Neel Sheth
Samarth Mishra
Srividya Rayaprolu
Tanushree Devi Balaji

# Predicting Bank Churn

## Description of Project Goals:

Description: The dataset we are investigating is a dataset of 10,000 rows and 14 columns that includes bank customer attributes and whether or not they "churn", or left the bank. We are attempting to find patterns in the customer attributes that will allow us to identify customers with waning engagement rates early on.
Importance:

- The primary use case would be identifying potential churners & formulating retention strategies, thus protecting the customer base.
- Although this dataset is only for a specific bank, our analysis could be extrapolated and employed by other banks as well to a wider customer base as a preliminary way to identify risk of churn
- Furthermore, our dataset and analysis is useful to even the customers themselves because customers could see if other customers who churned have similar attributes to them, and decide whether or not to join a certain bank depending on that.

## Exploratory Data Analysis:

Our exploratory data analysis began with checking the basics of our dataset and understanding each of the variables.
Sanity Checks: We found out that there were no null values, and we decided that we should try to bucket some of our attributes for enhanced readability and for use in some of the classification models.
Feature Creation: The variables we bucketed were Credit Score, Age, Balance, and Salary. We also experimented with a new metric that we named '% Credit Card Ownership', which was a simple ratio of the amount of credit cards a customer had to the number of accounts they had.
Variable Analysis: We also wanted to view a correlation matrix of our data to get a preliminary understanding of which of our variables were correlated with our target variable, the churn rate ('Exited'). After looking at the correlation matrix, we looked at the distributions of each variable individually (as they were originally and with our buckets and created variables), and their relationships to our target variable.
Findings: Churners tend to be between 40-50 years old, are female, and have higher bank balances. Furthermore, despite a higher customer base in

France, Germany has by far the highest churn rate. Other trends included the fact that customers who own only 1-2 products churn the least, longer tenured customers seem to churn slightly less (with a big dip at 7 tenured years), active customers tend to churn less (unsurprisingly) and when 1 out of 4 of the products owned by a customer is a credit card, the churn rate is high.

## Solution and Insights:

### K-Nearest Neighbors:

K-Nearest-Neighbors(KNN) is a classification model which uses distance (Euclidean distance, in our case), and a set of input values to try and predict the predictor value based on a 'K' amount of similar points to a given set of inputs in feature space. KNN is a simple model to run, easy to understand, and has low computational power needed, but is usually not the most accurate model; thus, making KNN a **good model to run first**, to get a baseline for the solution to our question. For our purposes, we first ran KNN with every variable to see what the accuracy score would be. In order to find out what the best attributes to include were, we ran the KNN model 15 different times, with a different combination of variables each time, to see which combination of variables performed the best. The subset that performed the best ended up **being every variable, except 'Has Credit Card', 'Tenure', and 'Salary'**, which were subsequently removed. To decide the optimal value of "K", or the nearest neighbors, we plotted the accuracy score against the K, and chose the point that looked to be the best, which ended up being 15 (**Fig 2.1**). Finally, the final model was created, with the optimal variable subset and the **optimal K (15)**. With these parameters, a confusion matrix was created to show our result, the best **test set accuracy** for the KNN model, at **80.4%** (**Fig 2.2**).

### Logistic Regression:

Logistic regression is a classification model that uses input variables (features) to predict a categorical outcome variable (label) that can take on one of a limited set of class values. 'Exited' has only two states and hence, binomial logistic regression is used. We split the data into training and test in a 70-30 ratio and used all the variables for the first model. This gave an accuracy of 78.76% with high specificity (98.6%) and very low sensitivity(3.67%). (**Fig 3.1**) The logistic regression model **(Fig 3.2)** shows us that Age, Credit Score and Location play a very important role in determining customer churn. The scikit uses a default threshold value of 0.5. We tried to find the optimum threshold value by maximizing the difference between True Positive rate and False Positive rate. This resulted in a threshold of 0.20 which gives a better True Positive Rate**(Fig 3.3).** But since reducing the threshold would ultimately reduce the precision, we tried other models of logistic regression using Ridge and Lasso. Ridge gave similar results and hence, we moved on to Lasso. Lasso gave a better accuracy of 81.2%**(Fig 3.4)**. Lasso displays a similar variable importance but also accounts for the percentage of credit cards in comparison to all other items**(Fig 3.5)**.

### Naive-Bayes:

Naive Bayes is a classification method whose predictions are driven by the product of **prior probabilities** (probabilities of the dependent class) and **likelihoods** (probabilities of combinations of predictors), assuming independence. For this specific data set, we split the data into training and test sets in a 70-30 ratio and used 9 categorical variables (37 features) to predict churn propensity. This yielded an accuracy of **82.4% on the train set** and an equally good accuracy of **82% on the test set**, eliminating concerns of overfitting.

**Fig 4.1** shows the most important predictors for Churn = 1. Usually when users own 3 or more products and only one of them is a credit card, the odds of them churning are high. These are possibly the earliest accounts of working professionals or older, who have since gravitated towards other banks with better offers.

**Fig 4.2** shows the most important predictors for churn = 0. Based on these variables, young users with 2 products, one of which is a credit card, have low odds of churning. These are probably students or people who are just starting off their careers with this new account & hence, are more engaged.

In summary, how many **credit cards you have in proportion to total products** & to an extent, **age** are key predictors of churn in the Naive Bayes Model.

### Decision Trees and Tree Ensemble Methods:

We ran the Decision Tree classifier and the various ensemble methods associated with trees (Bagging, Random Forest and Gradient Boosting) on the dataset to predict churning customers. Decision trees take many different features and split them one at a time to try and classify the target variable. They are then pruned to reduce height and complexity, but with ensemble methods, a tree can be built many different times and averaged out. Ensemble methods combine multiple classifiers to try and avoid model bias, but we do have to be careful of overfitting when utilizing ensemble methods.

1. Running models with no limit on depth of trees (**Fig 5.1**) : We observed that the models seem to overfit on the training data with almost ~100% accuracy for Decision Tree, Bagging and Random Forest Classifiers

2. Running models with Depth of Tree=2 (**Fig 5.2**): With a limit on depth of tree =2, we are able to prevent overfitting on training data with models giving similar accuracies for the training and test sets

3. Variation of Test Accuracies with Number and Depth of Trees Parameters

Number of Trees: Varied from 5 to 400

Depth of Trees: Varied from 1 to 10

Most Optimal Parameters Observed (**Fig 5.3**): We see that the Gradient Boosting performs the best among all the classifiers giving a test accuracy of 86.2%

Variable Importance (Gradient Boosting)(**Fig 5.4**)

While ***Age*** and ***NumOfProducts*** are the most important variables in predicting churn, ***EstimatedSalary***, ***HasCrCard***, and ***Tenure*** do not play a major role in predicting churn.

## Model Selection:

As we can see from **Fig 6.1**, ***Gradient Boosting*** performs the best among all the models giving the highest test accuracy between all the various models.

## Recommendations & Insights:

The variables that are most meaningful across models are Number of Products and Age.

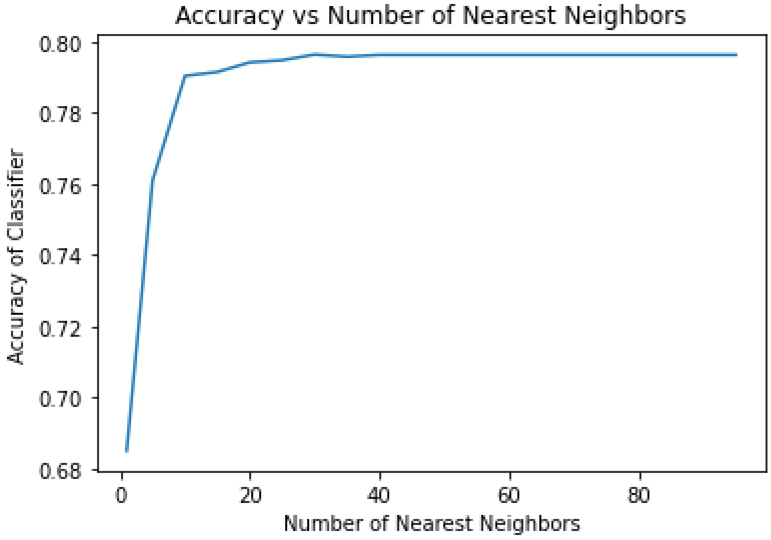
- Middle aged users with over 3 products are at the highest risk of churn. This group is a good starting point for loyalty programs.
- As seen from Naïve Bayes, Credit Card ownership is meaningful when interpreted independently. The same group could also be the ideal target demographic for credit card campaigns.

Caveats:

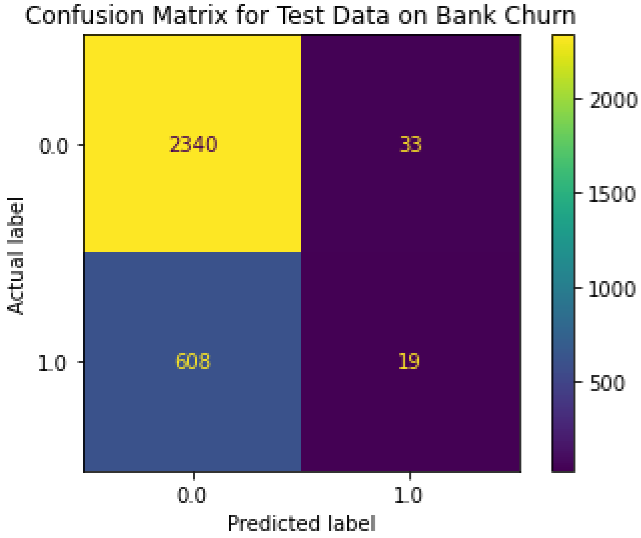
- While models are good indicators of relative trends, it can hard to define causal relationships – while utilizing the model, it would be a clever idea to perform A/B tests to validate our hypotheses
- The data size is specific to 1 bank and 3 regions. Testing out these models on datasets of higher granularity & larger size might result in more generalized outcomes

# Plots and Tables (Appendix)

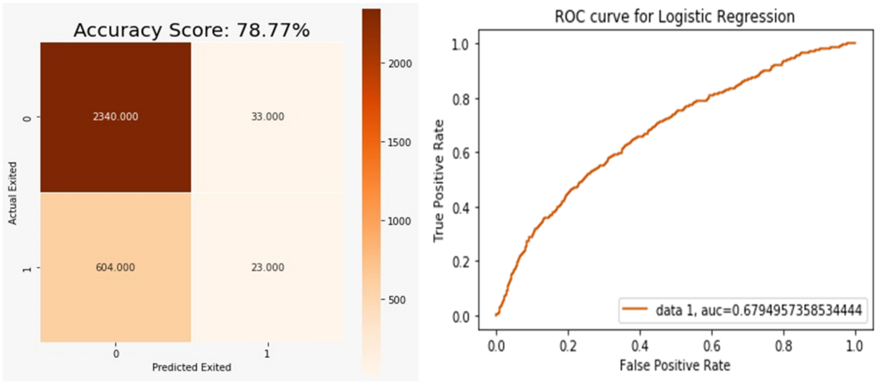
## Fig 2.1 - Find the optimal K for KNN



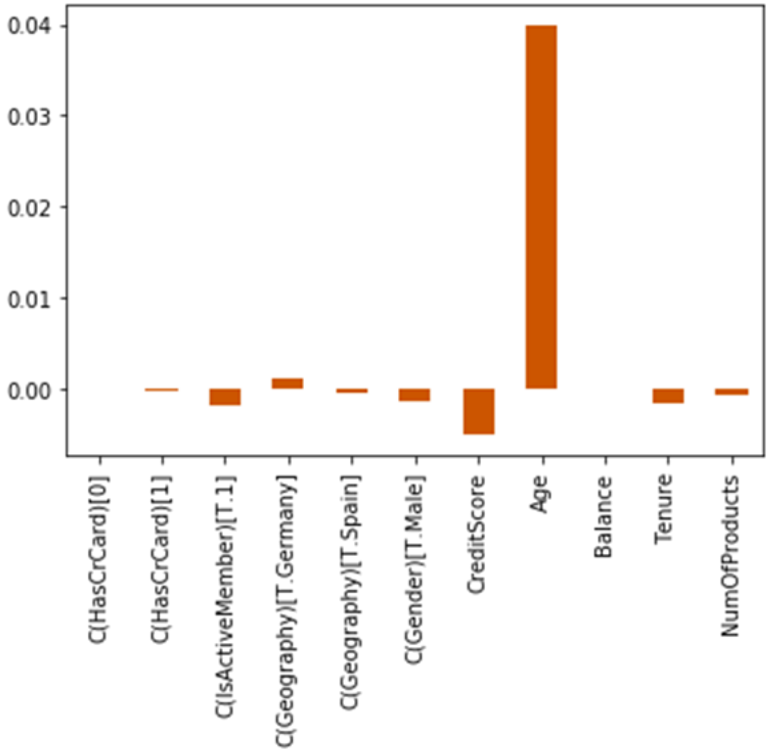
## Fig 2.2 - Confusion Matrix for KNN



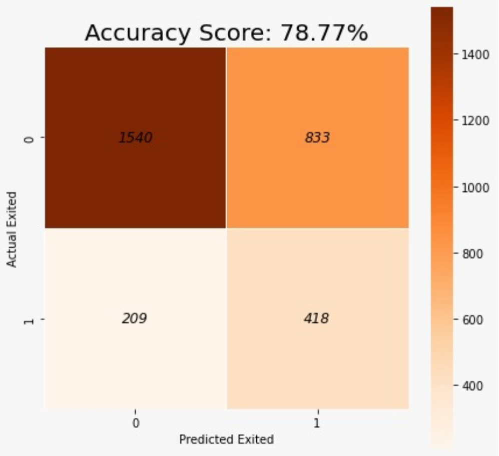
## Fig 3.1 - Confusion Matrix for Logistic Regression



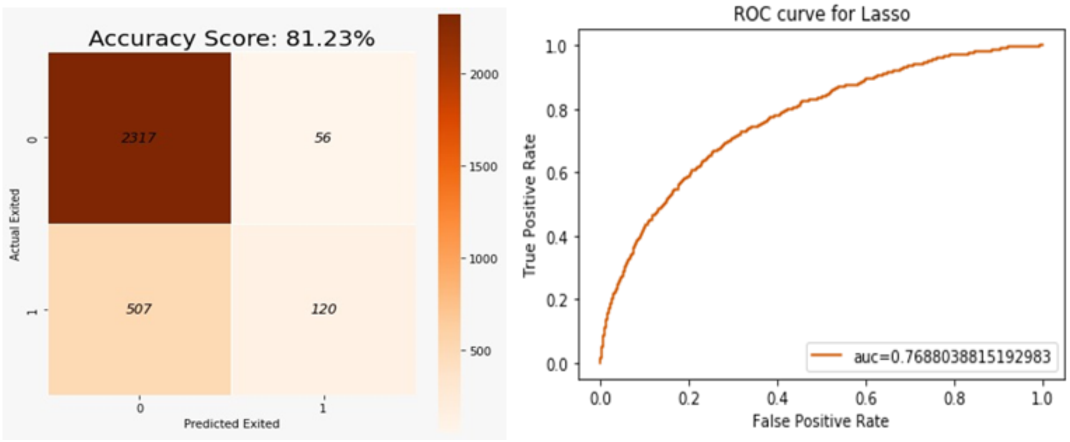
## Fig 3.2 - Feature Importance



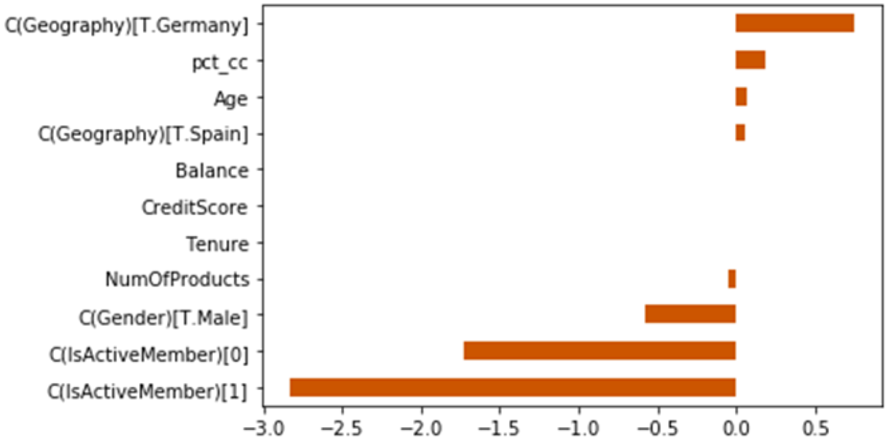
## Fig 3.3 - Threshold Optimization at 0.21



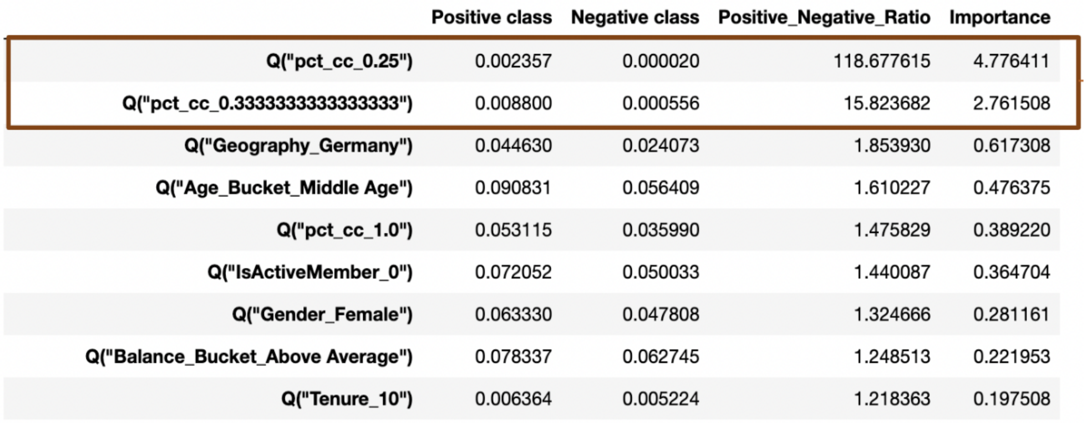
### Fig 3.4 - Lasso Confusion Matrix and ROC curve



### Fig 3.5 - Lasso Feature Importance

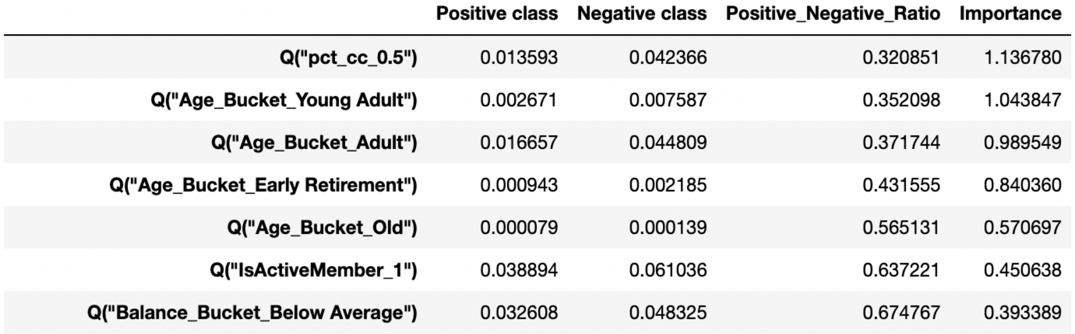


### Fig 4.1 - Best Predictors for Churn=1



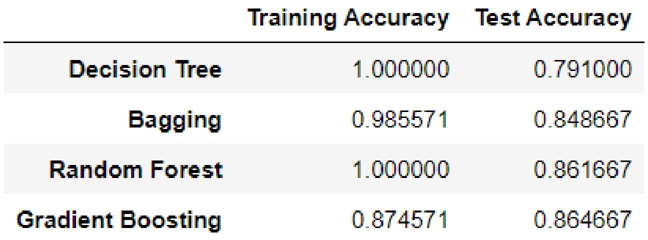
| | Positive class | Negative class | Positive_Negative_Ratio | Importance |
|---|---|---|---|---|
| **Q("pct_cc_0.25")** | 0.002357 | 0.000020 | 118.677615 | 4.776411 |
| **Q("pct_cc_0.3333333333333333")** | 0.008800 | 0.000556 | 15.823682 | 2.761508 |
| **Q("Geography_Germany")** | 0.044630 | 0.024073 | 1.853930 | 0.617308 |
| **Q("Age_Bucket_Middle Age")** | 0.090831 | 0.056409 | 1.610227 | 0.476375 |
| **Q("pct_cc_1.0")** | 0.053115 | 0.035990 | 1.475829 | 0.389220 |
| **Q("IsActiveMember_0")** | 0.072052 | 0.050033 | 1.440087 | 0.364704 |
| **Q("Gender_Female")** | 0.063330 | 0.047808 | 1.324666 | 0.281161 |
| **Q("Balance_Bucket_Above Average")** | 0.078337 | 0.062745 | 1.248513 | 0.221953 |
| **Q("Tenure_10")** | 0.006364 | 0.005224 | 1.218363 | 0.197508 |

### Fig 4.2 - Best Predictors for Churn=0



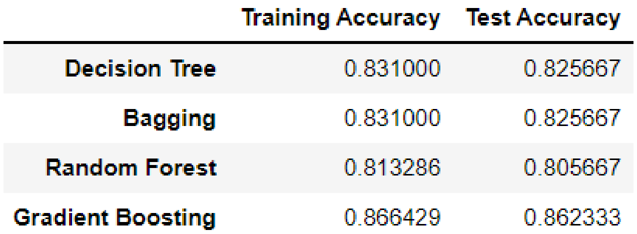
| | Positive class | Negative class | Positive_Negative_Ratio | Importance |
|---|---|---|---|---|
| Q("pct_cc_0.5") | 0.013593 | 0.042366 | 0.320851 | 1.136780 |
| Q("Age_Bucket_Young Adult") | 0.002671 | 0.007587 | 0.352098 | 1.043847 |
| Q("Age_Bucket_Adult") | 0.016657 | 0.044809 | 0.371744 | 0.989549 |
| Q("Age_Bucket_Early Retirement") | 0.000943 | 0.002185 | 0.431555 | 0.840360 |
| Q("Age_Bucket_Old") | 0.000079 | 0.000139 | 0.565131 | 0.570697 |
| Q("IsActiveMember_1") | 0.038894 | 0.061036 | 0.637221 | 0.450638 |
| Q("Balance_Bucket_Below Average") | 0.032608 | 0.048325 | 0.674767 | 0.393389 |

### Fig 5.1 - Running models with no limit on depth of trees



| | Training Accuracy | Test Accuracy |
|---|---|---|
| Decision Tree | 1.000000 | 0.791000 |
| Bagging | 0.985571 | 0.848667 |
| Random Forest | 1.000000 | 0.861667 |
| Gradient Boosting | 0.874571 | 0.864667 |

### Fig 5.2 - Running models with Depth of Tree=2



| | Training Accuracy | Test Accuracy |
|---|---|---|
| Decision Tree | 0.831000 | 0.825667 |
| Bagging | 0.831000 | 0.825667 |
| Random Forest | 0.813286 | 0.805667 |
| Gradient Boosting | 0.866429 | 0.862333 |

### Fig 5.3 - Variation of Test Accuracies with Number and Depth of Trees Parameters

| Model | Number of Trees | Depth of Tree | Training Accuracy | Test Accuracy |
|---|---|---|---|---|
| Decision Tree | **-** | **7** | **87.11%** | **85.90%** |
| Bagging | **10** | **8** | **88.47%** | **85.93%** |

| | | | | |
|---|---|---|---|---|
| Random Forest | **10** | **2** | **87.61%** | **85.93%** |
| Gradient Boosting | **50** | **2** | **86.23%** | **86.20%** |

**Fig 5.4 - Variable Importance (Gradient Boosting)**

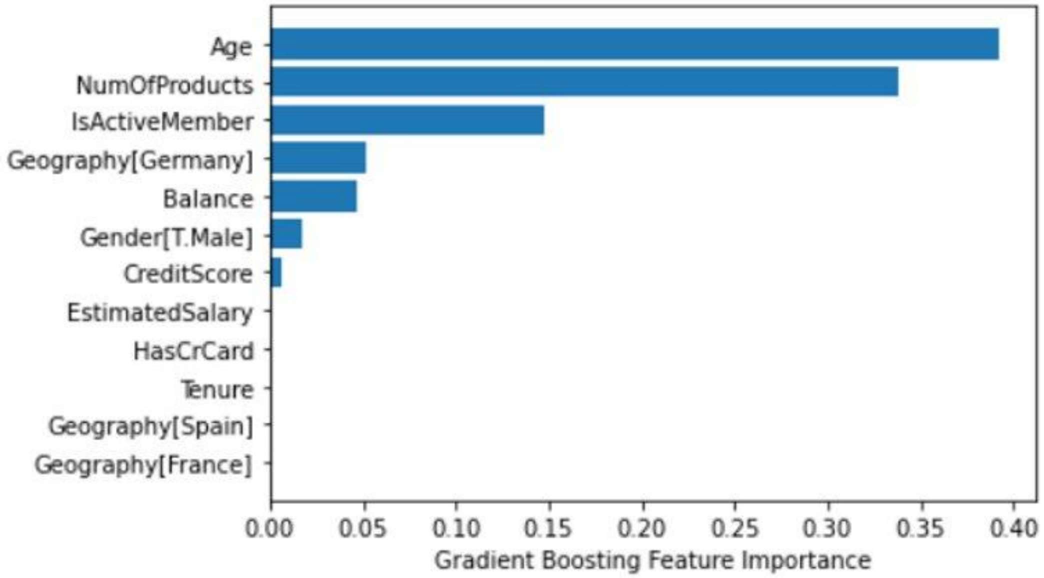


**Fig 6.1 - Model Selection**

| Model | Logistic Regression | KNN | Naive Bayes | Decision Tree | Bagging | Random Forest | Boosting |
|---|---|---|---|---|---|---|---|
| Test Accuracy | **78.7%** | **80.4%** | **82.0%** | **85.9%** | **85.9%** | **85.9%** | **86.2%** |