Jedidiah Webster

Introduction to Machine Learning Applications

Final Project

Churn Modeling - Categorization of Bank Customers

I. Executive Summary

The problem detailed in this Kaggle dataset is a common issue for many businesses today. Churn activity can disrupt budgeting processes, expected cash flow estimates, and customer lifecycle understanding. Churning is commonly used to refer to gaming a rewards system by repeatedly signing up for promotional products only to shortly cancel the account or subscription after the promotional reward is received. Churn can also simply refer to customers who are cancelling their subscription or ending their consumer relationship with an organization, and so churn rate is often synonymous with customer turnover rate. In this particular case, customers of a bank are presented and the challenge is to classify them as someone who kept their account(s) open, or someone who churned, or closed their account(s).

This is a binary classification problem, much like the Titanic dataset, and therefore the splitting methodology and scoring metric used in this problem will be analogous to the competition involving that data. In the Titanic dataset, the training set gives information about 891 passengers while the testing set provides details for 418. That corresponds to about a 70/30 train/test split, and so the same split will be employed in the attempt to solve this bank customer churning challenge. The scoring metric for the Titanic dataset is the percentage of the test set that was correctly categorized, and so the same scoring metric will be used for this problem, potentially with some extra cross validation requirements imposed.

The dataset gives information about 10000 bank customers from Spain, Germany, and France. It provides a customer ID, a surname, credit score, country of residence, stated gender, age, tenure as a customer, total balance, number of banking products owned, a flag for having a credit card with the bank, a flag for being an “active” member (with active not explicitly defined), and estimated salary. The categorization of churn or no churn is given as ‘Exited’, and there is an entry for each of the 10000 customers, providing an opportunity for robust cross validation with potential models.

The final models took as inputs the following continuous features: credit score, age, tenure, balance, number of products, and estimated salary. One hot encoding was used to transform the categorical features location and gender into binary variables, and then the final models took the following binary features as inputs: has credit card, is active member, male, female, France, Germany, Spain.

The final three models after the implementation of cross validation techniques were a decision tree, a random forest, and an adaptive gradient boosting classifier. The most accurate model was the random forest, but the most generalizable was the decision tree, with less than 0.5%

difference between the training and testing set accuracy scores. The final accuracy scores ranged between 84%-86% for both the testing and training sets.

II. Benchmarking of Other Solutions

Notebook Name	Feature Approach	Model Approach	Train/Test Perf
Churn Deep-Learning VS XGBoost	Correlation analysis	Artificial neural network and gradient boosted decision tree	ANN: ~0.845/0.8615 XGB: 0.852/0.8635
Churn prediction by selecting from 11 tuned models	Distribution plots	11 different classification models	Ensemble: 0.867/0.8685
Classification Model Comparison-Banks & Customers	Standardization	Every sklearn classifier	Best performance: Not given/0.862

a) Notebook 1:

This notebook compares a deep learning approach with a gradient boosted decision tree. The features are mean centered and standardized using the StandardScaler from sklearn.preprocessing, and a correlation matrix gives the information for feature selection. Geography and Gender are transformed into binary inputs with label encoding and one hot encoding. The train/test split is 80/20.

For the artificial neural network, keras is used to create a model with two Rectified Linear Unit hidden layers and a sigmoid function output layer. The optimizer used is Adamax, and there is hyper parameter tuning using GridSearchCV from sklearn.model_selection.

For the gradient boosted decision tree, XGBClassifier is used with hyper parameter tuning using RandomizedGridSearch from sklearn.model_selection with area under the ROC curve as the scoring metric.

In this particular instance, the decision tree outperformed the neural network, displaying the efficacy of using a more classical approach.

b) Notebook 2:

This notebook compares 11 different classifiers including decision trees, nearest neighbors classifiers, logistic regression, and more. A correlation matrix and factor plot are both used to visualize the features, as well as distribution plots of several features. Once again, label encoding and one hot encoding are used for the categorical variables. The train/test split is 80/20.

The classifiers are tuned for optimal hyperparameters, and then the top three performers are put together in an ensemble model using the VotingClassifier from sklearn.ensemble. These turned out to be CatBoost, RandomForest, and Gradient Boosting Machine.

c) Notebook 3:

This notebook compares every classifier in the sklearn library. The exploratory data analysis is performed in a similar manner to the previous two notebooks. The train/test split is 90/10.

Performance is visualized using a confusion matrix in addition to just looking at the percentage of correct predictions.

III. Data Description and Initial Processing

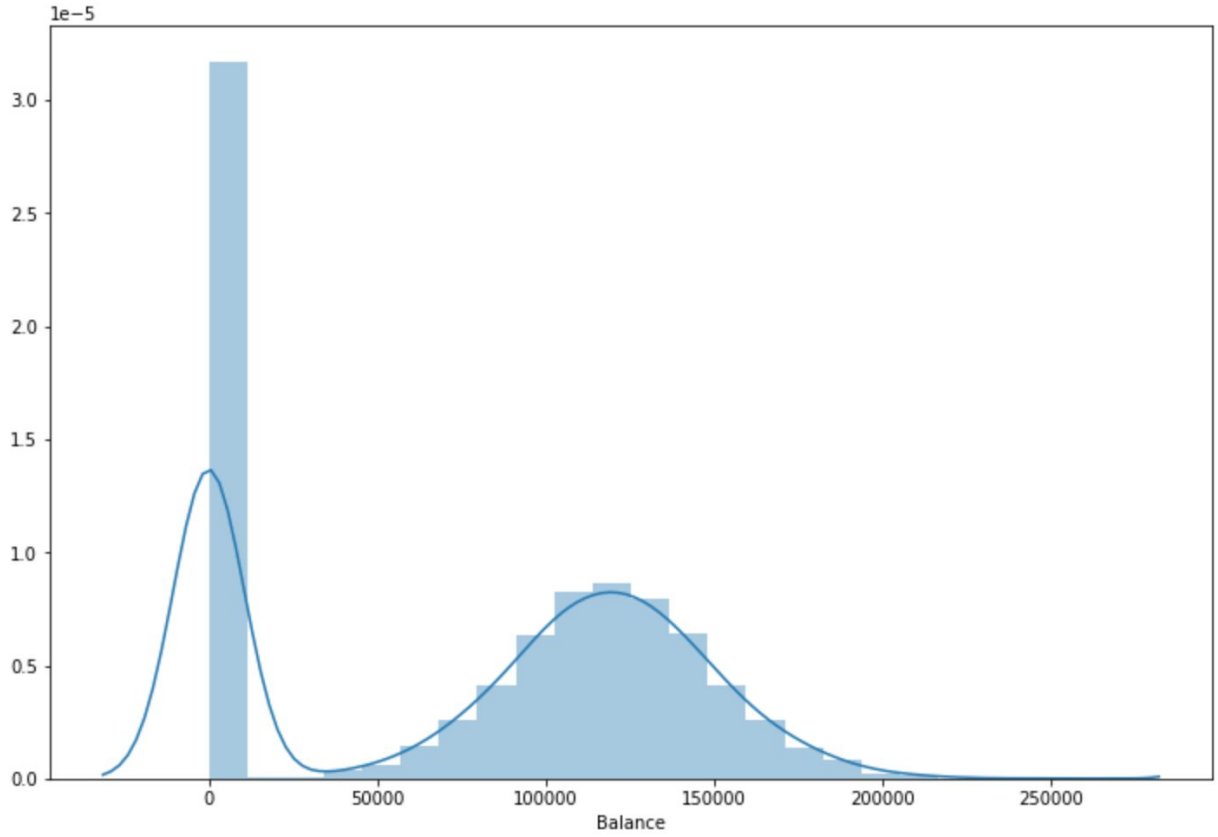
Taking a look at the relevant data:

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...
9995	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

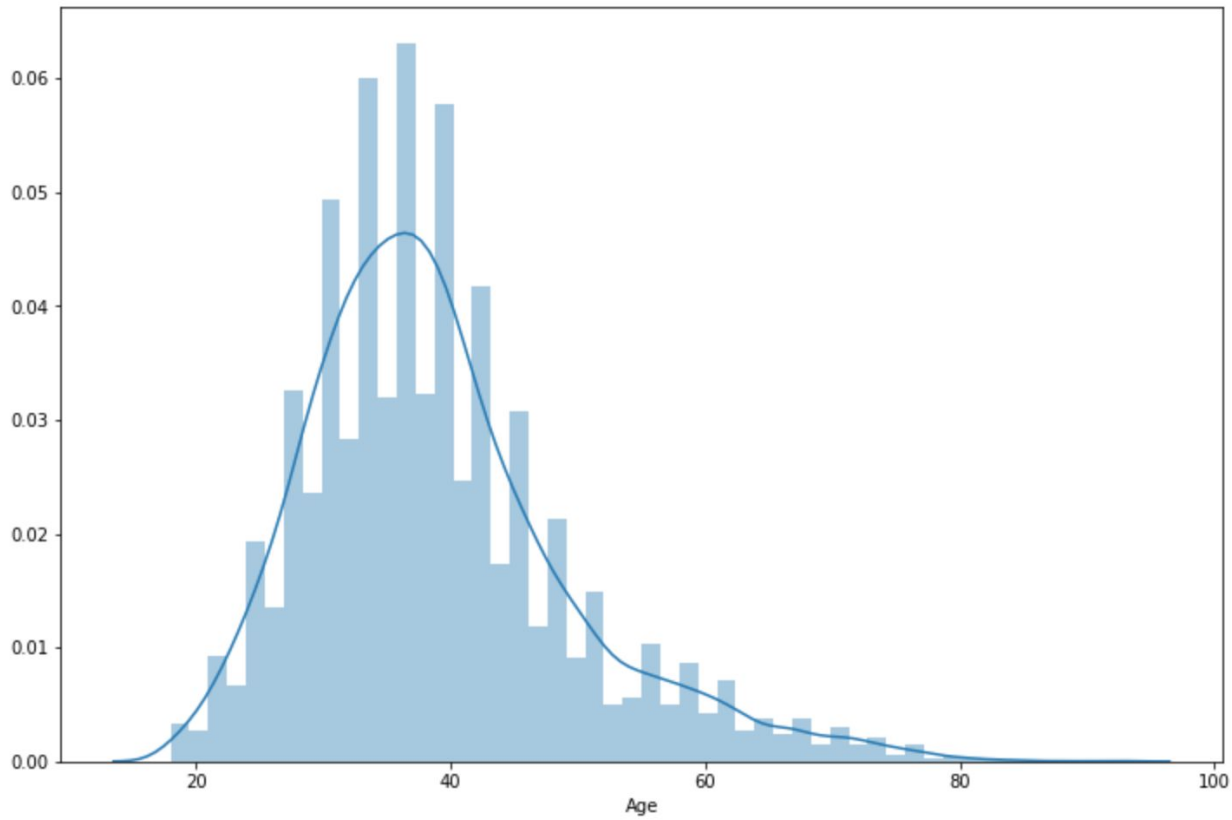
Here we have already removed the columns 'RowNumber', 'CustomerId', and 'Surname'. 'Exited' is the target variable in this binary classification problem. We now look at summary statistics for the numerical data to get a better sense of it.

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.00000	10000.000000	10000.000000
mean	650.528800	38.921800	5.012800	76485.889288	1.530200	0.70550	0.515100	100090.239881
std	96.653299	10.487806	2.892174	62397.405202	0.581654	0.45584	0.499797	57510.492818
min	350.000000	18.000000	0.000000	0.000000	1.000000	0.00000	0.000000	11.580000
25%	584.000000	32.000000	3.000000	0.000000	1.000000	0.00000	0.000000	51002.110000
50%	652.000000	37.000000	5.000000	97198.540000	1.000000	1.00000	1.000000	100193.915000
75%	718.000000	44.000000	7.000000	127644.240000	2.000000	1.00000	1.000000	149388.247500
max	850.000000	92.000000	10.000000	250898.090000	4.000000	1.00000	1.000000	199992.480000

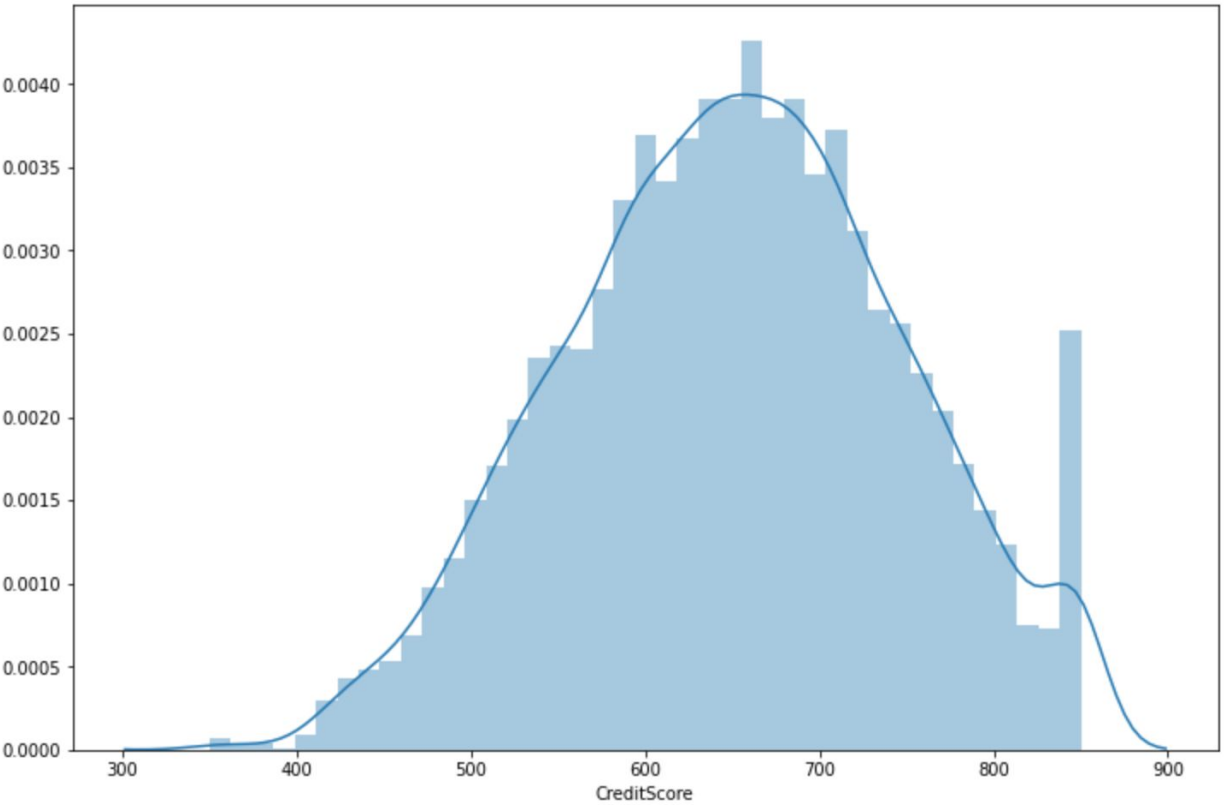
We look at distribution plots for certain features in order to get a better sense of the customer base for the bank. The y-axis represents density.



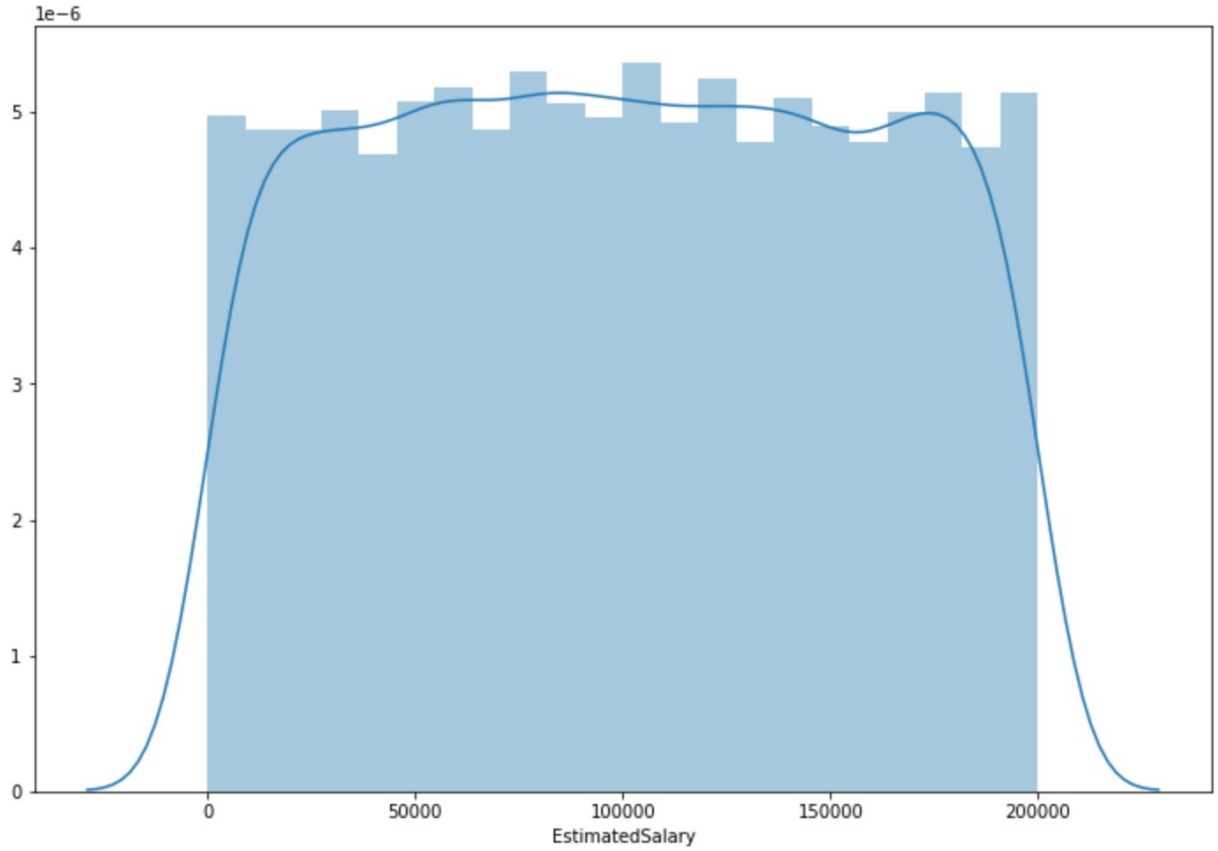
We see a normal distribution for average balance with the exception of those customers with a zero balance.



Here we see a positively skewed distribution for age.

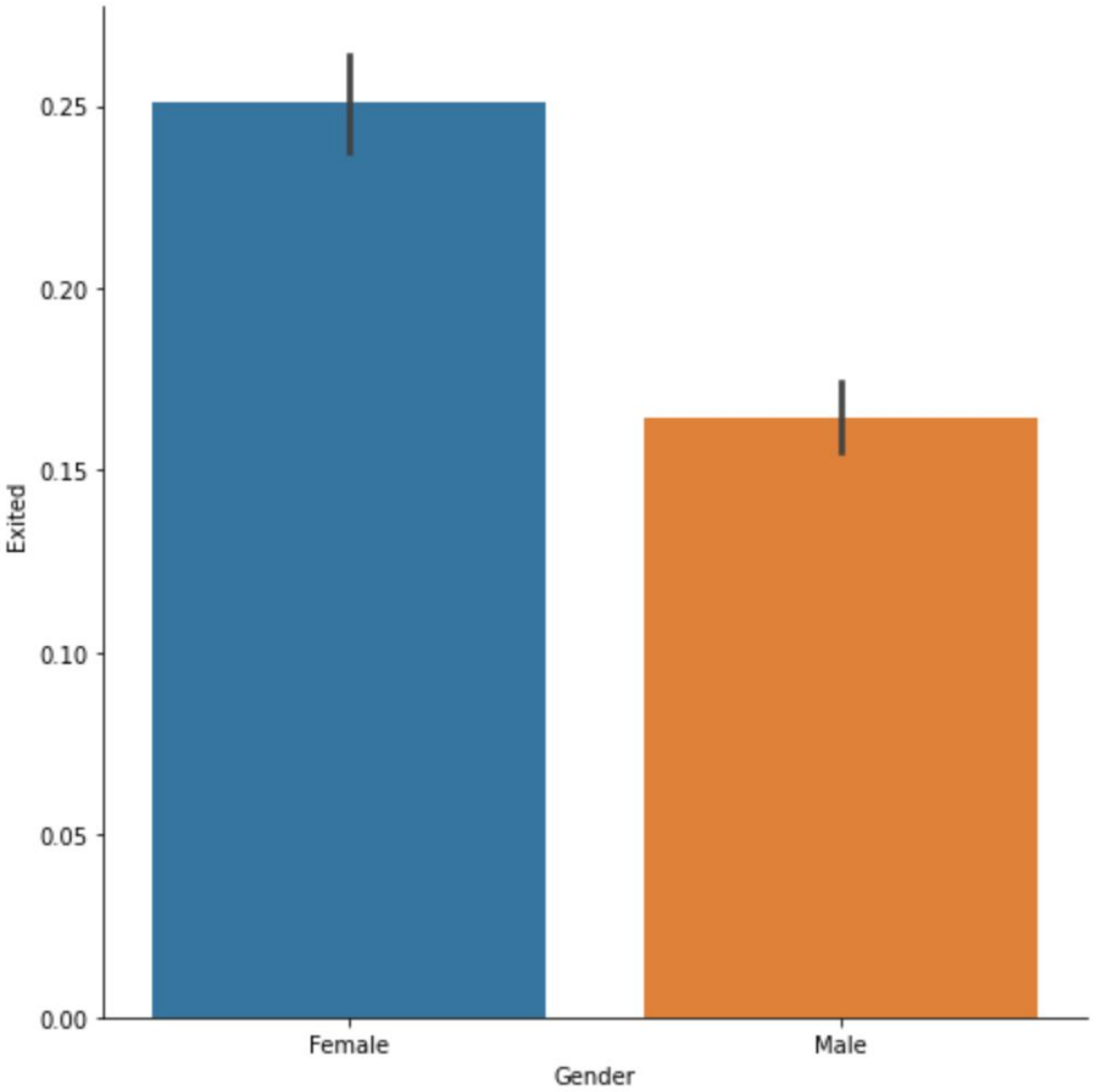


We see a negatively skewed bimodal distribution for credit score, with a relatively high density of perfect scores.

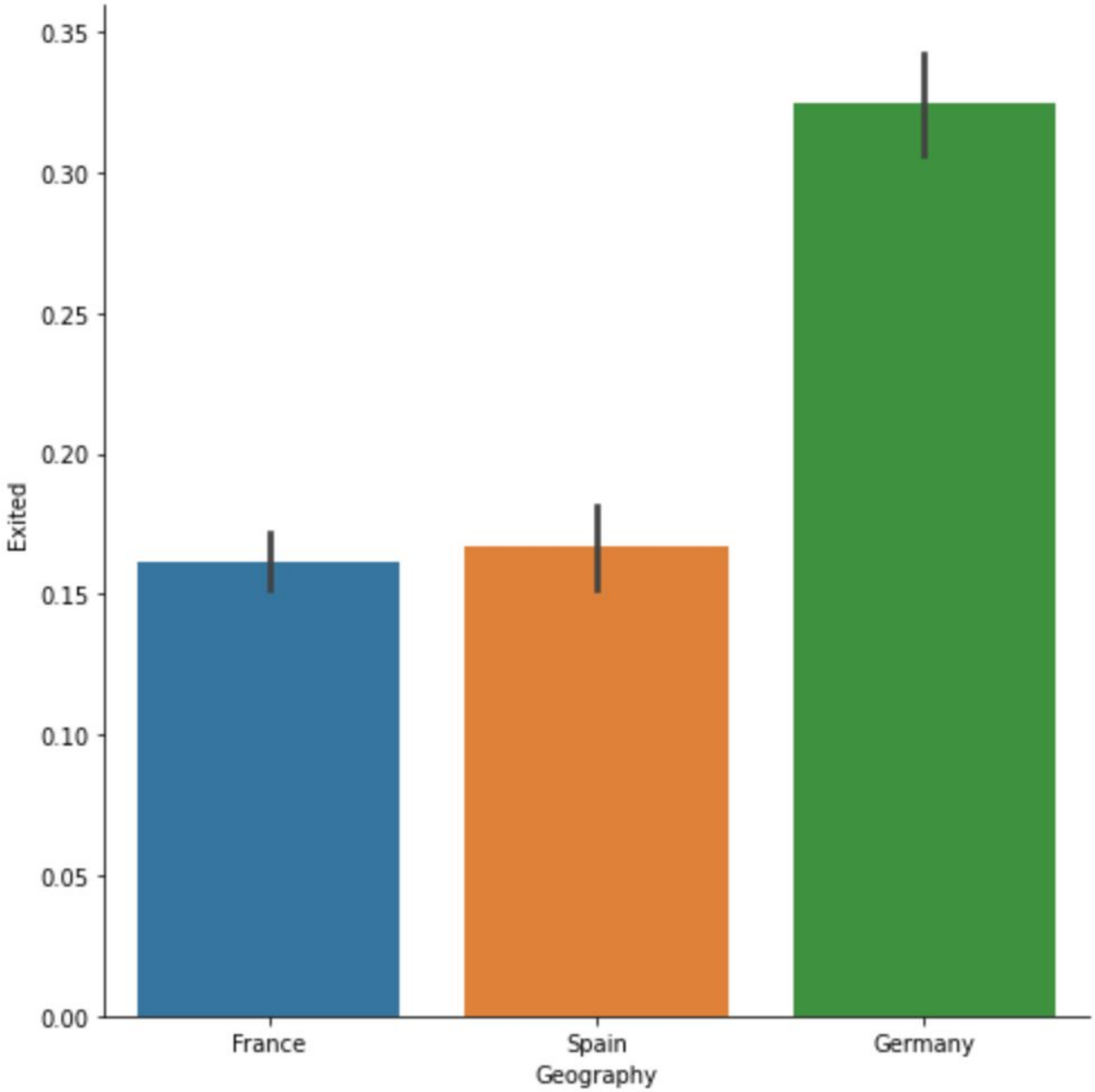


Lastly, we see a totally uniform distribution for estimated salary.

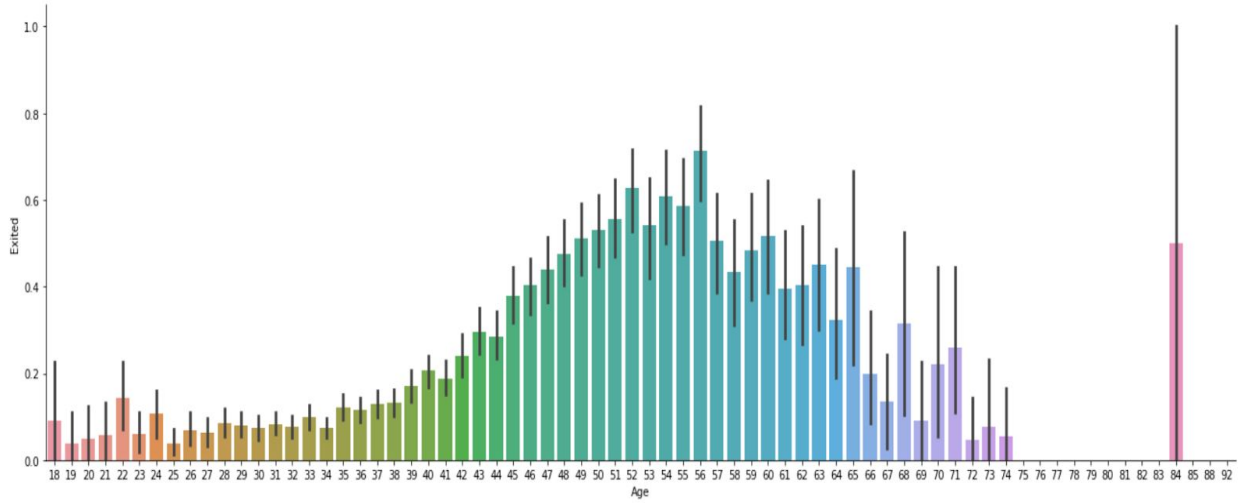
We see an interesting result looking at a categorical plot of gender with the target:



Could this mean that women are more frequent churners?

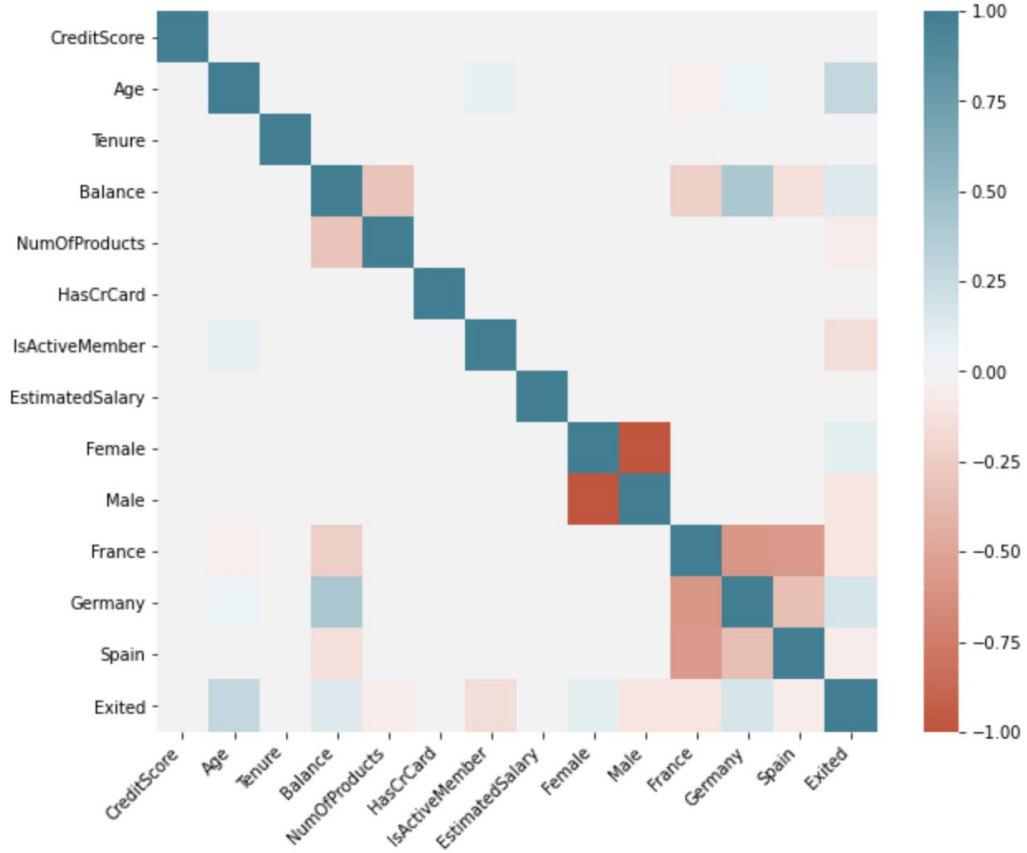


It seems that people in Germany are more likely to churn for some reason.



From this categorical plot we can see that age has a clear relationship with the propensity to churn, with the center of the distribution in the upper 50's. We can expect that age will be an important feature in many of our models.

We can also visualize how important features might be by looking at a correlation matrix which will tell us the correlation of each feature with the target.



From this we can see that age, being an active member, number of products, gender, country, and balance are all somewhat correlated with churn.

IV. Modeling

In order to prevent overfitting and ensure that the models would be robust and generalizable, a cross validation technique centered around ShuffleSplit was employed. Using a preset train/test split ratio, the data was randomly split ten times for each model, the model was retrained on the new training set after each split, and an accuracy score was computed for both the train and the test sets. The average of these scores was used to give a generalized accuracy score which displayed the consistency of model performance. At first a 70/30 train test split was implemented, but this led to issues with overfitting, and ultimately the models were able to be trained optimally with a 50/50 split.

Output of cross validation code:

Number of random splits: 10

Classifier: LogisticRegression()
Train/test split percentage: 50/50
Average train accuracy: 79.05%
Average test accuracy: 79.1%
Variance in train scores: 1.016039999999998e-05
Variance in test scores: 1.892639999999997e-05

Classifier: KNeighborsClassifier(n_neighbors=4)
Train/test split percentage: 50/50
Average train accuracy: 81.55%
Average test accuracy: 78.03%
Variance in train scores: 1.414440000000003e-05
Variance in test scores: 8.72960000000005e-06

Classifier: SVC()
Train/test split percentage: 50/50
Average train accuracy: 79.59%
Average test accuracy: 79.67%
Variance in train scores: 8.617999999999945e-06
Variance in test scores: 8.618000000000096e-06

Classifier: GaussianNB()
Train/test split percentage: 50/50
Average train accuracy: 78.55%
Average test accuracy: 78.53%
Variance in train scores: 3.3860000000000024e-06
Variance in test scores: 1.559359999999992e-05

Classifier: DecisionTreeClassifier(max_depth=5)
Train/test split percentage: 50/50
Average train accuracy: 86.07%
Average test accuracy: 85.39%

Variance in train scores: 1.517759999999985e-05
Variance in test scores: 1.4360399999999987e-05

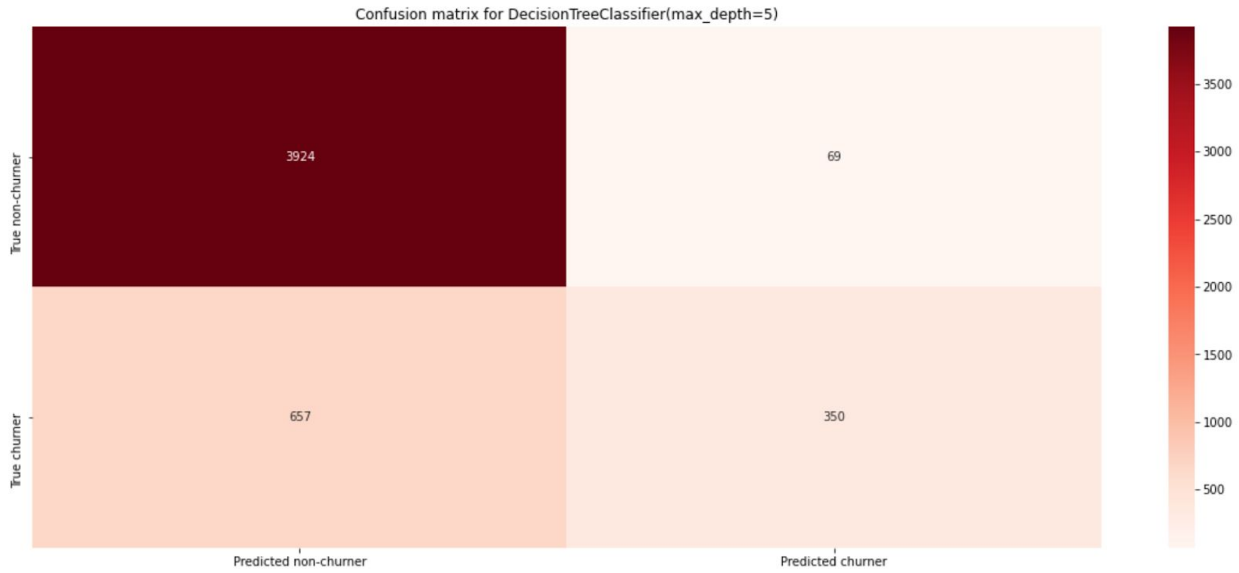
Classifier: RandomForestClassifier(max_depth=6)
Train/test split percentage: 50/50
Average train accuracy: 86.67%
Average test accuracy: 85.25%
Variance in train scores: 7.3856000000000085e-06
Variance in test scores: 9.961600000000064e-06

Classifier: AdaBoostClassifier()
Train/test split percentage: 50/50
Average train accuracy: 86.2%
Average test accuracy: 85.27%
Variance in train scores: 1.9280000000000006e-06
Variance in test scores: 1.4921600000000125e-05

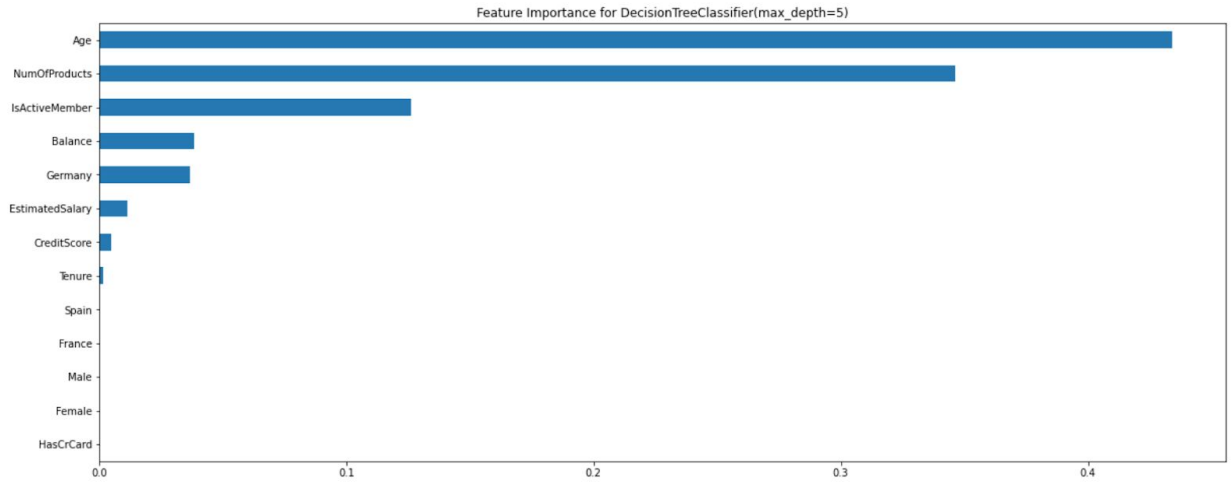
Hyperparameters were tuned by hand which wasn't burdensome due to the simple nature of the models that were employed. The best performers were chosen as the finalists, and the optimal train/test split of 50/50 was employed before a final fit was performed for model evaluation.

Model 1: Decision Tree Classifier

The first model used was a decision tree algorithm. Maximum depth was chosen to be 5 which minimized overfitting and led to the highest accuracy score. A final accuracy score of 85.48% was achieved for the test set, with a score of 85.84% for the train set, showing that there wasn't overfitting. A confusion matrix is shown below.



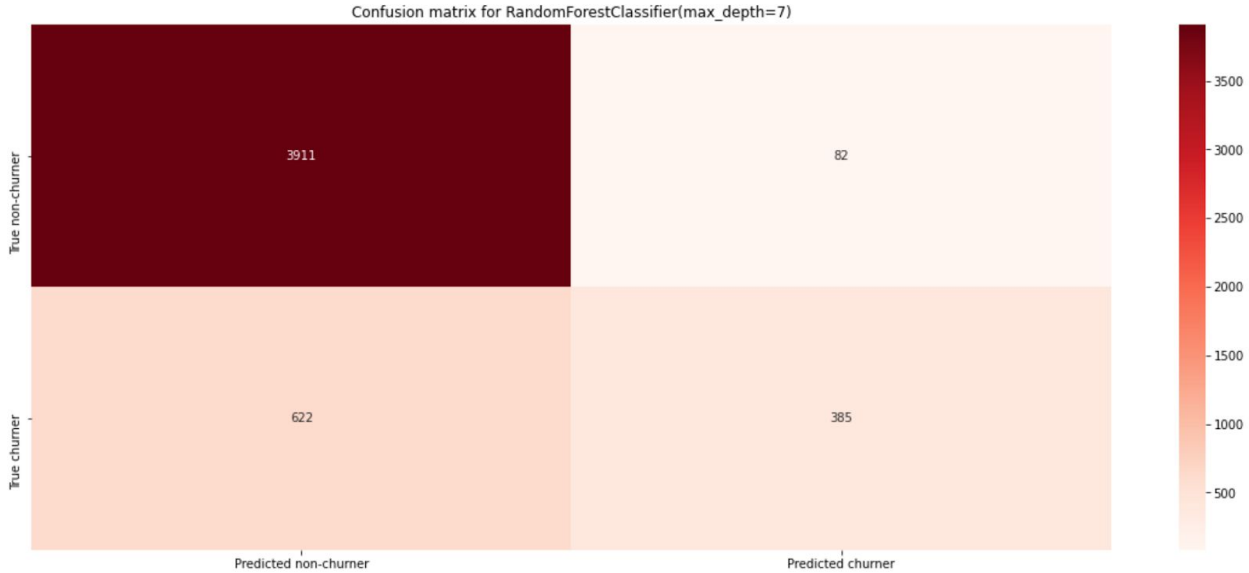
We can see that the model tends to predict that most people are non-churners, and therefore it predicted that 657 true churners were non-churners, and only got 350 true churners correct. If we wanted to penalize missing out on catching a true churner more strictly, additional regularization could likely be employed to punish miscategorization of customers in this group.



Here we have a figure depicting feature importance for the decision tree classifier. As one might have suspected from our exploratory data analysis, age is the most important feature by far, followed closely by the number of products. Active status also plays a key role in determining someone's likelihood to churn. We can see that the decision tree did not use location or gender to arrive at any of its predictions, which may be a surprise given that both of these features were somewhat correlated with the target.

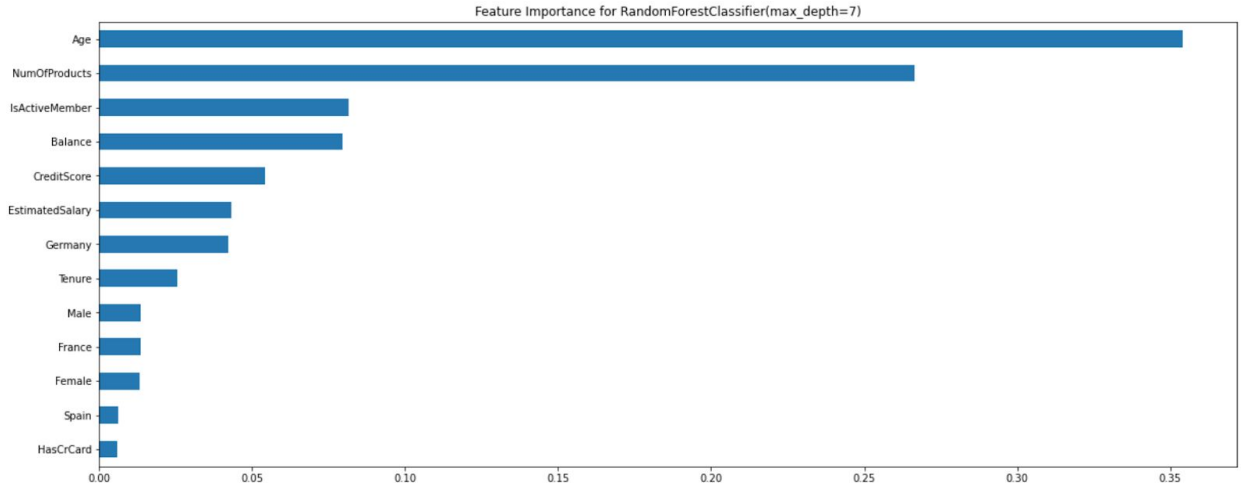
Model 2: Random Forest

The best performer of the final three models was a random forest. Maximum depth was chosen to be 7 this time. A final accuracy score of 85.92% was achieved for the test set, with a score of 87.86% for the train set. Less consistency between these two scores shows a potential for overfitting that may need to be watched closely, but in this case the model was still generalizable. A confusion matrix is once again shown below.



We can once again see that the model over-predicts that customers are non-churners. A positive aspect of this is the fact that only 82 of the 5000 customers in the test set were miscategorized as

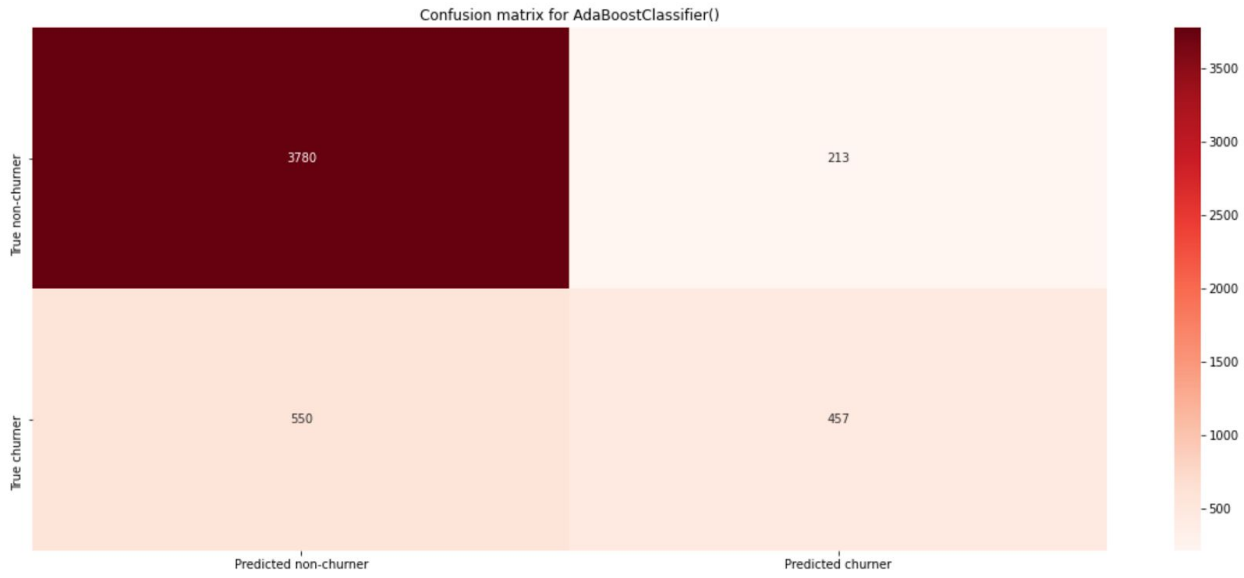
churners. Advanced regularization techniques could still be employed to increase the number of churner categorizations. A figure depicting feature importance is shown below.



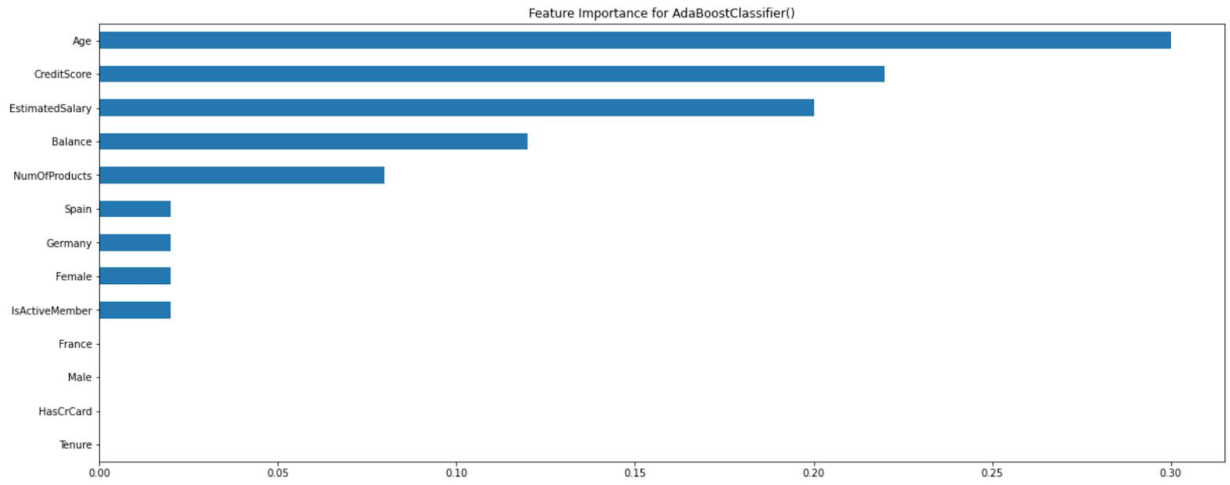
This time all features were used unlike in the decision tree. We can also see balance playing more of a role, which means that the amalgamation of the 100 (the default setting) trees in our forest led to that feature becoming almost as important as active member status.

Model 3: AdaBoost

The final model used was adaboost. Default parameters led to the best results, and the accuracy scores were on par with the tree-based approaches. Final accuracy scores of 84.74% for the train set and 86.08% for the test set were recorded, making AdaBoost the worst performing model in terms of test accuracy.



The confusion matrix shows that AdaBoost performed slightly better than the trees at classifying true churners approaching a 50/50 ratio of incorrect/correct categorizations of that group. This came at the cost of miscategorizing a higher proportion of the non-churner.



Once again age dominates feature importance, but here we diverge from the features selected by the tree-based approaches. The second and third most important features for AdaBoost were credit score and salary, with active status and number of products shunted towards the bottom of the list. It is possible that the gradient boosting nature of this algorithm led to the difference in feature importance, and this could be investigated by implementing a gradient boosted tree-based algorithm like XGBoost and comparing feature importance. Either way, AdaBoost was able to achieve respectable accuracy using very different feature weights than those of the trees.

V. Appendix

Github Repository**:**

https://github.com/fall2020-intro-ml-apps/final-project-jwebster47

Kaggle Notebook Links**:**

https://www.kaggle.com/snanilim/churn-deep-learning-vs-xgboost/comments
https://www.kaggle.com/berkanacar/churn-prediction-by-selecting-from-11-tuned-models
https://www.kaggle.com/akdagmelih/classification-model-comparison-banks-customers