi.org/10.1145/2365324.2365338
15. David M. W. Powers, 2020. “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation”. https://doi.org/10.48550/arXiv.2010.16061
16. Davide Chicco & Giuseppe Jurman (2020). “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. DOI: https://doi.org/10.1186/s12864-019-6413-7
17. Douglas Curran-Everett (2018). “Explorations in statistics: the log transformation”. DOI: 10.1152/advan.00018.2018
18. Eugenio Jay Zuccarelli (2020). “Handling Categorical Data, The Right Way”. Link: https://towardsdatascience.com/handling-categorical-data-the-right-way-9d1279956fc6
19. Famili A., Shen Wei-Min, Weber Richard, Simoudis Evangelos (1997). “Data Preprocessing and Intelligent Data Analysis”. DOI : 10.3233/IDA-1997-1102
20. Hastie T., Tibshirani R. & Friedman J. (2001). “The Elements of Statistical Learning”. Link: https://link.springer.com/book/10.1007/978-0-387-21606-5 .
21. Hua Chai et al. (2018). “A novel logistic regression model combining semi-supervised learning and active learning for disease classification”. Link: https://www.nature.com/articles/s41598-018-31395-5/figures/3
22. Ivan Izonin et al. (2022). “A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain”. DOI: https://doi.org/10.3390/math10111942
23. J. Burez & D. Van den Poel (2009). “Handling class imbalance in customer churn prediction”. DOI: https://doi.org/10.1016/j.eswa.2008.05.027

24. J. Ross Quinlan (1992). “C4.5: Programs for Machine Learning 1st Edition”. ISBN-13 : 978-1558602380
25. Javier M. Moguerza & Alberto Muñoz (2006). “Support Vector Machines with Applications”. DOI: 10.1214/088342306000000493
26. Jesse Davis & Mark Goadrich (2006). “The relationship between Precision-Recall and ROC curves”. DOI: https://doi.org/10.1145/1143844.1143874
27. Jia-Bao Wang et al. (2021). “AWSMOTE: An SVM-Based Adaptive Weighted SMOTE for Class-Imbalance Learning”. DOI: https://doi.org/10.1155/2021/9947621
28. Jiawei Han, Micheline Kamber & Jian Pei (2012). “Data Mining: Concepts and Techniques Third Edition”. DOI: 10.1016/C2009-0-61819-5
29. Jireh Yi-Le Chan et al. (2022). “Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review”. DOI: https://doi.org/10.3390/math10081283
30. João B. G. Brito et al. (2024). “A framework to improve churn prediction performance in retail banking Research”. DOI: https://doi.org/10.1186/s40854-023-00558-3
31. John Hadden, Ashutosh Tiwari, Rajkumar Roy & Dymitr Ruta (2007). “Computer Assisted Customer Churn Management: State-Of-The-Art and Future Trends”. DOI: https://doi.org/10.1016/j.cor.2005.11.007
32. John T. Hancock & Taghi M. Khoshgoftaar (2020). “Survey on categorical data for neural networks”. DOI: https://doi.org/10.1186/s40537-020-00305-w
33. Kazumi Wada (2020). “Outliers in official statistics”. DOI: 10.1007/s42081-020-00091-y
34. Ke Peng,Yan Peng & Wenguang Li (2023). “Research on customer churn prediction and model interpretability analysis”. DOI: https://doi.org/10.1371/journal.pone.0289724
35. Kiran Bangalore Ravi & Jean Serra (2017). Cost-complexity pruning of random forests. DOI: https://doi.org/10.48550/arXiv.1703.05430
36. Kirchner, Antje & Curtis S. Signorino (2018). “Using Support Vector Machines for Survey Research”. DOI: https://doi.org/10.29115/SP-2018-0001
37. Lionel Cucala et al. (2009). “A Bayesian Reassessment of Nearest-Neighbor Classification”. DOI: https://doi.org/10.1198/jasa.2009.0125
38. Manas Rahman & V Kumar (2020). “Machine Learning Based Customer Churn Prediction In Banking”. DOI : 10.1109/ICECA49313.2020.9297529
39. Marco Sandri & Paola Zuccolotto (2009). “Analysis and correction of bias in Total Decrease in Node Impurity measures for tree-based algorithms”. DOI: https://link.springer.com/article/10.1007/s11222-009-9132-0
40. Marina Sokolova & Guy Lapalme (2009). “A systematic analysis of performance measures for classification tasks”. DOI: https://doi.org/10.1016/j.ipm.2009.03.002
41. Marina Sokolova, Nathalie Japkowicz & Stan Szpakowicz (2006). “Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation”. DOI: https://link.springer.com/chapter/10.1007/11941439_114
42. Md Manjurul et al. (2021). “Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance”. DOI: https://doi.org/10.3390/technologies9030052
43. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2018). “Foundations of Machine Learning, second edition”. ISBN : 9780262039406. Link : https://books.google.gr/books?hl=el&lr=&id=dWB9DwAAQBAJ&oi=fnd&pg=PR5&dq=Foundations+of+Machine+Learning&ots=AzoP0Pv-o2&sig=YhhovcGyXezfgd0eg8uaAbgp7KQ&redir_esc=y#v=onepage&q=Foundations%20of%20Machine%20Learning&f=false
44. Mike West. “What is a node in a decision tree?”. Link: https://www.quora.com/What-is-a-node-in-a-decision-tree
45. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer (2002). “SMOTE: Synthetic Minority Over-sampling Technique”. DOI: https://doi.org/10.1613/jair.953
46. Ngai Li Xiu & D.C.K. Chau (2009). “Application of data mining techniques in customer relationship management: A literature review and classification”. DOI: https://doi.org/10.1016/j.eswa.2008.02.021
47. Olivier Caelen (2017). “A Bayesian interpretation of the confusion matrix”. DOI: 10.1007/s10472-017-9564-8

48. Osval Antonio Montesinos López, Abelardo Montesinos López & Jose Crossa (2022). “Overfitting, Model Tuning, and Evaluation of Prediction Performance”. DOI: 10.1007/978-3-030-89010-0_4
49. Palak Mahajan et al. (2023). “Ensemble Learning for Disease Prediction: A Review”. DOI: https://doi.org/10.3390/healthcare11121808
50. Paul D. Allison (1999). “Logistic Regression”. Link: https://statisticalhorizons.com/wp-content/uploads/2022/01/LR-Sample-Materials.pdf
51. Pedro Domingos (2012). “A Few Useful Things to Know About Machine Learning”. DOI: 10.1145/2347736.2347755
52. Phuong Bich Le & Zung Tien Nguyen (2022). “ROC Curves, Loss Functions, and Distorted Probabilities in Binary Classification”. DOI: https://doi.org/10.3390/math10091410
53. R. Iniesta, D. Stahl & P. McGuffin (2016). “Machine learning, statistical learning and the future of biological research in psychiatry”. DOI:10.1017/S0033291716001367
54. Rand R. Wilcox (2011). “Introduction to robust estimation and hypothesis testing (3rd ed.). Academic Press”. Hardback ISBN: 9780123869838. eBook ISBN: 9780123870155.
55. Rezvan Ehsani and Finn Drabløs (2020). “Robust Distance Measures for kNN Classification of Cancer Data”. DOI: https://doi.org/10.1177/1176935120965542
56. Rokach L, Maimon O. (2014). “Data Mining with Decision - Trees Theory and Applications”. Link: https://www.worldscientific.com/worldscibooks/10.1142/9097#t=aboutBook
57. Ron Kohavi (1995). “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. DOI: https://dl.acm.org/doi/10.5555/1643031.1643047
58. S. B. Kotsiantis (2011). “Decision trees: a recent overview”. Link: https://link.springer.com/article/10.1007/s10462-011-9272-4 . DOI: 10.1007/s10462-011-9272-4
59. S. James Press & Sandra Wilson (2007). “Choosing Between Logistic Regression and Discriminant Analysis”. Link: https://people.stat.sc.edu/hoyen/PastTeaching/STAT705-2019/Presentation/LogitOrLDA.pdf
60. Sarang Narkhede (2018). “Understanding Confusion Matrix”. Link: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
61. Schölkopf B. & Smola A. J. (2002). “Learning with Kernels: Support Vector”. ISBN: 0-262-19475-9
62. scikit-learn Machine Learning in Python”. Link: https://scikit-learn.org/stable/
63. Sebastian Raschka (2020). “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning”. DOI: https://doi.org/10.48550/arXiv.1811.12808
64. Seema Singh (2018). “Understanding the Bias-Variance Tradeoff”. Link: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229
65. Shichao Zhang, Chengqi Zhang & Qiang Yan (2010). “Data preparation for data mining”. DOI: https://doi.org/10.1080/713827180
66. Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas (2006). “Handling imbalanced datasets: A review”. Link: https://www.researchgate.net/publication/228084509_Handling_imbalanced_datasets_A_review
67. Statistics By Jim. “QQ Plot: Uses, Benefits & Interpreting. Link: https://statisticsbyjim.com/graphs/qq-plot/
68. Sylvain Arlot & Alain Celisse (2010). “A survey of cross-validation procedures for model selection”. DOI : https://doi.org/10.1214/09-SS054
69. T.Vafeiadis et al. (2015). “A comparison of machine learning techniques for customer churn prediction”. DOI: https://doi.org/10.1016/j.simpat.2015.03.003
70. Thomas J. Ostrand & Elaine J. Weyuker (2007). “How to Measure Success of Fault Prediction Models”. DOI: https://doi.org/10.1145/1295074.1295080
71. Tom Fawcett (2006). An introduction to ROC analysis. DOI: https://doi.org/10.1016/j.patrec.2005.10.010
72. Vapnik V. ,1995. “The Nature of Statistical Learning Theory”. DOI: 10.1007/978-1-4419-3160-1
73. W. Nor Haizan et al. (2012). “A comparative study of Reduced Error Pruning method in decision tree algorithms”. DOI: 10.1109/ICCSCE.2012.6487177
74. Wee How Khoh et al. (2023). “Predictive Churn Modeling for Sustainable Business in the Telecommunication Industry: Optimized Weighted Ensemble Machine Learning”. DOI: https://doi.org/10.3390/su15118631

75. Xihai Zhang et al. (2021). “Dissolved Gas Analysis for Transformer Fault Based on Learning Spiking Neural P System with Belief AdaBoost”. Link: https://www.researchgate.net/figure/A-common-ensemble-learning-architecture_fig1_352212528
76. Yaya Xie , Xiu Li , Ngai & Weiyun Ying (2009). “Customer churn prediction using improved balanced random forests”. DOI: https://doi.org/10.1016/j.eswa.2008.06.121
77. Zach Bobbitt (2021). “Statology”. Link: https://www.statology.org/sigmoid-function-excel/. Link2: https://www.statology.org/mean-greater-than-median/