Machine Learning in Finance

Summer Term 2025

Chair of Finance

Prof. Dr. Martin Hibbeln

Assistant: Noah Urban, M.Sc.

Predicting Bank Customer Churn Using Machine Learning Techniques

Name:	**Thi Hong Cam, Nguyen**	**Ayda Beiram Zadeh**
Matriculation-No:	3236351	3207263
Location:	47057 Duisburg, Germany	45127 Essen, Germany

Table of Contents

List of abbreviations *3*

List of Figures *4*

List of Tables *4*

1. Introduction *5*

2. Data *5*

3. Methodology *6*

3.1 Logistic Regression **6**

3.2. Regularization **8**

3.3 Random Forest and Boosting **9**

3.3. Model Evaluation **10**

4. Data Preparation and Exploratory Analysis *13*

4.1 Dataset Structure and Variables **13**

4.2 Missing Values **13**

4.3 Class Imbalance **14**

4.4 Exploratory Data Analysis **14**

5. Model Training and Hyperparamenter Tuning *18*

6. Result and Discussion *19*

7. Conclusion *26*

References *27*

List of abbreviations

AUC: Area Under Curve

ROC: Receiver Operating Characteristic

SMOTE: Synthetic Minority Oversampling Technique

List of Figures

Figure 1: Logistic Model for the dataset Bank Customer Churn 8
Figure 2: Random Forest and Boosting 10
Figure 3: Simple Confusion Matrix 11
Figure 4: Roc Curve 12
Figure 5: Class Distribution of the Target Variable Exited (0 = Stayed, 1 = Churned) 14
Figure 6: Customer Age 15
Figure 7: Customer by Geography and Gender 16
Figure 8: Descriptive Statistic 16
Figure 9: Distribution of categorical variables 17
Figure 10: Correlation Matrix 17
Figure 11: Parameter Tuning for XGBoost 19
Figure 12: ROC Curve for Logistic Regression 20
Figure 13: Ridge Regression Performance 21
Figure 14: ROC Curve for Random Forest 22
Figure 15: Random Forest outperforms Decision Tree 23
Figure 16: ROC Curve for XGBoost with AUC=0.886 24
Figure 17: Variable importance of the models 25

List of Tables

Table 1 Performance metrics of the models 24

1. Introduction

Customer churn is an important topic for many industries, especially for the banking industry, since clients are the main source of bank income.[1] The phenomenon of customer churn, known as "churn," i.e., the transition from one service provider to another occurs due to reasons such as availability of the latest technology, bank staff are friendly to customers, low interest rates, close geographical location, diverse services offered.[2] The longer the customer has been in the bank, the higher the customer worth and it is such expensive to obtain new clients. Therefore, the analysis of bank customer churn can reflect which factors affect the customer's choice of retention, and in the later stage, it can provide corresponding solutions and plans to guarantee the bank's income. The accurate customer prediction helps the company to persuade the appropriate customer to stay at the right time.[3]

The advancement of technology, especially the rapid evolution of machine learning algorithms and theories, has introduced innovative approaches to predicting customer churn. Machine learning techniques enable the automatic extraction of meaningful information from extensive historical datasets, allowing for the identification of underlying patterns and relationships that support the prediction of future trends and customer behaviors.[4]

This term paper introduces several machine learning techniques applied to the dataset to solve the given problem and predict customers who exit, including their mathematical foundations and practical implementations of machine learning models. Furthermore, it evaluates the models based on performance metrics—specifically the Area Under the Curve (AUC)—to determine the most effective approach.

2. Data

The bank customer churn dataset is used for predicting customer churn in the banking industry. The data was collected in Kaggle and has 165.034 observations and 14 variables including:

Customer ID: A unique identifier for each customer
Surname: The customer's surname or last name
Credit Score: A numerical value representing the customer's credit score

[1] Li *et al.*, 2023: p.1065
[2] Nguyen *et al.*, 2024: p.368.
[3] Li *et al.*, 2023: p.1066.
[4] Li and Yan, 2025: p.20

Geography: The country where the customer resides (France, Spain or Germany)

Gender: The customer's gender (Male or Female)

Age: The customer's age.

Tenure: The number of years the customer has been with the bank

Balance: The customer's account balance

NumOfProducts: The number of bank products the customer uses (e.g., savings account, credit card)

HasCrCard: Whether the customer has a credit card (1 = yes, 0 = no)

IsActiveMember: Whether the customer is an active member (1 = yes, 0 = no)

EstimatedSalary: The estimated salary of the customer

Exited: Whether the customer has churned (1 = yes, 0 = no)

This study uses the "Exited" as dependent variable, and "Gender"."Age", "Tenure", "Geography", "Balance", "Num Of Products", "Has Cr Card", "Is Active Member" and "Estimated Salary" as covariates (independent variable) respectively.

3. Methodology

This section presents the machine learning methods used in the analysis. The objective of this study is to predict whether bank credit card customers are likely to churn, using various machine learning techniques. In addition to making predictions, the models identify and explain the key factors that influence customer churn. These insights can support banks in proactively managing customer relationships and improving management strategies. Each model is assessed based on its predictive performance, with particular focus on AUC. The model with the highest AUC value is selected as the optimal one for this task.

3.1 Logistic Regression

Classification methods play a vital role in machine learning and data mining tasks. It is estimated that around 70% of data science challenges involve classification. Among the various techniques used for classification, logistic regression is one of the most widely applied and effective approaches, especially for binary classification problems. In contrast, multinomial classification is used when the target variable consists of more than two categories, allowing for the prediction of multiple class labels.[5]

[5] Navlani, 2024

Logistic regression can be used for various classification problems, such as spam detection, default customer prediction, cancer prediction or customer churn prediction in this term paper. Logistic regression describes and estimates the relationship between one dependent binary variable and independent variables.[6] In logistic regression, the independent variables can either be continuous or categorical, while the dependent variable is binary. Consider the dataset Customer Churn where the reponse fall into 0 and 1 which are 1 for exited customer and 0 for not exited customers, logistic regression models the probability that the response belongs to one of these particular class. Since this is a probability, we would expect the value to range between 0 and 1, and we would expect to interpret the value as the prediction of how likely it is that the response variable is true.[7]

The probability of an existing customer can be written as:

$$Pr(exited = 1|X)$$

In logistic regression, we use function that gives probability between 0 and 1 for variables X:[8]

$$\mathrm{P}(X) = \frac{e^{\beta_0+\beta_1X_1+\ldots+\beta_pX_p}}{1+e^{\beta_0+\beta_1X_1+\ldots+\beta_pX_p}}$$

After manipulation, we find that :

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1X_1+\ldots+\beta_pX_p}$$

The quantity p(X)/[1−p(X)] is called the odds, and can take on any value between 0 and ∞. Values of the odds close to 0 and ∞ indicate very low and very high probabilities of exited. Odds and probabilities are closely related but represent different concepts. While probability measures the likelihood of an event occurring out of all possible outcomes, odds express the ratio of the event happening to it not happening. Odds ratios are widely used in contexts such as sports betting, epidemiology, and gambling. For example, in sports, one might refer to the odds of a team winning rather than its probability. Suppose Team A has won 6 out of the last 10 basketball games against Team B. The probability of Team A winning a future game is 0.6 (6/10), whereas the odds are calculated as 0.6 / (1 - 0.6) = 1.5. This means Team A is 1.5 times more likely to win than to lose. Understanding this distinction is particularly important in

[6] Navlani, 2024
[7] Nwanganga and Chapple, 2020
[8] James *et al.*, 2013: p.134

logistic regression, where the model estimates the log-odds of an outcome rather than the probability directly.[9]

By taking the logarithm of both sides, we have:

$$\log \frac{p(X)}{1-p(X)} = \beta_0 + \beta_1 X$$

In a logistic regression model, increasing X by one unit changes the log odds by β_1.

```
Call:
stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)

Coefficients:
                     Estimate Std. Error  z value Pr(>|z|)
(Intercept)         -0.142401   0.005533  -25.738  < 2e-16 ***
credit_score        -0.063575   0.005499  -11.560  < 2e-16 ***
age                  0.988912   0.006388  154.814  < 2e-16 ***
tenure              -0.044288   0.005523   -8.019 1.07e-15 ***
balance             -0.093574   0.006963  -13.438  < 2e-16 ***
num_of_products     -0.487987   0.005688  -85.789  < 2e-16 ***
estimated_salary     0.036958   0.005502    6.718 1.85e-11 ***
geography_Germany    0.518028   0.007192   72.031  < 2e-16 ***
geography_Spain      0.017266   0.005769    2.993  0.00276 **
gender_Male         -0.357575   0.005542  -64.517  < 2e-16 ***
has_cr_card_X1      -0.084202   0.005500  -15.310  < 2e-16 ***
is_active_member_X1 -0.638591   0.005731 -111.423  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 273621  on 197770  degrees of freedom
Residual deviance: 199477  on 197759  degrees of freedom
AIC: 199501

Number of Fisher Scoring iterations: 4
```

Figure 1: Logistic Model for the dataset Bank Customer Churn

A positive coefficient estimate increases the chance of churn; a negative estimate reduces it, as shown in Figure 1, older customers are much more likely to churn and more products leads to lower churn probability. However, the coefficient for the dummy variable is negative, indicating that males are less likely to churn than female.

3.2. Regularization

A simple linear model tends to perform poorly in the presence of a large number of predictors. As model complexity increases, it becomes more susceptible to overfitting—capturing random

[9] Nwanganga and Chapple, 2020

noise in the data rather than the underlying signal. To mitigate overfitting, it is crucial to reduce the effective number of parameters being estimated.[10] One common way to make a model simpler and reduce overfitting is to add a penalty to the objective function. This helps limit the size of the model's coefficients and encourages simpler solutions. Removing predictors completely is like setting their coefficients to zero. Instead of doing that, we can gently schrink the coefficients toward zero by penalizing them when they get too large. This keeps all variables in the model but makes their influence smaller (Oleszak, 2019). This idea is the basis of ridge regression, which reduces model complexity by shrinking the coefficients without dropping any predictors..

Ridge Regression is very similar to least squares, but it not only minimize the sum of squared residuals but also penalizes the size of parameter estimates, in order to shrink them towards zero (James *et al.*, 2013) p.237:

$$\sum_{i=1}^{n}(y_i - \beta_o - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2 = RSS + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

The $\lambda\sum_{j=1}^{p}{\beta_j}^2$ is a schinkage penalty and is small when coeficcient β_j is close to zero, so it has the effect of schrinking the β_j towards zero. When $\lambda = 0$, the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as $\lambda \rightarrow \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero.[11]

3.3 Random Forest and Boosting

Another machine learning methode is decision tree. At a basic level, trees are designed to find groups of observations that behave similarly to each. A tree "grows" in a sequence of steps. At each step, a new "branch" sorts the data leftover from the preceding step into bins based on one of the predictor variables (Gu, Kelly and Xiu, 2020). P.2239

Boosting and random forest are tree based methods and approaches that involve producing multiple trees, which are then combined to yield a single consensus prediction. Only one tree is simple and easy interpreted but perform not well. Howerver, combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss in interpretation.[12]

---

[10] Gu, Kelly and Xiu, 2020: p.2234
[11] James *et al.*, 2013: p.237
[12] James *et al.*, 2013: p.327

Random Forest like bagging are resemble method use different training data in parallel to create independent models that use the wisdom of the crowd. As in bagging, a number of decision trees are built on bootstrapped training samples, it simply means sampling rows at random from the training dataset, with replacement.. In bagging, all features are used but in a random forest, only a subset of features is selected at random at each split in a decision tree (James *et al.*, 2013) p.343. Predictions from each of the trees will be generated and then simply be aggregated to get a final prediction. The prediction is the average prediction or majority vote across the bootstrapped trees. Bagging can dramatically reduce the variance of unstable models such as trees, leading to improved performance.

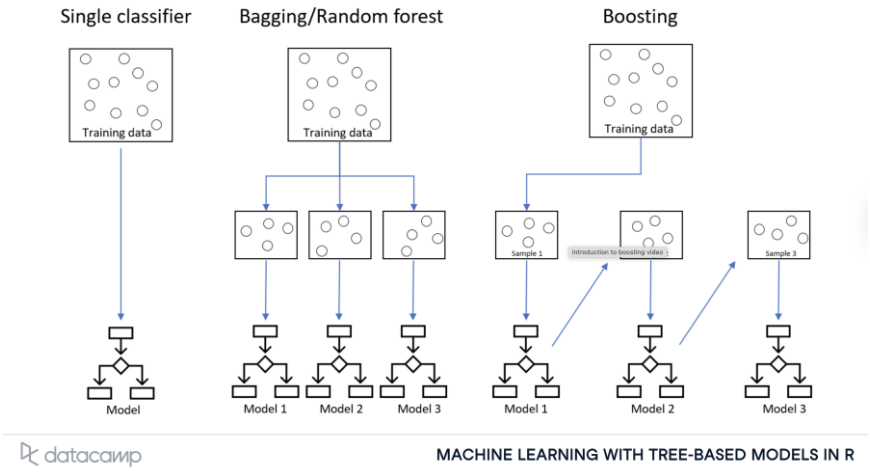

Figure 2: Random Forest and Boosting

Boosting is the improvenment of Random Forest, with Boosting, the subsequent model will try to fix the previous model each model would take advantage of the previous estimator's knowledge. This way, models cannot be trained in parallel, but the result should be better because each model is an improvement of its predecessor as shown in Figure 2.

3.3. Model Evaluation

This study uses 4 machine learning method to predict banking customer churn. The prediction will result in 4 possible cases:

True Positive (TP): The model correctly identify churned customer

True negative (TN): the model correctly identify not churned customer

False Positive (FP): The model predicted customer as churned but they are not churned

False Negative (FN): The model predicted customer as not churned but they are churned

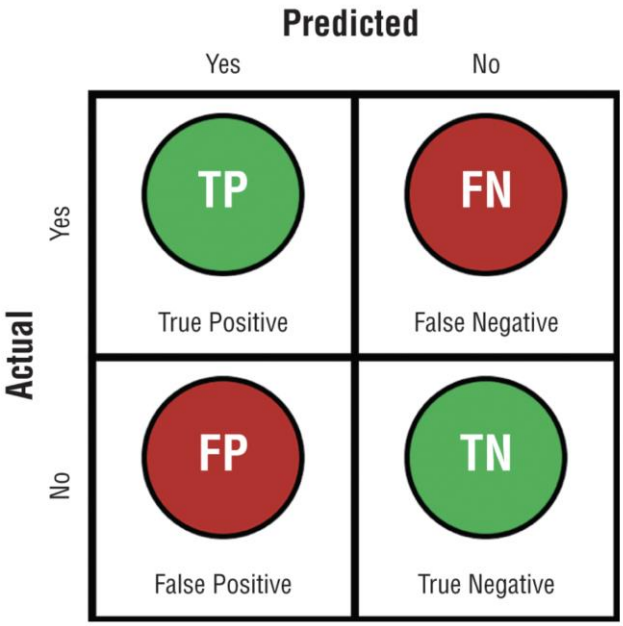

Figure 3: Simple Confusion Matrix[13]

Sensitivity or true positive rate is the proportion of all positive outcomes that were found by your model. For example, of the credit card customers that did churn, how many did our model predict correctly?

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

Specificity or true negative rate measures the proportion of all negative outcomes that were correctly classified. For example, of the credit card customers that did not churn, what proportion did our model predict correctly?

$$specificity = \frac{TN}{TN + FP}$$

The *receiver operating characteristic (ROC) curve* is commonly used to visually represent the relationship between a model's true positive rate (TPR) and false positive rate (FPR) for all possible cutoff values. The ROC curve shows the true positive rate of a classifier on the y-axis against the false positive rate on the x-axis. Note that the false positive rate is the same as 1 minus the true negative rate (or 1 – specificity).

[13] Nwanganga and Chapple, p. 2020

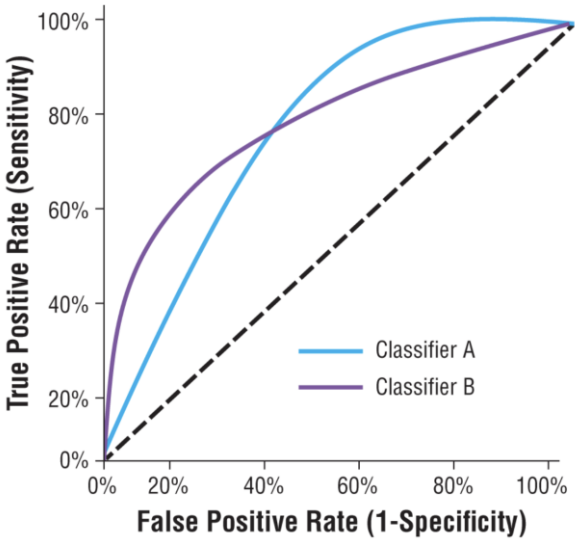

Figure 4: Roc Curve [14]

The ROC curve is sometimes summarized into a single quantity known as the *area under the curve* (AUC). As the name implies, the AUC is a measure of the total surface area under the ROC curve. AUC values range from 0.5 (for a classifier with no predictive value) to 1.0 (for a perfect classifier). The AUC of a classifier can be interpreted as the probability that a classifier ranks a randomly chosen positive instance above a randomly chosen negative instance.

It is important to note that it is possible for two different classifiers to have similar AUC values but have ROC curves that are shaped differently (as illustrated in Figure 3)[15]. So, it is important to not only use the AUC metric when evaluating model performance, but also combine it with an examination of the ROC curve to determine which classifier better meets the business objective.

Moreover, there are still two metrics precision and recall and accuracy. Precision, which is also known as the positive predictive value, is the proportion of positive predictions made by a model that are indeed truly positive. Recall is the proportion of positive examples in a dataset that were correctly predicted by a model.

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

[14] Nwanganga and Chapple, 2020

[15] Nwanganga and Chapple, 2020

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

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

4. Data Preparation and Exploratory Analysis

In this project, we were provided with two datasets: one for training and another for prediction. Our analysis begins with the training dataset, which contains 165,034 observations and 14 variables. Each observation represents a customer, and the goal is to predict whether a customer will churn, indicated by the binary target variable Exited.

4.1 Dataset Structure and Variables

The dataset comprises both numerical and categorical features. The columns ID, CustomerId, and Surname function primarily as identifiers and are not considered predictive. The remaining variables can be categorized as follows:

- **Numerical Variables**: CreditScore, Age, Tenure, Balance, EstimatedSalary, NumOfProducts
- **Binary Variables**: HasCrCard, IsActiveMember
- **Categorical Variables**: Geography, Gender
- **Target Variable**: Exited (1 = churned, 0 = not churned)

All variables were correctly typed in the dataset, with categorical features appearing as character strings and numerical features as continuous or integer values. The structure of the dataset allows for immediate preprocessing and model development.

4.2 Missing Values

A check for missing values revealed a complete dataset. None of the 14 variables contained any NA values, eliminating the need for imputation or deletion of incomplete records.

4.3 Class Imbalance

While the data quality is high in terms of completeness, the target variable is imbalanced. Specifically, out of 165,034 customers, 130,113 (approximately 78.8%) did not churn, while 34,921 (approximately 21.2%) did (Figure 5). This class imbalance is a common challenge in real-world datasets and can significantly impact model performance. As noted by Li and Yan (2025, p. 34), machine learning algorithms tend to be biased toward the majority class, leading to overfitting and reduced generalizability. Addressing this imbalance is therefore essential during model training and evaluation, therefore the balancing step will be included in the preprocessing step.

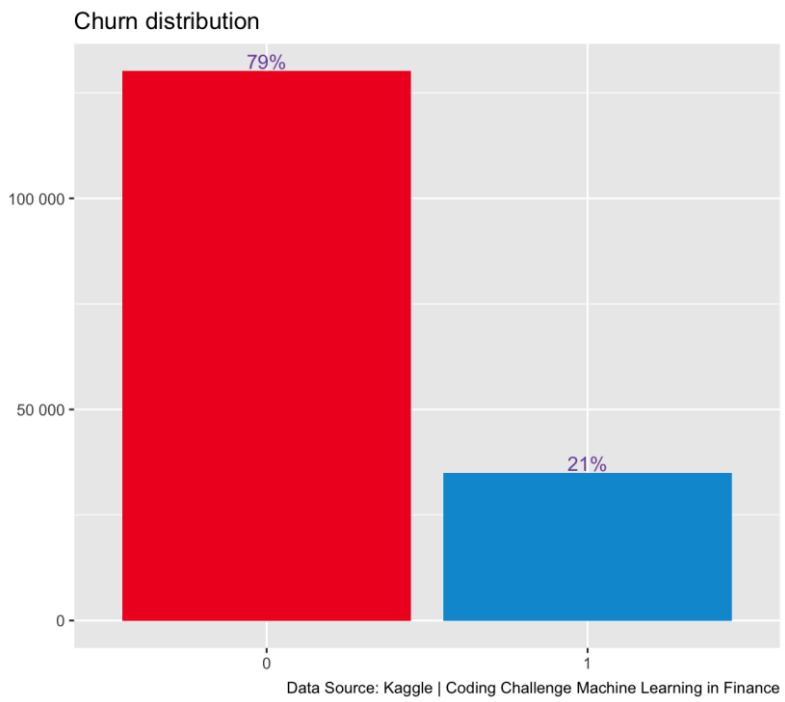

Figure 5: Class Distribution of the Target Variable Exited (0 = Stayed, 1 = Churned)

As visualized in Figure 5, the majority class (Exited = 0) substantially outweighs the minority class (Exited = 1), emphasizing the need for class-balancing techniques during model training.

4.4 Exploratory Data Analysis

To gain an initial understanding of the data, exploratory data analysis will be conducted, focusing on variable distributions and relationships such as numerical distribution, categorical distribution.

An examination of the *Age* variable through histogram analysis revealed that the majority of customers fall within the 30 to 40-year-old range. The age distribution exhibits a slight right skew, indicating a larger proportion of younger customers compared to older ones. Notably, customers aged over 60 are significantly underrepresented within the dataset, suggesting that older individuals are less likely to be part of the current customer base (Figure 6).

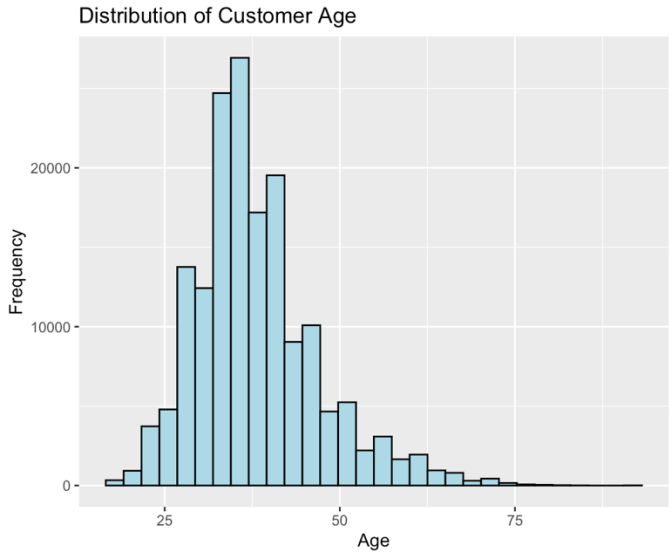

Figure 6: Customer Age

Bar plots showed that most customers were from France, followed by Spain and Germany. The gender distribution was approximately balanced across the dataset (Figure 7).

Figure 7: Customer by Geography and Gender

Basic descriptive statistics were calculated to understand central tendencies and variability among key features. These included the mean, median, and standard deviation for variables such as Age and CreditScore (Figure 8).

```
> summary(select(data, Age, Balance, CreditScore, EstimatedSalary))
      Age              Balance         CreditScore    EstimatedSalary
 Min.   :18.00   Min.   :     0   Min.   :350.0   Min.   :    11.58
 1st Qu.:32.00   1st Qu.:     0   1st Qu.:597.0   1st Qu.: 74637.57
 Median :37.00   Median :     0   Median :659.0   Median :117948.00
 Mean   :38.13   Mean   : 55478   Mean   :656.5   Mean   :112574.82
 3rd Qu.:42.00   3rd Qu.:119940   3rd Qu.:710.0   3rd Qu.:155152.47
 Max.   :92.00   Max.   :250898   Max.   :850.0   Max.   :199992.48
```

Figure 8: Descriptive Statistic

These exploratory steps helped inform our later choices regarding preprocessing, feature importance, and model selection strategies. For example as shown in Figure 9 the balance of churned and stayed customer are not balanced.

```
> truth %>%
+   keep(is.factor) %>%
+   summary()
   geography        gender      has_cr_card is_active_member exited
 France :94215   Female:71884   0: 40606    0:82885          0:130113
 Germany:34606   Male  :93150   1:124428    1:82149          1: 34921
 Spain  :36213
```

Figure 9: Distribution of categorical variables

The majority of variables in the dataset demonstrate either very weak or negligible linear relationships with one another (Figure 10). This suggests that the features are largely independent, which is favorable for building predictive models. The only moderately notable correlation identified was between **Number of Products** and **Balance**, with a Pearson coefficient of $r = -0.36$. This negative relationship indicates that customers holding fewer products tend to have higher account balances. The low levels of correlation across predictors point to minimal multicollinearity, an important consideration for regression modeling. Reduced multicollinearity enhances model stability and interpretability by minimizing redundancy among explanatory variables.

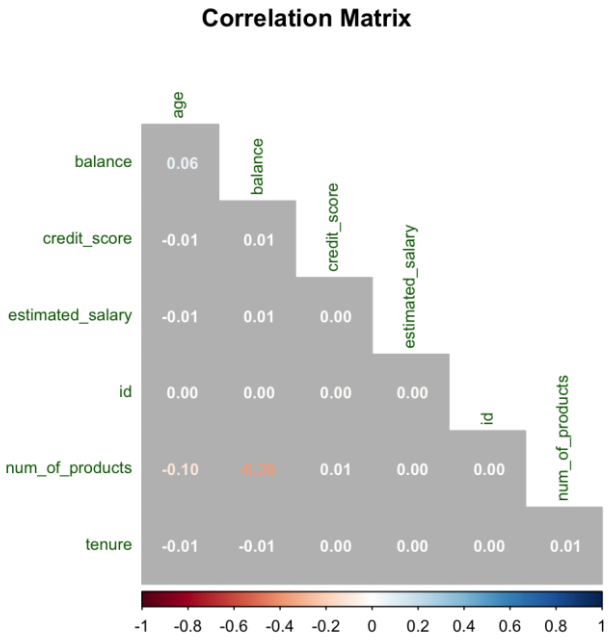

Figure 10: Correlation Matrix

5. Model Training and Hyperparamenter Tuning

To ensure robust model performance and minimize the risk of overfitting, we implemented a machine learning pipeline using the tidymodels framework in R. This modular and reproducible pipeline begins with a comprehensive data preprocessing recipe that includes:

- step_novel(): Handles novel factor levels that may appear in future or test data but were not observed during training.
- step_dummy(): Encodes categorical variables into binary (dummy) variables to allow compatibility with tree-based models.
- step_zv(): Removes predictors with zero variance, which do not contribute to the model's predictive power.
- step_normalize(): Normalizes numeric features to ensure comparability and improve optimization during model training.

Given the imbalance in the target variable Exited (i.e., far more non-churned than churned customers), we applied Synthetic Over Sampling Technique (SMOTE) via step_smote(). SMOTE synthetically generates new instances of the minority class by interpolating between existing ones. This balances the dataset and helps prevent models from being biased toward the majority class.

To evaluate model performance, 5-fold cross-validation will be used. In stratified cross-validation, the dataset is split into five folds (subsets), ensuring that each fold retains the same proportion of classes (churned vs. non-churned) as the original dataset. The model is trained on four folds and validated on the remaining one, and this process is repeated five times, with each fold used exactly once as the validation set. This approach yields reliable and unbiased performance estimates by mitigating variance due to random sampling and ensuring that all data is used for both training and validation.

For many machine learning problems, simply running a model out-of-the-box and getting a prediction is not enough. One way to find the best model with the most accurate prediction is with hyperparameter tuning, which means optimizing the settings for that specific model. This tuning was carried out using grid search in conjunction with cross-validation.

Each model has key hyperparameters. For the XGBoost classifier, we tuned the following key hyperparameters (Figure 11):

- trees: The number of boosting iterations (i.e., the number of trees in the ensemble). Increasing this can improve performance but may lead to overfitting if too large.

- learn_rate: Controls the contribution of each new tree to the ensemble. A smaller learning rate allows for more refined and stable learning but requires more trees to converge.
- mtry: The number of features randomly selected at each tree split (similar to colsample_bytree in XGBoost). This parameter helps to reduce correlation among trees and improve generalization.

```
> show_best(xgb_tune, metric = "roc_auc")
# A tibble: 5 × 10
   mtry trees tree_depth learn_rate .metric .estimator  mean     n  std_err .config
  <int> <int>      <int>      <dbl> <chr>   <chr>      <dbl> <int>    <dbl> <chr>
1    11  1000          5    0.01    roc_auc binary     0.888     5 0.000457 Preprocessor1_Model3
2     5   440         10    0.00316 roc_auc binary     0.887     5 0.000413 Preprocessor1_Model6
3     7   300          3    0.0316  roc_auc binary     0.885     5 0.000504 Preprocessor1_Model1
4     2   860          7    0.1     roc_auc binary     0.882     5 0.000509 Preprocessor1_Model4
5     3   720          4    0.001   roc_auc binary     0.874     5 0.000786 Preprocessor1_Model2
```

Figure 11: Parameter Tuning for XGBoost

The optimal combination of these hyperparameters was selected based on the highest **ROC AUC** across the validation folds, ensuring that the final model generalizes well to unseen data. As seen in Figure 11, The combination of mtry =11, trees=1000, tree_depth=5 and learn_rate =0.01 generates the best performance with AUC=0.888

6. Result and Discussion

In this section, the performance of the 5 machine learning models will be presented.

Logistic Regression

The logistic regression model achieved an accuracy of 75.3% and an AUC of 0.183 on the test dataset. While the accuracy may appear high at first glance, it is misleading due to class imbalance -the majority of customers in the dataset did not churn. As a result, the model achieves high accuracy by predominantly predicting the majority class ("no churn") without learning meaningful patterns. In this context, the AUC (Area Under the ROC Curve) is a more informative metric. An AUC of 0.5 represents random guessing, whereas higher values reflect improved class separation. An AUC of only 0.183 suggests the model performs worse than random, indicating a failure to identify customers at risk of churning. This poor performance is visually confirmed in Figure 12 (ROC Curve), where the ROC curve lies mostly below the diagonal, reinforcing the model's weak ability to separate churners from non-churners.

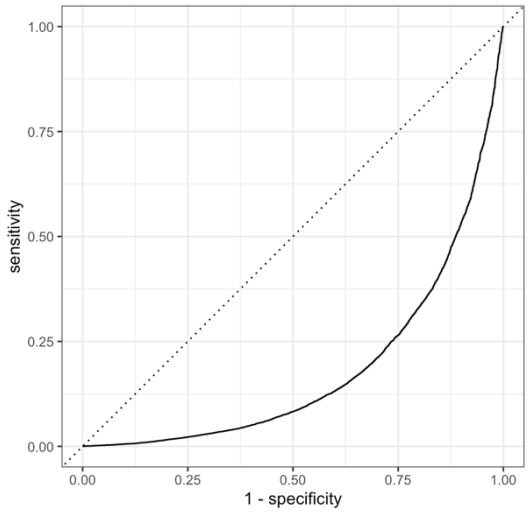

Figure 12: ROC Curve for Logistic Regression

Despite limited predictive power, the logistic model provides interpretable insights. For example, age was positively associated with churn (coefficient +0.836), suggesting older customers are more likely to exit. German customers also had a higher likelihood of churn (+0.465). In contrast, being an active member (−0.644), holding more products (−0.496), and being male (−0.331) were all associated with a lower churn probability. These findings align with exploratory data analysis and help identify potential churn drivers, even if the model itself is insufficient for prediction.

Although logistic regression offered interpretability, its performance—particularly the low AUC—was inadequate. For improved predictions, more advanced models such as regularized regressions for example Ridge or nonlinear approaches like Random Forest or Gradient Boosting are recommended.

Ridge Regression

Ridge Regression was employed to manage multicollinearity and improve generalization. The model was tuned across a range of penalty values (λ), with results indicating high ROC AUC stability for small penalties. For instance, λ values below 0.05 yielded AUC scores around 0.818, demonstrating the model's robustness to tuning.

Figure X illustrates the ROC AUC across penalty values, confirming that light regularization maintains predictive strength.

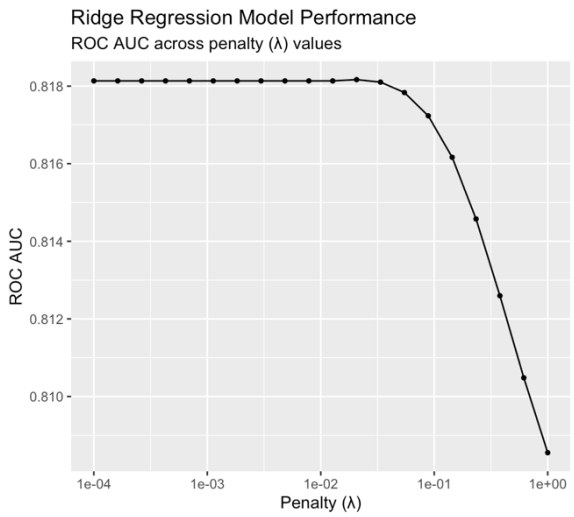

Figure 13: Ridge Regression Performance

The optimal penalty value was $\lambda = 0.0207$. However, models with lower λ values (e.g., 0.0001) showed nearly identical performance, highlighting the model's flexibility.

To mitigate class imbalance, we applied SMOTE (Synthetic Minority Oversampling Technique), which improved balance between classes. The Ridge + SMOTE model achieved an AUC of 0.818, with an overall accuracy of 75.2%, recall of 75.7%, and specificity of 73.1%. By comparison, the Ridge model without SMOTE yielded very high recall (96.6%) but low specificity (31.8%), resulting in too many false positives. SMOTE provided a better trade-off.

Ridge Regression offers strong baseline performance and interpretability, especially when paired with SMOTE. However, if the goal is to capture nonlinear effects or further improve classification, more flexible models like Random Forest or XGBoost are recommended.

Decision Tree & Random Forest

The Decision Tree model delivered high recall (91.6%) and precision (90.1%), with a ROC AUC of 0.872 and an overall accuracy of 85.4%. These results suggest reliable performance in identifying churners while minimizing false alarms.

Random Forest outperformed the Decision Tree across all metrics. At the default threshold (0.5), the model achieved:

- Accuracy: 86.3%
- Recall: 93.9%
- Precision: 89.3%
- AUC: 0.886

Lowering the threshold to 0.4 resulted in:

- Accuracy: 85.4%
- Recall: 90.6%
- Precision: 90.8%

This adjustment offered a more balanced classification with similar AUC performance. As shown in Figure 14, the ROC curve for the Random Forest lies well above the diagonal, indicating strong class separation.

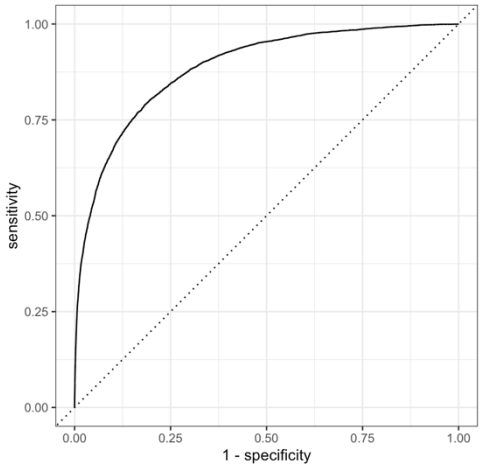

Figure 14: ROC Curve for Random Forest

Model Comparison

Compared to the Decision Tree, Random Forest consistently delivered superior results with:

- AUC: 0.886 (vs. 0.872)
- Recall: 93.9% (vs. 91.6%)
- Slightly higher accuracy and better robustness through ensemble learning

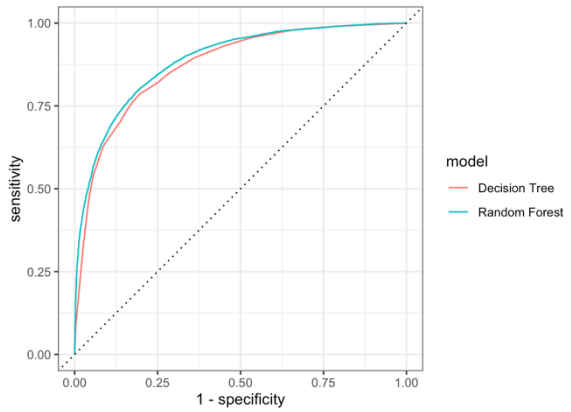

Figure 15: Random Forest outperforms Decision Tree

Random Forest provides excellent predictive performance and should be preferred over individual Decision Trees. However, the model still generates some false positives and false negatives, especially at the 0.5 threshold. For further improvement, boosting methods such as XGBoost are suggested.

XGBoost

XGBoost (Extreme Gradient Boosting) was implemented as the most advanced model in this analysis. It was selected as the final model for predicting customer exits due to its superior performance in accuracy, recall, precision, and overall discriminatory power compared to the other models.

On the test dataset, XGBoost achieved Accuracy: 86.1%, Recall: 93.2%, Precision: 89.5%, F1 Score: 91.3%, ROC AUC: 0.886. These results indicate that the model correctly identified most customers who exited (high recall), while maintaining a low false alarm rate (high precision). The high AUC value of 0.886 confirms the model's excellent ability to distinguish between churners and non-churners, ranking customers effectively by their likelihood to exit.

Figure 16 presents the ROC curve for the XGBoost model. The curve lies well above the diagonal, indicating strong discriminative performance across different classification thresholds.

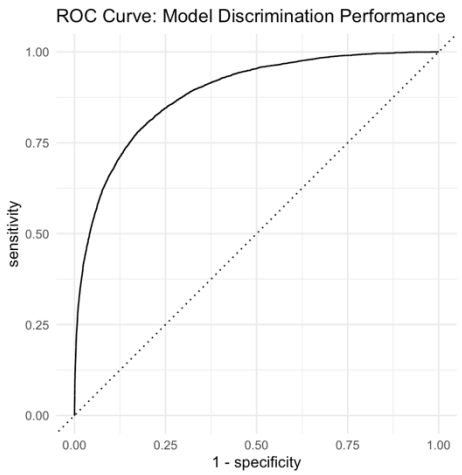

Figure 16: ROC Curve for XGBoost with AUC=0.886

Discussion

Table 1 illustrates all the important metrics of all models. Random Forest and XGBoost have the excellent class separation (AUC = 0.886).

Model	*Accuracy*	*Recall*	*Precision*	*AUC*
Logistic Regression	75.3%	37.7%	69.6%	0.183
Ridge Regression	75.2%	75.7%	91.3%	0.818
Decision Tree	85.4%	91.6%	90.1%	0.872
Random Forest	86.3%	93.9%	89.3%	0.886
XGBoost	86.1%	93.2%	89.5%	0.886

Table 1 Performance metrics of the models

Moreover, depends on which model train, the feature important of the models also changes as see in Figure 17. While by XgBoost, the most influential predictors of customer churn are: Age, Number of Products and Tenure, and for the Random Forest are Number of Product, Age and Active Member.

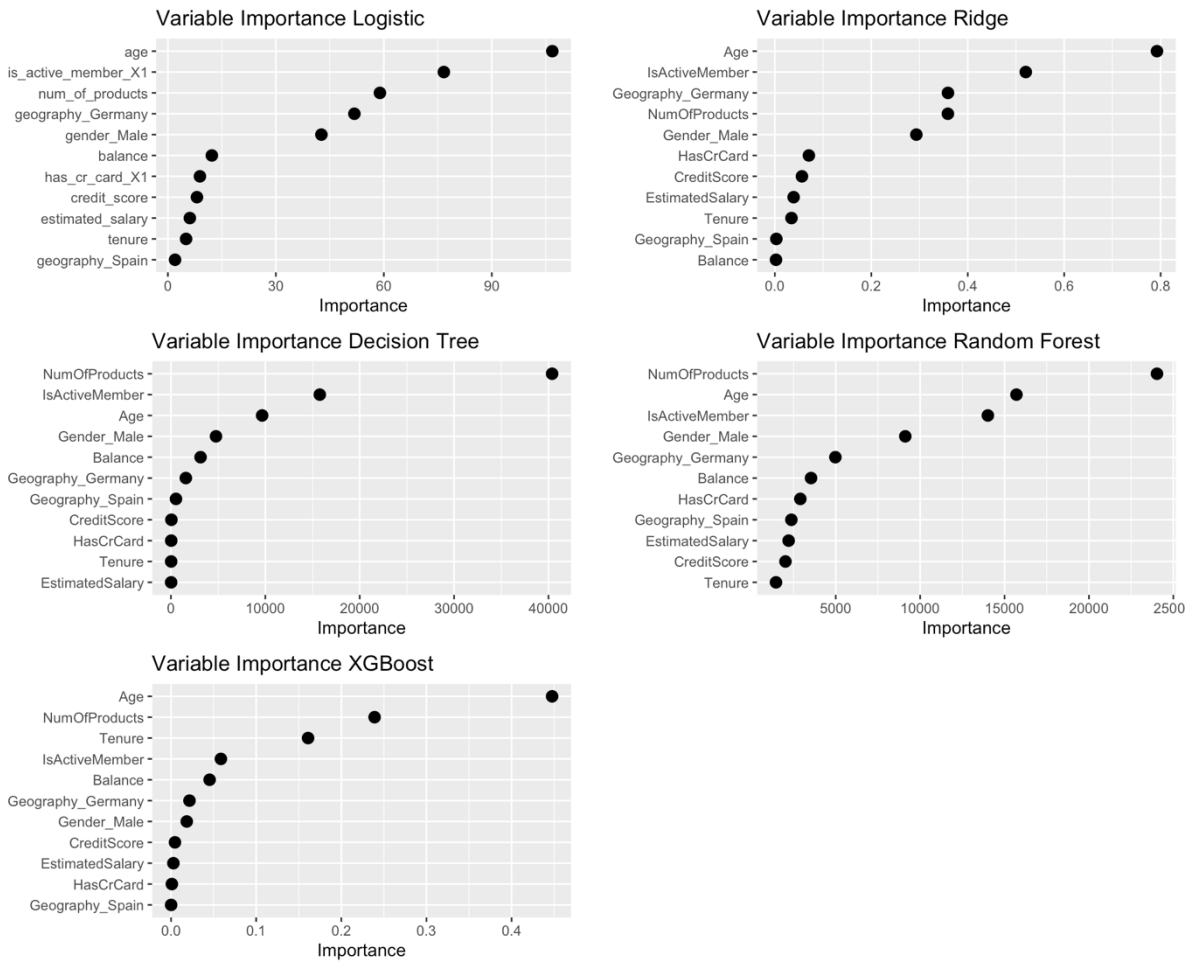

Figure 17: Variable importance of the models

Considering both statistical performance and business alignment, XGBoost is recommended as the final model for customer churn prediction. Its combination of **high recall (93.2%)**, **strong precision (89.5%)**, and **excellent class separation (AUC = 0.886)** ensures that the model can reliably identify at-risk customers while minimizing unnecessary retention costs. Unlike Random Forests, which can sometimes overfit without proper tuning, XGBoost includes built-in regularization that improved generalization to unseen data. Despite achieving an identical AUC of 0.886, the XGBoost model offered more controlled trade-offs through threshold tuning and subsampling strategies

To reduce false positives and better align the model with business needs, the default classification threshold of 0.5 was increased to **0.6**. This adjustment slightly reduced recall but improved precision, ensuring that fewer loyal customers were incorrectly classified as churners. This change supports more efficient resource allocation in customer retention strategies.

Furthermore, the model's output was filtered to prioritize high-value customers, specifically those with balances above 50,000. By concentrating on this segment, the business can focus retention efforts on customers whose departure would have the greatest financial impact.

7. Conclusion

In this term paper, we explored and implemented five supervised machine learning algorithms logistic regression, ridge regression, decision tree, random forest, and XGBoost to address the problem of customer churn prediction in a banking dataset.

In conclusion, this study highlights the importance of model selection and evaluation in machine learning workflows. While linear models offer interpretability and simplicity, more advanced tree-based models like Random Forest and XGBoost deliver superior predictive performance, especially in high-dimensional and nonlinear scenarios. Due to the limited number of minority class samples in the dataset, sampling techniques were employed to generate synthetic instances. This strategy helps mitigate class imbalance, allowing the predictive model to better distinguish between classes and improve its ability to correctly identify minority class observations. For business applications such as churn prediction, where identifying at-risk customers is crucial, these ensemble methods provide a highly effective solution.

References

Gu, S., Kelly, B. and Xiu, D. (2020) 'Empirical Asset Pricing via Machine Learning', *The Review of financial studies*, 33(5), pp. 2223–2273.

James, G. *et al.* (2013) *An Introduction to Statistical Learning: with Applications in R*. 1st edn. New York: Springer Nature (Springer texts in statistics).

Li, Y. *et al.* (2023) 'Bank Customer Churn Prediction Based on Correlation Analysis and Multiple Linear Regression', in *Proceedings of the 2nd International Conference on Business and Policy Studies*. Singapore: Springer (Applied Economics and Policy Studies), pp. 1065–1072.

Li, Y. and Yan, K. (2025) 'Prediction of bank credit customers churn based on machine learning and interpretability analysis', *Data science in finance and economics*, 5(1), pp. 19–34.

Navlani, A. (2024) 'Understanding Logistic Regression in Python'. Available at: https://www.datacamp.com/tutorial/understanding-logistic-regression-python.

Nguyen, Q.H. *et al.* (2024) 'The Proposed Model Machine Learning of Predicting Bank Churn Customer', in *Innovative Computing and Communications*. Singapore: Springer (Lecture Notes in Networks and Systems), pp. 367–374.

Nwanganga, F.C. and Chapple, M. (2020) *Practical machine learning in R*. Indianapolis: John Wiley and Sons.

Oleszak, M. (2019) 'Regularization in R Tutorial: Ridge, Lasso and Elastic Net', 12 November. Available at: https://www.datacamp.com/tutorial/tutorial-ridge-lasso-elastic-net.