Crypto Anomaly Detection

NYU Center for Data Science

Changhyun Lee, Howell Lu, Oussama Fatri, Victoria Xie

Supervised by Jon Wedrogowski

ripple

Abstract

In cryptocurrency markets, the price of crypto assets can diverge across markets due to numerous reasons (e.g. exchange downtime and trade volumes). Therefore, outlier detection is extremely important for ensuring that erroneous market data does not distort price feeds. This project aims to detect anomaly (outliers) in cryptocurrency prices across exchanges and cryptocurrencies using various models including Z-score thresholds, Logistic Regression, Random Forest, Ensemble Voting, and XGBoost. XGBoost achieves a highest AUC of 0.99. The final model could potentially be used by Ripple as an inference layer on top of various financial models to ensure data quality.

Research Question & Related Work

Objective: build a model to detect outliers in cryptocurrency transactions across different exchanges and cryptocurrencies.

Use Case: feed Ripple's financial models better quality data

Evaluation metric: AUC (assign a probability of being an outlier to each data point in the sample)

Past Literature: Commonly implemented applications of data validation include credit card fraud detection and and financial model outlier detection. Both are highly mature fields, and techniques from credit card fraud detection are commonly used to analyze anomalous wallet-to-wallet transactions of different cryptocurrencies. However, there is limited related literature regarding anomalous cryptocurrency exchange data.

Data Processing

Dataset:
- Minute by minute (time series) data from 41 exchanges
- Information about volume, pricing, commissions and exchange details
- Labelled data points which could generate arbitrage profits

Data Cleaning:
- Removed exchanges with substantial missing data

Labeling:
- Labeling was done through a LSTM model which predicts volume. Exchange tickers are labeled as anomalies if their price differed significantly from their predicted value

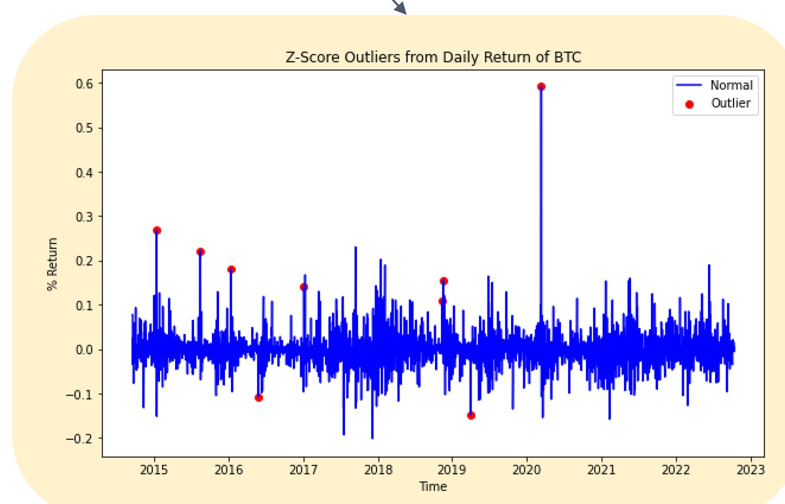
Feature Engineering:
- Identified and labeled transactions that could potentate arbitrage
- 60-20-20 train-val-test split for supervised models
- Calculated difference between high and low for minute intervals
- Encoded independent variables (crypto-exchange size, arbitrage-viability, exchange name, high-low difference, exchange size, and commission size) into numerical values
- Checked feature correlation - low

List of Features:
- Opening time
- Exchange at which the cryptocurrency is trading
- Base currency volume
- Counter
- Opening & closing price
- Highest & lowest price
- Volume of base that is available at the current price
- Taker fees
- Largest cryptocurrency exchanges based on 24h volume location
- Size
- Arbitrage (transaction price discrepancy that can generate profit)

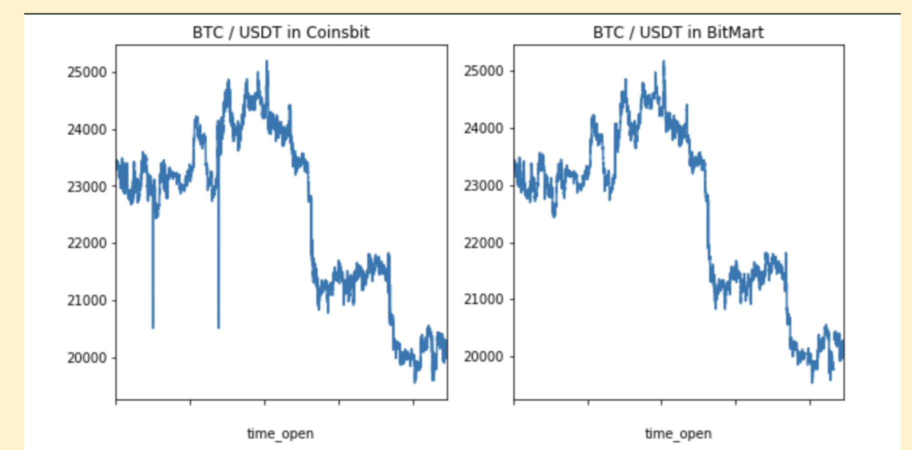
Definitions and Methods

Definition of Anomalies

- Outliers in % change of crypto prices
- Outliers in cryptocurrency volumes and prices
- Significant differences in exchanges' prices
- Triangular arbitrage between different currencies



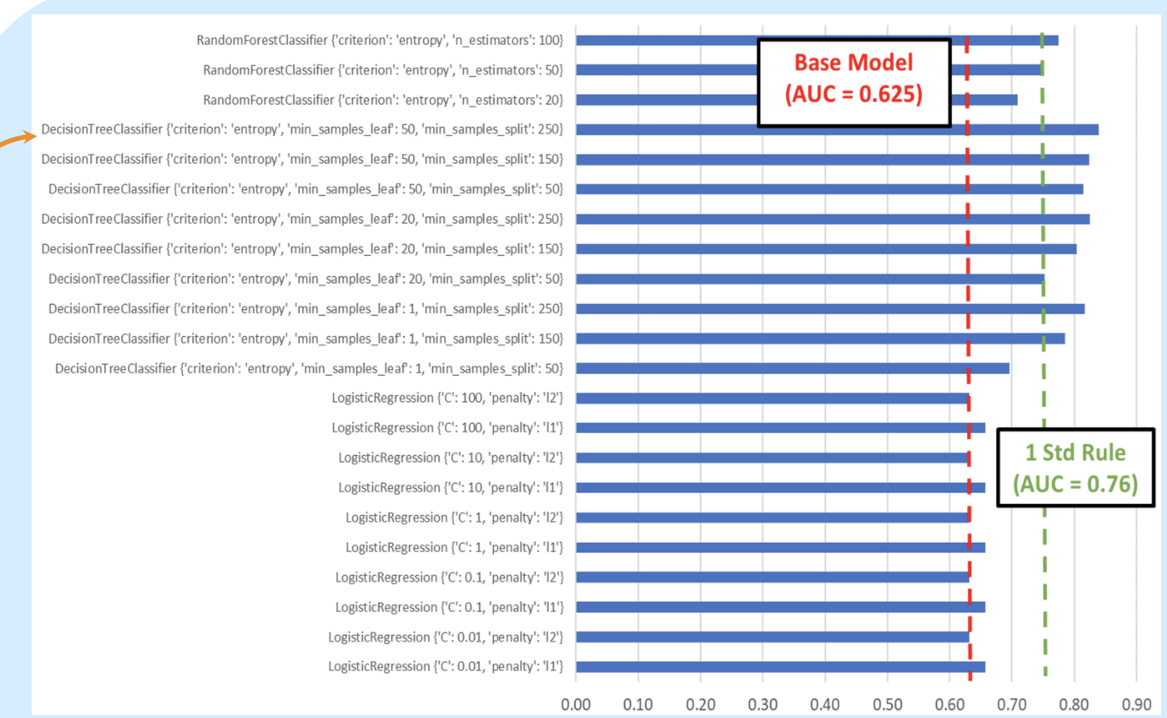
LSTM Time Series Model



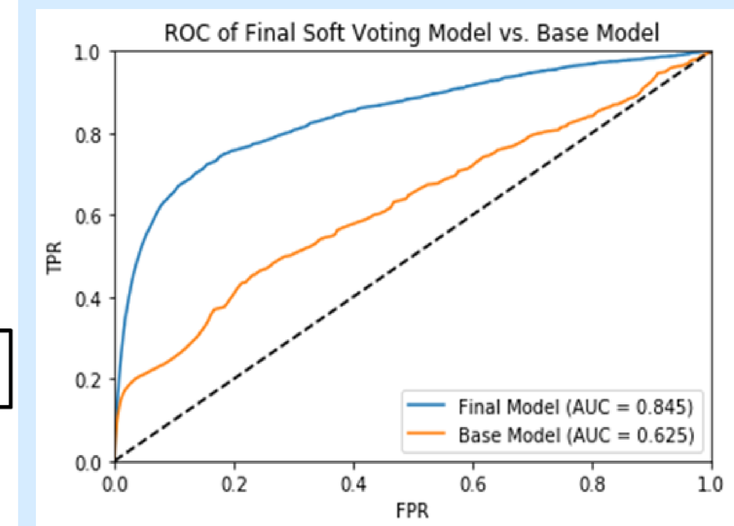
Supervised Classification Models
- Logistic Regression
- Random Forest
- XGBoost

Models and Experiments

Approach 1: Z-Score Labeling
- Logistic Regression, Decision Trees, Random Forest
  - Hyperparameter-tuning with 10-fold CV and Grid Search
  - Avoid overfitting using the one-standard-error rule
- Scikit-Learn Ensemble Voting
  - Combine logistic regression, decision trees, and random forest using soft voting
  - Predict with soft voting based on the argmax of the sums of the predicted probabilities

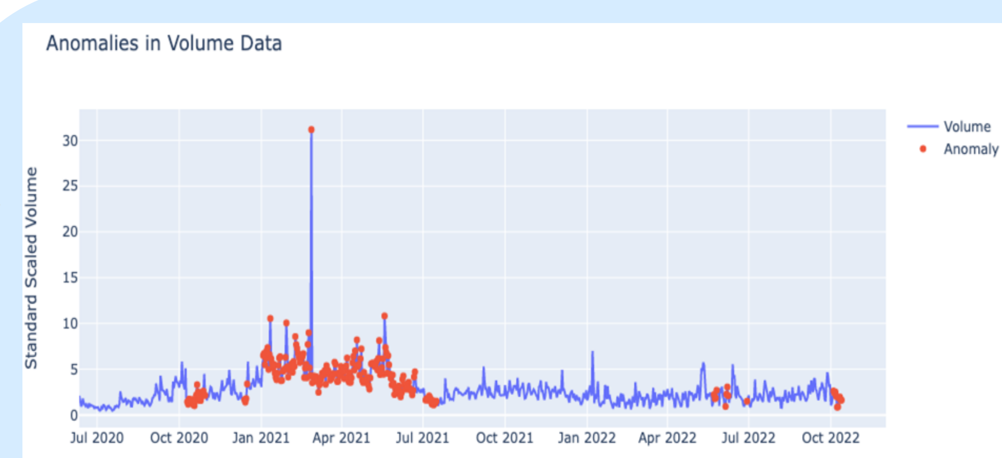


Grid Search and corresponding AUC

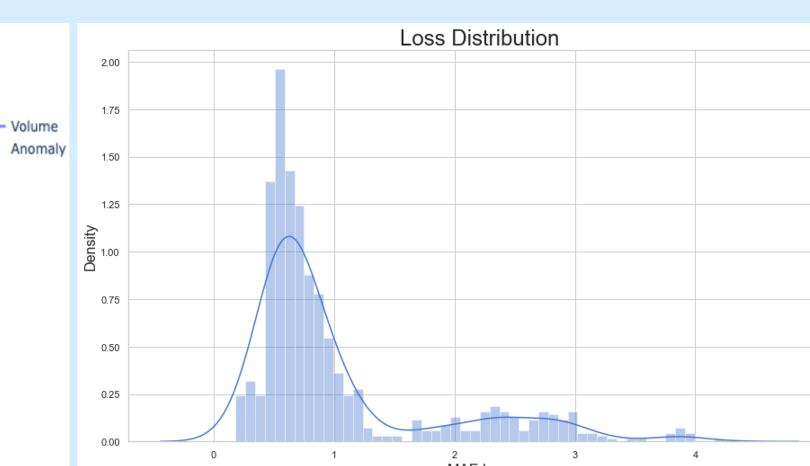


ROC of Baseline vs. Soft Voting Classifier

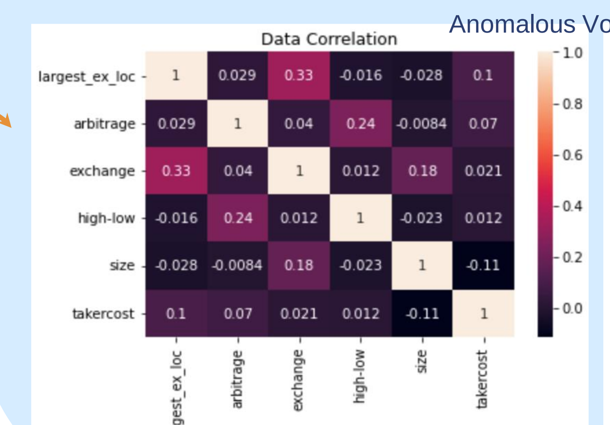
Approach 2: LSTM Labeling and Exchange Data
- Labeling with LSTM
  - Utilize anomalous volumes from each exchange
  - Two LSTM and two dropout layers
  - Optimize with MAE loss
  - Assign anomaly if loss exceeds z-score threshold
- Classification with XGBoost
  - Encoded categoricals; standardized continuous variables
  - CV with stratified K-fold
  - Random search parameter tuning



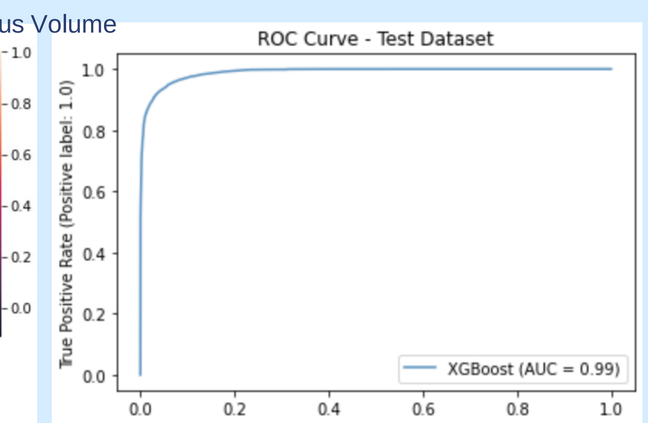
Anomalous Volume



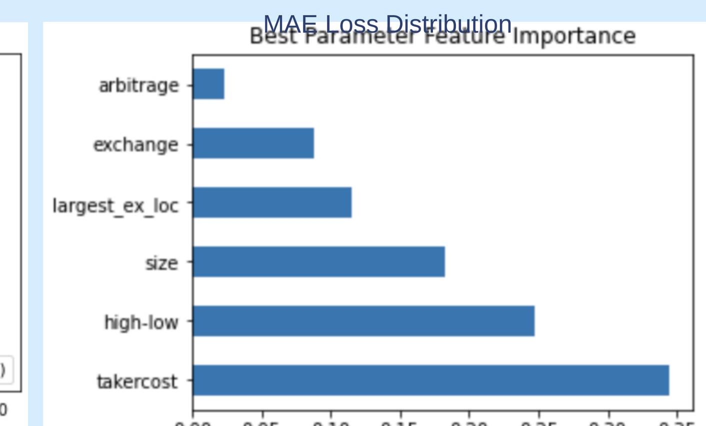
MAE Loss Distribution



Correlation among Features



Test Set Performance



Feature Importance

Results and Analysis

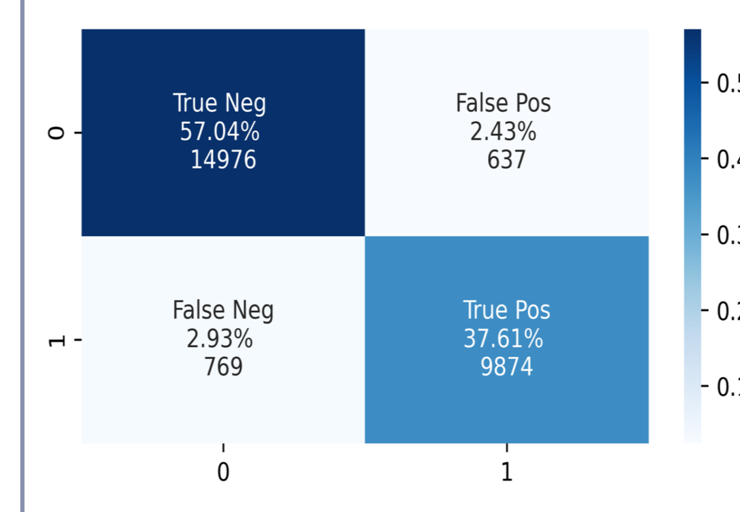
A horse race was ran with multiple machine learning algorithms, experimenting with logistic regression and ensemble models with varying specifications.
- More complex models improved performance
- Best model was XGBoost with this set of parameters: {'subsample': 0.8, 'min_child_weight': 10, 'max_depth': 5, 'gamma': 2, 'colsample_bytree': 1.0}

Model	AUC
Logistic Regression	0.653
Decision Trees	0.798
Random Forest	0.789
Ensemble Voting	0.845
XGBoost	0.99

Model Performance Comparison

	Precision	Recall	F-1	Support	Accuracy
Non-anomaly	0.95	0.96	0.96	15613	0.95
Anomaly	0.94	0.93	0.93	10643	

Evaluation Summary



Confusion Matrix for XGBoost

Results show a high level of accuracy, precision and recall in predicting anomalous and non-anomalous data. Further analysis showed the most important feature for estimating anomalous data are the *transaction commissions taken by exchanges*, further research is needed on feature attribution.

XGBoost successfully found the volumetric anomaly patterns using a small number of exchange-related features (i.e. location, size, price, etc.)

Conclusion

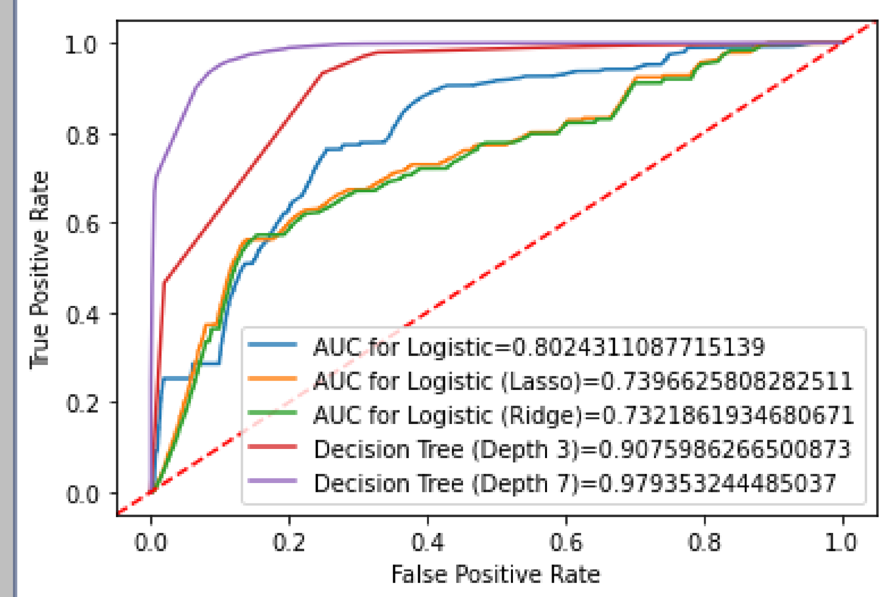
Due to the immutability of all blockchain transactions, firms that deal with blockchains such as Ripple need to be able to monitor and evaluate transactions in a timely and accurate manner. The cryptocurrency market is highly volatile, and there still exists many unregulated aspects in crypto exchanges and transactions. Our experiments show opportunities for abnormal behaviors from price, volume, and arbitrage. The classification models accurately determined anomalies based on Z-scores and LSTM. However, we also warn that these models are likely to fluctuate and their performance may decrease as all models incorporate historical data. Moreover, our labeling methods may not truly represent all anomalies as its definition is subjective. Nonetheless, we believe our models provide a strong foundation for crypto anomaly detection.

Future Work

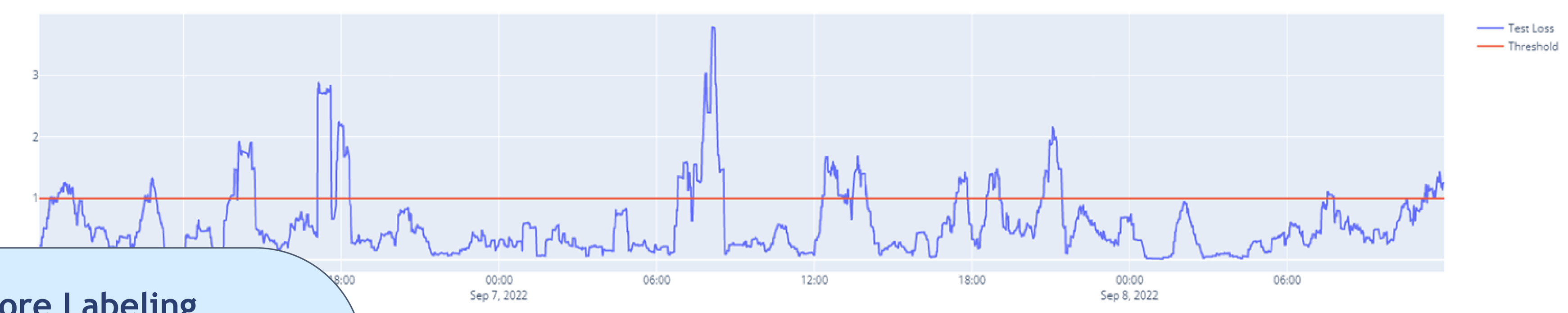
- Incorporate up-to-date data that reflects recent crypto market turbulence in order to expand training set diversity
- Backtest model on data from recently insolvent exchanges to see if model can utilize predictions of anomalous data to predict exchange insolvency
- Further expand list of features (e.g. commission tiers, tenure, volatility, volumes in context, etc.)
- Experiment with clustering algorithms such as K-means
- Further investigate attribution of each feature and look at potential interactions between features

1. **Baseline: Logistic Regression**
   a. Default parameters w/o tuning
2. **Decision Trees (Random Forest)**
   a. Slight improvement
   b. Hyperparameter-tuning with 10-fold CV and Grid Search
   c. Apply the one standard error rule to avoid overfitting
3. **Ensemble Voting**
   a. Combining logistic regression, decision trees, and RF using soft voting
   b. Significant improvement
4. **XGBoost**
   a. CV with stratified K-fold and random search
   b. Most important features include taker fees, difference between highest and lowest price points, and exchange size

Plots:

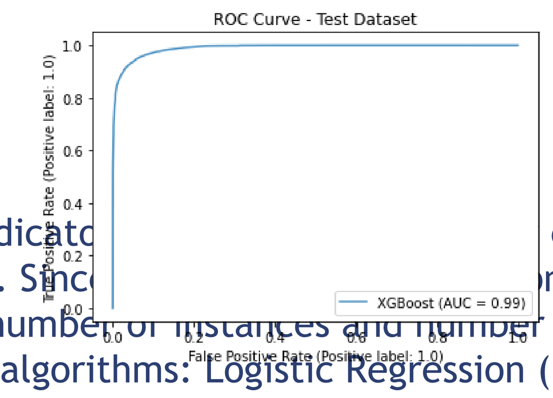


Ensemble Learning Improvement over Base Model



Approach 1: Z-Score Labeling

- Logistic Regression, Decision Tree, Random Forest
  - Hyperparameter-tuning with 10-fold CV and Grid Search
  - Avoid overfitting with one standard error rule to
- Scikit-Learn Ensemble Voting
  - Combine logistic regression, decision trees, and random forest using soft voting
  - Predict with soft voting based on the argmax of the sums of the pred_proba



From exploratory data analysis, we observed that the target variable (binary indicato[...] correlated to the base currency volume and the difference between the high and low prices for that timestamp. Sinc[...]nstraints that dictate what type of algorithms we should be using - our dataset has a manageable size both in the number of instances and number of features dimensions - we ran a horse race between different algorithms. We first explored the following three algorithms: Logistic Regression (baseline), Decision Trees, and Random Forest. For each one of these, we ran a grid search over an appropriate range of hyperparameters, and picked the specification using the "one-standard error rule". Next, We decided to go one step further and chose the best model within each family of learning algorithms so far and combine them using the Scikit-Learn ensemble soft voting classifier. [Next: xgboost…]

- However, upon seeing the results from decision trees, it was obvious to build on from decision trees to ensemble methods. XGBoost was used as it allows for boosting, incrementally improving the results of each decision tree and squeezing out all possible improvements.

Metric	Precision	Recall	F-1 score	Support
Non-Anomaly	0.95	0.96	0.96	15613
Anomaly	0.94	0.93	0.93	10643
Accuracy			0.95	26256
Macro Average	0.95	0.94	0.94	26256
Weighted Average	0.95	0.95	0.95	26256

	Precision	Recall	F-1	Support	Accuracy
Non-anomaly	0.95	0.96	0.96	15613	0.95
Anomaly	0.94	0.93	0.93	10643	