ethereum

Ethereum Fraud Detection

Kai Hayden, Huizhu Liu, Melody Feng, Siyuan Shao, Camille Xia, Xiao Zhang

Agenda

I. **Business Problem**
   - Business Use Case

II. **Data Preprocessing**
   - Data Overview
   - Data Cleaning
   - Exploratory Data Analysis

III. **Feature Engineering**

IV. **Data Balancing**

V. **Modeling**

VI. **Conclusions**

THE UNIVERSITY OF CHICAGO

ethereum

ethereum

Business Problem

Business Problem

ethereum

Fraud in Blockchain

- Scammers are relying on the **anonymity** from blockchain to hide fraudulent transactions
- Crypto transactions are **irreversible** - added to the next block and is not reversible
- 2021: Cryptocurrency scams increased **516%** from 2020 → **$3.2 billion** worth of cryptocurrency

Ethereum

- Blockchain-based software platform used for sending and receiving value globally with native cryptocurrency (**Ether**)
- Ether is second to Bitcoin in market capitalization
- Common scams include: giveaway scams, support scams, phishing scams, crypto trading broker scams

Business Use Case

ethereum

*" In 2016, **false positives** amounted to **19 percent** of **loss** while **actual fraud** represented **7 percent** of total cost of fraud."*

According to JPMorgan

Our model aims to identify fraudulent transactions on the ethereum blockchain before they are written to the blockchain and reduce false positives

- Precision (TP / TP + FP) : of all predicted real frauds, how many are actual real frauds
  - Business focus
- Recall (TP / TP + FN) : of all real frauds, how many are we predicting correctly
  - User focus
- We will be focusing on **precision** to **decrease false positives** for our business, Ethereum

Data Preprocessing

Data Overview

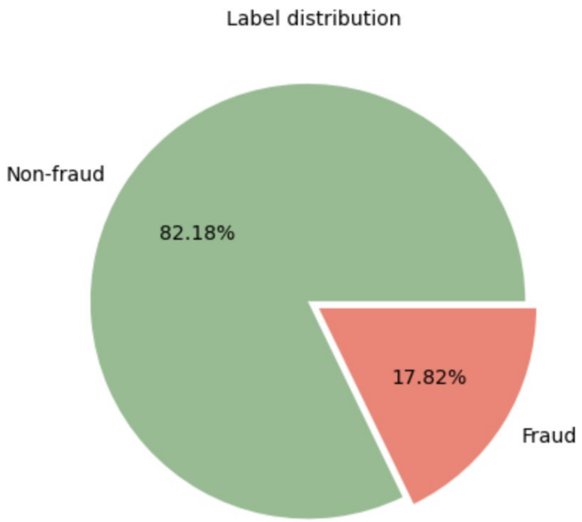
ethereum

Data Source: https://www.kaggle.com/vagifa/ethereum-frauddetection-dataset
Data Size:
1 csv file with 9841 rows and 51 columns (2.88 MB)

This dataset is imbalanced (Fraud:18%; Non-fraud:82%)

	Unnamed: 0	Index	Address	FLAG	Avg min between sent tnx	Avg min between received tnx	Time Diff between first and last (Mins)	Sent tnx	Received Tnx	Number of Created Contracts	Unique Received From Addresses
0	0	1	0x00009277775ac7d0d59eaad8fee3d10ac6c805e8	0	844.26	1093.71	704785.63	721	89	0	40
1	1	2	0x0002b44ddb1476db43c868bd494422ee4c136fed	0	12709.07	2958.44	1218216.73	94	8	0	5
2	2	3	0x0002bda54cb772d040f779e88eb453cac0daa244	0	246194.54	2434.02	516729.30	2	10	0	10
3	3	4	0x00038e6ba2fd5c09aedb96697c8d7b8fa6632e5e	0	10219.60	15785.09	397555.90	25	9	0	7
4	4	5	0x00062d1dd1afb6fb02540ddad9cdebfe568e0d89	0	36.61	10707.77	382472.42	4598	20	1	7
...	...	...	...	...	...	...	...	...	...	...	...
9836	9836	2175	0xff481ca14e6c16b79fc8ab299b4d2387ec8ecdd2	1	12635.10	631.39	58748.48	4	13	0	11
9837	9837	2176	0xff718805bb9199ebf024ab6acd333e603ad77c85	1	0.00	0.00	0.00	0	0	0	0
9838	9838	2177	0xff8e6af02d41a576a0c82f7835535193e1a6bccc	1	2499.44	2189.29	261601.88	67	43	0	31
9839	9839	2178	0xffde23396d57e10abf58bd929bb1e856c7718218	1	0.00	0.00	0.00	0	1	0	1
9840	9840	2179	0xd624d046edbdef805c5e4140dce5fb5ec1b39a3c	1	37242.70	149.56	670817.33	18	3	0	1

9841 rows × 51 columns



Data Cleaning

ethereum

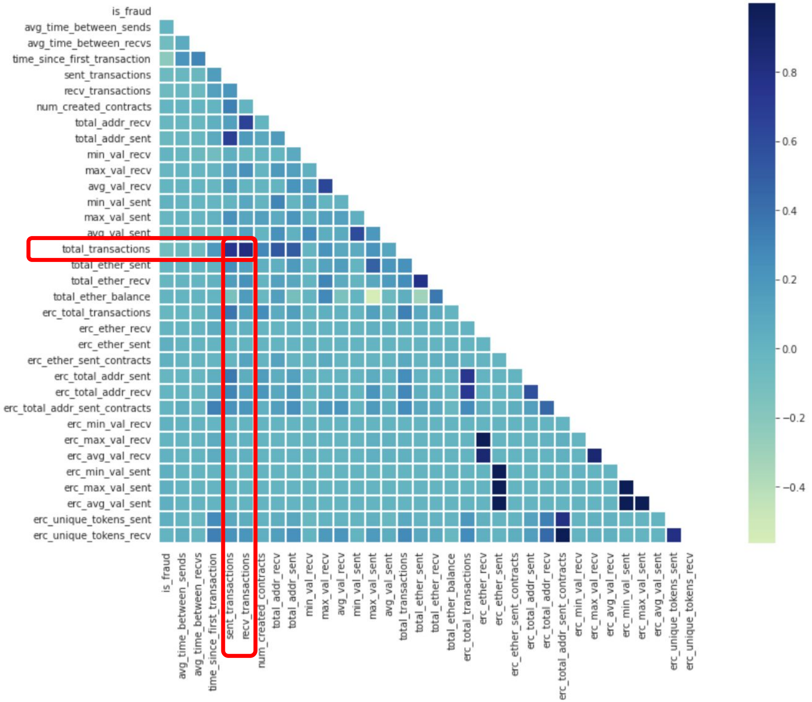
- Drop **Index Columns**
- Drop **Columns which have only one distinct value**
- Drop Duplicated Rows
- Drop Missing Values Data
(25 columns with about 8% missing values)

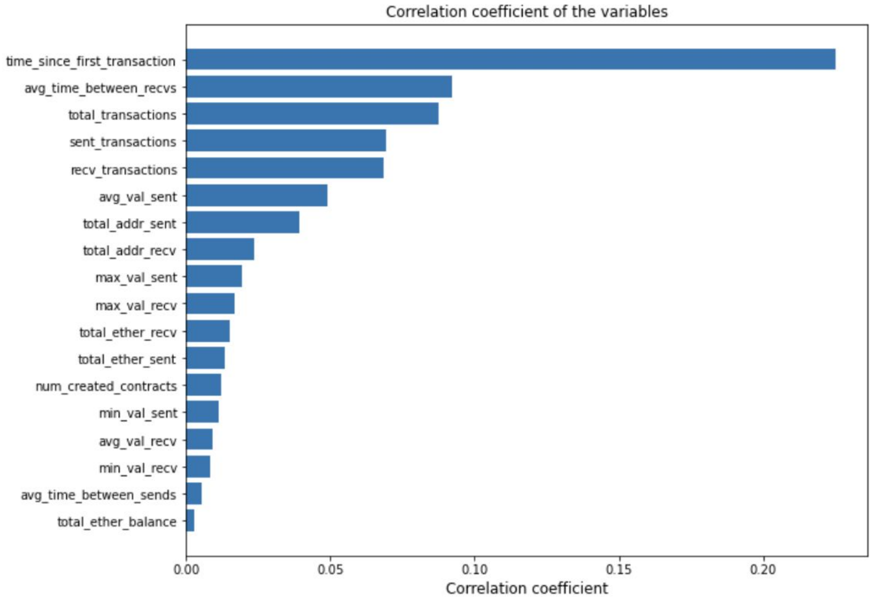
Null: number of missing values
Uniques: number of distinct values

	Nulls	Uniques
Unnamed: 0	0	9841
Index	0	4729
Address	0	9816
FLAG	0	2
Avg min between sent tnx	0	5013
Avg min between received tnx	0	6223
Time Diff between first and last (Mins)	0	7810
Sent tnx	0	641
Received Tnx	0	727
Number of Created Contracts	0	20
Unique Received From Addresses	0	256
Unique Sent To Addresses	0	258
min value received	0	4589
max value received	0	6302
avg val received	0	6767
min val sent	0	4719
max val sent	0	6647
avg val sent	0	5854
min value sent to contract	0	3
max val sent to contract	0	4
avg value sent to contract	0	4
total transactions (including tnx to create contract	0	897
total Ether sent	0	5868
total ether received	0	6728
total ether sent contracts	0	4
total ether balance	0	5717
Total ERC20 tnxs	829	300
ERC20 total Ether received	829	3460
ERC20 total ether sent	829	1415
ERC20 total Ether sent contract	829	29
ERC20 uniq sent addr	829	107
ERC20 uniq rec addr	829	147
ERC20 uniq sent addr.1	829	4
ERC20 uniq rec contract addr	829	123
ERC20 avg time between sent tnx	829	1
ERC20 avg time between rec tnx	829	1
ERC20 avg time between rec 2 tnx	829	1
ERC20 avg time between contract tnx	829	1
ERC20 min val rec	829	1276
ERC20 max val rec	829	2647
ERC20 avg val rec	829	3380
ERC20 min val sent	829	476
ERC20 max val sent	829	1130
ERC20 avg val sent	829	1309
ERC20 min val sent contract	829	1
ERC20 max val sent contract	829	1
ERC20 avg val sent contract	829	1
ERC20 uniq sent token name	829	70
ERC20 uniq rec token name	829	121
ERC20 most sent token type	841	305
ERC20_most_rec_token_type	851	467

EDA

ethereum





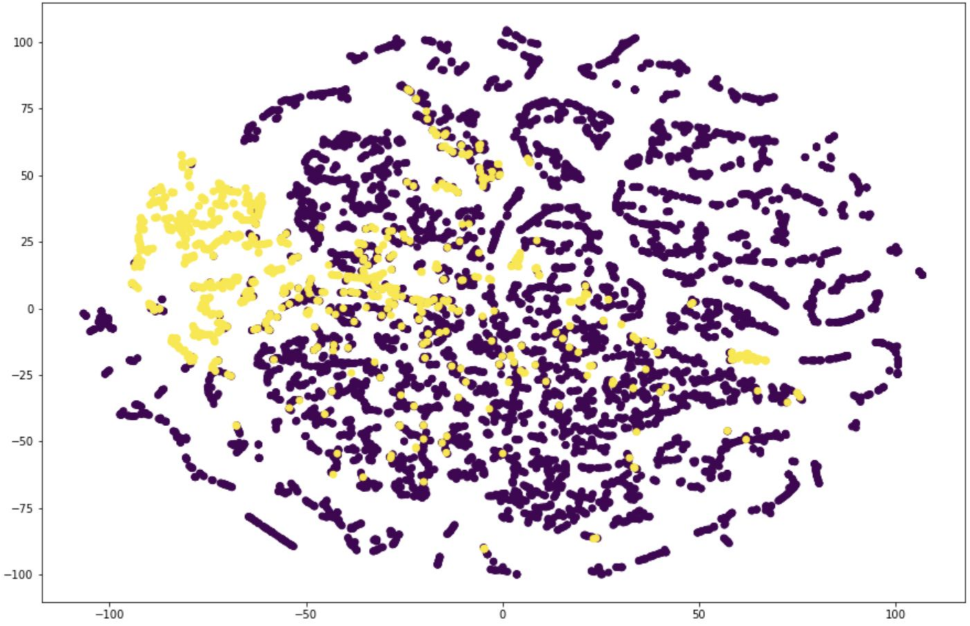
Multicollinearity: several independent variables in a model are correlated which will affect the linear model results such as logistic regression and support vector machine.

THE UNIVERSITY OF CHICAGO

Visualizing High-Dimensional Dataset

ethereum

t-SNE



- Yellow: Fraud Transactions
- Purple: Non-fraud Transactions

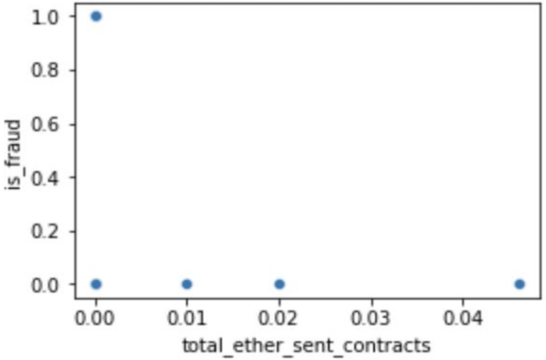
Yellow dots are scattered without a specific pattern

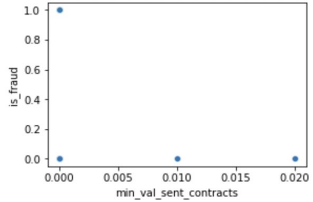
EDA

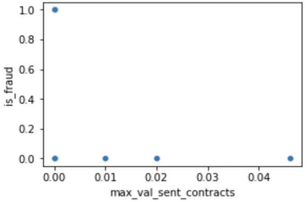
ethereum

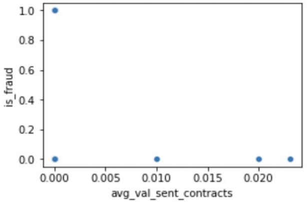
The unique values for min/max/avg_val_sent_contracts and total ether sent contracts are very low and the distributions of the them are very small, therefore, we decided to drop them.

min value sent to contract	0	3
max val sent to contract	0	4
avg value sent to contract	0	4
total transactions (including tnx to create contract	0	897
total Ether sent	0	5868
total ether received	0	6728
total ether sent contracts	0	4





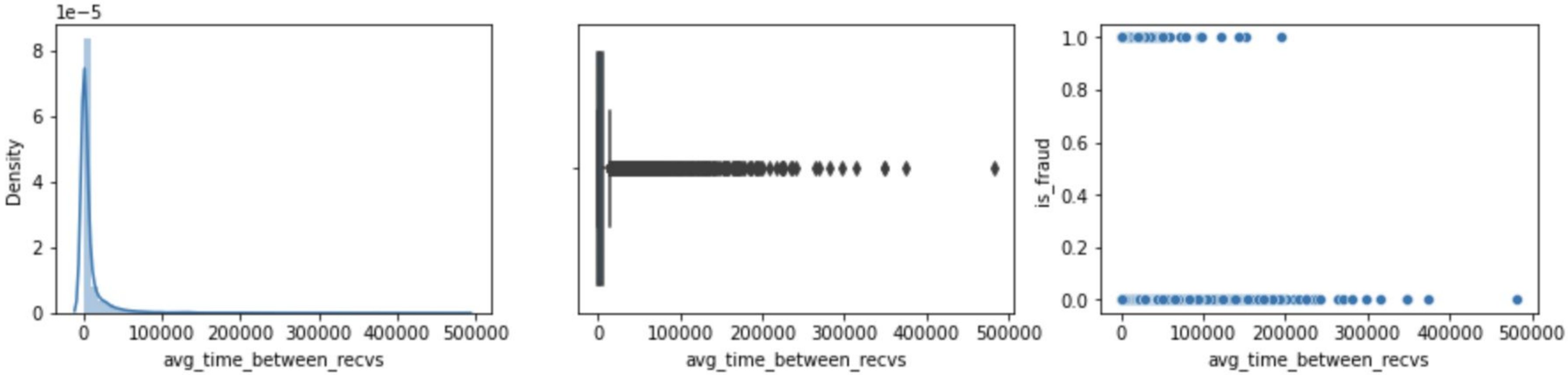





```
The column min_val_sent_contracts has the following distribution:
0.00      9293
0.02         1
0.01         1
Name: min_val_sent_contracts, dtype: int64
=========================================
The column max_val_sent_contracts has the following distribution:
0.000000     9292
0.020000        1
0.010000        1
0.046029        1
Name: max_val_sent_contracts, dtype: int64
=========================================
The column avg_val_sent_contracts has the following distribution:
0.000000     9292
0.020000        1
0.010000        1
0.023014        1
Name: avg_val_sent_contracts, dtype: int64
=========================================
The column total_ether_sent_contracts has the following distribution:
0.000000     9292
0.020000        1
0.010000        1
0.046029        1
Name: total_ether_sent_contracts, dtype: int64
=========================================
```

EDA

Three different types of graphs were drawn to help analyze the data, however, both distribution and box plots were not as helpful as the scatter plot due to most of the major are very right skewed.

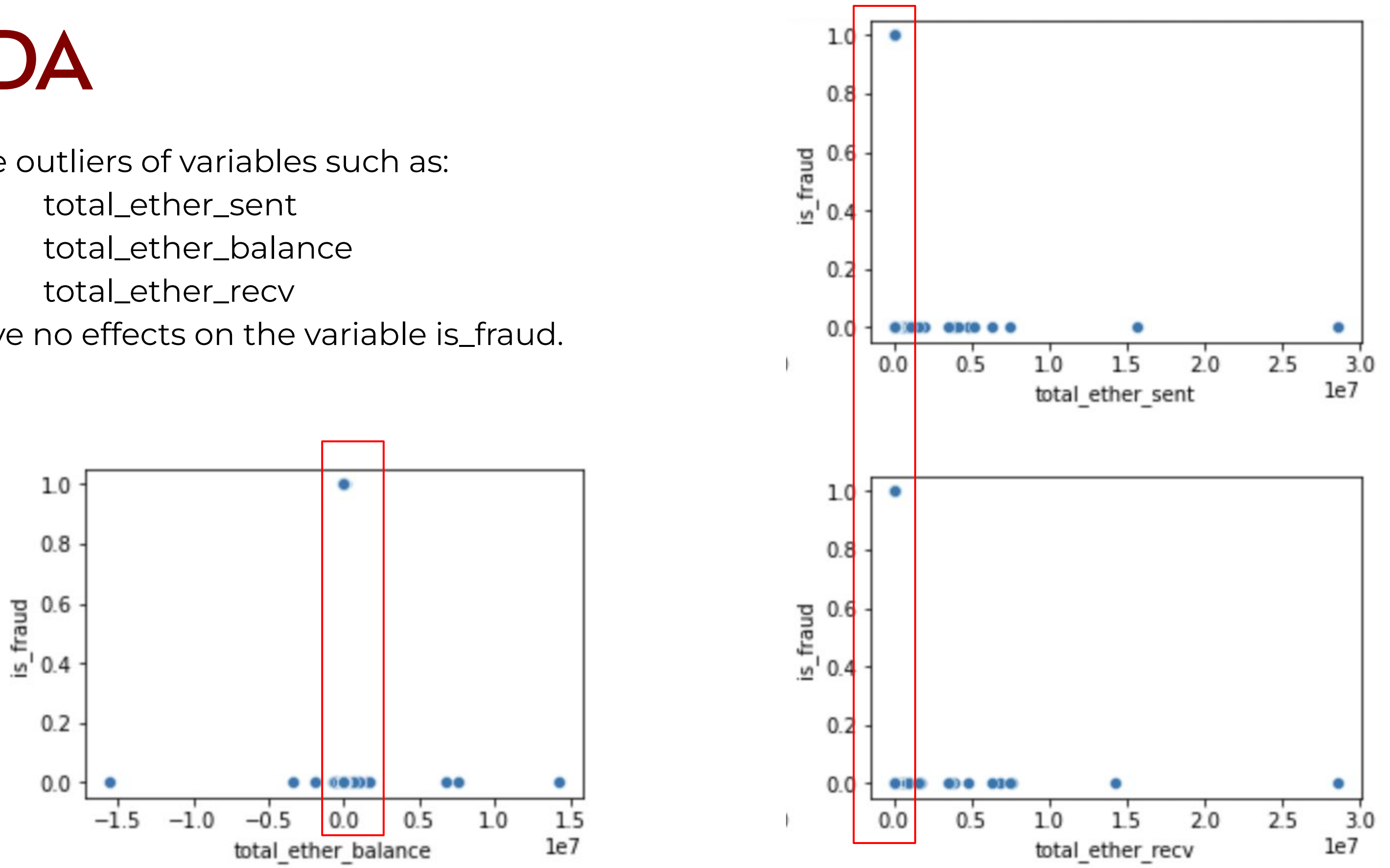


EDA

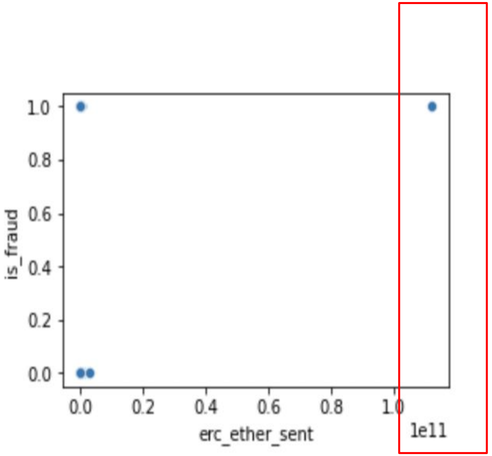
The outliers of variables such as:

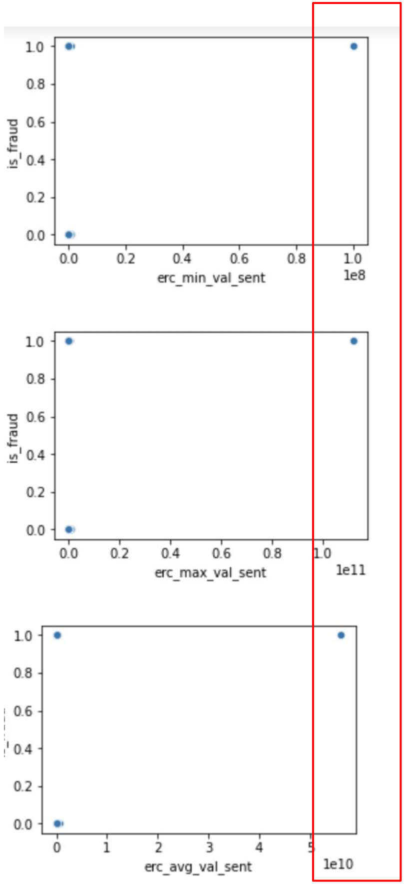
- total_ether_sent
- total_ether_balance
- total_ether_recv

have no effects on the variable is_fraud.



EDA



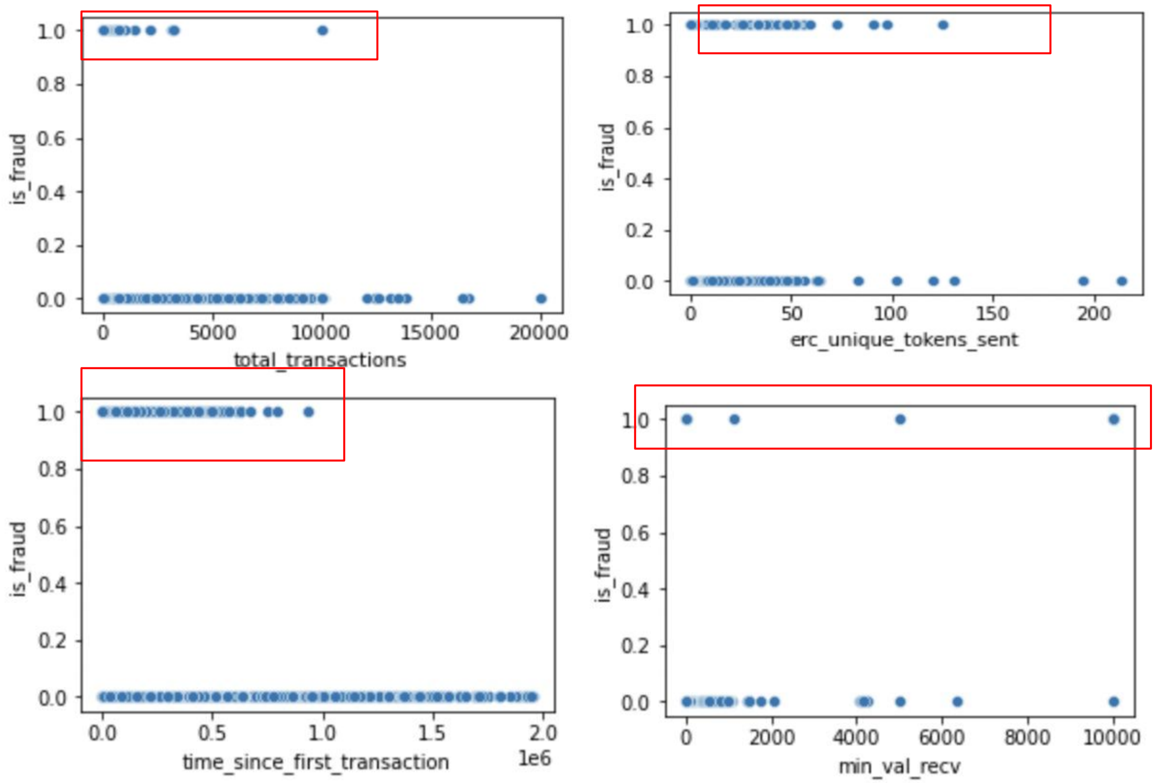


However, on the other hand, the outliers of variables such as:

- erc_ether_sent
- erc_min_val_sent
- erc_max_val_sent
- erc_avg_val_sent

have major effects on the variable is_fraud.

EDA



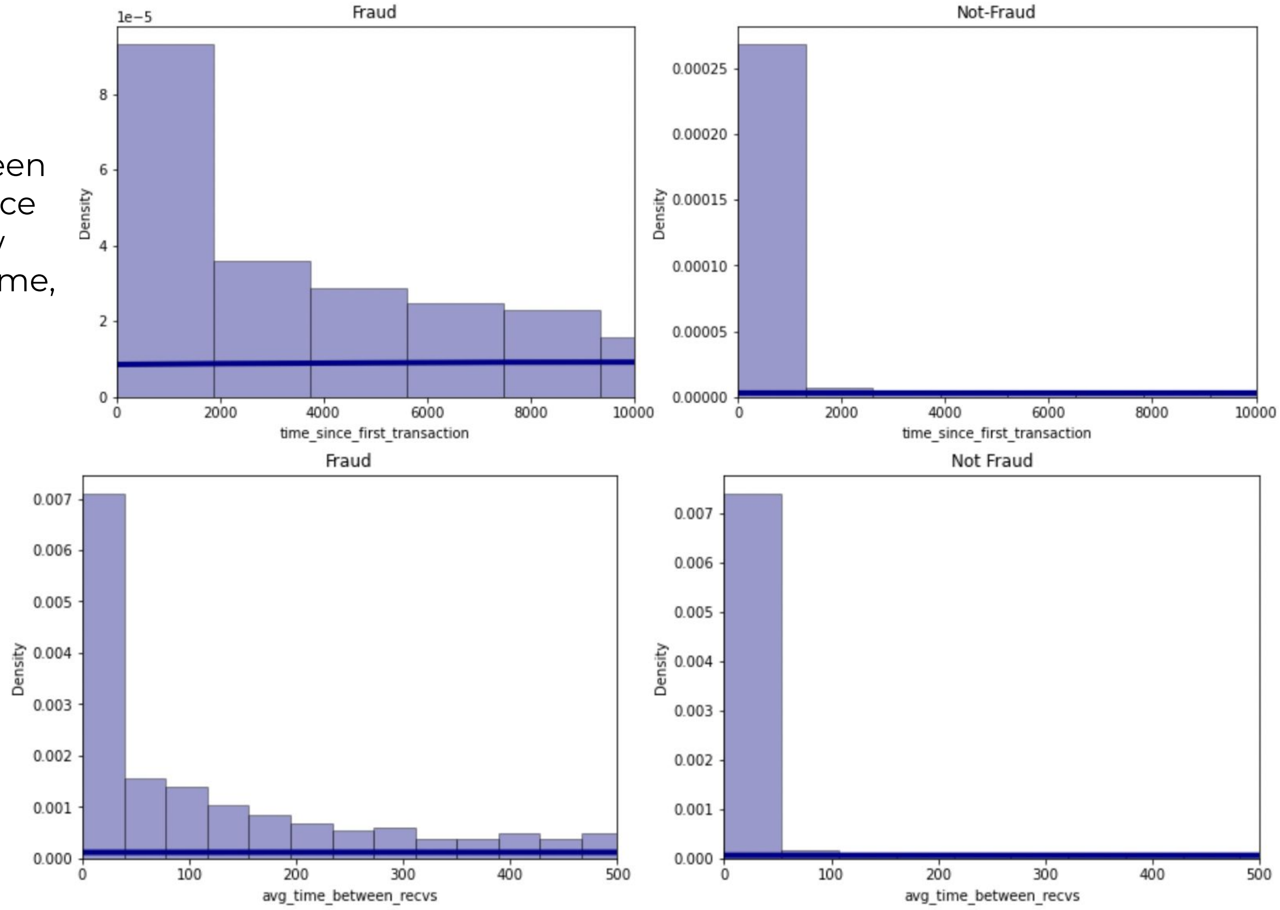
For variables such as:

- total_transcations
- erc_unique_tokens_sent
- time_since_first_transcation
- min_val_sent

their fraud data points and outliers are evenly distributed, so these variables outlier do not have big influence.

EDA

Both avg_time between receives and time since first transaction show that the longer the time, the more likely the transaction is fraud



ethereum

Feature Engineering

Feature Engineering

ERC20 Tokens

ERC20 is a standard protocol for smart contracts on the Ethereum blockchain. ERC20 tokens follow ERC20.

- Ether: USD
- ERC Tokens: credit card, vouchers, real world assets

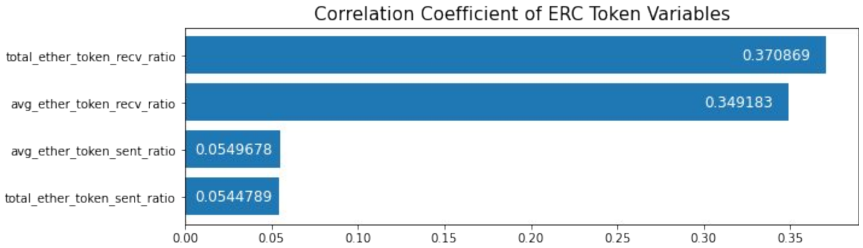
Comparing ERC Token & Ether Transactions

- Total Amount Sent
- Total Amount Received
- Average Amount Sent
- Average Amount Received

Feature Crossing

- Divide Ether value by ERC Token value
- Replace infinity values with -1
- Replace -1 with the maximum value of the column
- Fill NAs with 0

Correlation Coefficient of New Variables



Feature Engineering

ethereum

Most Sent/Received ERC20 Token

- Most sent and received ERC tokens for each account
- High cardinality (70 unique sent, 120 unique received)
- Missing/null data (829 missing values)

original

is_fraud	token
1	fraudcoin
1	fraudcoin
0	fraudcoin
1	fraudcoin

groupby

token	mean fraud
fraudcoin	0.75

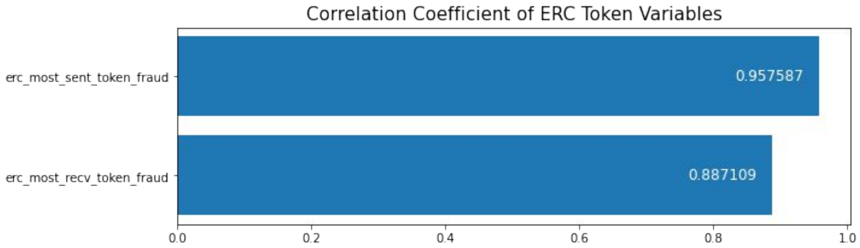
transformed

is_fraud	token
1	0.75
1	0.75
0	0.75
1	0.75

Converting String to Numeric Column

- Groupby most sent/received ERC token
- Calculate mean fraud rate for each token
- Left join new column with the original dataframe
- Drop the original column

Correlation Coefficient of New Variables



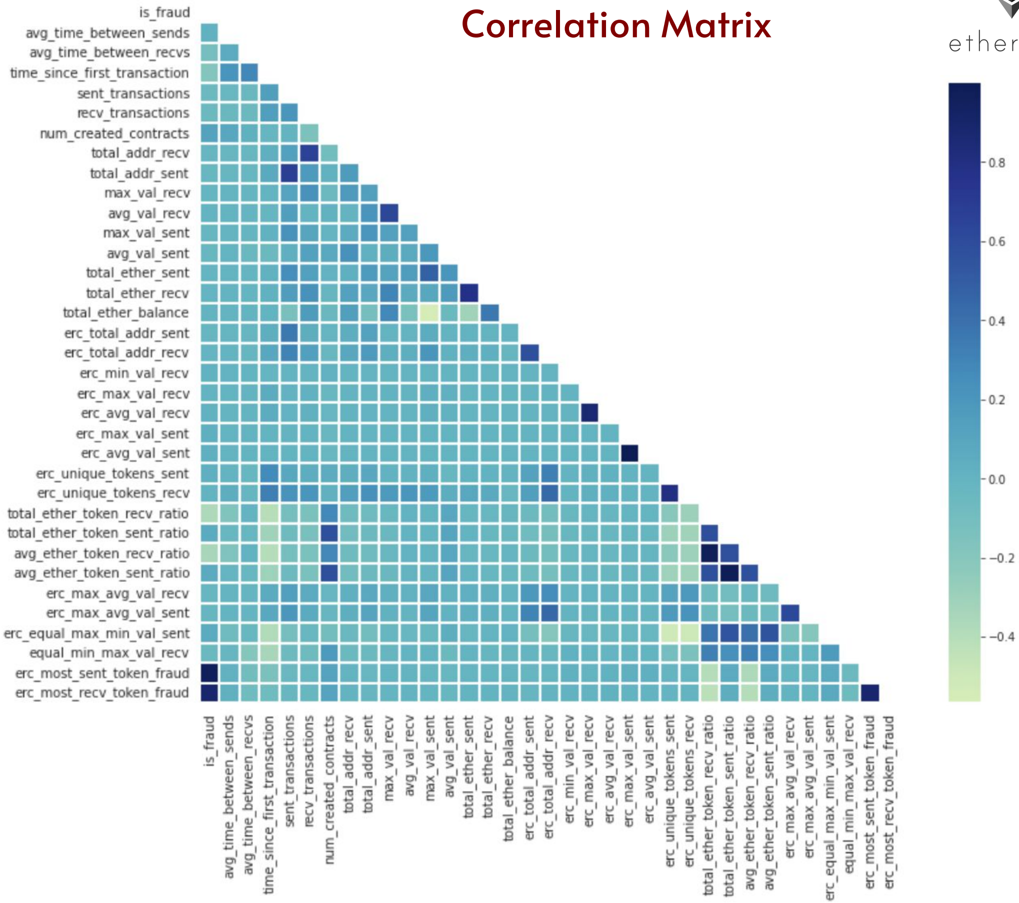
Final Dataset

Dataset Summary

	Original Dataset	Final Dataset
Columns	51	35
Corr > 0.7	21	9
Rows	9841	8746
Not Fraud	7662	7639
Fraud	2179	1107
Nulls	20759	0

- Rows dropped: 829
- Columns dropped: 25
- Features engineered: 9

Correlation Matrix



Data Balancing

SMOTE-ENN Resampling

ethereum

SMOTE Oversampling

- Choose random data from minority class (is_fraud)
- Calculate distance between random data and k-nearest neighbors
- Multiply difference with random number between 0 and 1, add result to minority class

ENN Undersampling

- Remove samples whose class label differs from the class of the majority of their k-nearest neighbors
- An observation that is not fraud is removed if its predominant k-nearest neighbor class is 'is_fraud'

Train Test Split

- Stratified shuffle split - 80% train, 20% test
- Training data **is** resampled and used for modelling
- Test data **is not** resampled and used for validation

Train & Test Size

	Train (Original)	**Train (Resampled)**	**Test**
Not Fraud	6111	4853	1528
Fraud	885	5727	222
Total	6996	10580	1750

Modeling

Model Comparison

ethereum

Model	Accuracy	Precision	AUC	Recall	F1
Logistic Regression	0.993	0.969	0.984	0.973	0.971
Decision Tree	0.991	0.960	0.981	0.968	0.964
Random Forest	0.992	0.995	0.970	0.941	0.968
XGBoost	0.992	0.995	0.970	0.941	0.968
SVM	0.994	0.977	0.985	0.973	0.975

Model Selection

ethereum

- We pick **Random Forest** as our final model.
- The advantages of Random Forest:
  - Highest Precision score
  - Flexibility in user-defined hyperparameters
  - High Interpretability
  - Use Ensemble Learning: Prevent Overfitting
  - Run in parallel: Computationally fast
- The disadvantages of Random Forest:
  - Little control over what the model does (Black-box)

Hyperparameter Tuning

ethereum

- We use **GridSearchCV** to find the best parameters for Random Forest Model
- Then we use the best parameters to train the final model

```
rf = RandomForestClassifier(oob_score = True, random_state = RANDOM_STATE, n_jobs = -1)
param_grid = {'n_estimators': [200, 500],
              'max_features': ['auto', 'sqrt', 'log2'],
              'max_depth': [2, 4, 6, 8],
              'criterion': ['gini', 'entropy']}
CV_rf = GridSearchCV(estimator = rf, param_grid = param_grid, cv = 10)
CV_rf.fit(X_train_scaled, y_smt)
```

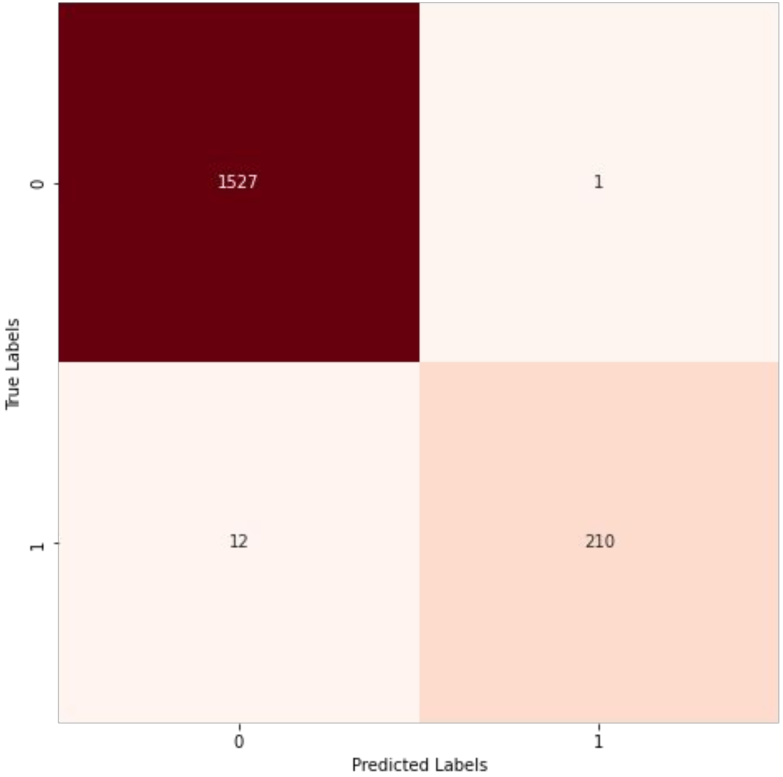
```
GridSearchCV(cv=10,
             estimator=RandomForestClassifier(n_jobs=-1, oob_score=True,
                                              random_state=42),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'n_estimators': [200, 500]})
```

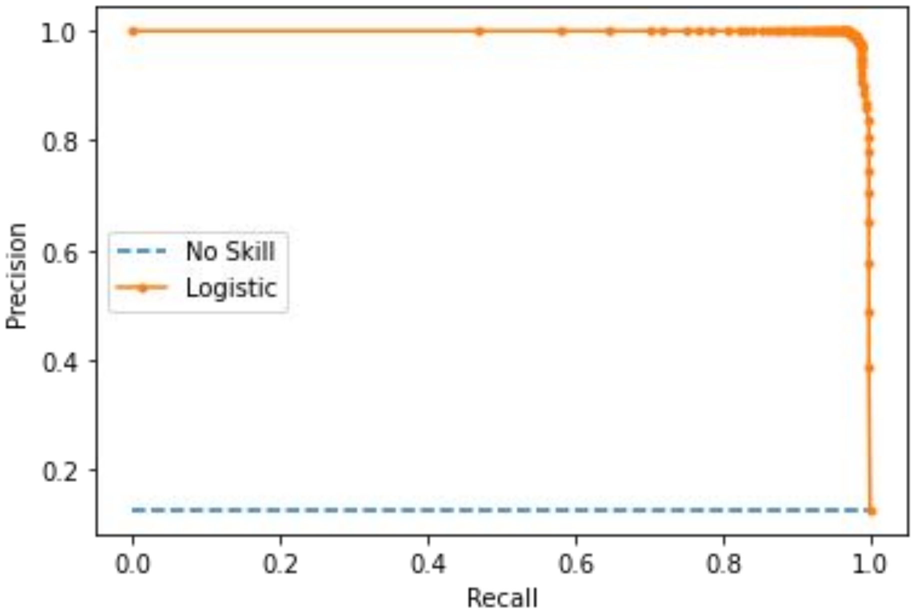
```
print(CV_rf.best_params_)
```

```
{'criterion': 'entropy', 'max_depth': 8, 'max_features': 'auto', 'n_estimators': 500}
```

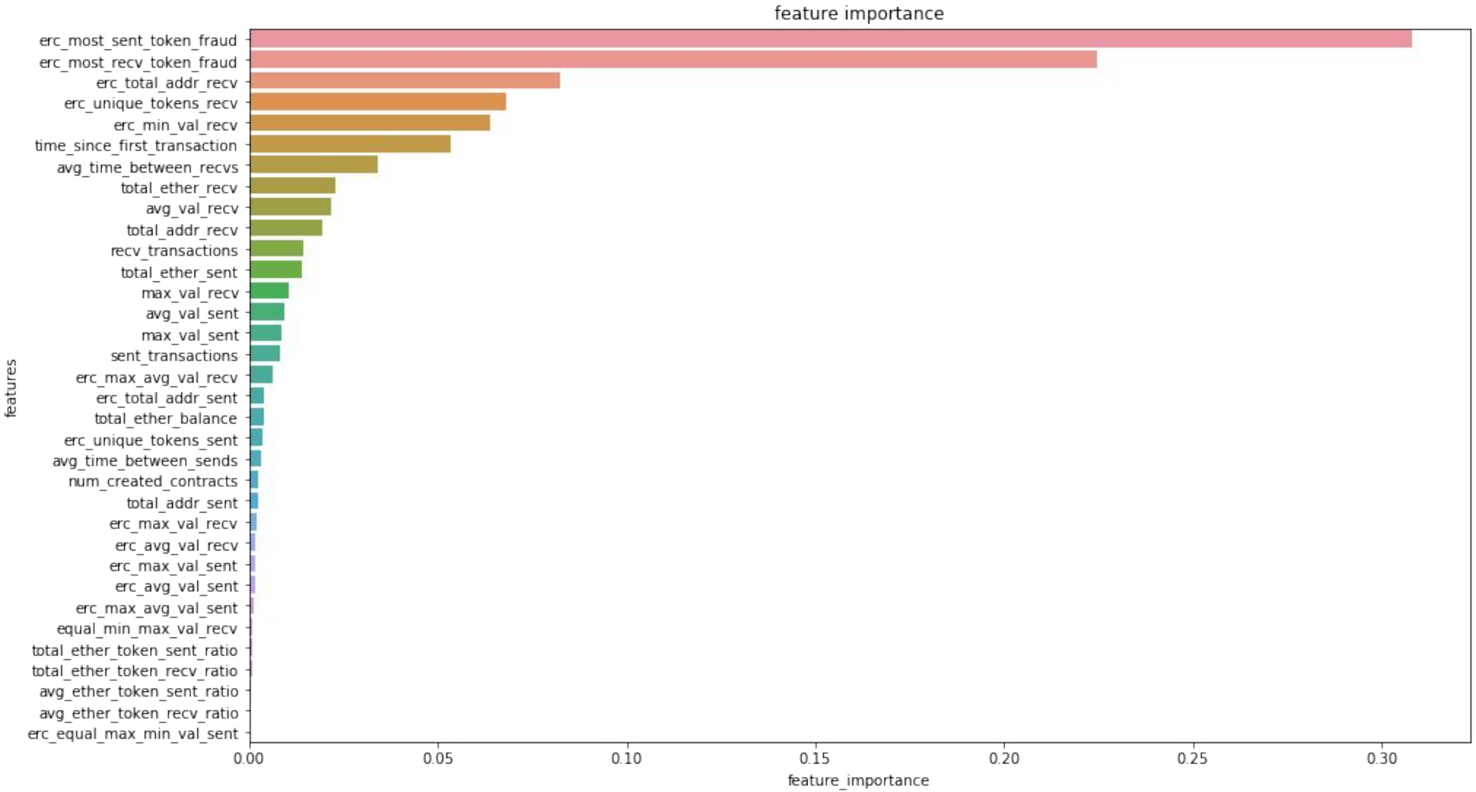
Conclusions

Final Results





Feature Importance



Improvements

ethereum

Runtime

Ethereum blockchain has an average block time of **13 - 14 seconds**, which means transactions must be verified quickly.

Accuracy

1.2 million transactions are processed on Ethereum every day, low error rates can still have significant repercussions.

Scalability

Solutions are being researched to increase the scalability of Ethereum, and allow **higher transaction throughput**.

Updatability

New blocks on the Ethereum blockchain currently average **170 transactions** of data which can be used for training.

LightGBM

Runtime

- Almost 10x speed of XGBoost
- Leaf-wise growth converges faster than level-wise

Accuracy

- High number of tunable parameters
- Similar or slightly higher accuracy than XGBoost

Scalability

- Supports parallel distributed learning
- Better suited to handle large datasets

Updateability

- Supports incremental training, saves time and resources
- Model can be continuously updated without retraining

Thank You!