Credit Card Fraud

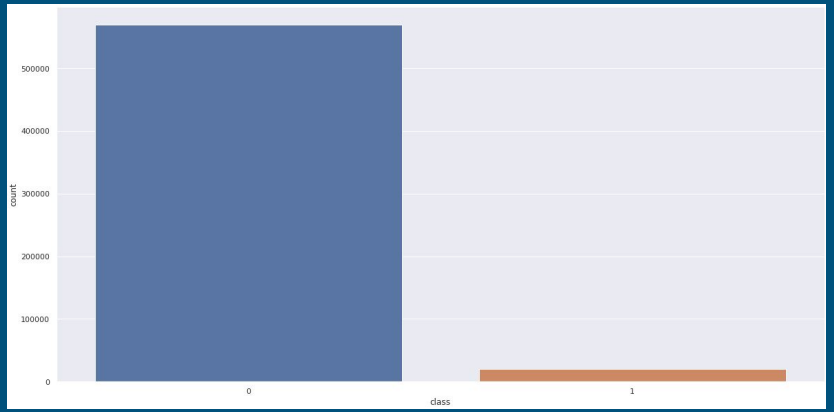
Kevin Gui, Steven Ha

Intro

Our Overarching Idea: Detect Fraudulent Credit Card Transactions

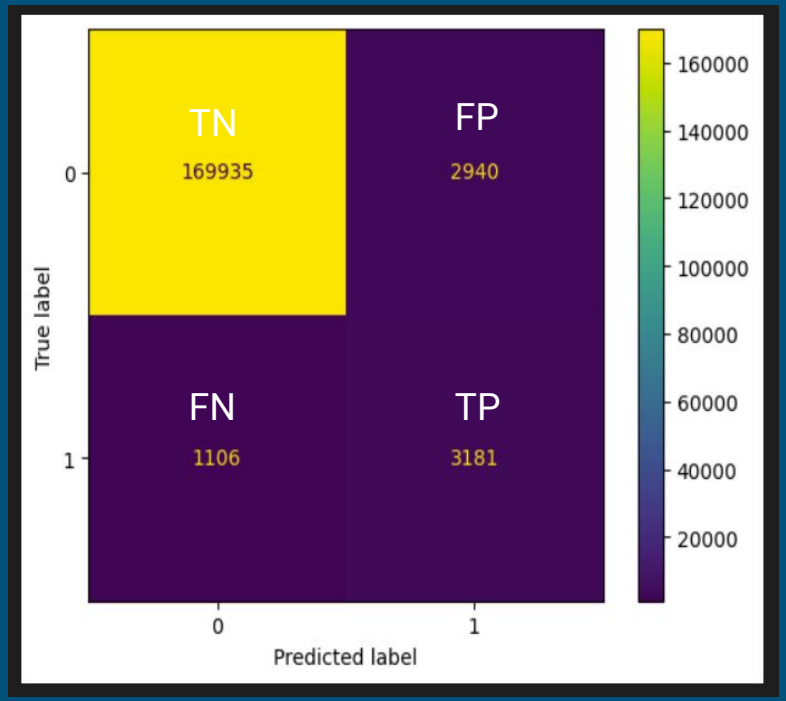
The data is very imbalanced

- Finding the best ML model
- How we improved our model
- Conclusions and Results



Model Evaluation and Metrics

Confusion Matrix



Accuracy: $\frac{TP + TN}{TP + TN + FP + FN}$

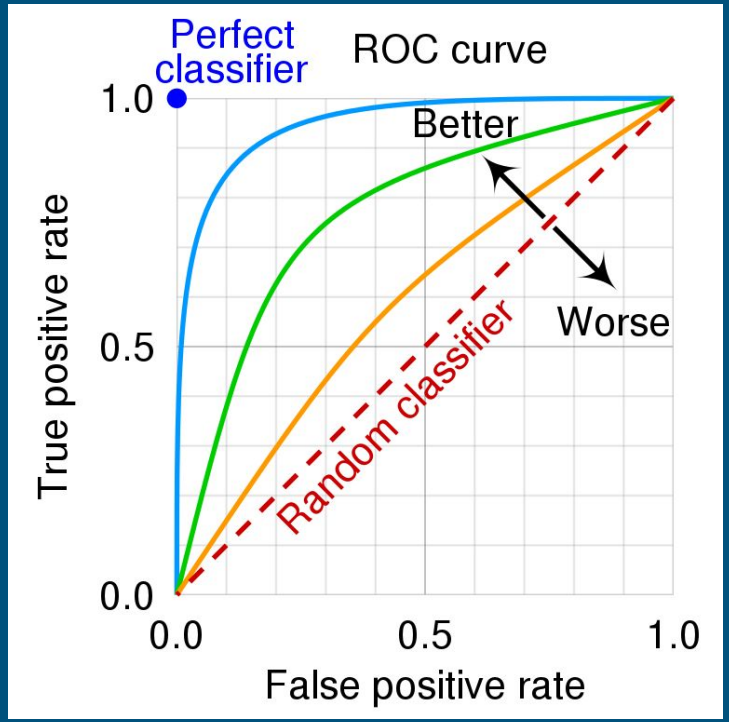
Precision: $\frac{TP}{TP+FP}$

Recall: $\frac{TP}{TP+FN}$

F1 Score: $2 \times \frac{Precision \times Recall}{Precision + Recall}$

AUC/ROC Curve

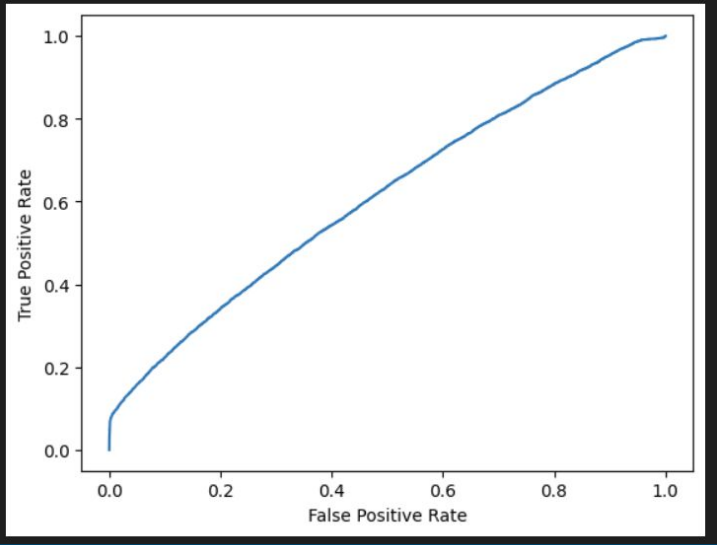
- Plots TP vs. FP
- Higher AUC score = more accurate model



Model 1

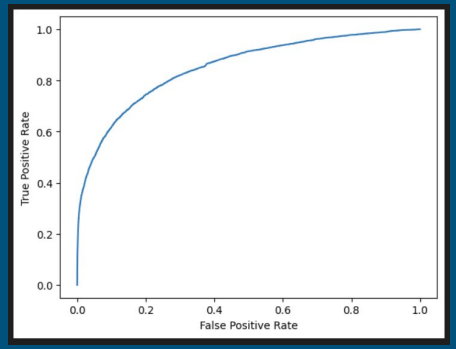
Logistic Regression

```
Accuracy Score :  0.9663415405109448

AUC Score :  0.6107526375093872

Confusion Matrix :
[[170884    157]
[  5806    315]]

Classification Report :
              precision    recall  f1-score   support

       False       0.97      1.00      0.98    171041
        True       0.67      0.05      0.10      6121
```



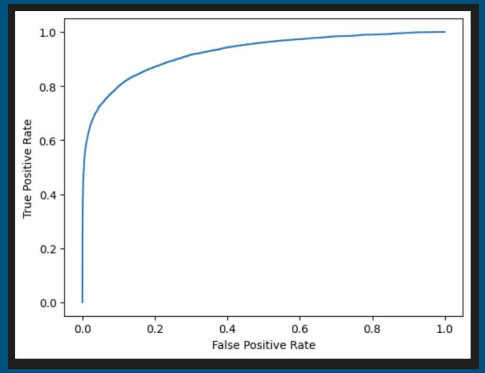
Models 2/3: Boosting

AdaBoost




```
Accuracy Score :  0.970484641175873

AUC Score :  0.8517222221643287

Confusion Matrix :
[[170626    415]
[  4814   1307]]

Classification Report :
              precision    recall  f1-score   support

       False       0.97      1.00      0.98    171041
        True       0.76      0.21      0.33      6121
```

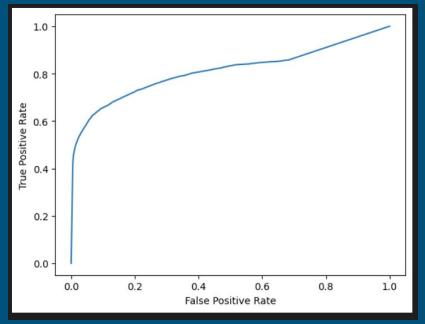
XGBoost




```
Accuracy Score :  0.9786579514794369

AUC Score :  0.9242500893514191

Confusion Matrix :
[[170676    365]
[  3416   2705]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      1.00      0.99    171041
        True       0.88      0.44      0.59      6121
```

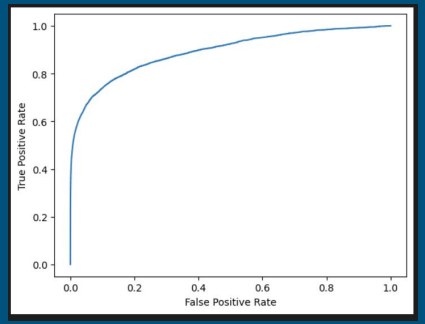
Models 4/5: Trees

Decision Tree Classifier



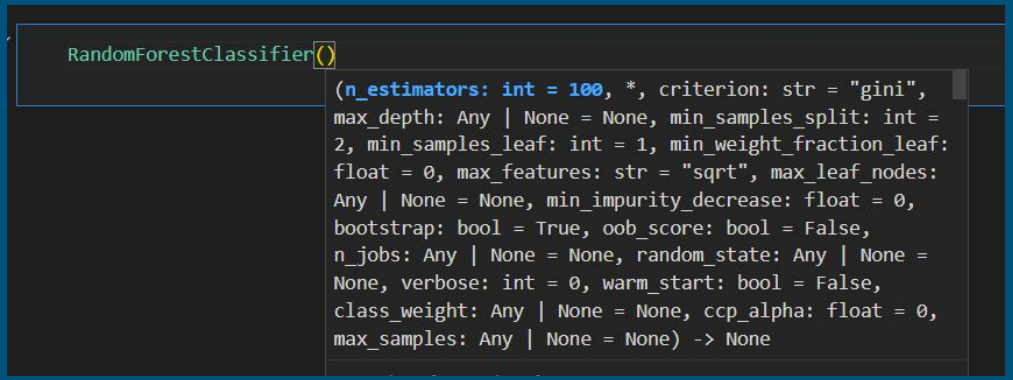

```
Accuracy Score :  0.974627741840801

AUC Score :  0.8109088713848963

Confusion Matrix :
[[169996   1045]
[  3450   2671]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      0.99      0.99    171041
        True       0.72      0.44      0.54      6121
```

Random Forest




```
Accuracy Score :  0.9764001309535905

AUC Score :  0.8924669134548138

Confusion Matrix :
[[170862    179]
[  4002   2119]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      1.00      0.99    171041
        True       0.92      0.35      0.50      6121
```

Hyperparameter Tuning




```
# Parameters of Interest
criterion = ["gini", "entropy", "log_loss"]
bootstrap = [True, False]

# Grid Search
grid = dict(criterion=criterion, bootstrap=bootstrap)
grid_search = GridSearchCV(estimator=rfor, param_grid=grid, n_jobs=-1, cv=2, scoring='roc_auc')
grid_result = grid_search.fit(x_train, y_train)

print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
```

```
Best: 0.909344 using {'bootstrap': False, 'criterion': 'entropy'}
```

Results

Original Random Forest

```
Accuracy Score :  0.9764001309535905

AUC Score :  0.8924669134548138

Confusion Matrix :
[[170862    179]
[  4002   2119]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      1.00      0.99    171041
        True       0.92      0.35      0.50      6121
```

Tuned Random Forest

```
Accuracy Score :  0.9777378896151545

AUC Score :  0.9138097188178325

Confusion Matrix :
[[170866    175]
[  3769   2352]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      1.00      0.99    171041
        True       0.93      0.38      0.54      6121
```

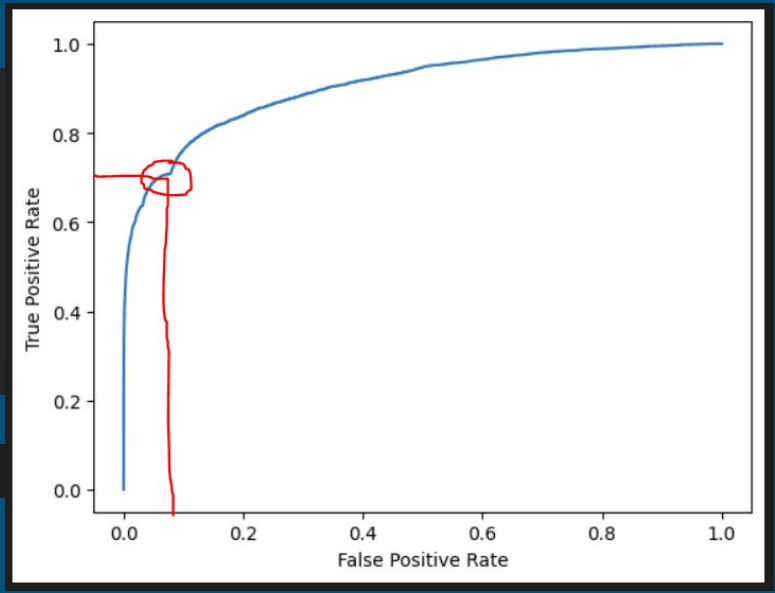
AUC Score: 0.89 -> 0.91

F1 Score: 0.50 -> 0.54

Thresholds

```python
def to_labels(pos_probs, threshold):
    return (pos_probs >= threshold)

# Define thresholds
thresholds = np.arange(0, 1, 0.01)
# Evaluate each threshold
scores = [f1_score(y_test, to_labels(y_prob_pred_rfor, t)) for t in thresholds]
# Get the best threshold
ix = np.argmax(np.array(scores))
print(f"The best f1 score is {scores[ix]} and the threshold is {thresholds[ix]}")
```

```
The best f1 score is 0.611260568793236 and the threshold is 0.13
```



Results

Tuned Random Forest (w/o thresholds)

```
Accuracy Score :  0.9777378896151545

AUC Score :  0.9138097188178325

Confusion Matrix :
[[170866    175]
[  3769   2352]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      1.00      0.99    171041
        True       0.93      0.38      0.54      6121
```

Tuned Random Forest (w/ thresholds)

```
Accuracy Score :  0.9771621453810636

AUC Score :  0.9071698020326076

Confusion Matrix :
[[169935   1106]
[  2940   3181]]

Classification Report :
              precision    recall  f1-score   support

       False       0.98      0.99      0.99    171041
        True       0.74      0.52      0.61      6121
```

AUC Score: 0.91 -> 0.90

F-1 Score: 0.54 -> 0.61