# Fraud Detection of Mobile Platform Transaction Data

Ariel Li, Divya Ravindran,
Glory Scheel, Sneha Vasudevan

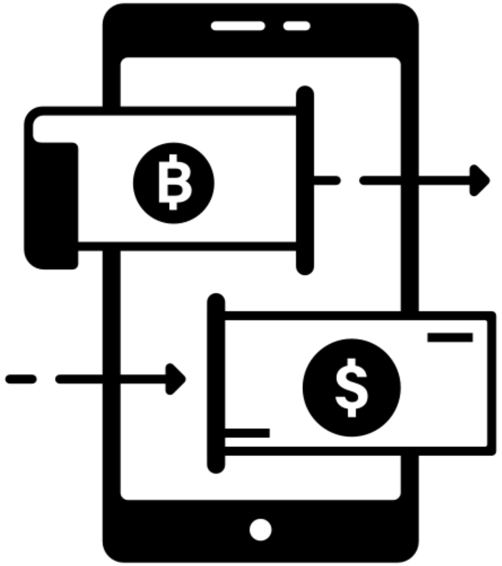
# Agenda

- Problem Statement
- Exploratory Data Analysis
- Feature Engineering
- Data Transformation and Assumptions
- Modeling
- Future Work

# Problem Statement

As online transactions become prevalent, so do frauds. Fraudulent transactions are detrimental to customers' accounts as well as the reputation of the transactional platform, so fraud detection has become paramount in financial services.

To protect customer accounts, this project aims to develop machine learning models to accurately detect and intercept frauds in mobile transactions.



# EDA and
# Feature Engineering

# Data Summary

Data Source:
Kaggle Synthetic Financial Dataset for Fraud Detection

Dataset:
PaySim-generated mobile money transactions (6.3+ million records)

Time Range:
1-month simulation, 743 unit hours

Number of Variables:
11

Target Variable: Is_Fraud

Valid Transactions: 6,354,407

Fraudulent Transactions: 8,213

0.13% of the transactions are frauds.

Highly imbalanced

# Exploratory Data Analysis

**Total Transactions**

Cash-in : 1399284

Cash-out : 2237500

Debit: 41432

Payment : 2151495

Transfer : 532909

**Frauds**

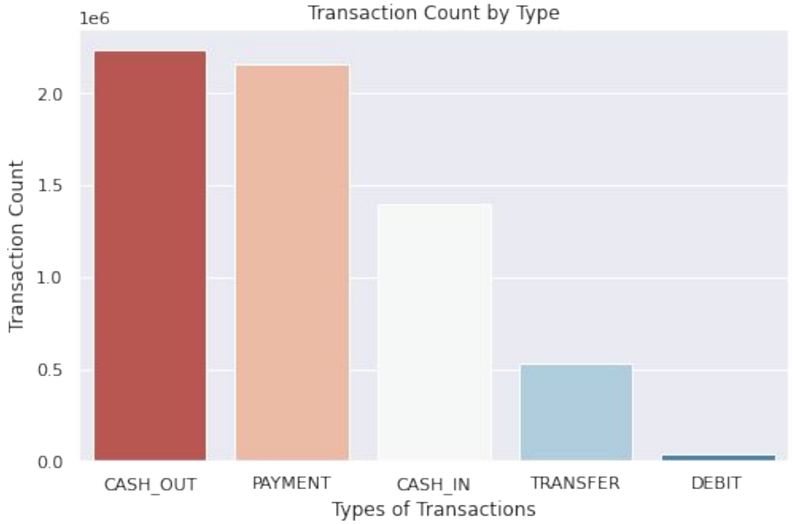
Cash-in : 0

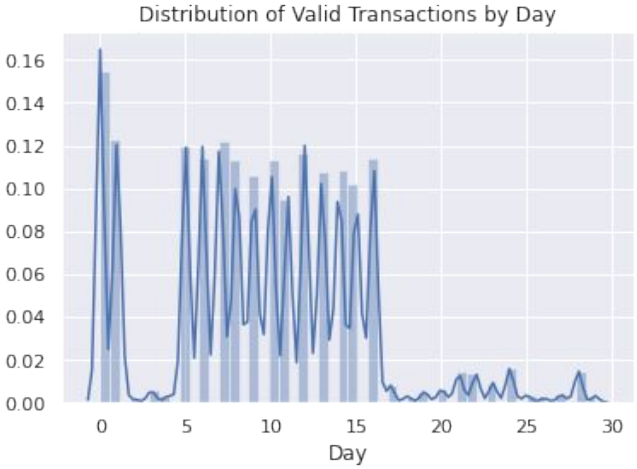
Cash-out : 4116

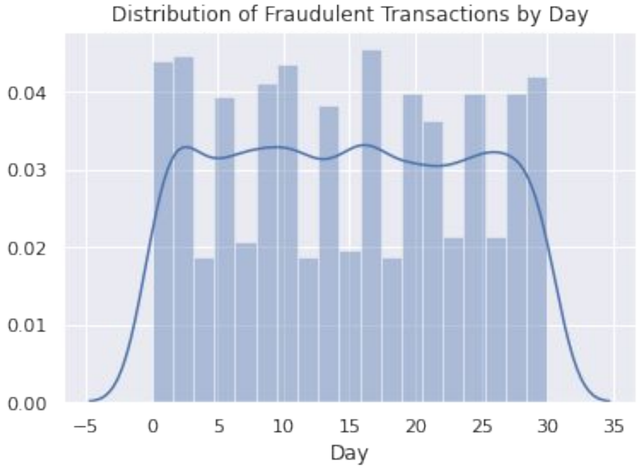
Debit: 0

Payment : 0

Transfer : 4097

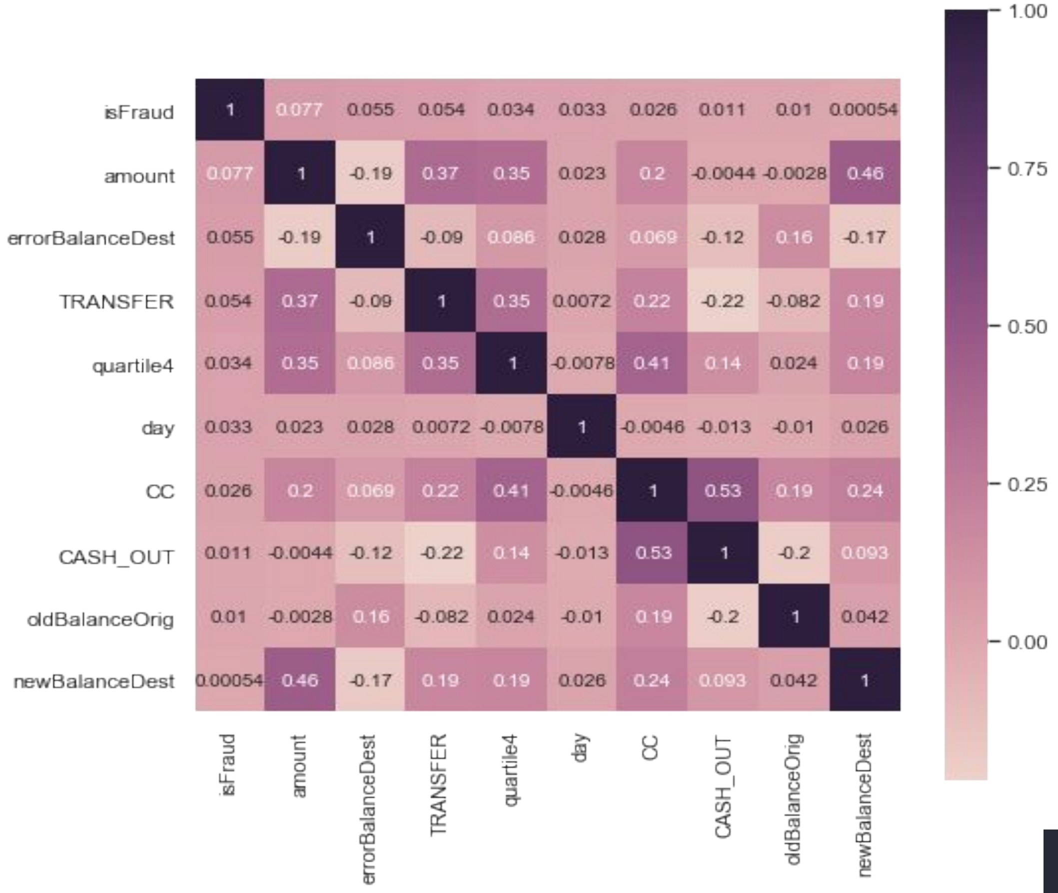


# Feature Engineering

- Error Amount in Balance
  - Error in origination account
  - Error in destination account
- Day (of data simulation)
- One-hot encoding of transaction types
  - Cash-in, cash-out, payment, transfer, debit
- Account Types
  - "CC" (Customer to Customer)
  - "CM" (Customer to Merchant)
  - "MC" (Merchant to Customer
  - "MM" (Merchant to Merchant)



# Data Transformation and Assumptions

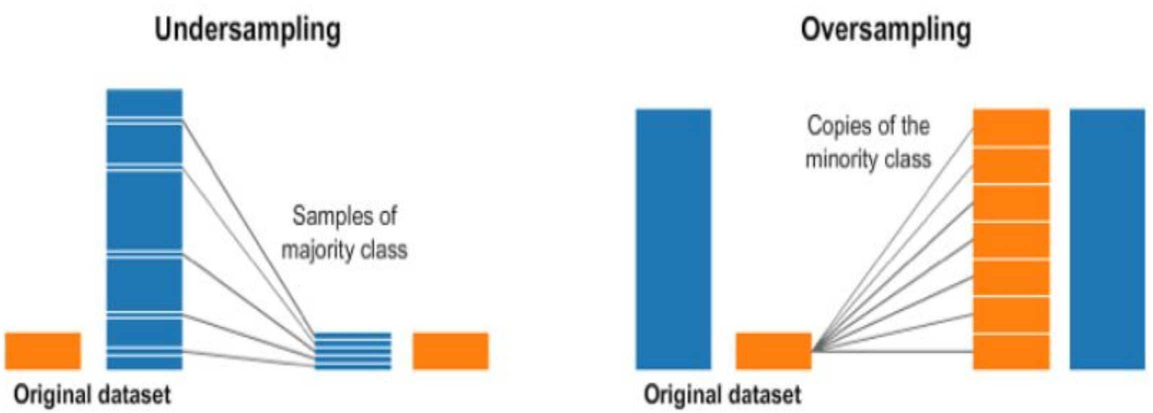
# Sampling Techniques

- **Undersampling Technique Used :**

  Random Undersampling (XGBoost)

- **Oversampling Technique Used:**

  ADASYN (Logistic Regression, MLP)




Source: https://medium.com/quantyca/oversampling-and-undersampling-adasyn-vs-enn-60828a58db39

# Assumptions and Considerations

## Assumptions on Simulated Data

- Simulated data is as prudent as the original dataset
- Data simulated preserves the multivariate relationships between variables

## Modelling Considerations

- Sampling techniques address imbalance without bias
- Outliers are correlated with the anomalies (outlier detection models)
- Normalization over standardization as the latter assumes Gaussian distribution (MLP, Autoencoder, Log Reg)

# Modeling

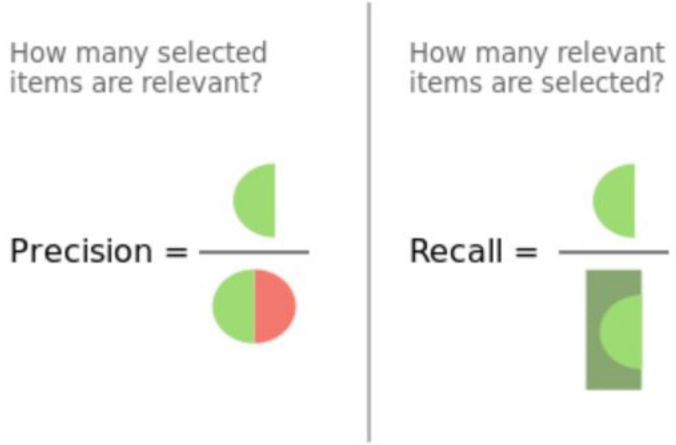
# Evaluation Metrics

**Recall (class 1):**

Tells us the proportion of frauds correctly classified

**Precision (class 0):**

Tells us the proportion of true valid transactions out of the predicted valid transactions



# Tree-Based Models with Undersampling

| Train Model | True Positive Rate | False Positive Rate | True Negative Rate | False Negative Rate |
|---|---|---|---|---|
| Random Forest | 0.999242 | 0.00000 | 1.00000 | 0.000758 |
| AdaBoost | 0.996208 | 0.00091 | 0.99909 | 0.003792 |
| GradientBoost | 0.996208 | 0.00091 | 0.99909 | 0.003792 |
| XGBoost | 0.995601 | 0.00000 | 1.00000 | 0.004399 |

| Test Model | True Positive Rate | False Positive Rate | True Negative Rate | False Negative Rate |
|---|---|---|---|---|
| Random Forest | 0.997531 | 0.000148 | 0.999852 | 0.002469 |
| AdaBoost | 0.997531 | 0.002050 | 0.997950 | 0.002469 |
| GradientBoost | 0.998148 | 0.000901 | 0.999099 | 0.001852 |
| XGBoost | 0.997531 | 0.000124 | 0.999876 | 0.002469 |

Best Model (XGBoost): Gamma : 4.5, learning rate = 0.3, max_depth : 2, n_estimators = 700

# XGBoost with Weight Adjustment

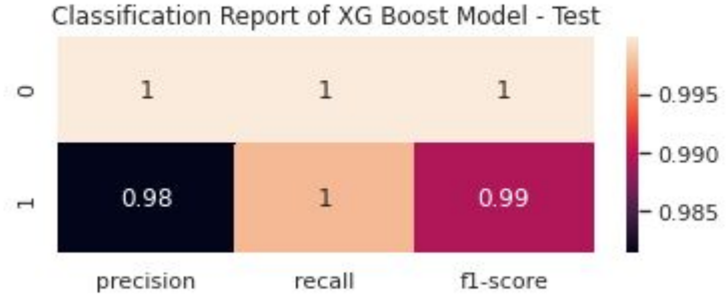
**Parameter Tuning :**
scale_pos_weight = sqrt(ratio(non-fraud/fraud))

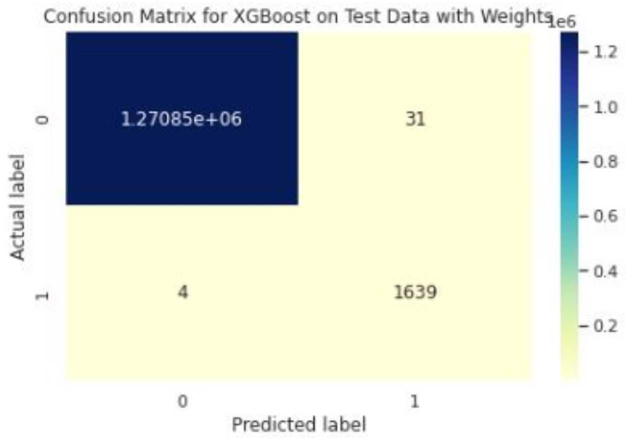
**Method :** Randomized search

Best model parameters :

- gamma : 4
- learning rate  = 0.8
- max_depth : 1
- n_estimators = 550

| Test Model | True Positive Rate | False Positive Rate | True Negative Rate | False Negative Rate |
|---|---|---|---|---|
| XGBoost | 0.997565 | 0.000024 | 0.999976 | 0.002435 |

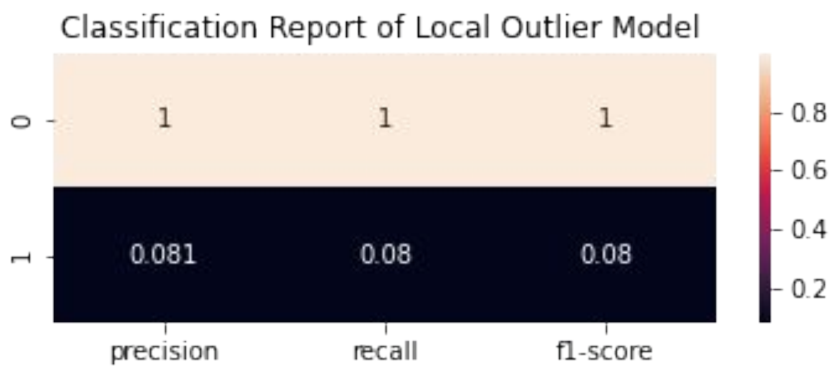


# Outlier Detection

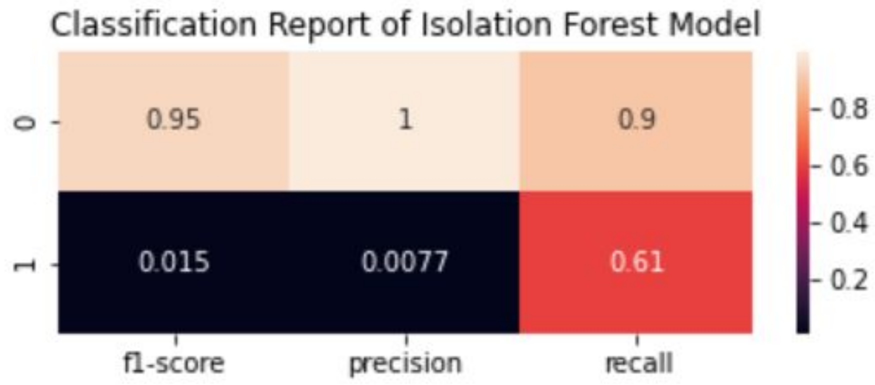
### Local Outlier Factor

- Unsupervised detection method
- Samples that are substantially lower in density are considered outliers




### Isolation Forest

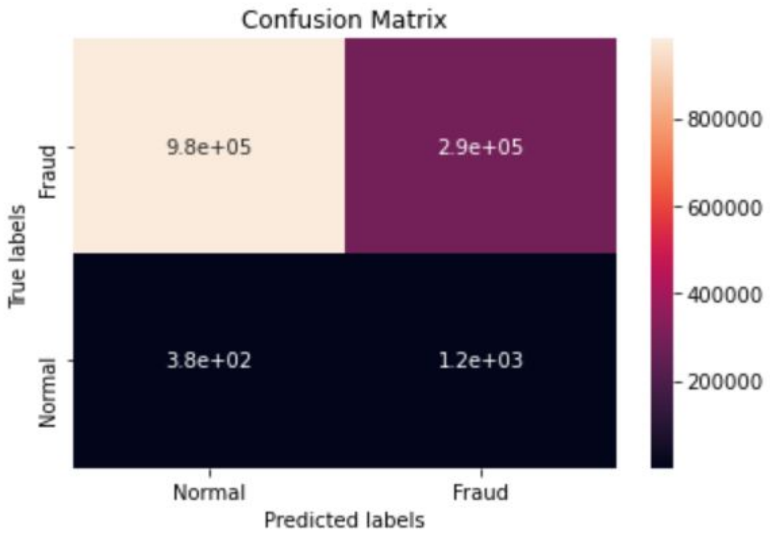
- Identifies normal vs. abnormal observations and classifies abnormal as outliers
- Calculates anomaly score - scores closer to 1 are considered anomalies, below 0.5 is normal

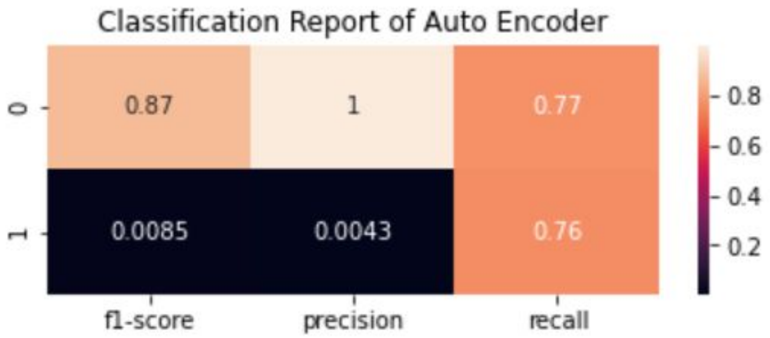


# Auto Encoder

**Process**

1. Train network on the training data without any instances of fraud
2. Predict on test data
3. Calculate error
4. Where error passes certain threshold (MSE>0.007) we predict as anomaly.



# Weighted Logistic Regression

Base model

- Low recall for class 1

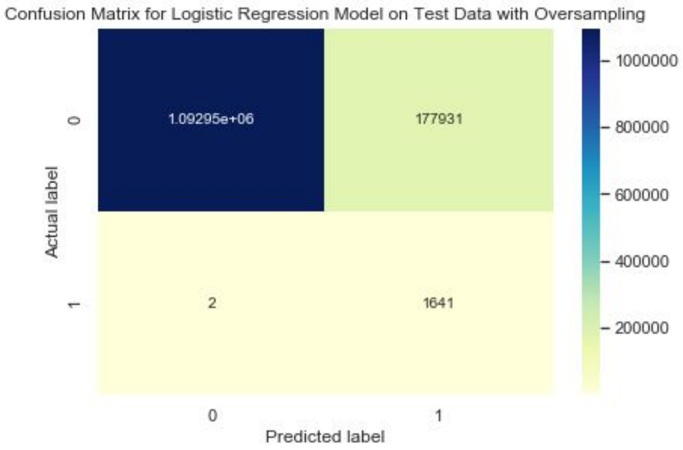
Weighted logistic regression

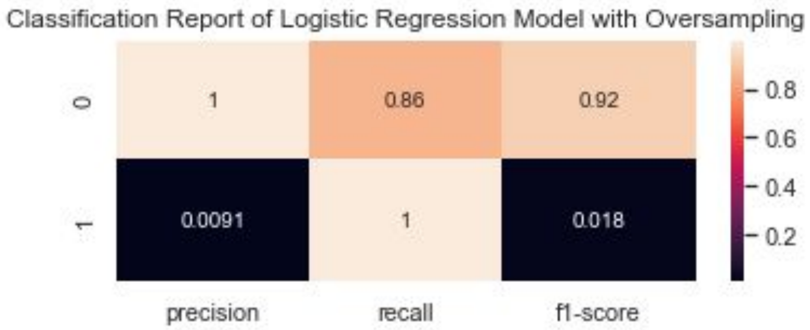
- Assign more weights to the minor class to heavily penalize misclassification
- Grid search for best class weights: {0: 1, 1: 1200}

Logistic regression on oversampled data

- Almost perfect class 1 recall





# Multi-Layer Perceptron

Base model

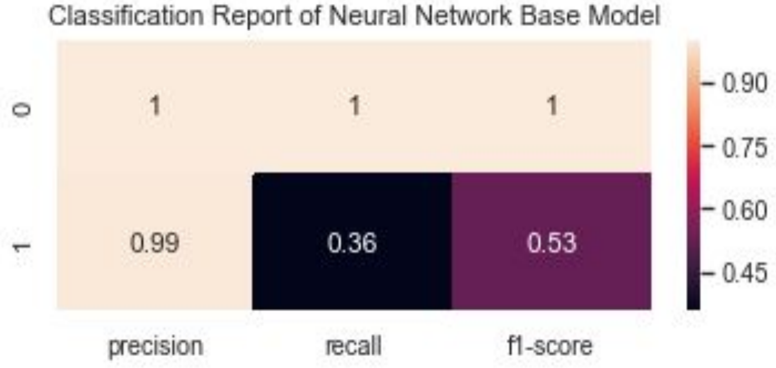
- Low recall for class 1

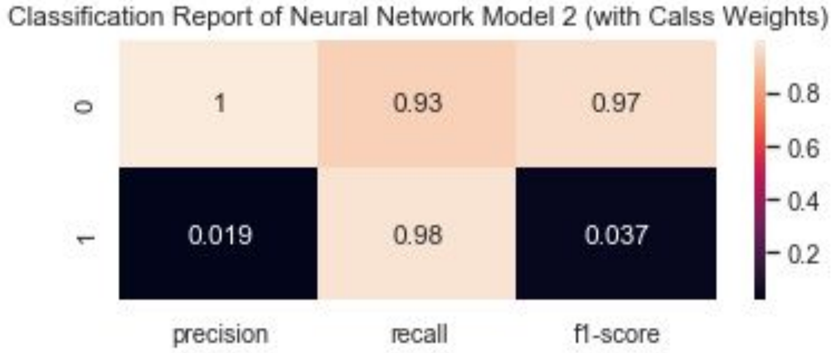
Weighted MLP

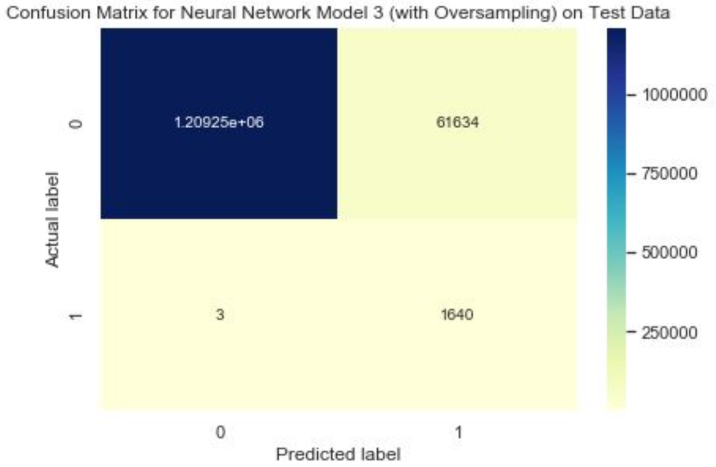
- Assign more weights to the minor class to heavily penalize misclassification
- Class weights: {0: 1, 1: 780}

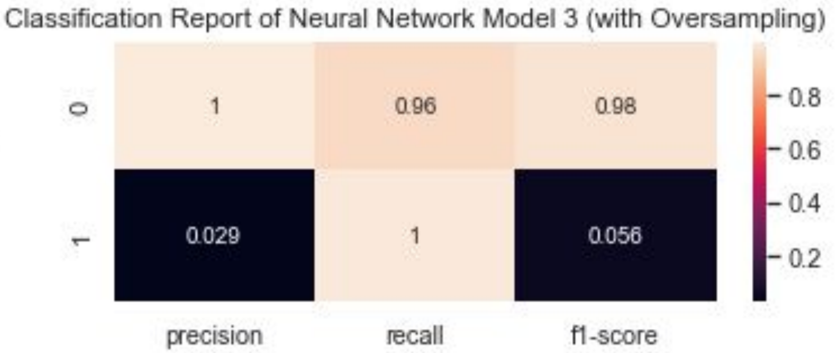
MLP on oversampled data

- Almost perfect class 1 recall



# Model Selection

Best performance on both classes:
XGBoost Classifier

High performance and efficiency:
Multi-Layer Perceptron

# Challenges and Future Work

## Challenges

- Struggle of classification on extremely imbalanced datasets (<1:100)
- Oversampling on 6+ million data points is computationally intensive

## Future Work

- Use ensemble modeling to improve prediction performance
- Modify cost function to reflect real costs to the stakeholder, i.e. punish false negatives and false positives differently

# Thank you!