MSDS 422 Final Project - Customer Churning Prediction

Zachary Cmiel and Jerry Zhao

MSDS 422: Practical Machine Learning

May 30, 2024

Executive Summary

Customer data can be used to identify whether someone has exited, whether that means something like unsubscribing or moving to a different service. By using the Bank Customer Churn dataset we tuned four classification models to obtain the most accurate results when predicting customer “Exited” categories. Initial EDA revealed that 11 features were quantitative and 3 were qualitative. The data was clean with no null values in any of the columns. Plotting the correlations of the quantitative data revealed that balance and age had the strongest correlation to whether or not a customer exited whereas several products and whether the customer was active showed the least correlation.

We tested a total of six models on our classification problem to try and help banks identify customer churn more accurately. These models included logistic regression, k-nearest neighbors, random forest, unbalanced random forest, and a multi-layer perceptron neural network. After appropriate hyperparameter tuning using Grid Search for the first five models and a completely crossed 2x2 experiment for the MLP neural network, our best models performed at an 87% test accuracy level. We suggest going forward to use these models as a great starting point to reach out to customers that have a high risk of leaving to give them new offers on different products or better offers on the products they already own. This can help retain them and thus save the bank money and effort. While this dataset is a bit limited in terms of numbers of features, this is a promising start and in the future other datasets that might better capture a user’s socioeconomic background along with the current state of the economy can provide more accurate predictions. Economic data like interest rates, rising and falling GDP, or unemployment can all lead to bank customer churn and thus be valuable for models.

Dataset: https://www.kaggle.com/competitions/playground-series-s4e1/data

Github: https://github.com/RamenNoodleJerry/My-Projects/tree/main/MSDS%20422%20Project/Final%20Project

Problem Statement and Research Objectives

For businesses, analyzing Customer Lifetime Value (CLV) and ways to retain customers is essential to creating a profitable and sustainable business. Continually acquiring new customers is critical, but successful businesses must know when to instead maximize the value of existing ones and focus on customer retention. In general, it is 5 times more expensive to acquire a new customer than it is to retain an existing one[1]. By exploring customer data, companies and organizations can benefit from a model that can categorize customers as likely to exit in some way i.e. unsubscribe, quit, or move to a competitor. They can use these predictions as a guideline for which customers to focus retention efforts. Furthermore, classification models can help identify which features of a customer correlate to exiting thus allowing them to optimize their retention plan.

By using the Bank Customer Churn dataset we hope to be able to create a model that can categorize and predict whether or not a customer has exited based on various features such as credit score, balance, and salary. Significant exploratory data analysis along with hyperparameter tuning will allow us to identify important features of customers that we can recommend to banks to focus on to retain current customers and maximize CLV. After using the training and test sets to build and optimize our models we will be able to report on what these features are and deliver a tool that can be used to analyze a customer base and provide insights to banks on ways to reduce customer churn, a key action to increase profitability.

[1] Tidey, Will. “Acquisition vs Retention: The Importance of Customer Lifetime Value.”

Exploratory Data Analysis

We will split our analysis into three parts, univariate, bivariate, and multivariate. We started by checking the missing value of the data. Fortunately, the data is clean enough with no outliers across all variables. For univariate analysis, we analyze each variable separately and check their value distribution. This allows us to check if there is missing values and outliers of the data. We checked from boxplots (Figure 1), there are some outliers across “number of product“, “credit score“, “age“, and “has credit card flag” variables.

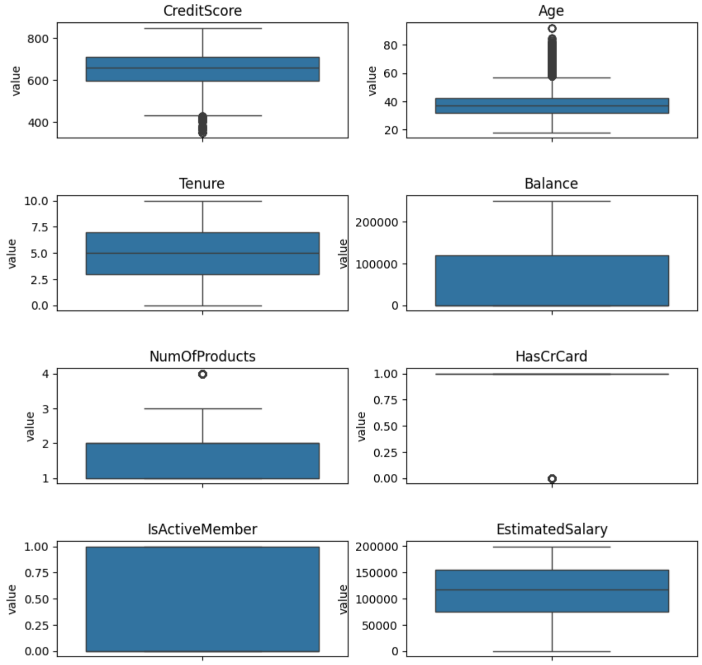


Figure 1

These outliers are natural variations and should not be dropped. We will keep those outliers. We also find some irrelevant variables like customer ID and surname, which should be dropped and not used in model training. Next we look at histogram (Figure 2) of the dependent variable “exited flag” and identify around 20% are churned customers. We can see there is indeed an imbalanced distribution of positive and negative. We may need to deal with the imbalance later otherwise models like Logistic Regression and Support Vector Machine (SVM) would produce biased results.

We then check the categorical variable using histogram (Figure 3). We find France takes the majority of the customer geographical status, more male than female in the dataset, majority of customer has credit card, and similar amount between active inactive customers.

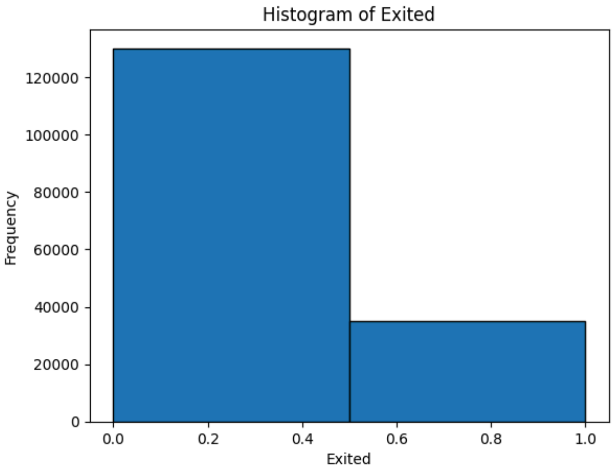


Figure 2

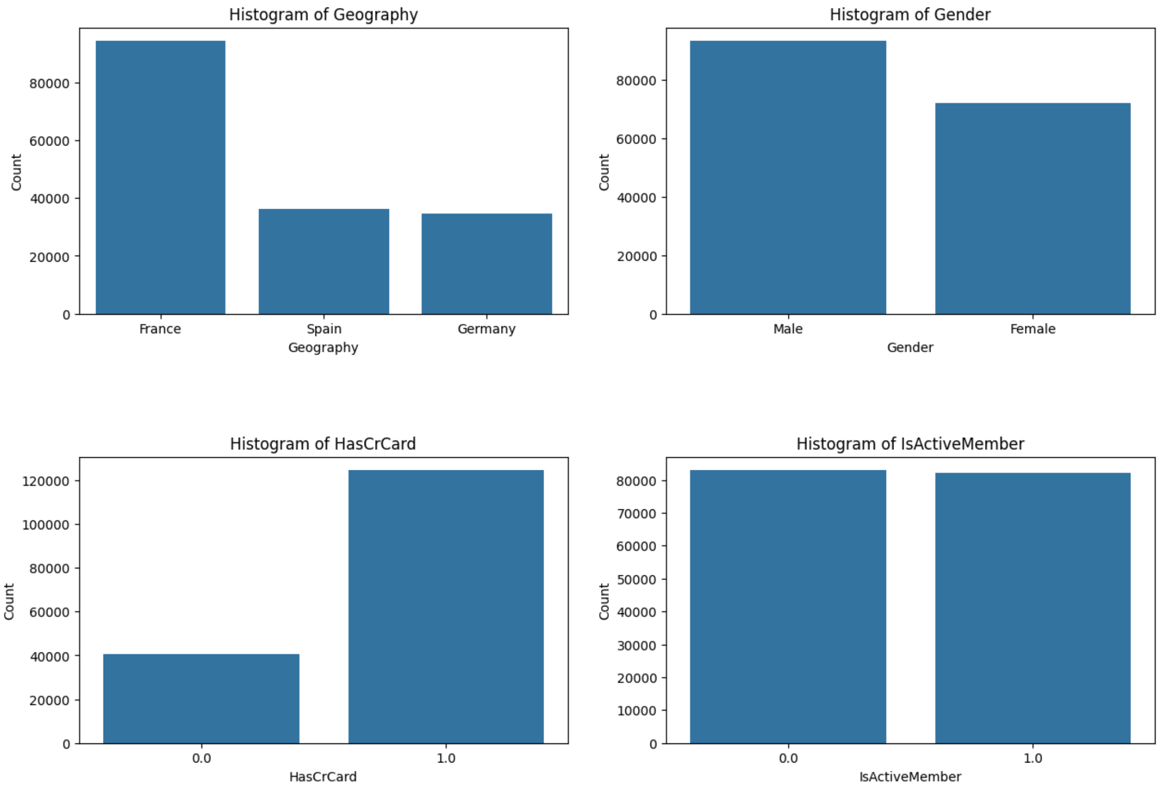


Figure 3

For bivariate analysis, we plot dependent variables against each independent variable and see their relationship (Figure 4). We first plot two categorical variables, geography and gender against the churn flag. The result shows Germany seems to have more churning than other countries and females are more likely to churn than male customers.We then plot numeric variables against churn flag using bar plot based on average. Some insights we find are: active member are less likely to churn, older people are more likely to churn, those who churn has higher balance, those who churn has fewer number of products,

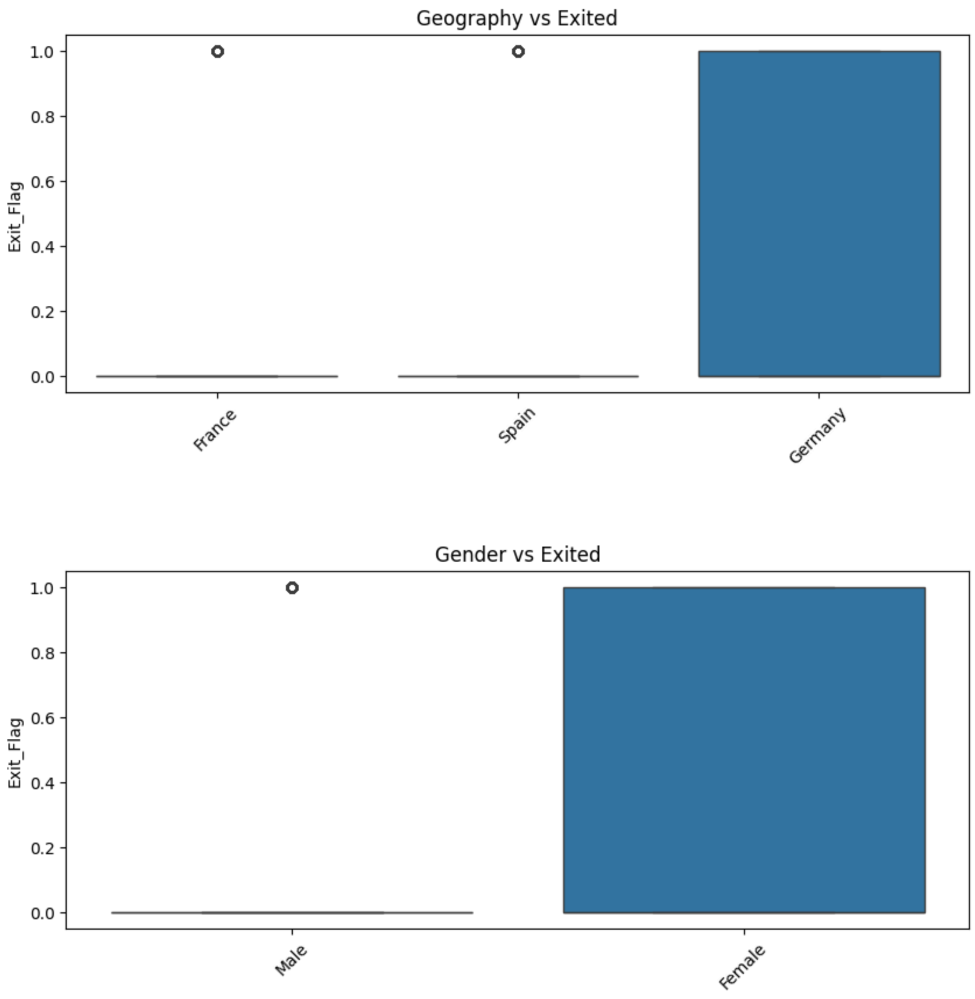

Geography vs Exited
Exit_Flag
1.0
0.8
0.6
0.4
0.2
0.0
France
Spain
Germany
Gender vs Exited
Exit_Flag
1.0
0.8
0.6
0.4
0.2
0.0
Male
Female

Figure 4

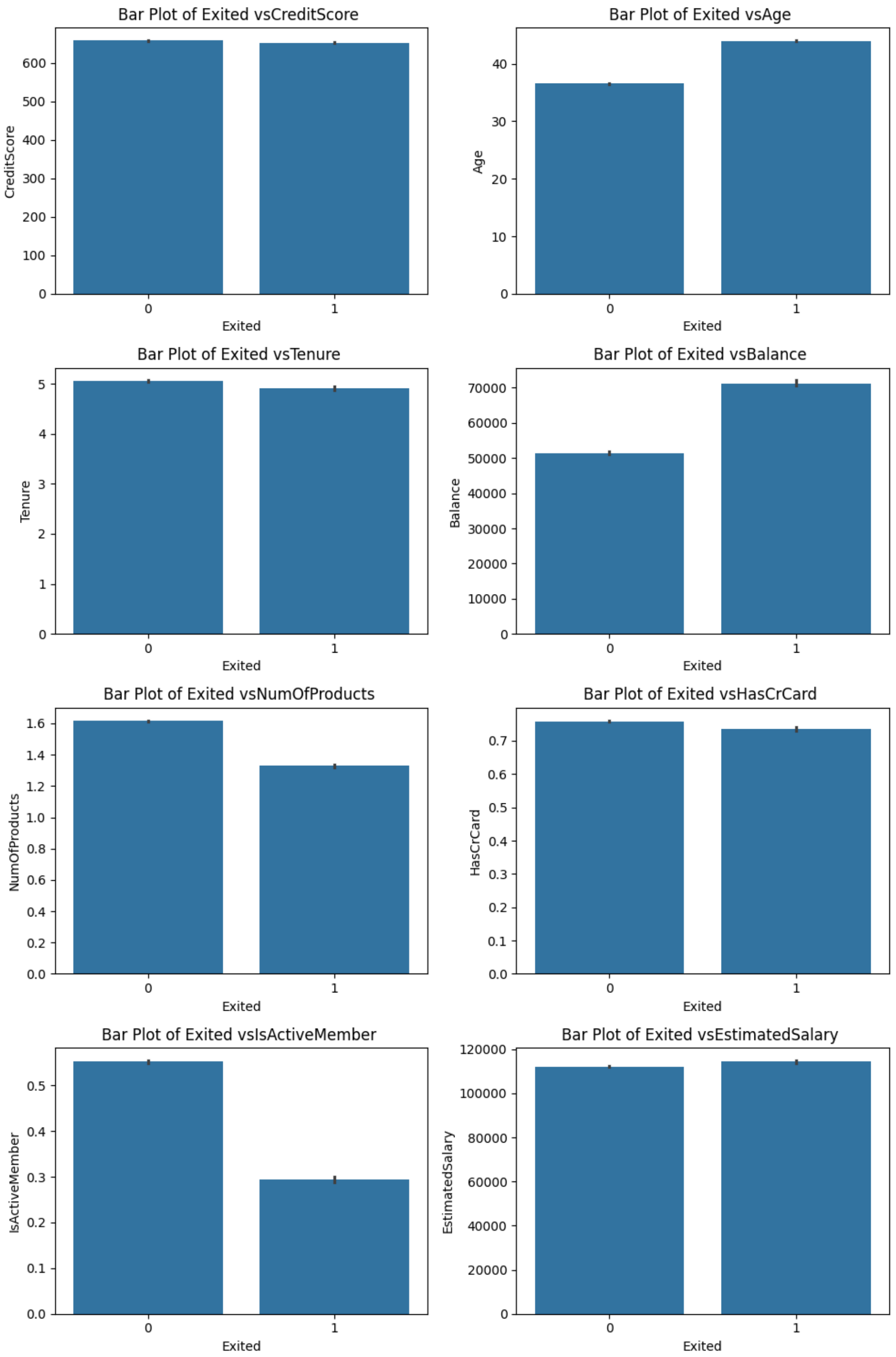


Figure 5

Next, we apply multivariate analysis to check correlation between variables. We check the correlation heat map (Figure 6) of numeric variables and customer churn flag. From the plot, we can see variables like the “number of products“ and “isactivemember“ are negatively correlated, and “age” and “balance” are positively correlated. This helps us confirm features we used are likely to be useful. We then use a heatmap (Figure 7) to further check correlations between each variables, and find generally there is no correlation between dependent variables, except some negative correlations between number of product and balance. We are less likely to deal with multicollinearity problems in this case.

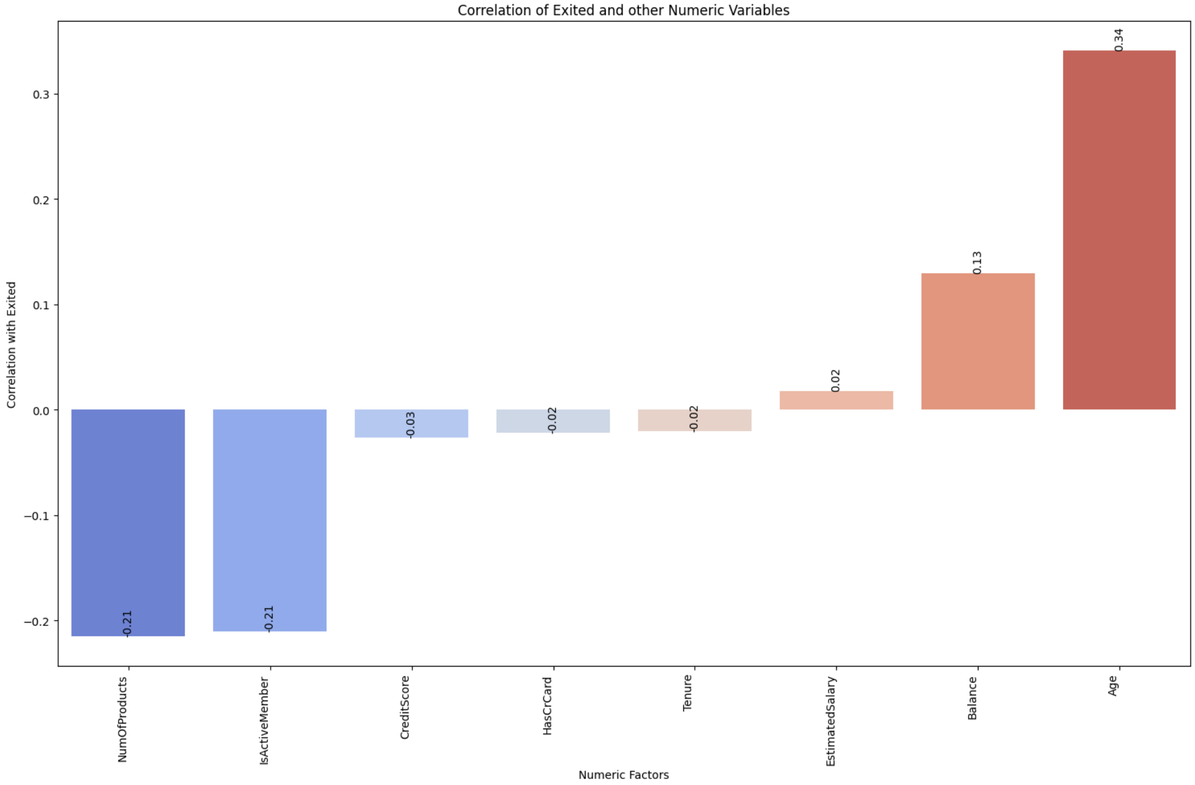


Figure 6

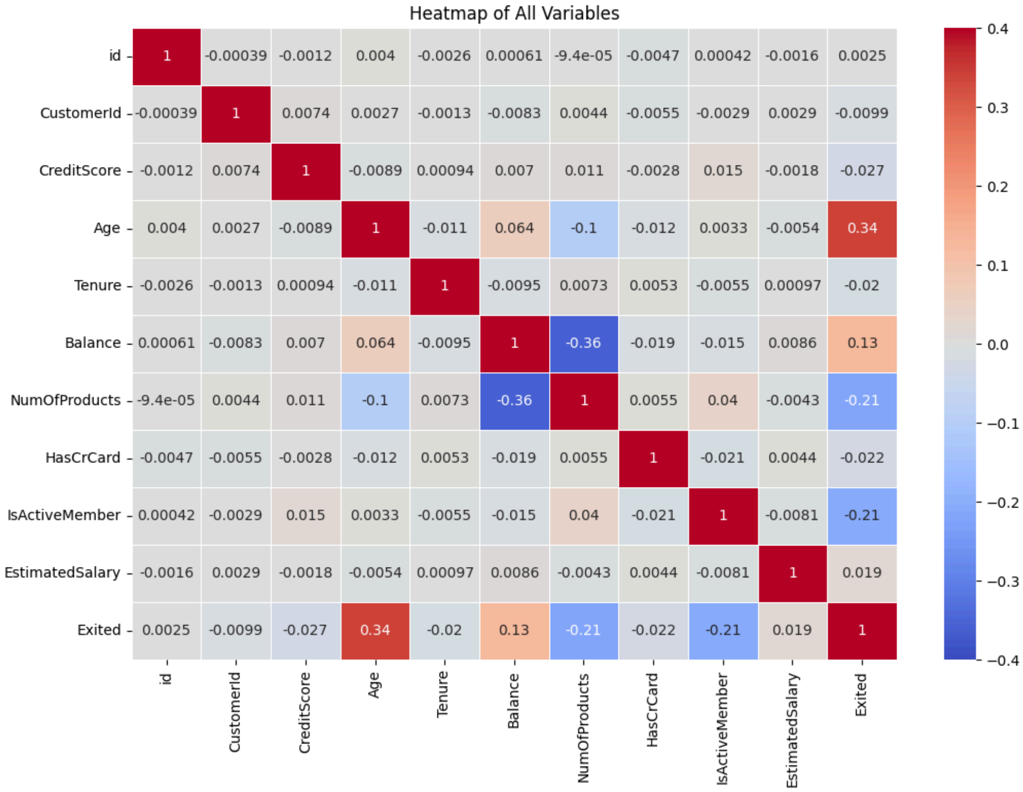

Heatmap of All Variables
id
CustomerId
CreditScore
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Exited
1 -0.00039 -0.0012 0.004 -0.0026 0.00061 -9.4e-05 -0.0047 0.00042 -0.0016 0.0025
-0.00039 1 0.0074 0.0027 -0.0013 -0.0083 0.0044 -0.0055 -0.0029 0.0029 -0.0099
-0.0012 0.0074 1 -0.0089 0.00094 0.007 0.011 -0.0028 0.015 -0.0018 -0.027
0.004 0.0027 -0.0089 1 -0.011 0.064 -0.1 -0.012 0.0033 -0.0054 0.34
-0.0026 -0.0013 0.00094 -0.011 1 -0.0095 0.0073 0.0053 -0.0055 0.00097 -0.02
0.00061 -0.0083 0.007 0.064 -0.0095 1 -0.36 -0.019 -0.015 0.0086 0.13
-9.4e-05 0.0044 0.011 -0.1 0.0073 -0.36 1 0.0055 0.04 -0.0043 -0.21
-0.0047 -0.0055 -0.0028 -0.012 0.0053 -0.019 0.0055 1 -0.021 0.0044 -0.022
0.00042 -0.0029 0.015 0.0033 -0.0055 -0.015 0.04 -0.021 1 -0.0081 -0.21
-0.0016 0.0029 -0.0018 -0.0054 0.00097 0.0086 -0.0043 0.0044 -0.0081 1 0.019
0.0025 -0.0099 -0.027 0.34 -0.02 0.13 -0.21 -0.022 -0.21 0.019 1
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4

Figure 7

Data Preparation and Feature Engineering

During the preparation stage, we will first drop irrelevant features like customerids and surname. Following on, we will keep outliers and split the dataset into 80% train and 20% test. Next, we encode every categorical feature into multiple boolean dummy variables using Sklearn one-hot encoding. As an example, geography location will be encoded into a binary True/False dummy based on the value. If a sample is in German, a flag of True will be assigned to Geography_German column, while the other Geography columns like Geography_Spain will be False. This allows us to include categorical features in model training. Finally, since the value of some features like estimated salary are significantly higher than others, when training linear / non-linear classifiers like Logistic regression and SVM we will use a standard scaler to make sure all features are on the same scale. This will drastically reduce these models' bias. For tree-based models like decision tree classifiers, we will not apply standard scaling as they are not based on absolute value of features.

Methodology and Tools

In order to test our models, we used Google Colab to run our EDA and model evaluation. In this notebook, we ran six models to try and classify our bank customer churn data. These included logistic regression, K Nearest Neighbors (KNN), a balanced random forest classifier, unbalanced random forest classifier, gradient boosting classifier, and finally a multi-layered perceptron (MLP) neural network using a completely crossed 2x2 experiment design. We felt that these six models gave us a wide variety of strategies to try and create an accurate classifier for our data. Each of these models have their own set of pros and cons and being able to compare each of their results would be helpful in achieving the best results. We used Grid Search to tune hyperparameters for all of the models other than the MLP. The dataset revealed an unbalanced classification label of ~80% for 0s and ~20% for 1s. Using an unbalanced random forest classifier could better account for this disparity.

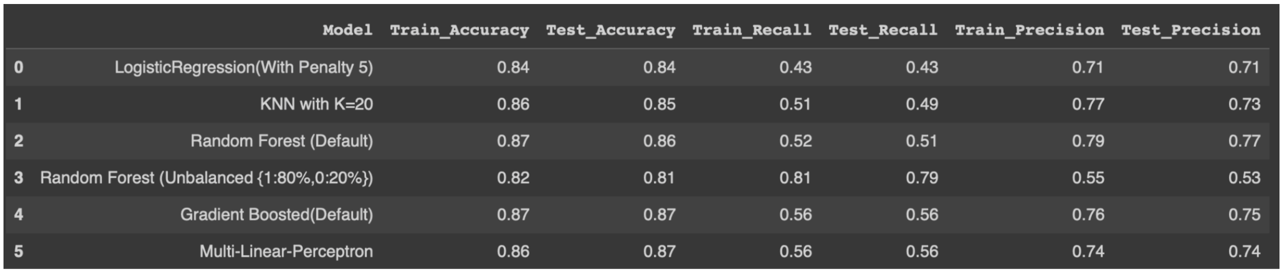
We chose these models to compare due to their differences and unique strengths. Logistic regression is a valid choice due to its ability to perform binary classification. The algorithm uses a sigmoid function, which is an extension of the basic linear regression function, that outputs one of two possible categories. By inputting the features of our datasets, the logistic regression model will output a value that is either above or below a threshold thus classifying as 0 or 1. Next, K-nearest neighbors can handle both qualitative and quantitative data. It is not affected severely by outliers and creates clusters of data by taking the distance to a certain data point. These clusters correspond to a classification label.

As mentioned before, random forest classifiers do not require the scaling of data since the trees are not based on the absolute value of the data. Decision trees are prone to certain machine learning downfalls such as bias and overfitting, however combining these into a random forest reduces the probability of these happening. Random forest follows the bagging ensemble learning method, as opposed to boosting which we will explain next. In this learning method, the trees are uncorrelated, and in random forests they focus on a subset of features. In our model testing we use both balanced and unbalanced random forest classifiers. Unbalanced random forest classifiers use class weights to better reflect the data. This can help prevent any biases towards the class that makes up the majority of the training data. Our fifth model, gradient boosting, takes on the other ensemble learning method - boosting. As the name suggests, it uses gradient descent to build upon the previous model and minimize some sort of error function. Because the algorithm updates weights based on the gradient values, it is less affected by outliers and can combine many weak learners to be strong learners.

Finally, multi layer perceptrons utilize input and hidden layers to create an output layer. By creating a 2x2 completely crossed experiment, we tuned the neural network's parameters of neurons and layers. We were careful to not have too many layers which can increase processing time along with becoming prone to overfitting. Lastly, we were careful to choose hyperparameters that avoided the vanishing gradient problem in which the gradient becomes so small the neurons stop backpropagating. Running all of these models in our Google colab, we were able to compare the accuracy, recall, and precision of our models to choose and recommend the most effective one.

Findings and Conclusions

Our models produced fairly similar results, however gradient boosting and the multi-layer perceptron performed the best on our test dataset by looking at the test accuracy, recall, and precision with about an ~87% accuracy score (Figure 8 and 9).



	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
0	LogisticRegression(With Penalty 5)	0.84	0.84	0.43	0.43	0.71	0.71
1	KNN with K=20	0.86	0.85	0.51	0.49	0.77	0.73
2	Random Forest (Default)	0.87	0.86	0.52	0.51	0.79	0.77
3	Random Forest (Unbalanced {1:80%,0:20%})	0.82	0.81	0.81	0.79	0.55	0.53
4	Gradient Boosted(Default)	0.87	0.87	0.56	0.56	0.76	0.75
5	Multi-Linear-Perceptron	0.86	0.87	0.56	0.56	0.74	0.74

Figure 8

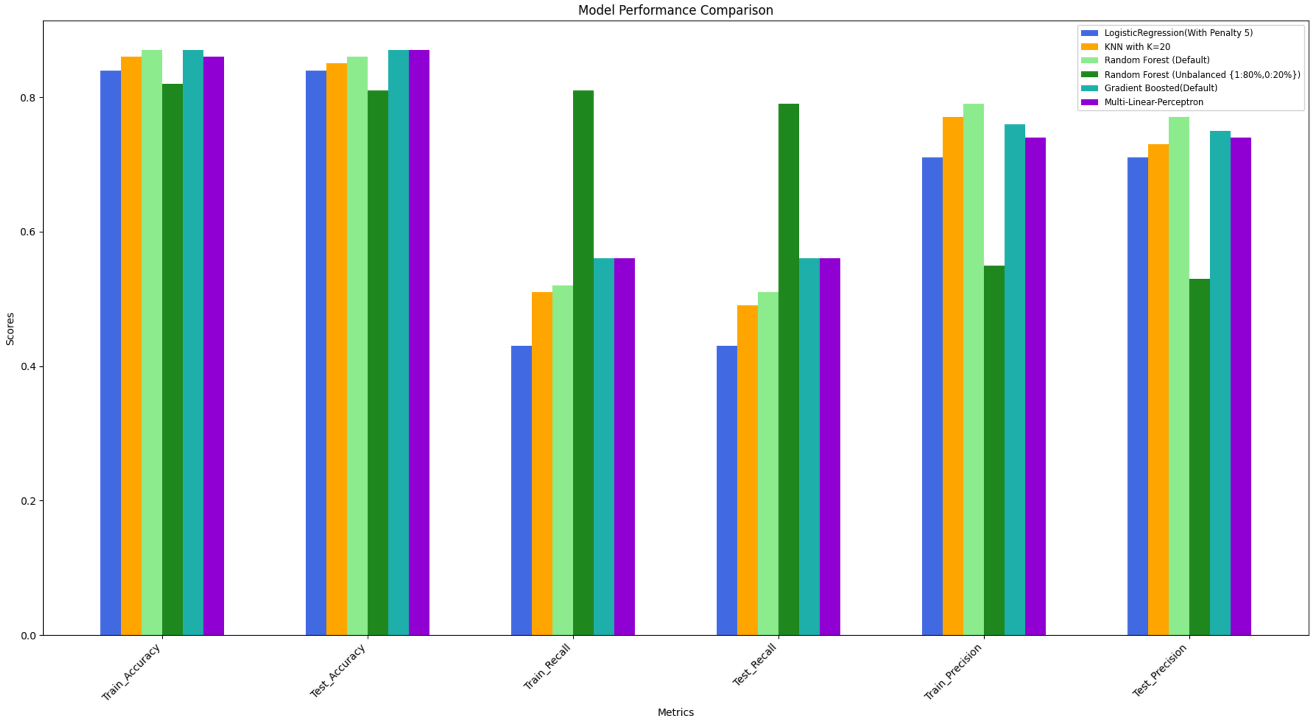


Figure 9

Checking the most important features from our random forest classifiers we can see that the number of products owned by a customer and their age are strong indicators of whether or not they exit from the bank. This also confirms our observation in Exploratory Data Analysis. Based on this data, banks can contact their customers and offer enticing deals on the products they might already own or offer more functionality. They can also look at a customer's age and devise ways to maximize the CLV. Either way, banks can use these models as a starting point for launching campaigns that target highly likely customers that will exit the bank soon. Focusing on retaining customers has high financial upside and as prior stated, is much more cost effective than finding new users.

Recommendations

Going forward, banks can use these models as a starting point for identifying customers to reach out to retain. We suggest that further modifications should be made that considers bringing in other data from outside datasets. This can include more economic data that reflects the current state of the country or world's economy such as unemployment, inflation, or interest rates. Or it can include more data points about a specific customer that our original dataset did not contain. Our dataset did not have a large number of features but if it did, principal component analysis can be conducted along with more training of our classifiers. We suggest using the MLP or gradient boosting classifier with this new data. Due to the regulation[2], banks need to use a model with good interpretability to label customers, and MLP, a black box model, may have limited usage in this scenario.

As for next steps, banks should use our classifiers that we have described here to identify potential customer risks for exiting the bank. Though the models are not perfect, it is more beneficial to cast a wider net when retaining users or offering enticing deals. Even if a customer is not contemplating exiting, a well timed or even proactive offer can lengthen their stay and thus help maximize their customer lifetime value. Through additional data collection and analysis, more indicators can be fed to our existing models to improve accuracy.

[2] Chien, Jennifer. "Transparency, Explainability, and Interpretability in AI/ML Credit Underwriting Models."

References

Tidey, Will. “Acquisition vs Retention: The Importance of Customer Lifetime Value.”

Inbound Marketing and Sales Partner, February 17, 2018.

https://www.huify.com/blog/acquisition-vs-retention-customer-lifetime-value.

Chien, Jennifer. "Transparency, Explainability, and Interpretability in AI/ML Credit Underwriting Models."

Innovation at Consumer Reports, 22 Mar. 2024,

https://innovation.consumerreports.org/transparency-explainability-and-interpretability-in-ai-ml-credit-underwriting-models/