ANOMALY DETECTION

PROBLEM DESCRIPTION

Detect credit card fraud by learning against an existing set of transactions to determine future anomalies

APPROACHES

- Supervised Algorithms
  - K-Nearest Neighbors
  - Binary Classifying Linear Learner
  - XGBoost
- Unsupervised Algorithms
  - Random Cut Forest

APPROACHES (CONT)

- Divide and Conquer
  - Group members pursued different approaches
  - Sharing of feature engineering for better outcome comparisons

FEATURES

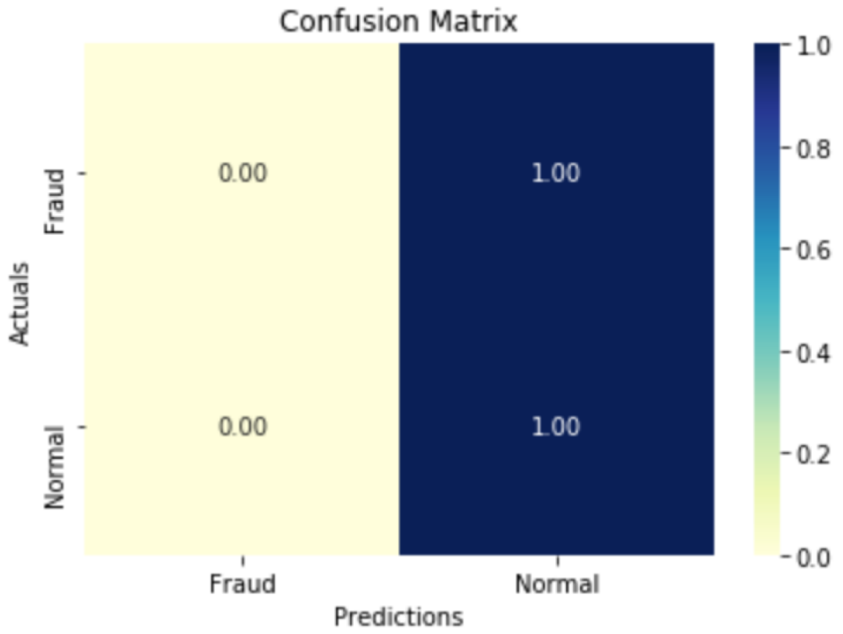
- Time representation
- transaction type
- amount
- origin
- old balance
- new balance
- destination
- destination old balance
- destination new balance
- whether or not the transaction is fraudulent
- whether other predefined rules marked the transaction as fraudulent

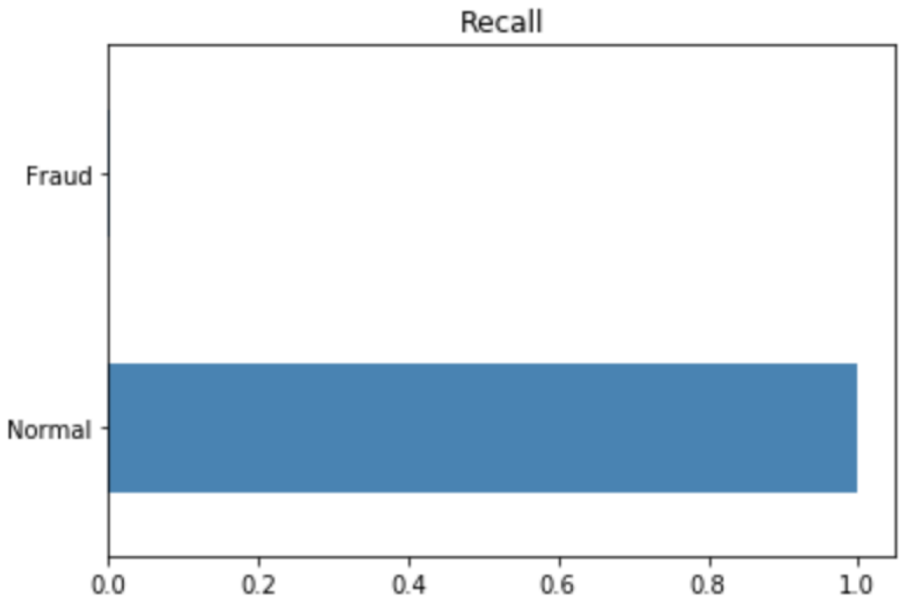
BASELINE

EXISTING COMPARISON

- Feature set flagging from a previous system
- Based on a set rule of ”illegal transactions of $200,000”
- The true metric to beat

```
Accuracy: 0.9987116942
F1 Score: 0.502
```

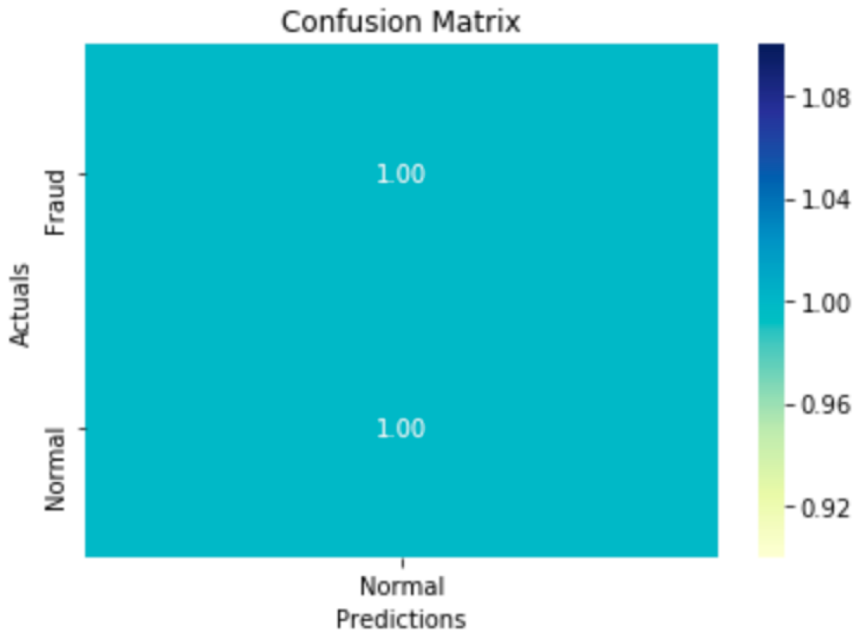


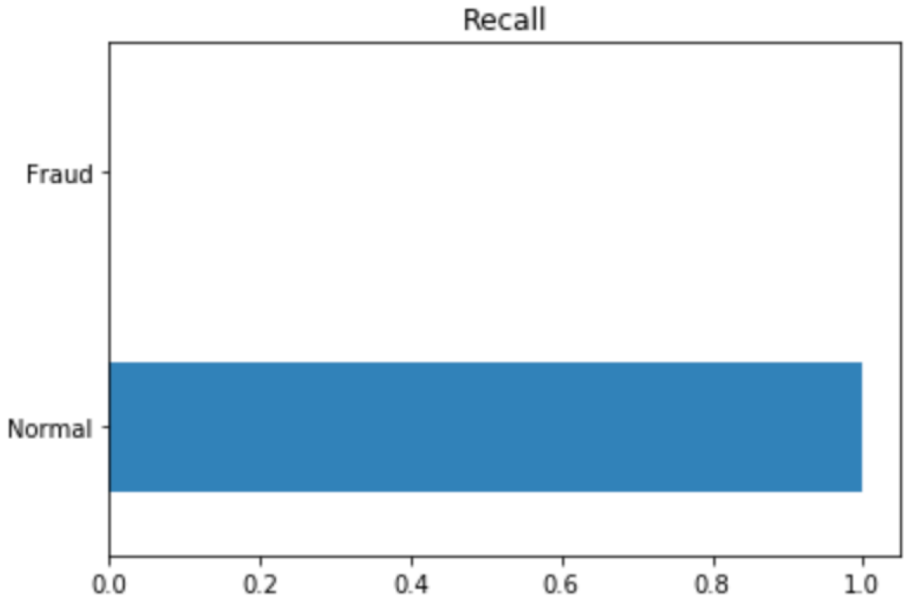


NAÏVE

- Assume nothing is fraud

```
Accuracy: 0.999
F1 Score: 0.500
```

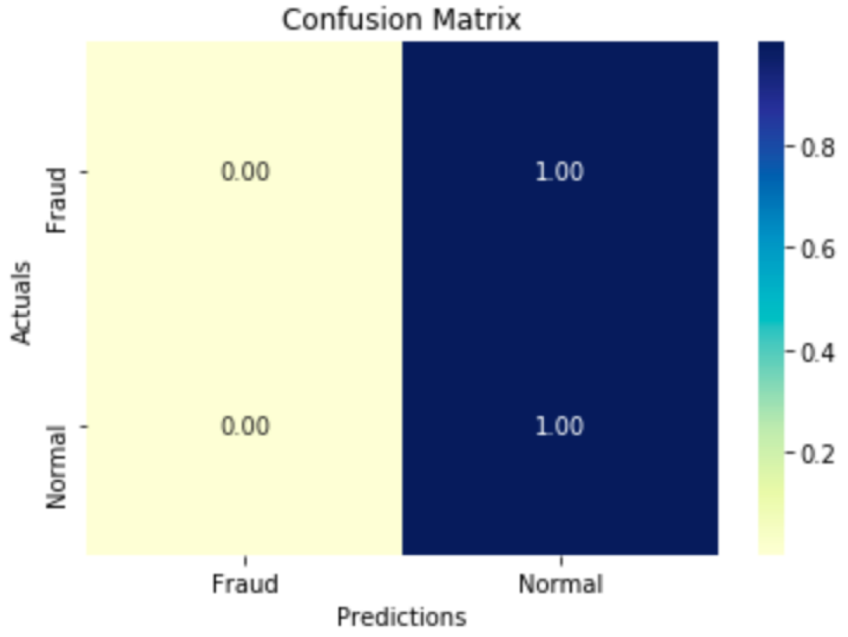


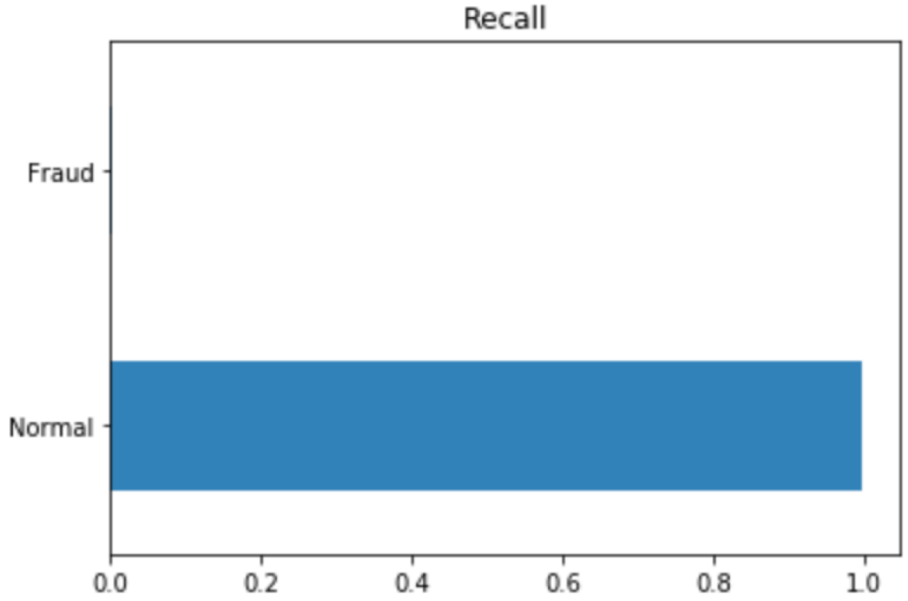


RANDOM

- Randomly generate the predicted label
- Probability based on occurrence of fraud in test set (1 in 775)

```
Accuracy: 0.997
F1 Score: 0.500
```



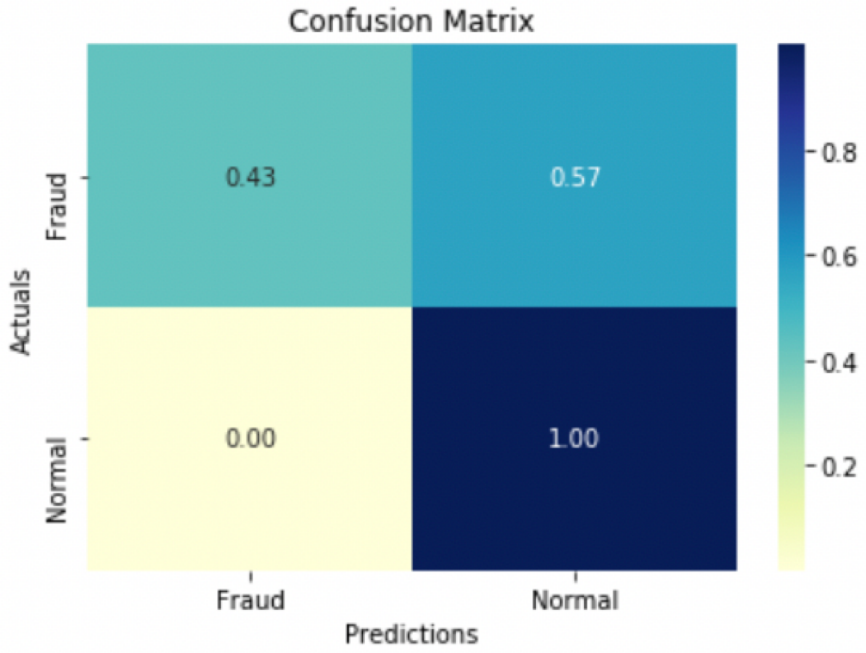


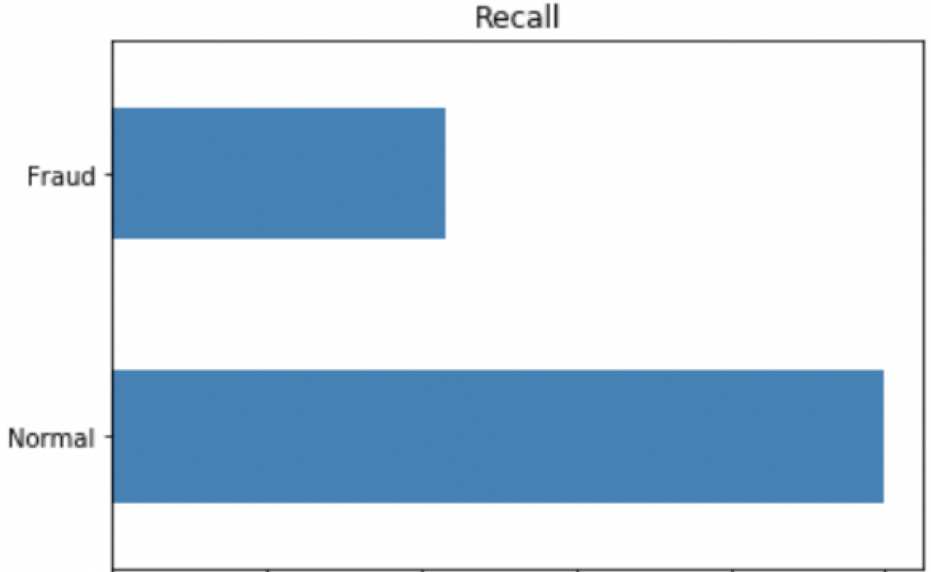
MODELING

K-NEAREST NEIGHBORS

- Flagged high-repeat destinations, one hot encoded type, dropped all other strings
- Trained a flat model and a clustered model
  For each, calculated F1 score across a range of k values (number of neighbors for majority comparison)
  For each, checked latency in calculation
  Confirmed that clustered model had similar metrics for accuracy, precision, and recall at much better latency
  Chose a k value corresponding to peak F1 for further use
- What I want now:
  Try(ing) similar approach with different feature set in hopes of even better outcomes.

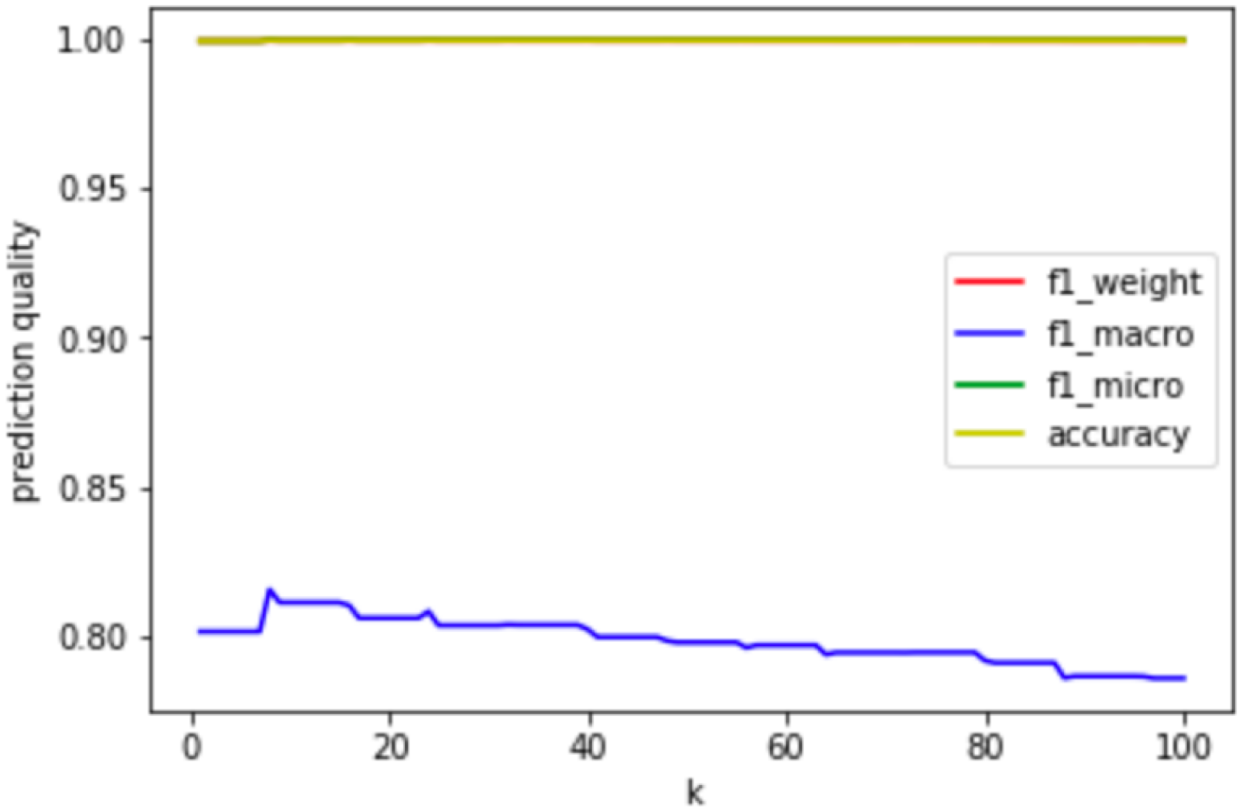
```
Accuracy: 0.999
F1 Score: 0.786
```





K-NEAREST NEIGHBORS

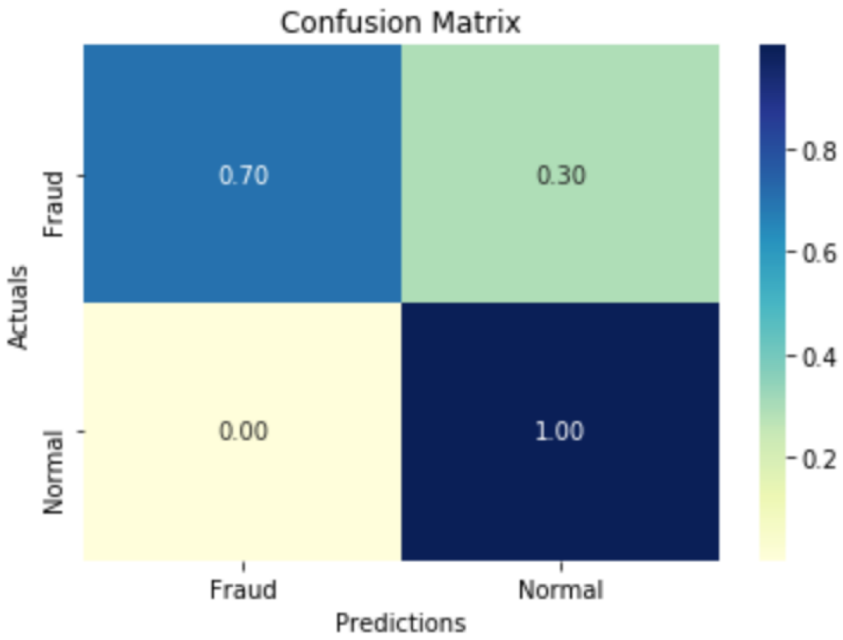
Quality of clustered model across different possible k values

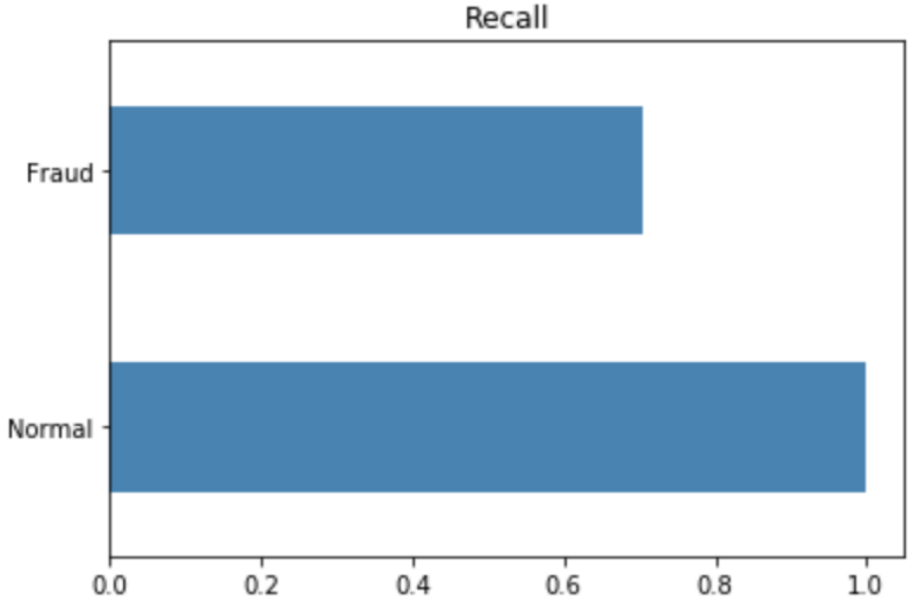


XGBOOST (BINARY:LOGISTIC)

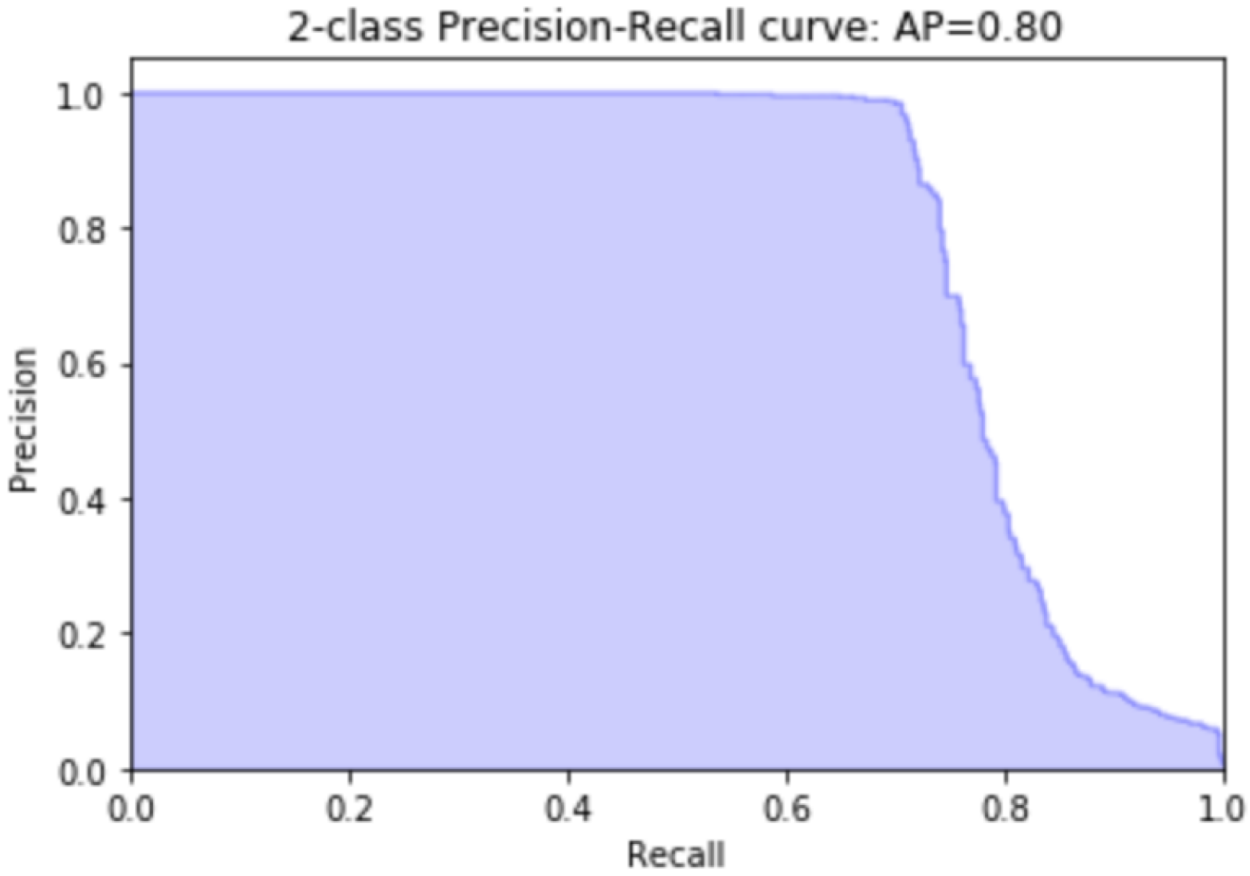
- Focused on the balance deltas as well as whether destination account was a Merchant or not
  Stored the balance deltas, a flag of whether they matched the transaction amount, and a flag for type of account, dropped names and balances, and cancelled transactions
- Started with a default hyper-parameters, later changed my objective metric, made some adjustments to the fixed hyper-parameters as well as ranges for tuning.
- Got a relatively high F1 score (0.91) compared to my initial training jobs. I think it was mostly to do with the feature engineering, but I likely could have spent a lot more time on hyper-parameters to get a better model, but didn't have much time to do so without guidance

```
Accuracy: 1.000
F1 Score: 0.910
```





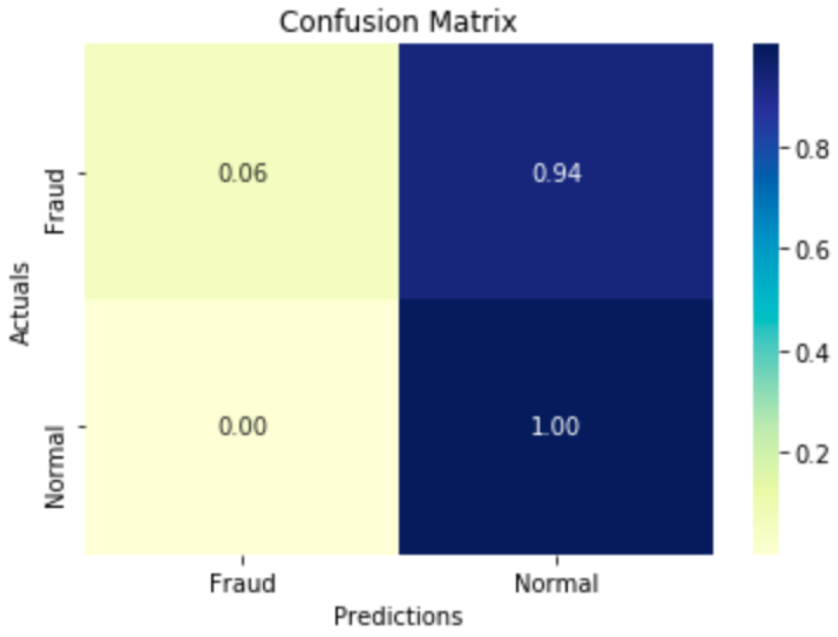
XGBOOST

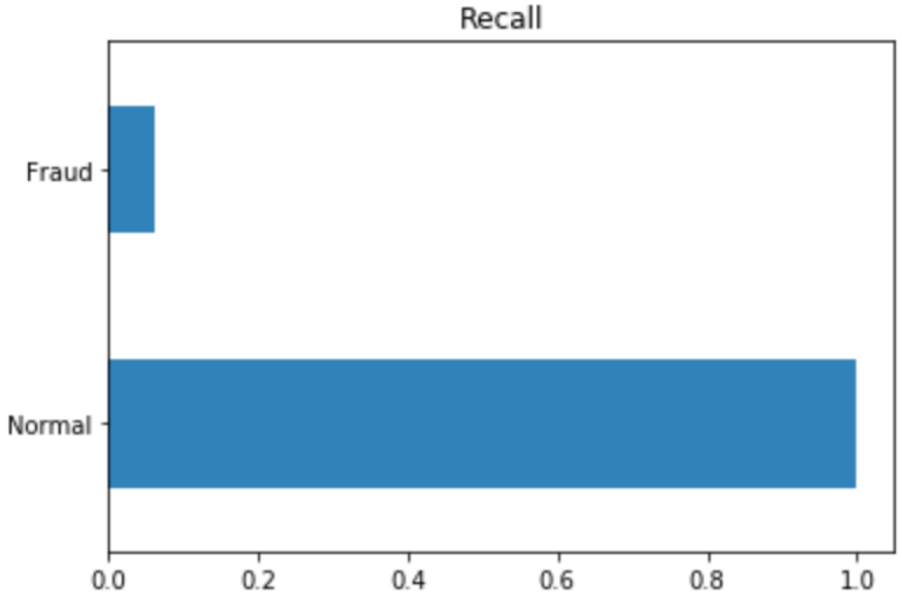

2-class Precision-Recall curve: AP=0.80
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

LINEAR LEARNER

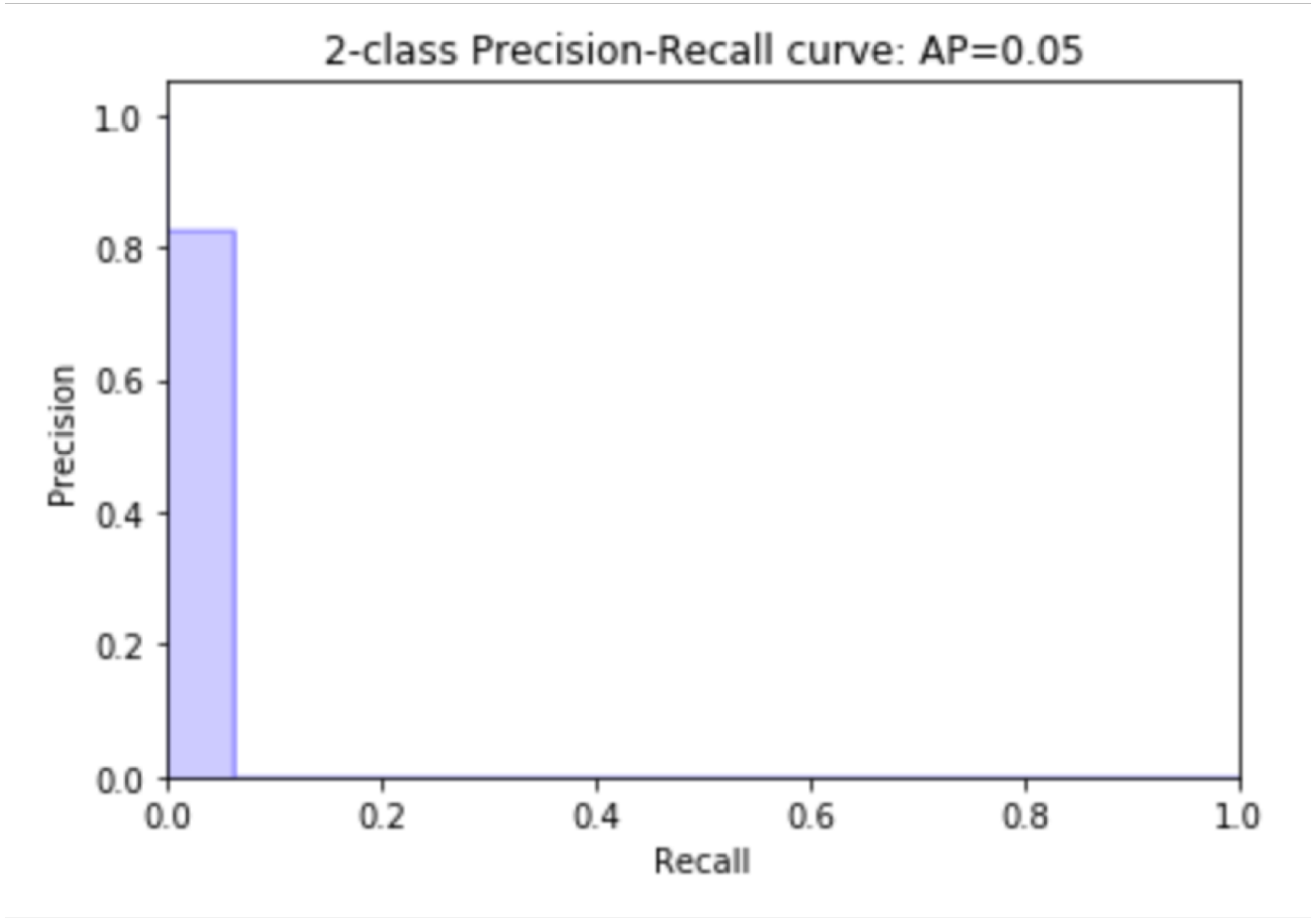
- Feature Engineering included dropping string fields, identifying merchant destinations, and having one-hot labels
- Hyperparams were tuned against a binary classifying validator
- Final model functions fairly abysmally, due probably to inadequate feature engineering
- Next time: spend more time feature engineering. Due to large sample size, model training took very long, so for better iterating use a subset for quick training

```
Accuracy: 0.999
F1 Score: 0.558
```



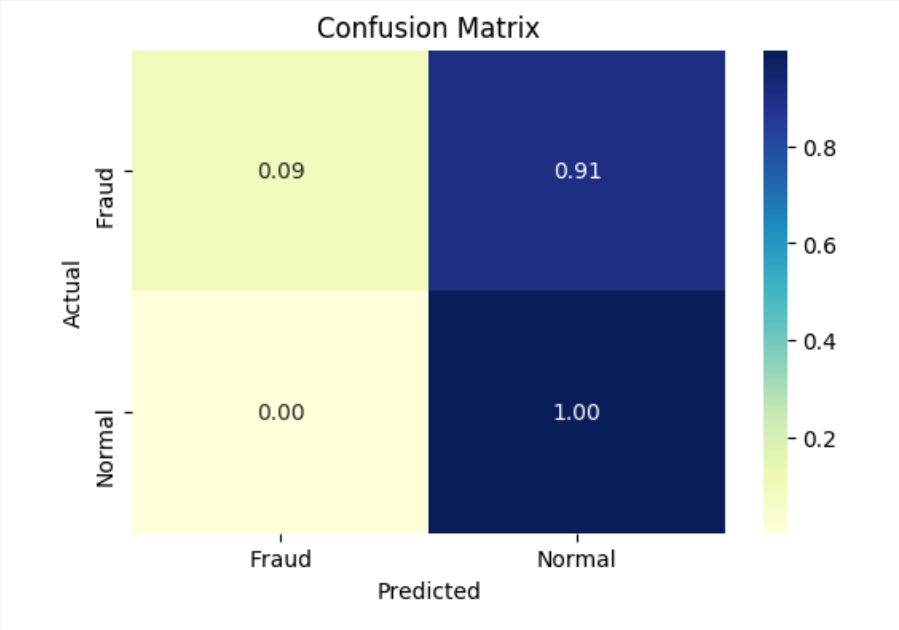


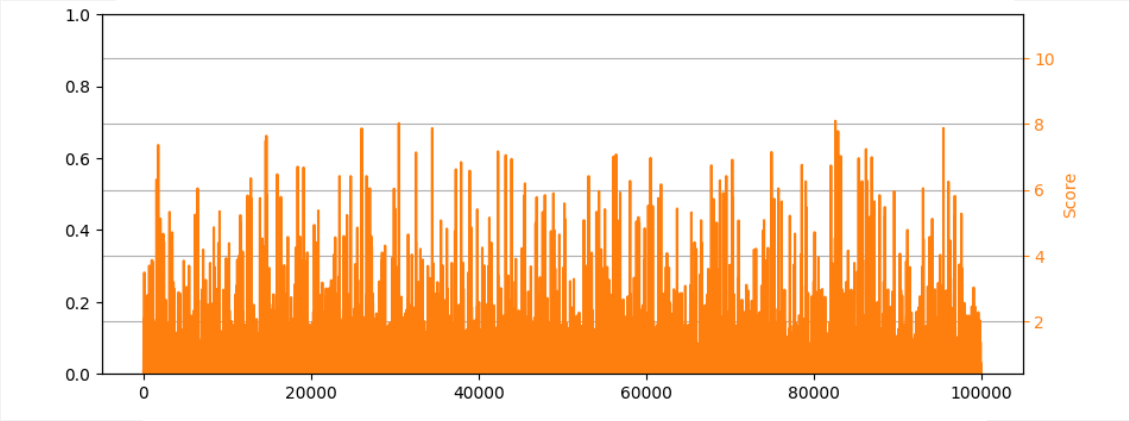
LINEAR LEARNER



RANDOM CUT FOREST

- Removed all values except for a calculated balance difference, the amount, the original balance and if it cleared the balance
- If I left all values in then the anomaly score scores would be closer together because of all the other noise. (score value chart posted above)
- I trained the RCF model on 60% of of random rows in the original data set.
- I tested with 100,00 random rows of the test data set
- Used a few different thresholds to calculate fraud using the score graph and checking the confusion matrix afterwards
- Better feature engineering would help RCF create outliers to help catch fraud.





CONCLUSIONS

- Of models tested, XGBoost faired the best
- Unanimously agree that more feature engineering was needed
- Better models could be built knowing business incentive (i.e. is reporting more False Positives better than False Negatives)
- Would time of transaction give insight to hot periods in the year?