BANK CUSTOMER CHURN CLASSIFICATION

PREPARED BY

Steve Githinji

1. Business Overview

Introduction

Customers discontinuing their relationship with a bank can have significant financial implications for the institution. Identifying customers who are likely to churn in advance can help banks implement targeted retention strategies, thereby reducing customer attrition and maximizing profitability. Machine learning techniques can play a crucial role in predicting customer churn, enabling banks to take proactive measures to retain valuable customers.

The key stakeholders in this project are the Bank Executives, Marketing and Retention Team and Customer Service Representatives.

Objectives

The objective of this machine learning project is to develop a classification model that accurately predicts customer churn in the bank. By leveraging historical customer data, including age, geographical location, account information, products and credit cards subscribed, and estimated salary, the model aims to identify patterns and indicators that suggest a higher likelihood of churn. The ultimate goal is to enable the stakeholders to prioritize retention efforts and develop tailored strategies to retain customers at risk of churn.

The objective of the study is::

> *"To accurately predict which customers are likely to churn based on their age, location and account information?"*

2. Data Understanding

Data Description

The dataset contains 10,000 entries and 13 columns of customer information. The *CustomerId* and *Surname* columns are data artifacts and may not be useful to our models. *Geography* and Gender columns are object datatypes while the rest are float64 and int64 datatypes. The *Exited* column is the target feature while the rest are the predictors.

The *Exited* column has about 80% of the entries as 0 (customers who did not churn) and 20% of entries as 1 (customers who churned). There is some class imbalance here which should be considered during modelling

From the *Geography* column, 50% of the records are for French customers, while the rest are for German and Spanish customers. The dataset contains customers whose salaries range from €11.58 to €199,992.48. Their account balances range from €0.0 to €250,898.09.

Data Preparation

The dataset did not contain any missing values. The target for this analysis is the churn status of a customer, described in the column *Exited*. Therefore the data was separated into X (predictors) and y (target feature) accordingly.. Next, the data was separated into a train set (75% of the full dataset) and a test set (25% of the full dataset) prior to performing any preprocessing steps. The split dataset was named X_train, X_test, y_train and y_test accordingly. This was done before data preparation to avoid data leakage. The treatment of the test data is therefore as similar as possible to how genuinely unknown data should be treated.

The dataset contained categorical features which would crash if fed into our scikit-learn Machine Learning, Hence we converted categorical variable into dummy/indicator variables using pandas' get_dummies.

The MinMaxScaler transformer was to scale the training set. Scaling is useful since there are features with different ranges and we want to bring them to a common scale so as to improve the performance of the machine learning algorithms that rely on distance calculations or feature comparisons. The same preprocessing steps were applied to the testing set separately.

3. Modelling

Evaluation Metrics

Precision was chosen as one of the evaluation metrics for the performance of models. Precision is the ratio between the True Positives and all the Predicted Positives. It allows us to answer the following question:

> *"Out of all the times the model said a customer churned, how many times did the customer in question actually churn?"*

Interpreting precision on its own can be misleading. This is why accuracy was also included as an evaluation metric. Accuracy is intuitive because it allows us to measure the total number of predictions a model gets right, including both True Positives and True Negatives. It allows us to answer the following question:

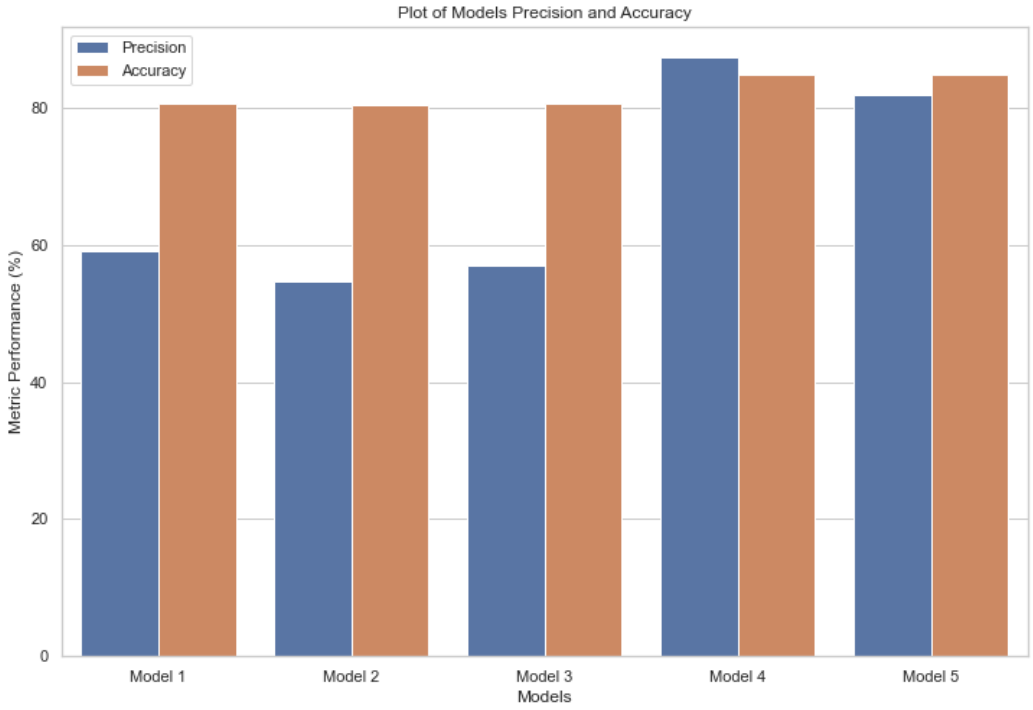
> *"Out of all the predictions our model made, what percentage were correct?"*

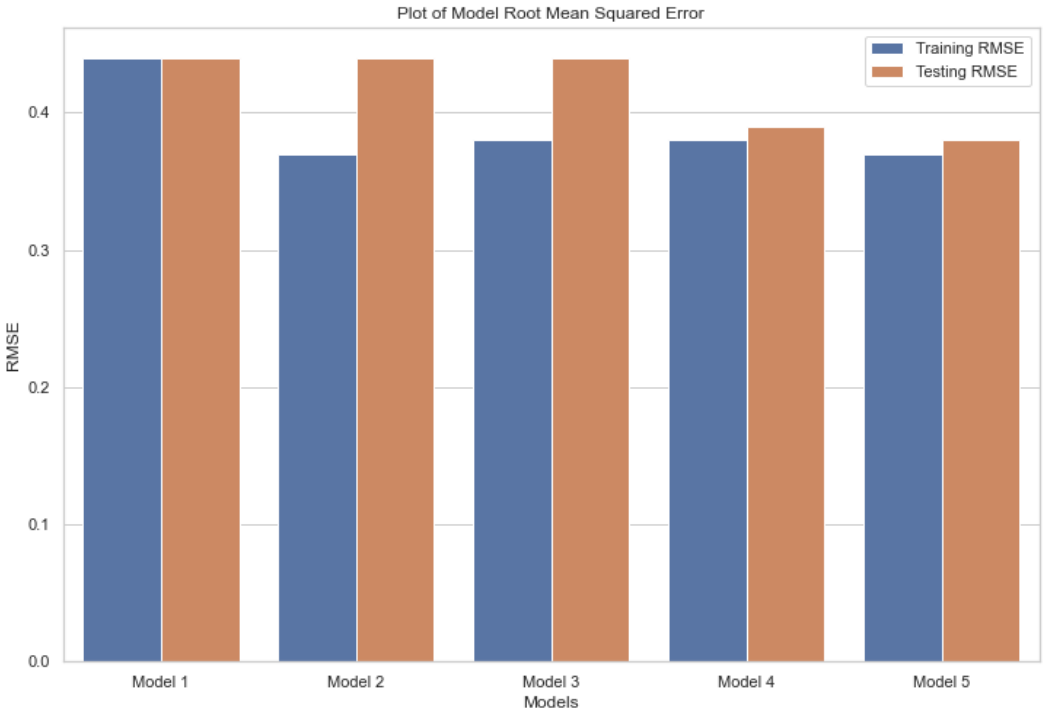
The error based metric used was Root Mean Squared Error (RMSE). It was used to measure the average magnitude of the differences between the predicted values and the actual (observed) values. It was also used to measure the level of overfitting in different models by looking at the magnitude of the difference between training set RMSE and testing set RMSE.

Modelling

This project took an iterative approach to modeling by building multiple models including a baseline logistic regression model, K-nearest neighhbors models, and random forests models. GridSearchCV was used to perform hyperparameter tuning and model selection by exhaustively searching through a specified hyperparameter grid to find the best combination of hyperparameters. Scikit-learn's SelectKBest was used to select the best performing features to use in modelling. In total, 5 models were built.

Below are plots of precion and accuracy of testing data for all the models and RMSE for training and testing sets for all the models:

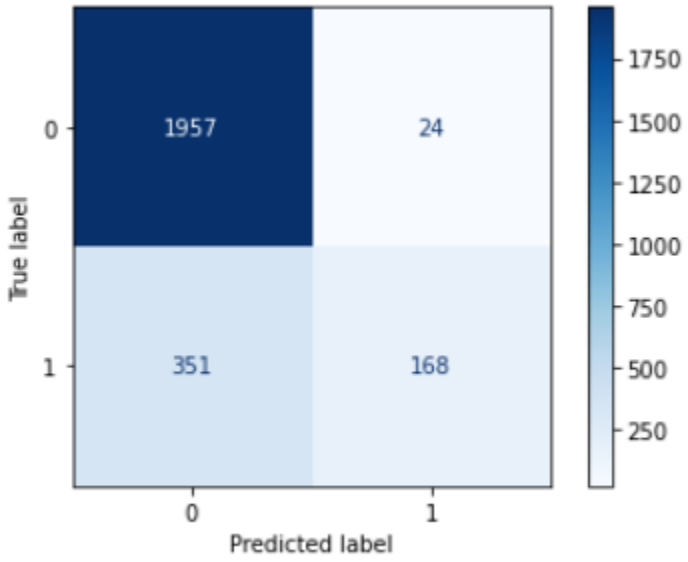

Plot of Models Precision and Accuracy
Precision
Accuracy
Metric Performance (%)
0
20
40
60
80
Model 1
Model 2
Model 3
Model 4
Model 5
Models


Plot of Model Root Mean Squared Error
Training RMSE
Testing RMSE
RMSE
0.0
0.1
0.2
0.3
0.4
Model 1
Model 2
Model 3
Model 4
Model 5
Models

Looking at the precision and accuarcy plot, model 4 had the highest accuracy and precision. In the RMSE plot, model 4 had the least amount of overfitting since it had fairly similar training and test RMSE.

Model 4 was therefore selected as the final model. It is an ensemble algorithm that used a Random Forest Classifier. It had the highest combined accuracy and precision and the least amount of overfitting. Model 4 had an accuracy of about 85% meaning that out of all the predictions this model made, 85% were correct. It had a precision of 87.5% meaning that out of all the times the model said a customer churned, the customer in question actually churned 87.5% of the time. It had a training set RMSE of 0.38 and testing set RMSE of 0;39.

Below is the final model's Confusion Matrix:



4. Conclusion

In conclusion, the final model demonstrated the power and effectiveness of analyzing the Bank Customer Churn dataset and predicting customer churn. Through careful hyperparameter tuning and model selection, a robust classification model capable of accurately predicting customer churn was successfully developed.

An analysis and evaluation of various machine models i.e. logistic regression, K-nearest neighhbors, decision trees and random forests, provided valuable insights into their performance. Ultimately, **model 4**, a random forest classifier, was chosen as the most suitable choice,

delivering the best performance metrics, including high accuracy and precision. By splitting the dataset into training and testing sets, the model's performance was validated on unseen data, minimizing the risk of overfitting. By using a Random Forest algorithm, various individual decision trees were built on different samples and their majority vote taken as the prediction. The model employed scikit-learn's GridSearchCV to perform hyperparameter tuning and model selection. It automated the process of exhaustively searching through a specified hyperparameter grid to find the best combination of hyperparameters for the given estimator. Cross-validation was done to ensure the generalizability of the model.

The deployment of this classification model has the potential to offer tangible benefits to the Bank Executives, Marketing and Retention Team and Customer Service Representives . By automating the process of categorizing new instances based on historical bank customer data, this model can assist the executives make decisions on proactive measures to take to mitigate customer churn such as targeted advertising and offers.

Nevertheless, it is essential to acknowledge the limitations of this study. These include class imbalance, overfitting and limited training data. Additionally, as the nature of the problem and data evolve, the model should be regularly re-evaluated and updated with more current data to maintain its efficacy.

In conclusion, our Random Forest classification classifier has demonstrated its capability to accurately classify customer churn instances based on the provided features. Its performance, generalizability, and potential for real-world application by the bank executives make it a valuable tool for decision-making processes.

5. Next Steps

The next step is to deploy the model to a production environment where it can be used to make predictions on new data. Here we will continuously monitor the performance of the churn classification model. We will keep track of evaluation metrics and periodically reevaluate the model's performance using new data. If the model's performance starts to degrade, we will consider retraining or updating the model with more recent data.