Predicting Customer Churn with Machine Learning:

Comparing Three Classification Models' Ability to Predict Customer Churn at a Bank

Avi Reissberg

Contents

Problem of Customer Churn

- **All the time businesses face the issue of customer churn**
- **Metrics like churn rate and retention rate are tracked and can be frequently reported by business intelligence**
- **Churn is generally considered to be quite costly, as research shows the net profit of retaining a customer is often significantly higher than gaining a brand new one**

What to Do?

- **Nothing/can’t be helped? One-size-fits-all solution/hope it works?**
- **No, instead businesses can attempt to predict whether a customer is likely to churn, and perform some intervention if they are**
- **They can do this with Machine Learning!**
- **But which methods are best?**
- **In the following slides we will look at 3 distinct classification models, and compare their performance, advantages and disadvantages**

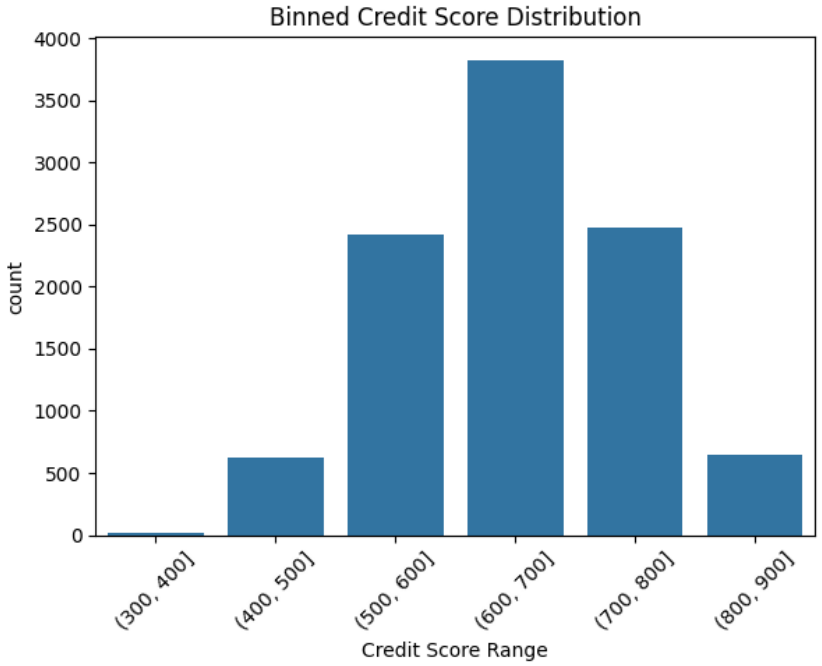
Overview of Important Features

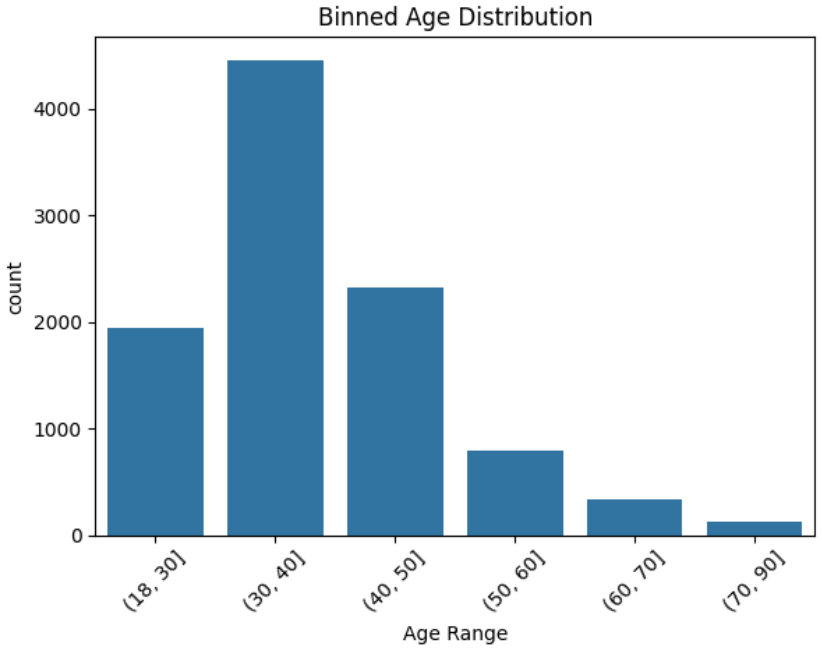
Column	Data Type	Note
CreditScore	Numerical	
Geography	Categorical	FR, DE, SP
Gender	Categorical	Male or Female
Age	Numerical	18-92
Tenure	Numerical	0-10
Balance	Numerical	
NumOfProducts	Numerical	1-4
HasCrCard	Binary	
IsActive	Binary	
Salary	Numerical	
Exited (Churned)	Binary	What we want to predict

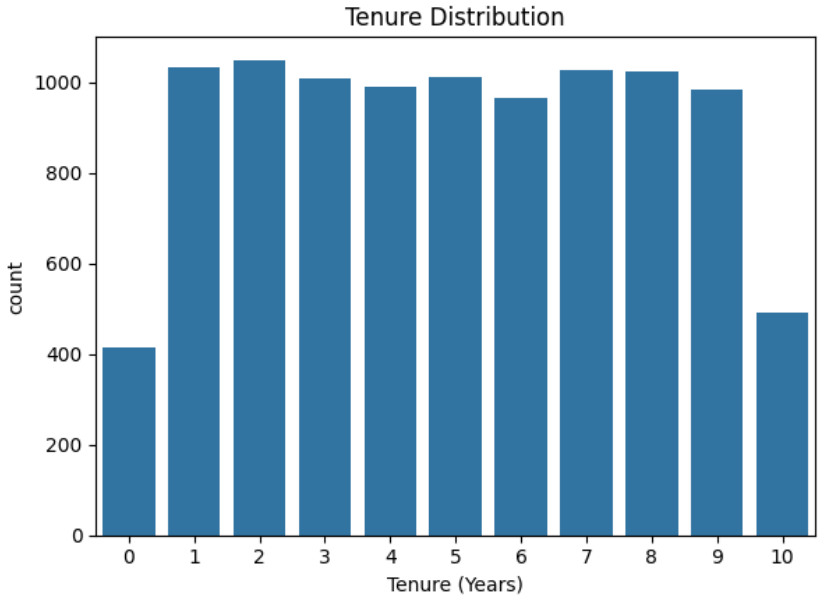
Not Exited	7,963
Exited	2,037

Uh oh! Above you can see that there are many more customers that didn't churn, than did churn. At a roughly 80/20 split this is what is considered a relatively bad class imbalance. The issue is the models will struggle to identify the class that is significantly underrepresented, and likely will report misleadingly good results if not addressed because a correct classification of not churned by chance is much more likely. As we'll see later, this is something we will correct for when building out our models.

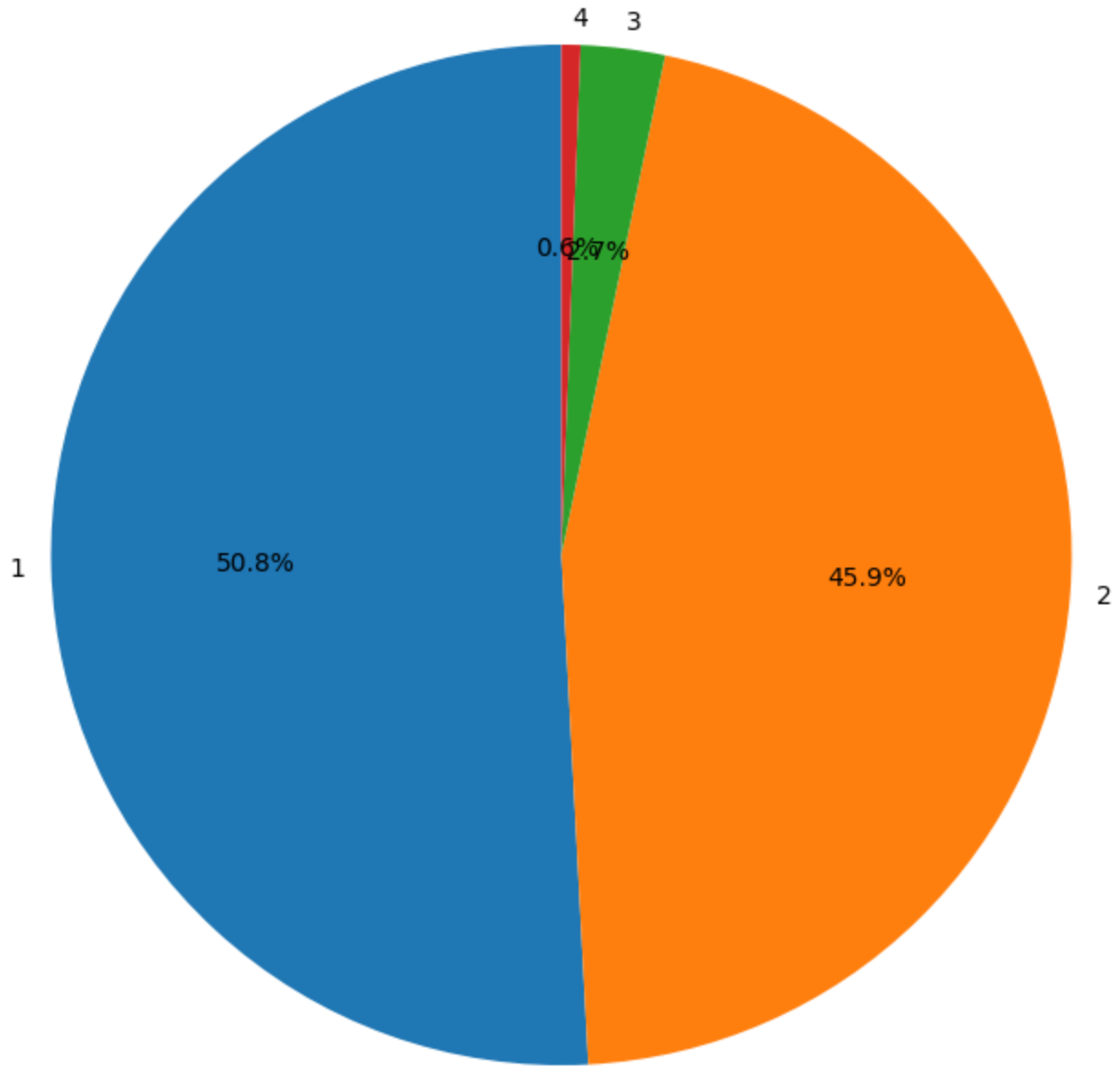
Feature Distributions



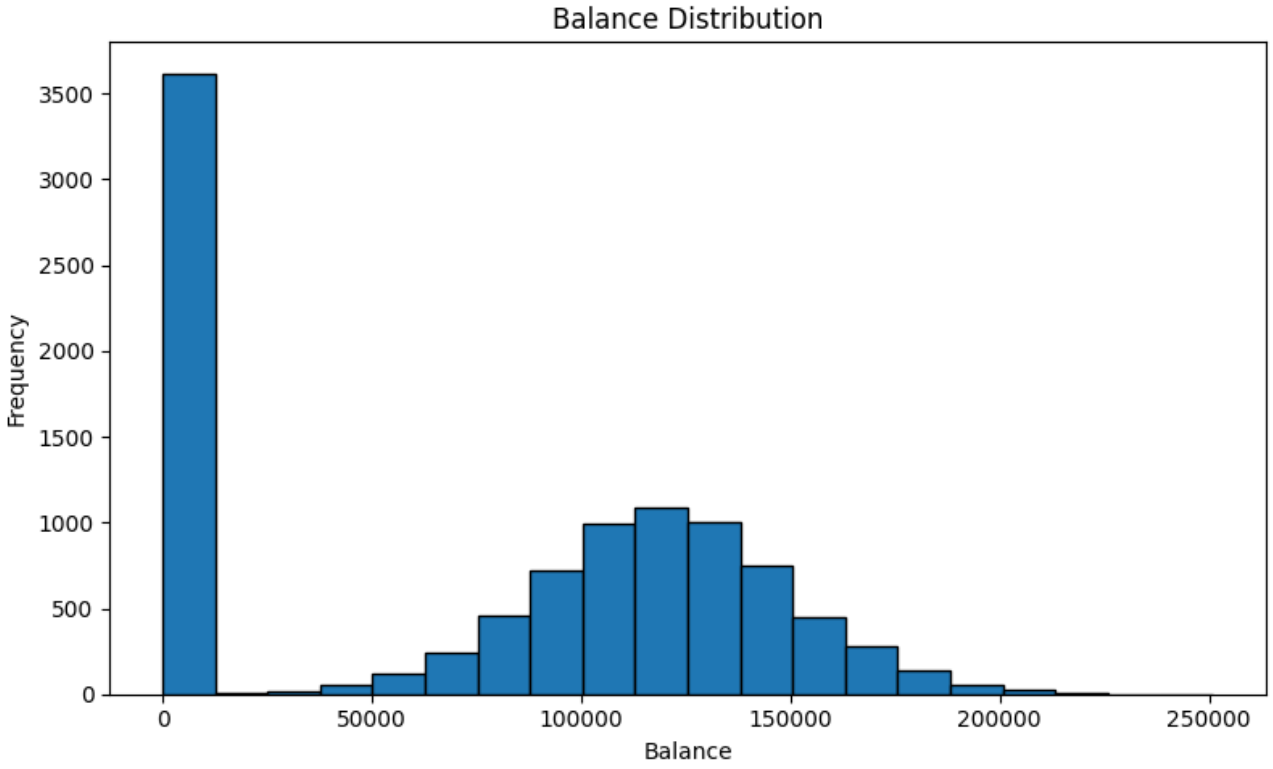


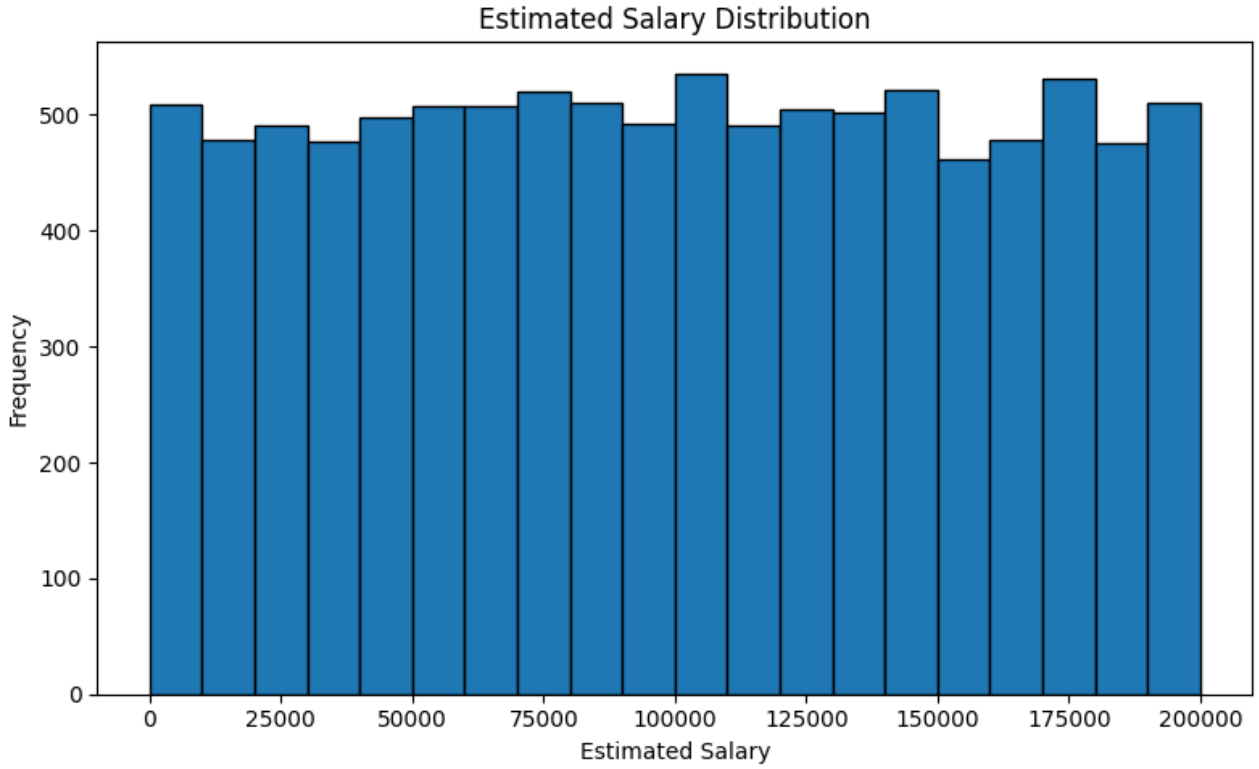


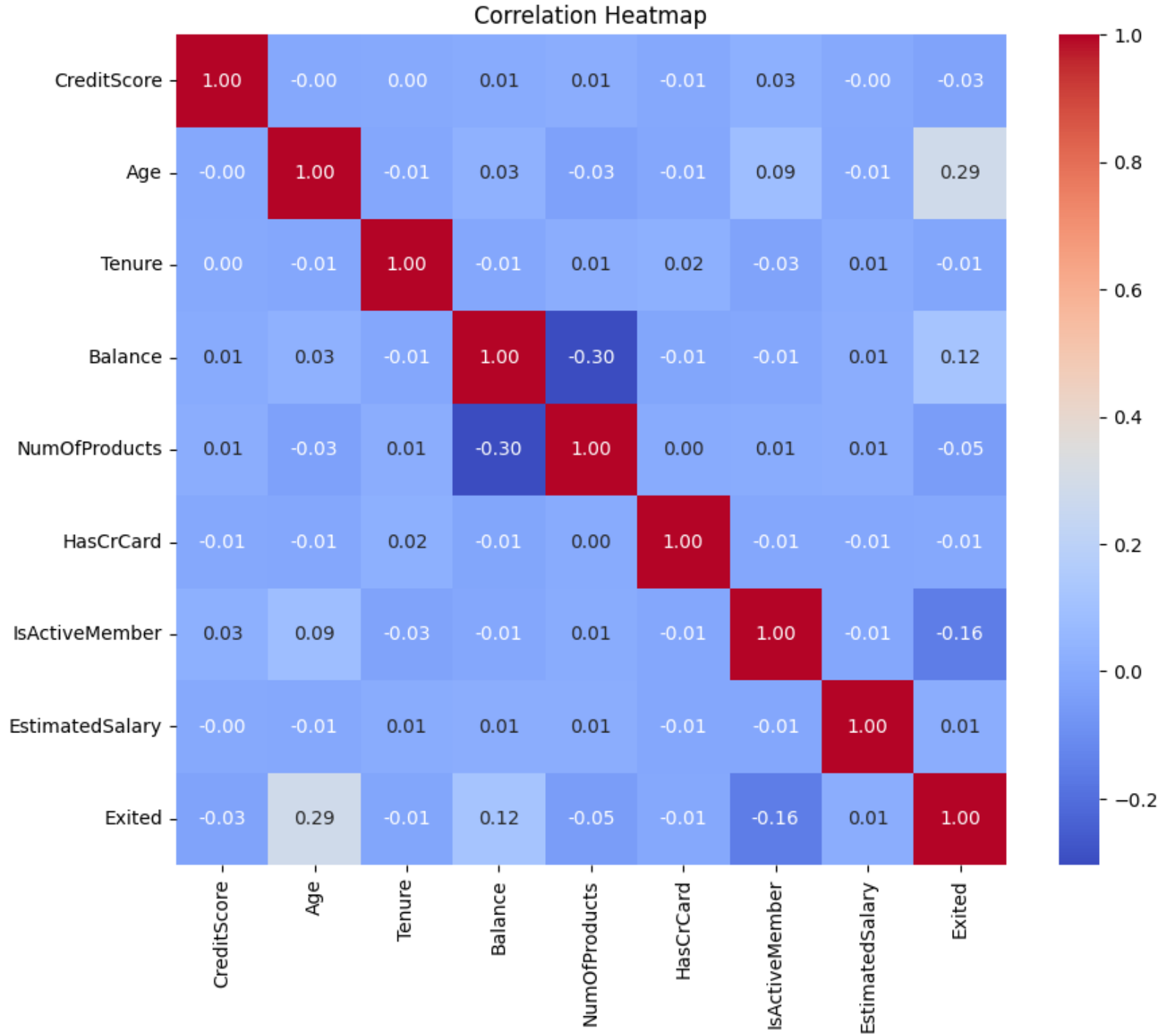
Number of Products Purchased



More Feature Distributions



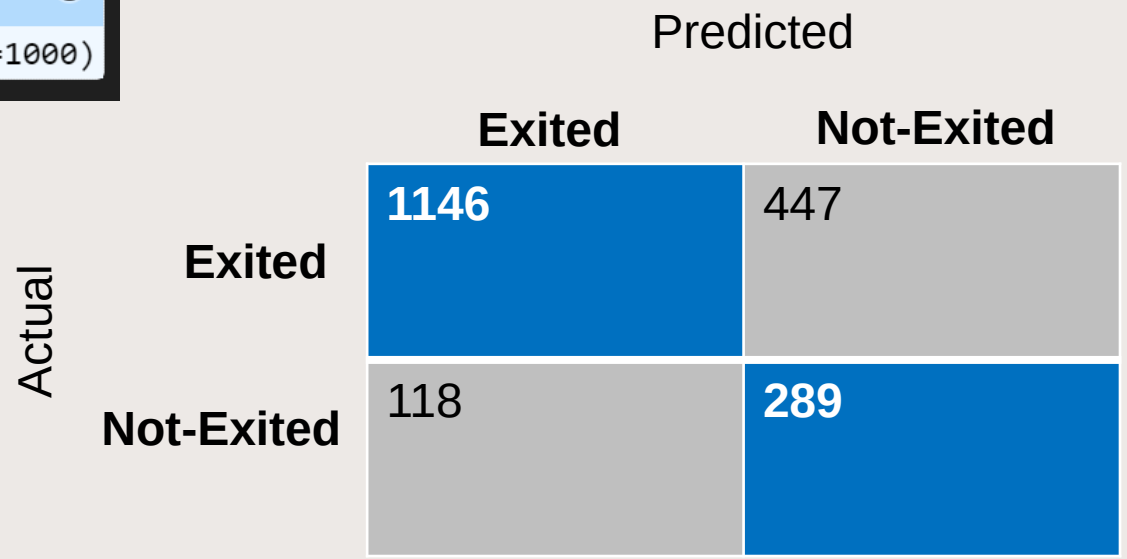



Correlation Heatmap
CreditScore 1.00 -0.00 0.00 0.01 0.01 -0.01 0.03 -0.00 -0.03
Age -0.00 1.00 -0.01 0.03 -0.03 -0.01 0.09 -0.01 0.29
Tenure 0.00 -0.01 1.00 -0.01 0.01 0.02 -0.03 0.01 -0.01
Balance 0.01 0.03 -0.01 1.00 -0.30 -0.01 -0.01 0.01 0.12
NumOfProducts 0.01 -0.03 0.01 -0.30 1.00 0.00 0.01 0.01 -0.05
HasCrCard -0.01 -0.01 0.02 -0.01 0.00 1.00 -0.01 -0.01 -0.01
IsActiveMember 0.03 0.09 -0.03 -0.01 0.01 -0.01 1.00 -0.01 -0.16
EstimatedSalary -0.00 -0.01 0.01 0.01 0.01 -0.01 -0.01 1.00 0.01
Exited -0.03 0.29 -0.01 0.12 -0.05 -0.01 -0.16 0.01 1.00
CreditScore
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Exited
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

LOGISTIC REGRESSION

```
LogisticRegression
LogisticRegression(C=0.01, class_weight='balanced', max_iter=1000)
```

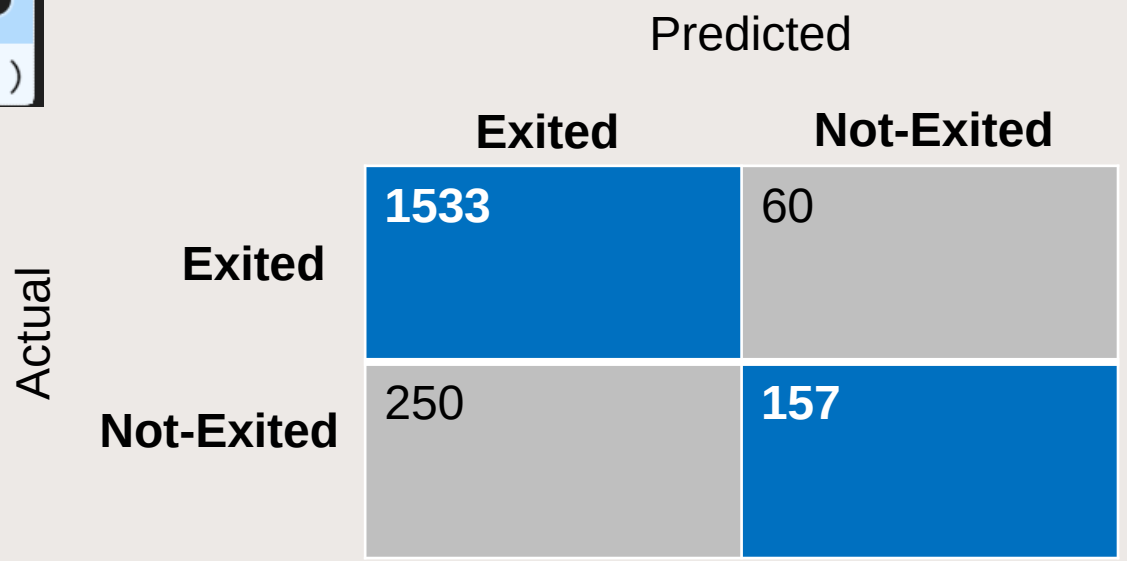
- Balanced class weights to correct for the imbalance
- Stratified sampling to preserve class ratio
- Inverse learning rate of C=0.01
- L2 (Ridge) Regularization
- Solver: Limited-Memory Broyden-Fletcher-Goldfarb-Shanno



K-NEAREST NEIGHBORS

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=9, weights='distance')
```

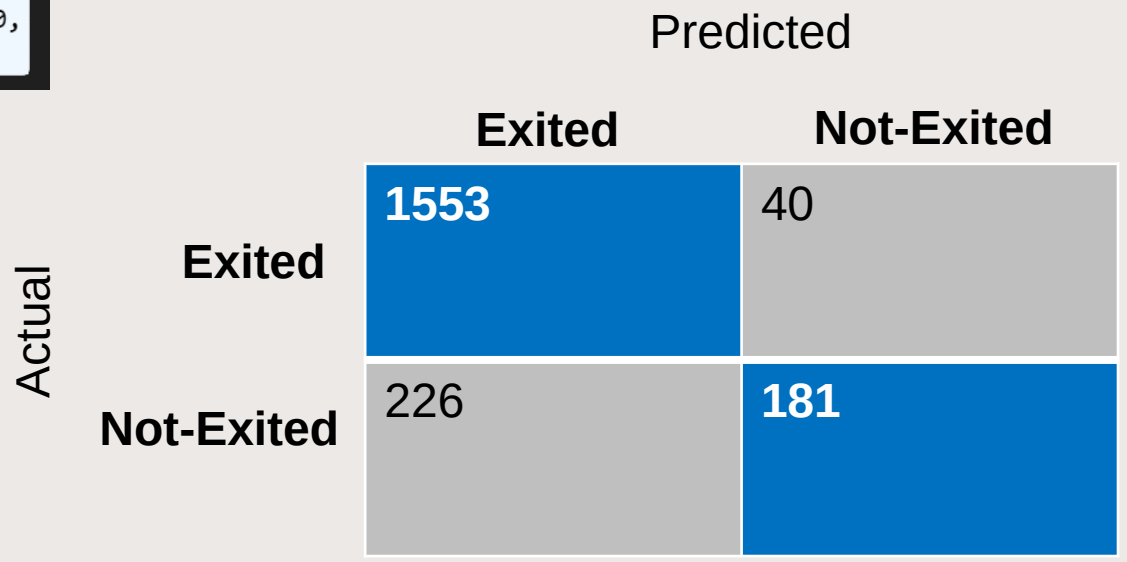
- Number of Neighbors: 9
- Neighbors weighted by distance
- L2 Norm (Euclidean Distance)



RANDOM FOREST ENSEMBLE

```
RandomForestClassifier
RandomForestClassifier(max_depth=10, min_samples_split=5, n_estimators=200,
                       random_state=42)
```

- Number of trees: 200
- Max depth of trees: 10 leaves
- Minimum samples to split on: 5
- No weights applied to classes



Model Comparison Summary

Model	Precision (Churn)	Recall (Churn)	F1 (Chrn)	Notes
Logistic Regression	0,393	0,710	0,506	Casts a wide net – catches many churners but misclassifies non–churners more often
K–NN	0,724	0,386	0,503	Selective and precise – avoids false alarms but misses many actual churners
Random Forest	0,819	0,445	0,576	Best all–around– strong balance between precision and recall, best F1 score

Final Recommendations

Random Forest or Logistic Regression

- We make a practical decision about whether we view false positives or false negatives as more costly/least desirable
- Most customers will likely have some non-zero probability of churning, therefore interventions such as targeted marketing, offering discounts, rewarding loyalty etc. won't be completely wasted on those we predict as non-churners.
- On the other hand, the cost of missing likely churners is high.
- Logistic Regression is preferable to K-NN as it is much less reluctant to make a positive classification and comes with the added benefit of perhaps the best interpretability.
- However, Random Forest performed the best overall.
- If we applied further techniques such as gradient boosting, we might see it outperform the other two on all key metrics
- If interpretability is deemed highly important, logistic regression could be chosen.
- Fortunately, good interpretability is not out of reach for Random Forest utilizing Shapley Additive Explanations (SHAP)

THANK YOU