Finance Challenge

Anomaly Detection

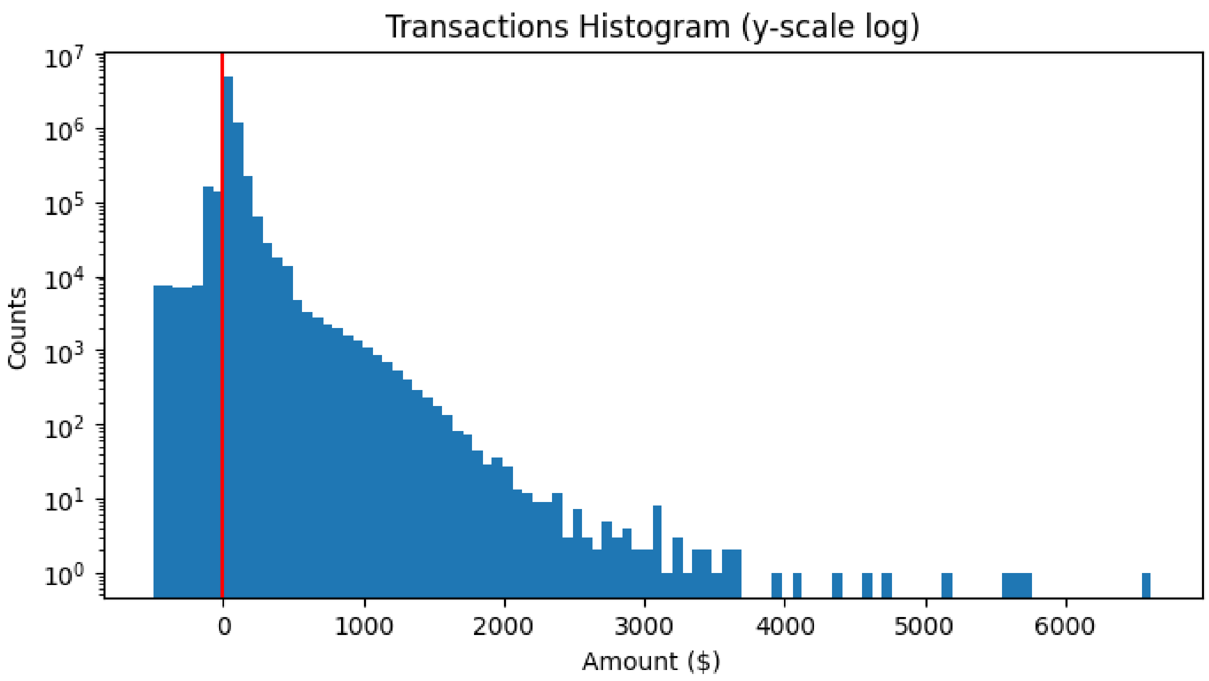
Overview

- Data on credit card transactions, composed of transaction logs, customer and card info.
- Spanning from Jan 2016 – December 2019
- Existing fraudulent transactions, interested in a model to identify future ones.

Image src: https://time.com/personal-finance/static/84016af8afe9681354d097200e07945e/57e17/credit-card-types.jpg

Data - EDA

- Around ~7 million transactions ranging from -$500-$6k
- 0.12% categorized as fraudulent.
- 2,000 customers, 1,610 with transactions.
- ~6k cards, customers can have multiple cards.



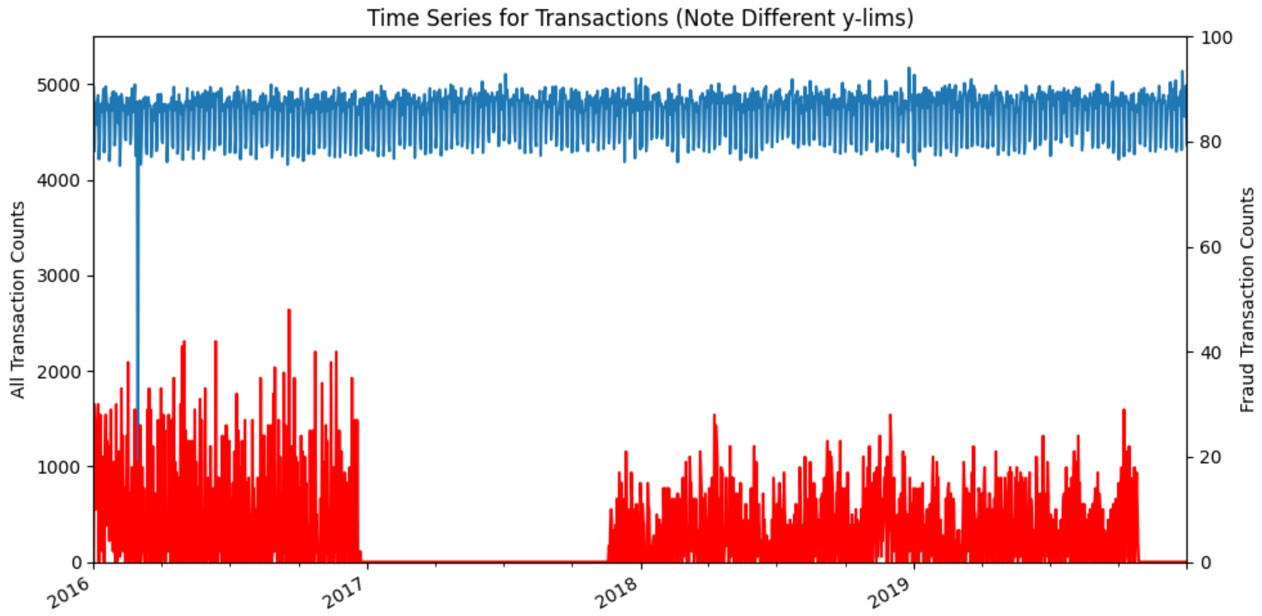
Data - EDA

- 70% of transactions used a Chip, 17% were swipes and ~13% online.
- ~2% of all invoices were cancellations
- ~89% of all cards have chips.
- Error types for transactions are listed.
- Some transactions are international.

Num Credit Cards	Counts	Percentage %
3	449	22.45
1	416	20.80
2	388	19.40
4	376	18.80
5	206	10.30
6	105	5.25
7	40	2.00
8	17	0.85
9	3	0.15

Data – Data Quality

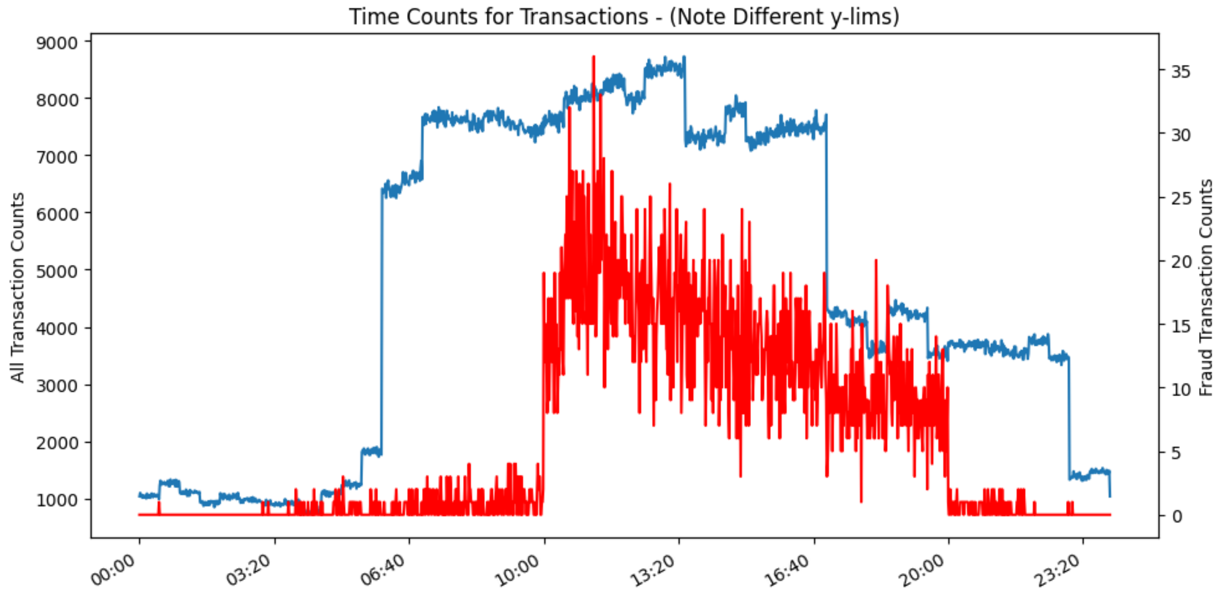
- ~12.5% merchants are missing location information.
- No fraudulent activity for most of 2017, this might've been a data truncation issue.



Data – Feature Engineering

Create additional features:

- Transaction took place between 10AM and 8PM.
- User is retired.
- Merchant and Customer States match
- International transaction
- Debt to Income Ratio
- Zip Median to Income Ratio



Model

- Need to use Recall as the preferred evaluation metric.
- XGBoost as choice of model, powerful library for gradient boosting.
- Why?
  - GPU acceleration
  - Class imbalance support
  - Works great with tabular data

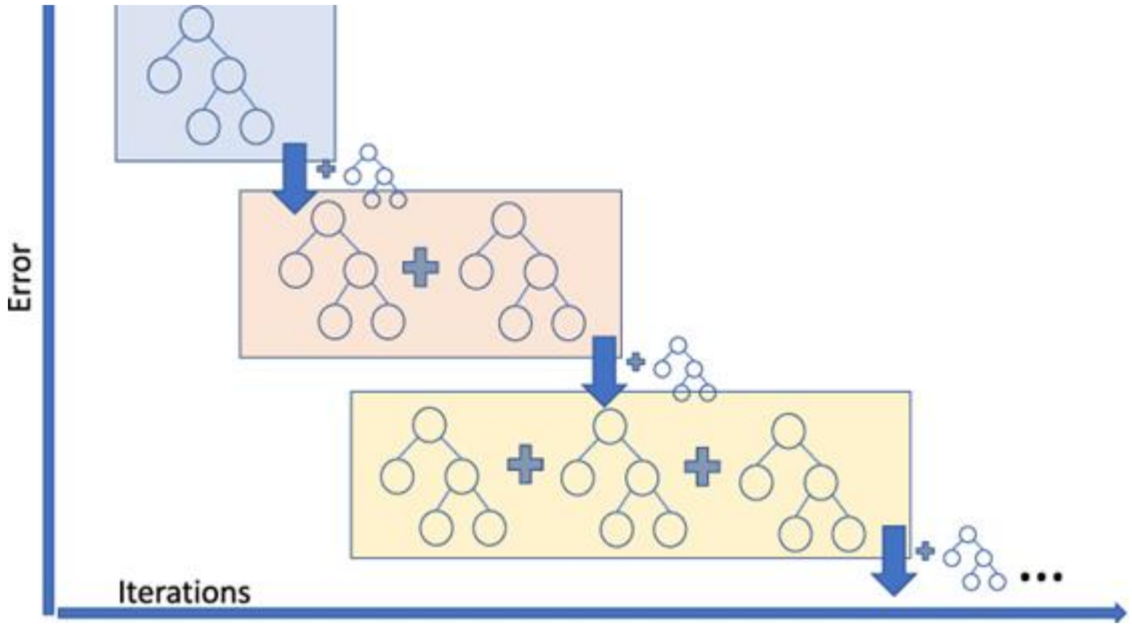
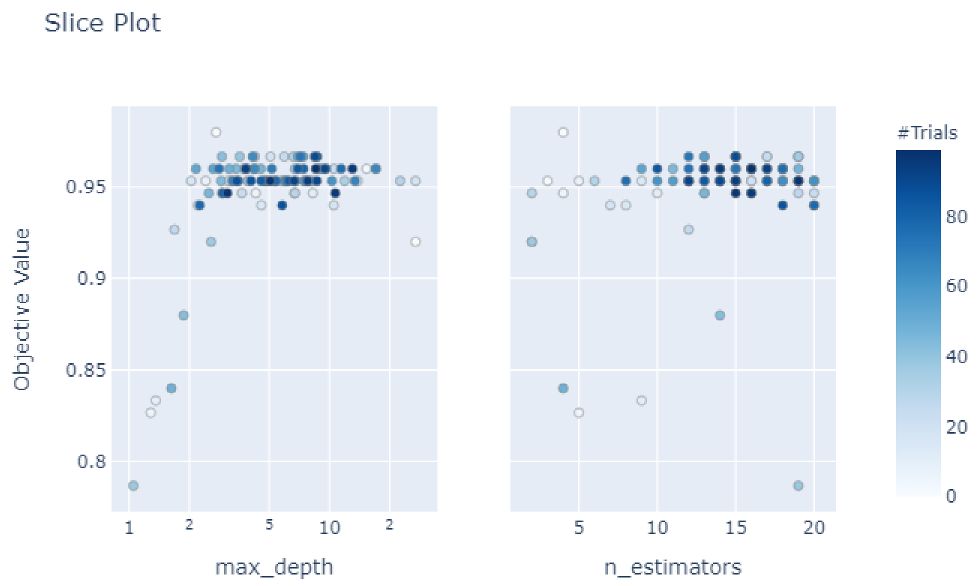


Image src: https://medium.com/analytics-vidhya/what-is-gradient-boosting-how-is-it-different-from-ada-boost-2d5ff5767cb2

Model – Training

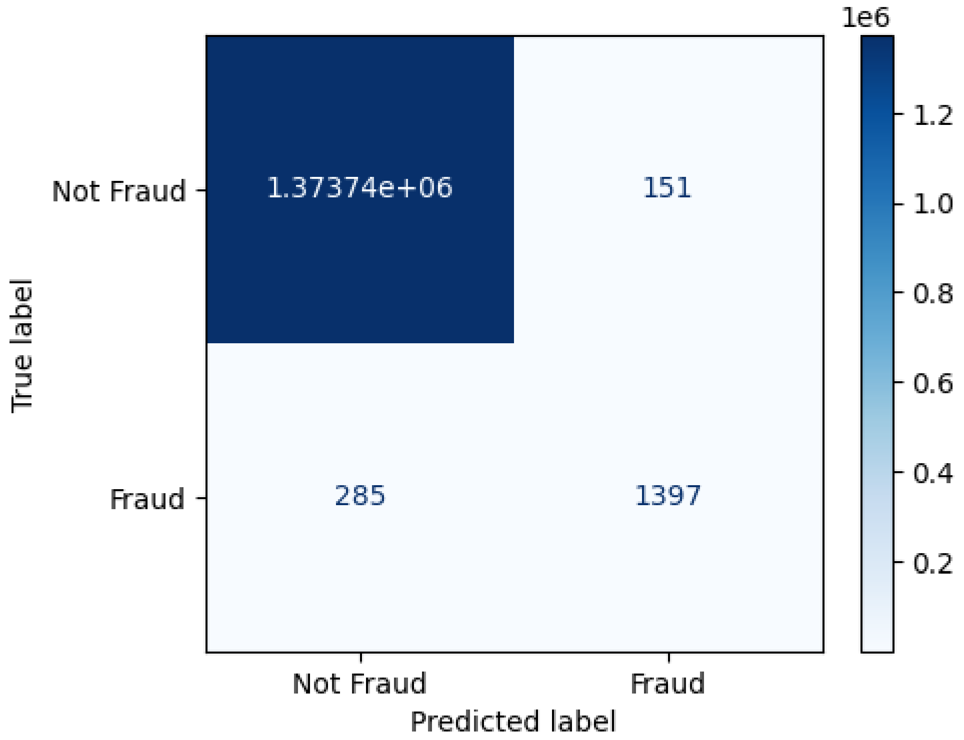
- Encode all the selected categorical columns.
- Split into Training, Testing and Validation datasets stratified on the fraud column.
- Tune hyperparameters, set recall as the preferred metric to maximize.



Results

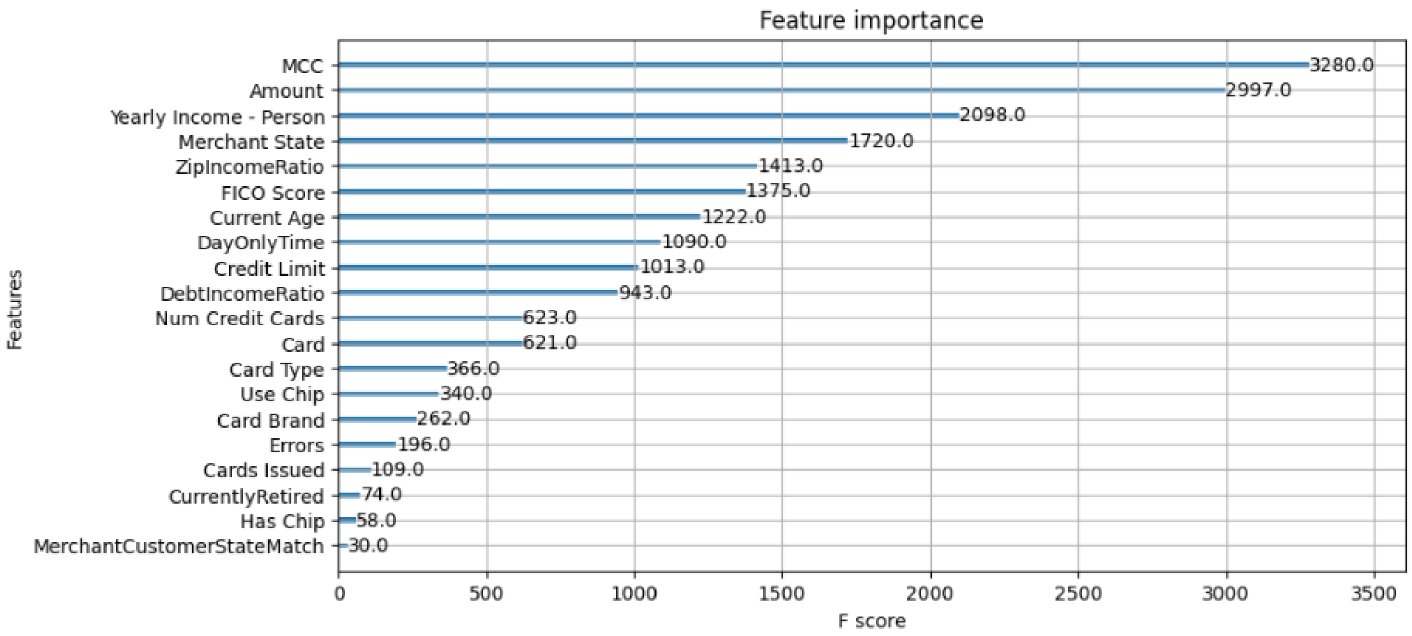
- The best model, based on the best tuning parameters had the following classification metrics:

Accuracy: 1.0
Precision: 0.902
Recall: 0.831
F-1 Score: 0.865



Results

- Feature importance for XGBoost model
- Suggests that the merchant code is the most important feature followed by the amount, income bracket and location.



Conclusion

- Model did a decent job at predicting fraudulent transactions.
- Improvements possible through additional features, such as customers state data during transaction rather than at present.
- Further explorations can include the use of other boosting algorithms, or advanced deep learning models such as GAN’s.

Questions?

Thank you