Report: Using Machine Learning to predict Bank Customer Churn

---

The Dataset is about bank customers churning and can be found on Kaggle:

https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling

Disclaimer: The dataset above is simulated.

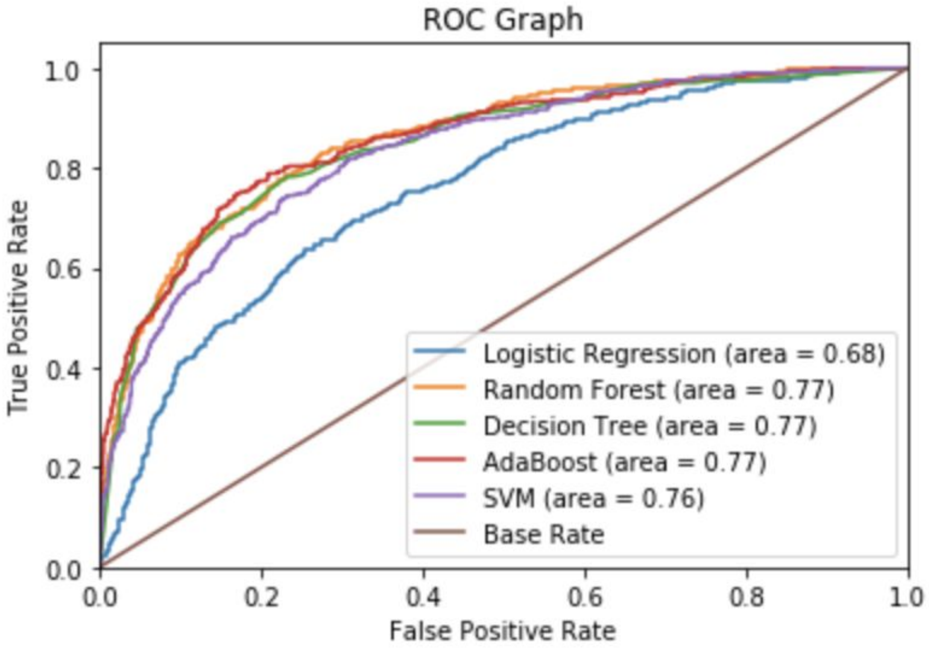
After performing statistical analysis in the last step, the next thing to do was choosing and training different machine learning classifiers on the data. First and foremost, the class balance of the response variable in the dataset was checked and it turned out that the response classes were imbalanced in the dataset. What this means is that 'Exited' (the response variable) has two classes 0 and 1 that are imbalanced in the dataset. In this particular case, there were 20.4% '1' and 79.6% '0' instances of the two classes in the 'Exited' field. This information is important as now we know that we should use a class balancing technique later on before training our models.

Another important step was to get rid of feature columns that do not give us any valuable information. In this case, we dropped the columns 'Surname' and 'CustomerId' with the rest of the data saved as 'X'. I wanted the independent variables to occupy the same scale so I used the robust scaler method on my data. The robust scaler removes the median and scales the data according to the quantile range while also being robust to outliers, hence the name robust scaler. After scaling the data, it was split into training and test sets with an 80-20 train test split.

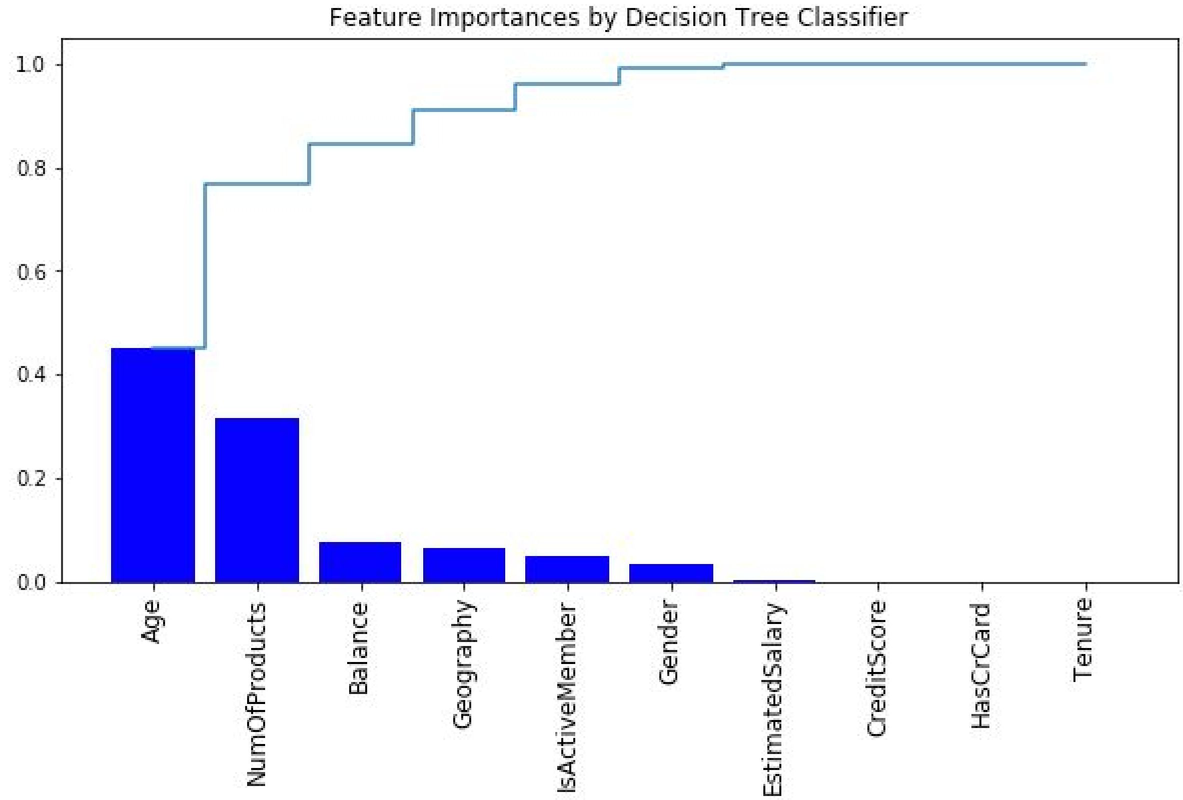
There was a need for a base classifier, which in this case I picked the Logistic Regression Classifier, so that the result of this classifier could serve as a baseline to check other models against. One thing that struck out about the logistic regression classifier was that its AUC - ROC score improved by 10% after changing the class_weight parameter to 'balanced'. This could be explained due to our data being imbalanced. The Logistic Regression Classifier got an AUC - ROC score of **0.68**, which now serves as our baseline AUC - ROC score to check against the other classifiers. The AUC - ROC curve is a performance measurement for classification problems at various threshold settings. ROC is a probability curve and AUC represents degree or measure of separability. Basically, it tells how much a model is capable of distinguishing between classes. The higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. The ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) with the TPR on the y-axis and FPR on the x-axis. Another thing that is important to note here is that we are using the AUC - ROC score as our metric of choice to assess the quality of the models and not the accuracy because we had an imbalanced dataset.

Next, it was time to handle the class imbalance in the dataset using SMOTENN which combines over- and under-sampling using SMOTE and Edited Nearest Neighbours. After transforming the data using SMOTEENN, there were 5551 **'1'** samples and 4309 **'0'** samples. It is important to mention that this transformation was only performed on the training set as we would like our test set to be as pure as possible. After getting the resampled data, hyper-parameter tuning was carried out for Decision Tree Classifier, AdaBoost Classifier, Random Forest Classifier and the Support Vector Machine Classifier to find the best hyper-parameters of each model.

All the 4 machine learning classifiers performed much better than the Logistic Regression base classifier. Subsequently, 3 of the 4 machine learning classifiers were tree based methods and they all performed equally, with an AUC score of **0.77**. The Support Vector Machine Classifier performed the poorest out of the 4 classifiers and got an AUC score of **0.76.** It would be best to go with the Decision Tree classifier as it is the least computationally expensive and performs at least as well as the other models.



One of the things that Decision Trees allows you to do is assess which features in your dataset were the most important and we can visualize each features contribution as it can be seen below. It appears that 'Age' and 'NumOfProducts' contributed more than 70% in terms of feature importance. 'Tenure', 'HasCrCard', and 'CreditScore' did not contribute at all which is surprising as one would think they would be good predictors of bank customers exiting or not.



The main takeaway from these analysis is to use one of the three tree based models as your final classifier for predicting if bank customers will exit or not. My recommendations for the future include the bank acquiring more data so other variants of tree based methods like Bagging Classifier, Gradient Boosting Classifier etc. can be trained to see how they perform. Another thing the bank could do is start collecting more data on more features for e.g. how long the customer has been with the bank, satisfaction score with the bank etc. These things might help us improve our model, especially collecting more data as this bank customer dataset was relatively small.

Now that we have our model and we can predict a customer exiting or not, how do we use that information to positively impact the bank’s business? It is important that rather than just predicting if the bank customer will exit the bank or not, we would rather have an estimate of the probability that the customer will exit the bank or not. We could rank the customers by their probability of leaving, and allocate an incentive budget to the highest probabilities in hopes of retaining them.

We can run into two sorts of problems with this approach. Firstly, consider that the customer is given incentives by the bank because they think the customer will leave at the end of the month, but in reality the customer doesn’t leave. This is called a false positive and this mistake could be expensive, inconvenient and time consuming for all parties. It is not all negative though as it can be seen as a good investment for relational growth between the bank and the customer.

The opposite scenario would be where the bank does not provide any incentive offer to the customer and they do leave. This is called a false negative and this error is more costly because the bank lost a customer, which translates to lost revenue for the bank. It also means that the more false negative cases the bank endures, the more it will have to spend on its marketing to attract new customers. Depending on these errors, the different costs are weighed based on the type of customer being treated. If it is a high-net worth customer then the loss is greater than losing a low-net worth customer. Hence, the cost for each error is different and this should be taken into account.