Mining Pool Detection and Identifying Anomalies.

Problem Statement : Bitcoin provides an extra level of anonymity for the identities of the users, since the user addresses are hashed. A Mining pool is the pooling of resources by miners, who share their processing power over a network, to split the reward equally, according to the amount of work they contributed to the probability of finding a block. I attempt to create a classifier model which identifies the mining pools from other users and then detects anomalies in these mining pools.

Dataset : The dataset for this purpose can be obtained from Big Query API provided by GCP. Google Cloud released datasets consisting of the blockchain transaction history for Bitcoin to help you better understand cryptocurrency. We can make different SQL queries at Google Big Query API and download the data as Csv files which can be used for further analysis. In this case, I have modified an example query given as an example (https://gist.github.com/allenday/16cf63fb6b3ed59b78903b2d414fe75b) in order to download the required numerical attributes and the addresses. After analysing all the available attributes, I shortlisted 21 attributes(including address) to be important for our purpose. The target variable "is_miner" is True for miner addresses and false for non-miners. I downloaded 2000 miner addresses and 12000 non-miner addresses for training purposes.

Model Selection and Training: I excluded the address column and two other columns (standard deviation of output idle time and input idle time per address) as these columns had nan values. Next, we split the data into training and test dataset and I train an Extreme gradient boosting Classifier model on the remaining attributes for the training dataset. Then this XGBClassifier model is used to classify the miner pool addresses from the test data.

Next, the address instances from test data which are classified as miners are taken separately to test for anomalies i.e. the test examples which are classified as miners are used to check for anomalies.

An Isolation forest model is used which classifies the selected miner dataset to detect anomalies. In order to get a better result, I used all the attributes so that any kind of outlier anomalies can be detected. The outlier percentage or the contamination was given to be 0.02, I.e. the outlier 2% of the miner datset are considered as anomalies.

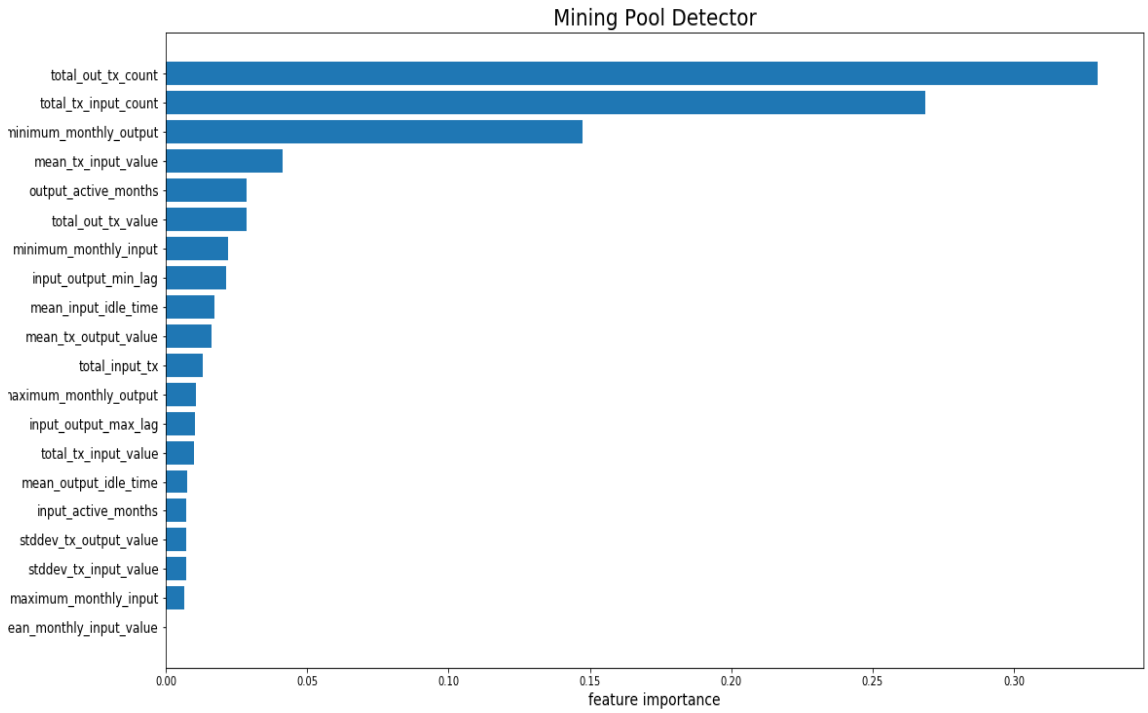
Results: Out of the 14000 entries that were collected, a train-test split was done at 0.4 of the total entries being test data. On predicting the miner/non-miner classification using the test data (5600 entries) with the help of the XGBClassifier model, we find the confusion matrix to be as follows,

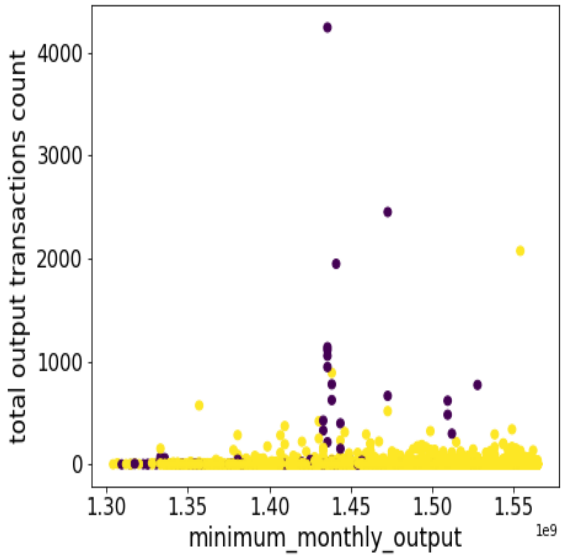
		Predicted Values	
		Non-mining pool	Mining pool
True values	Non-mining pool	4741	28
	Mining pool	21	810

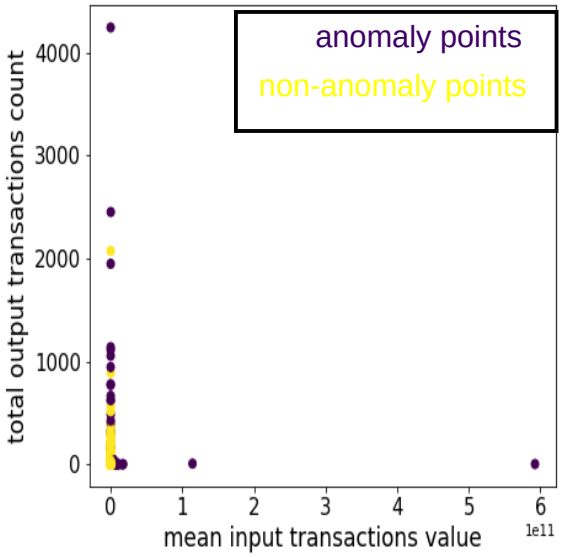
We find the final accuracy to be **99.125**%

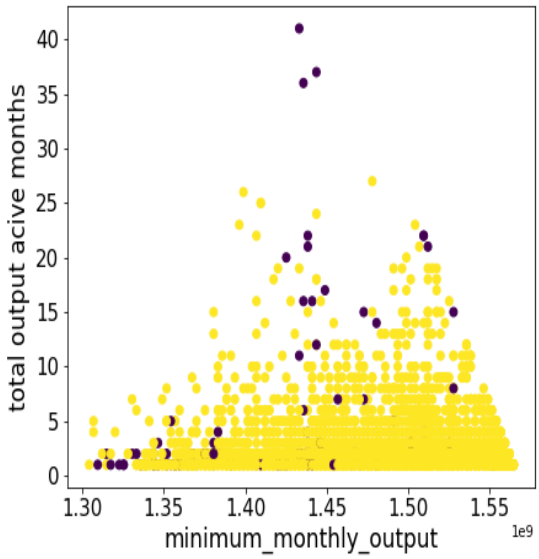
We can find the features which add most value to the classifier by plotting feature importance with respect to each of the features and we find that features: total output transactions, total input transactions and minimum monthly output value are the most important ones.

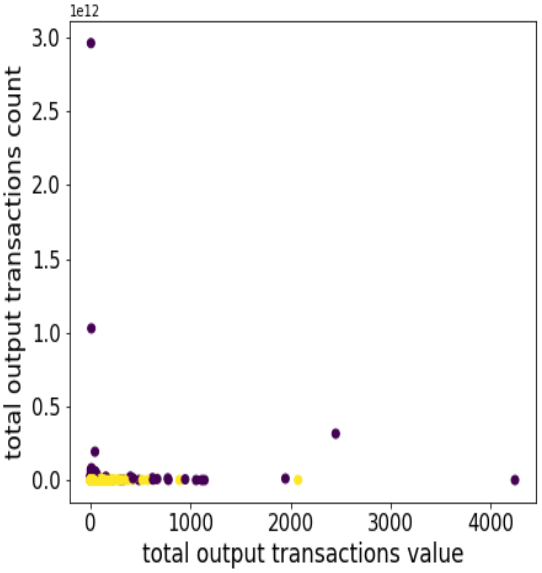
For Anomaly Detection, we found **2**% probable anomalies from the **838 miners** we found in the test dataset. From the plots below, we can see that these points are outliers considering any attributes combined. **The purple points are anomalies.**


Mining Pool Detector
total_out_tx_count
total_tx_input_count
minimum_monthly_output
mean_tx_input_value
output_active_months
total_out_tx_value
minimum_monthly_input
input_output_min_lag
mean_input_idle_time
mean_tx_output_value
total_input_tx
maximum_monthly_output
input_output_max_lag
total_tx_input_value
mean_output_idle_time
input_active_months
stddev_tx_output_value
stddev_tx_input_value
maximum_monthly_input
mean_monthly_input_value
0.00
0.05
0.10
0.15
0.20
0.25
0.30
feature importance


total output transactions count
4000
3000
2000
1000
0
1.30
1.35
1.40
1.45
1.50
1.55
1e9
minimum_monthly_output


anomaly points
non-anomaly points
total output transactions count
4000
3000
2000
1000
0
0
1
2
3
4
5
6
1e11
mean input transactions value


total output acive months
40
35
30
25
20
15
10
5
0
1.30
1.35
1.40
1.45
1.50
1.55
1e9
minimum_monthly_output


1e12
total output transactions count
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1000
2000
3000
4000
total output transactions value