CLUSTERING ETHEREUM BLOCKCHAIN SMART CONTRACT TOKEN BASED DECENTRALIZED FINANCE (DEFI)

RAJASEKARAN PARAMASIVAM
MSC DATA SCIENCE (DS)
STUDENT NUMBER 931058

Ethereum is a decentralized blockchain, let you send cryptocurrency to anyone for a small fee. Ethereum is a distributed public blockchain network comprises blockchain ledger, native cryptocurrency Ether (ETH) and robust ecosystem, aims to build a decentralized everything more democratized and more secure environment to the existing client-server models.

Ethereum blockchain built based on 3 pillars Decentralization, transparency and immutability. To run smart contracts, it must compile them to the low-level byte code that runs in the Ethereum Virtual Machine (EVM). It deploys them on the Ethereum platform using a special contract creation transaction once compiled.

Decentralized Finance (DeFi) smart contract token protocol built on Ethereum to build Money Legos breaks down into protocol layer: lending, debt Positions, debt Markets, derivatives, payments and assets.

Introduction

Smart contracts execute predetermined terms and condition, using tokens between addresses. Token value counters stored in a smart contract.

Tokenized assets and asset management are a booming sector of DeFi. Existing financial assets deployed to the blockchain as tokens fit nicely into DeFi protocols which extend their utility.

Asset management protocols allow investors to put their money in the hands of smart contracts or fund managers which manage their portfolio.

Introduction

Problem Statement

Ethereum blockchain program built based out of decentralized feature the transactions data stored on multiple devices in multiple location and data stored on the blockchain is converted into cryptography in such a way data cannot be easily understand, changed, forged or altered because of which limited analysis performed. There is prime need of analysing transaction level data.

Nobody really wants to send money to (or receive money from) a long hexadecimal number. This is where the Ethereum Name Service (ENS) comes in. ENS translates human-readable names to Ethereum addresses (and back)

The COVID-19 pandemic outbreak escalate surge in stock price across the globe. The increased market uncertainty led many diversify their investment into cryptocurrency as an alternative because of emerging growth of Decentralized Finance (DeFi) on Ethereum Blockchain, compare with tradition in financial markets

Aim & Objective

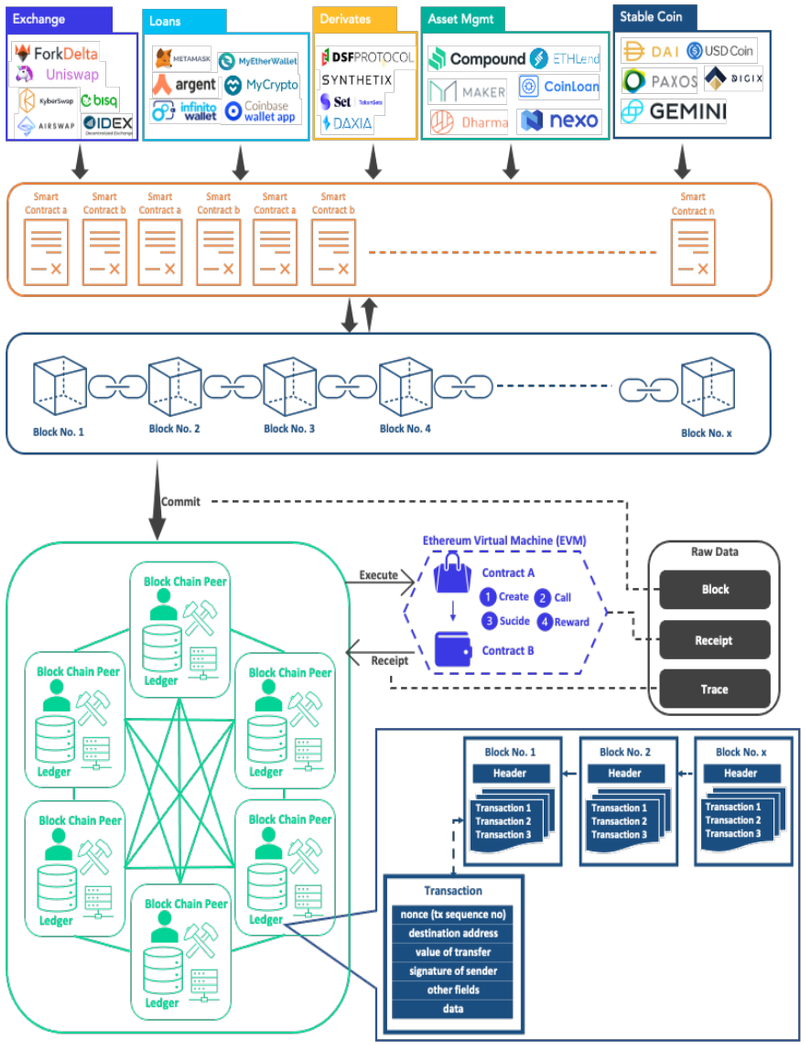
To suggest a suitable Smart Contract Token Cluster profiling and Multiple Classifier along with Class imbalance for Ethereum decentralized financial transaction analysis.

To analyze the pattern and relationship of Ethereum decentralized financial Smart Contract token transaction types Exchange, Lending, Assets management and Stable coins for the list of Ethereum External owned accounts (EOA's) first-time using Google Big Query Public Dataset

To propose feature engineering techniques by converting large data sets into smaller ones containing fewer variables, it helps in improving model performance, visualizing complex data sets, and capture the covariance or the correlations between the columns.

To compare between the classifier model technique to identify the best accurate model to classify Ethereum decentralized financial Smart Contract Token.

To evaluate the performance of the classifier based on the balancing techniques.



Literature Review

Vitalik Buterin's introduced Ethereum blockchain built based on 3 pillars: Decentralization, transparency and immutability. To run smart contracts, it must compile them to the low-level byte code that runs in the Ethereum Virtual Machine (EVM). It deploys them on the Ethereum platform using a special contract creation transaction once compiled.

Digital currency is currency available only in Digital or electronic form. In blockchain encrypted private key used to allow transaction and public key used for verification purpose.

Block Chain Transaction is in hashing data type representing contains set of inputs and outputs. One Blockchain can use transaction outputs as an input.

Externally Owned Account (EOAs): Participate in Ethereum network the private key is generated for each user account to perform transactions

Smart contract contains a set of rules built using Solidity programming language. Blockchain EVM Smart Contracts get executed when the required conditions met.

Bitcoin or Ethereum network Miner nodes get reward in Digital coins for the storing full ledger data and perform heavy computations to verify transaction.

Block constructed with a collection of Transactions. Each block comprises a collection of transactions with fields From address, To address in hashing format, Value amount transferred in Wei as a hexadecimal value and Input byte array data to this execution

Trace: Is detailed blockchain transaction activity captured contract execution in EVM at run-time. Trace categorized based on activities performed: 1. Smart Contract "Create" used to creator code and initial balance. 2. "Call" occurs during they transfer messages or ether between Ethereum addresses. 3. Suicide action occurs when refund the value and delete Smart Contract for given account. 4. Reward represents miners get the Ether reward for block mined.

Literature Review

Price prediction analysis being performed with several comparative study and there were no transaction related analysis or study being performed

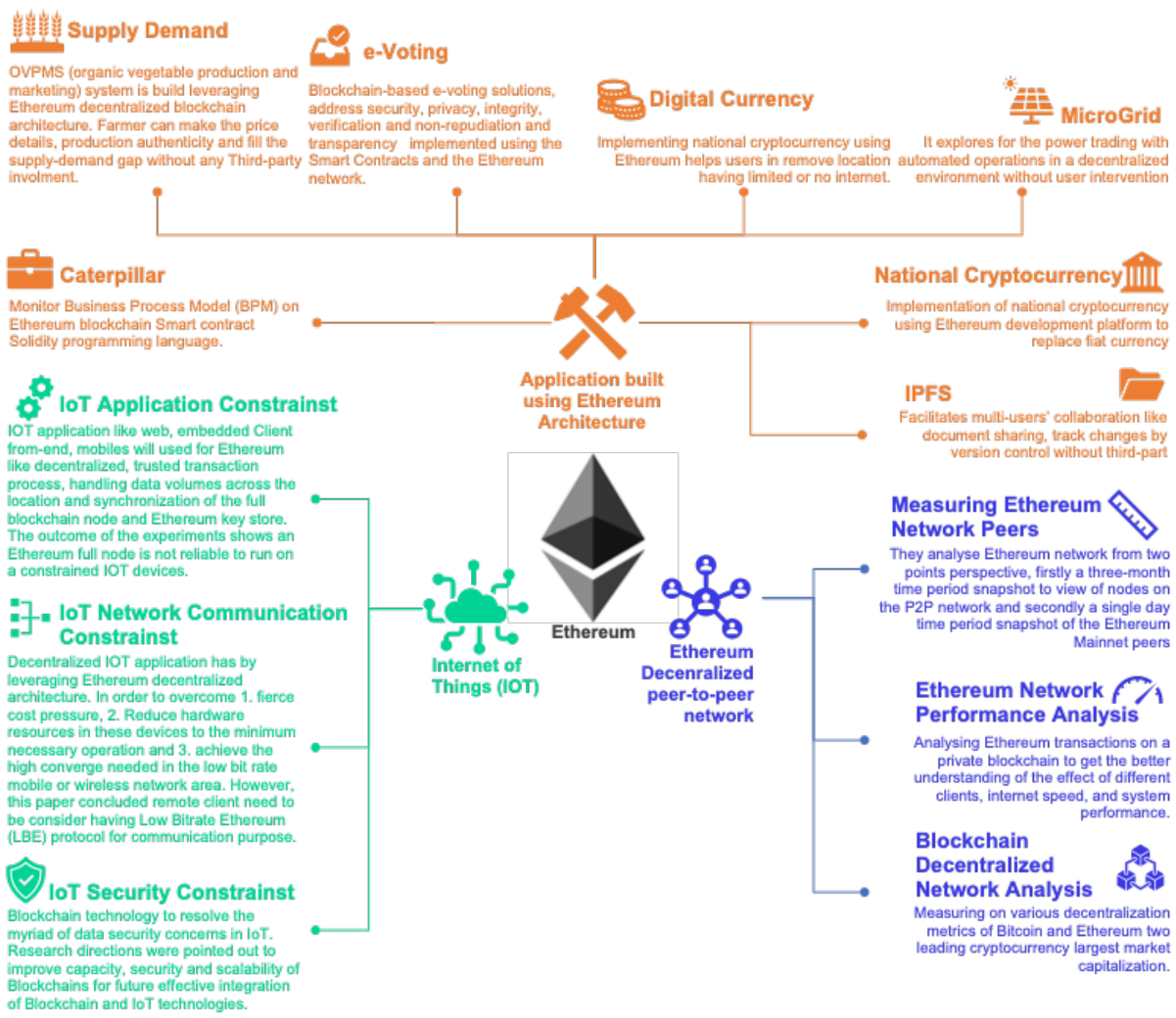
Lead-lag effect, especially in between Bitcoin and Ethereum. It describes the situation where one leading variable in cross-correlated with the values of another lagging.

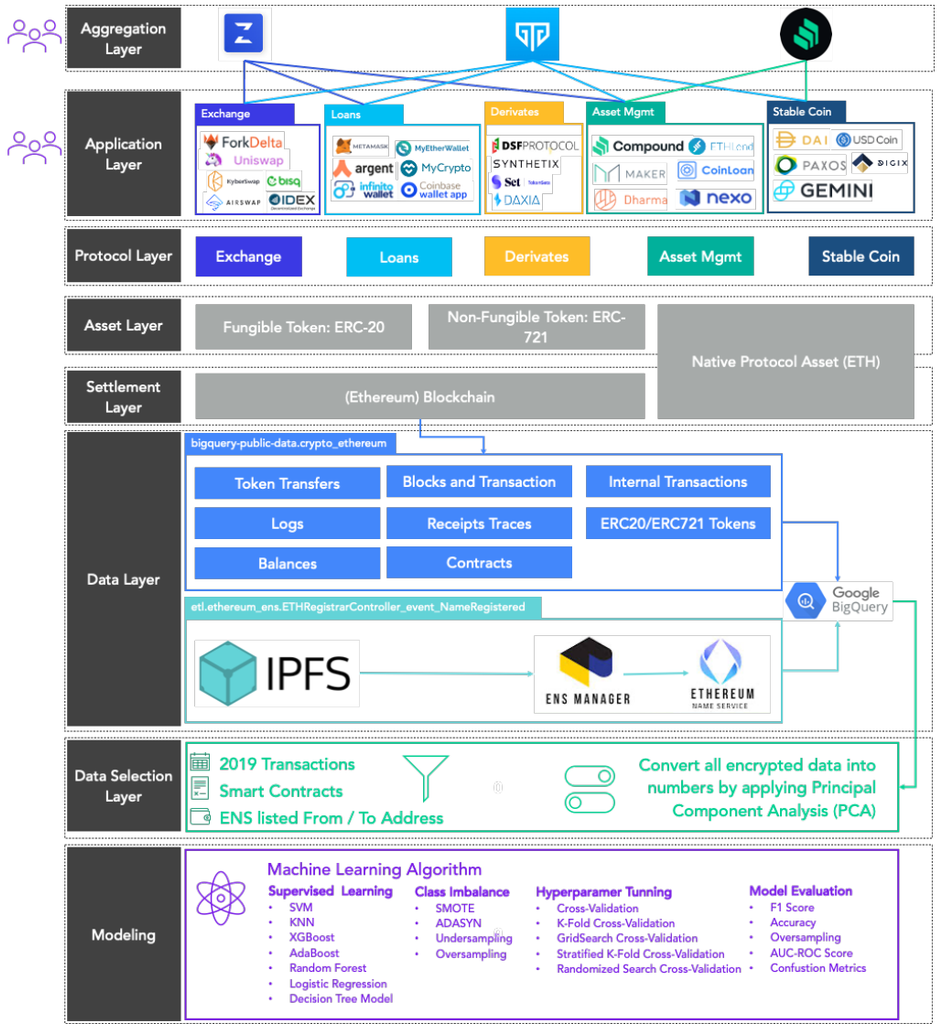
Traditional studies information flow between Bitcoin and Ethereum verses assets, Stock markets, ICO, Foreign exchange, Sentiment analysis, news, social media effect the price of Bitcoin and Ethereum

Model Method	Findings
Clustering Model	• Bitcoin price is inefficient however certain exchange trade are price clustering.
ARCH, GARCH, EGARCH, Threshold GARCH, IGARCH, DCC GARCH Model	• Bitcoin uncertainty about the future price of a commodity, share, or other financial product led to positive stock returns. • Bitcoin shows the capability of fence or boundary formed but had no effect in price movement due to other variables. • Bitcoin and Gold have similarities, sensitive to foreign exchange rate. Increase in interest rate demand on Bitcoin increases. No effect on Bitcoin returns based on mainstream media news.
Var and Granger:	• There is a price impact between the coins. • There is a negative relationship between Initial Coin Offerings and Bitcoin and Ethereum price, Bitcoin influence Ethereum price, Initial Coin Offerings affect Ethereum price. • Bitcoin exchange trade variables BTC-e and Mt.GOX are leader and others are followers. • Bitcoin is sensitive to public announcement, price and returns increase based on mainstream online media forums' messages and social network have positive comments. • Bitcoin is sensitive to good news and bad news in context with cryptocurrency. Bitcoin price and transaction increase based out of positive comments made in Ripple forum.
Ordinary least square (OLS) method	• Bitcoin and Ethereum cryptocurrency price are sensitive to good news and bad news made in mainstream media. • Altcoin miner have greater profitability. Cryptocurrency variables hardware, electricity, mining difficulty decrease then cost of production decreases
CNN, MLP, CLSTM, SVM, Random Forest	• Cryptocurrencies Bitcoin, Ether and Litecoin provide good results in CNN neural networks • This approach provides an accuracy of up to 99% for Bitcoin and Ethereum price prediction.

Literature Review

Ethereum blockchain program built based out of decentralized feature the transactions data stored on multiple devices in multiple location and data stored on the blockchain is converted into cryptography hash function mathematical algorithm in such a way data cannot be easily understand i.e. non-human-readable, changed, forged or altered because of which limited analyzing limited to peer-to-peer network analysis, leveraging Ethereum architecture in IoT, Building applications using Ethereum network or Smart Contracts, Detect Smart Ponzi or Phishing schemes, Lead-lag effect, especially in between Bitcoin and Ethereum, and Ethereum price movement or transaction prediction using ML model.





Methodology

- The aggregation layer is a combination of Application Layer to build Decentralized Financial (DeFi) Money Lego, can be handpick stand-alone product to easily integrated to build a new financial tool and accessed by any user.
- The application layer developer can build and launch a third-part Decentralized Application (DApps) that connects to individual protocols and accessed by any user.
- The protocol layer dictates the terms of contract and control the execution of Decentralized Financial (DeFi) like exchanges, debt markets, derivatives, Asset management and Exchanges. The protocol standards are achieved by combination of smart contracts Decentralized Application (DAppss) accessed by any user.
- The asset layer comprises many types of Ethereum tokens keep peer-to-peer trading in check. This includes synthetic assets and any additional token standards supported by the Blockchain.
- The settlement layer smart contract transaction processed by executing terms of contracts using computer code by developer and transactions data stored on multiple devices in multiple location converting into cryptography hash function mathematical algorithm in such a way data cannot be easily understand i.e., non-human-readable, changed, forged or altered.

Methodology

Data Layer: Ethereum ETL project on GitHub lets you convert blockchain data into convenient formats like CSVs and relational databases and load it into BigQuery Public dataset.

- ETHRegistrarController_event_NameRegistered: list of all ENS (~14673 records) from Google BigQuery Public Dataset etl.ethereum_ens
- crypto_ethereum.Blocks: It constructs Block with a collection of Transactions
- crypto_ethereum.Contracts: list Address of contract, byte code the EVM can natively execute, function signature, and ERC20 and ERC721 flag indicator and block number
- crypto_ethereum.Logs: A log record can be used to describe an event within a smart contract, like a token transfer or a change of ownership.
- crypto_ethereum.token_transfers: Token transfers capture only ERC20 (Ethereum Request for Comments) that transfer tokens between address, amount transferred, Transaction hash and block number.
- crypto_ethereum.tokens: Table list all ERC20 (Ethereum Request for Comments) Tokens created in Ethereum network and unique symbol and name associated along with decimal need to use total number of tokens supplied and the block number used in that transaction.
- crypto_ethereum.traces: Is detailed blockchain transaction activity captured contract execution in EVM at run-time. Trace is categorized based on activities performed
- crypto_ethereum.transactions:: Transaction identifier is in hashing data type representing contains set of inputs and outputs. One Blockchain can use transaction outputs as an input to another block transaction called Internal Transaction.

Cluster profiling:

- PCA for dimensions reduction
- SSD evaluation metric to test clustering algorithms
- Elbow curve shows good k number
- Silhouette analysis explains separation distance between the clusters
- Hopkins used to measure the cluster tendency.
- Clustering performed using K-Mean, Hierarchical.

After extracting the feature, 24 combinations of

- machine learning classifier Decision Tree, Random Forest and XGBoost
- Class imbalanced techniques Random oversampling, SMOTE and ADASYN
- GridSearch CV, Hyperparameter tunning evaluation metrics and accuracy performance in details specific to research study.

Analysis, Results and Discussions

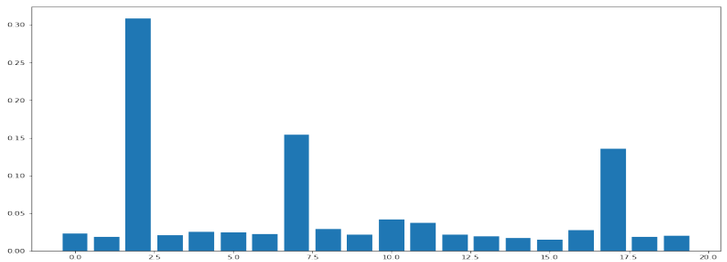
The Hopkins statistic our dataset returns 0.971214673264458 indicates the dataset is highly clustered

Silhouette score returns for this dataset is 0.6359613381214638 explain distance separation between the resulting clusters

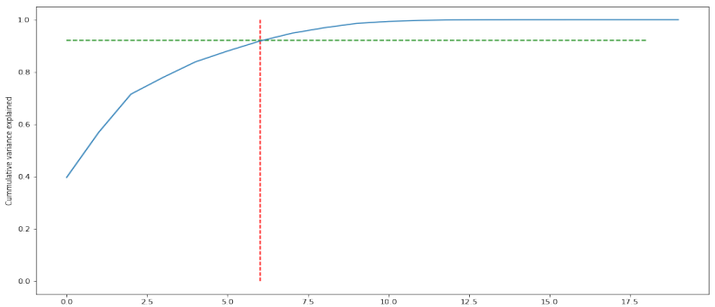
Feature Engineering

- Performing unique count on columns: trace_from_address, trace_to_address, token_transfer_from_address, token_transfer_to_address, transaction_from_address, transaction_to_address and Contract_address
- Performing sum on columns: trace_value, trace_gas, trace_gas_used, token_transfer_value, transaction_value, transaction_gas, transaction_gas_price and receipt_gas_used
- Performing max on columns: token_decimals, token_total_supplyand receipt_cumulative_gas_used

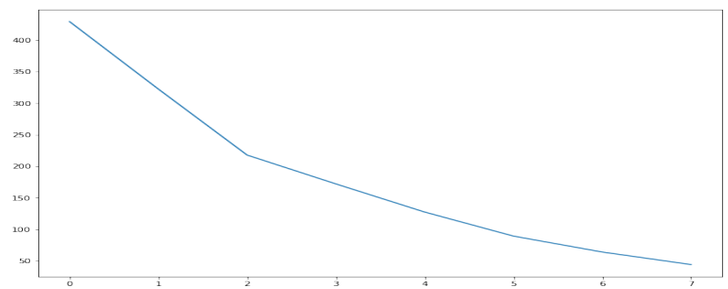
Feature Importance



The cumulative variables of score 92 determine the 6 principal components out of 20 features



Elbow curve shows that good k number 4.



S. No.	Variable Name	Value
Feature 1	'receipt_status'	0.02320103
Feature 2	Contract_is_erc20	0.01823772
Feature 3	trace_from_address	0.3086279
Feature 4	trace_to_address	0.0208992
Feature 5	trace_value	0.024868554
Feature 6	trace_gas	0.024106627
Feature 7	trace_gas_ used	0.02232336
Feature 8	token_transfer_from_address	0.1541869
Feature 9	token_transfer_to_address	0.028747967
Feature 10	token_transfer_value	0.021825304
Feature 11	transaction_from_address	0.041813623
Feature 12	transaction_to_address	0.036888547
Feature 13	transaction_value	0.021476977
Feature 14	transaction_gas	0.019385168
Feature 15	transaction_gas_price	0.01679462
Feature 16	receipt_cumulative_gas_used	0.015070091
Feature 17	receipt_gas_used	0.027688103
Feature 18	token_decimals	0.13520354
Feature 19	token_total_supply	0.01872537
Feature 20	Contract_address	0.019929413

Table Name	Record count (Year = 2019)
Balances	149,068,908
Blocks	2,280,958
Contracts	49,523,431
Logs	2,093,472
Token_transfer	224,175
Tokens	10,376
Traces	67,272,780
Transactions	15,423,875
ENS	28,825

Analysis, Results and Discussions (Evaluation Metrics)

	Clustered (Predicted)	In between Clusters (Predicted)
Clustered (Actual)	True Positive	False Negative
In between Clusters (Actual)	False Positive	True Negative

$$\text{Accuracy} = \frac{\text{Correctly Precdicted Labels}}{\text{Total Number of Labels}}$$

$$\text{Recall} = \frac{\textit{Correctly precited clusters}}{\textit{Correctly predicted cluster} + \textit{incorrectly preicted clusters}}$$

$$\text{Precision} = \frac{\textit{Correctly precited clusters}}{\textit{Correctly predicted cluster} + \textit{Predicted incorrectly between clusters}}$$

$$\text{Fscore} = 2 \text{ X } \frac{\textit{Precision X Recall}}{\textit{Precision+Recall}}$$

Metrics

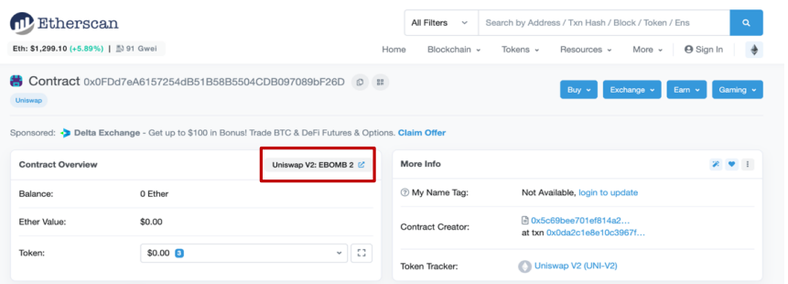
- **True Positive (TP):** classifier algorithms return correct multiclass clusters
- **False Positive (FP):** classifier algorithms return incorrect multiclass clusters during the cases like in between clusters.
- **True Negative (TN):** classifier algorithms return correct multiclass clusters during the cases like in between clusters
- **False Negative (FN):** classifier algorithms return incorrect multiclass clusters

AUC-ROC Curve

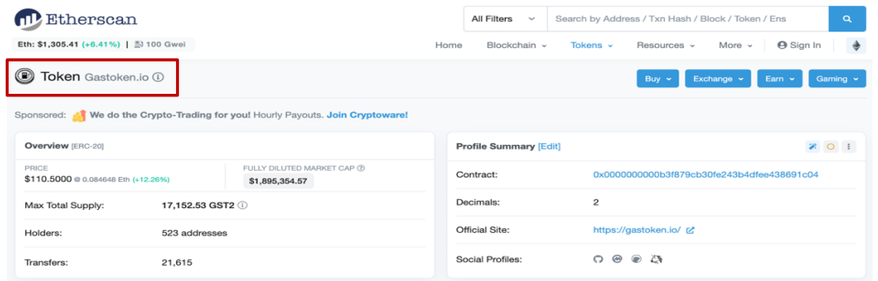
This experiment's evaluation purpose we have treated the function takes all call true outcomes (0,1,2,3,4) from the test set and the predicted probabilities for the predicted probabilities for the class.

Analysis, Results and Discussions (Cluster Profiling)

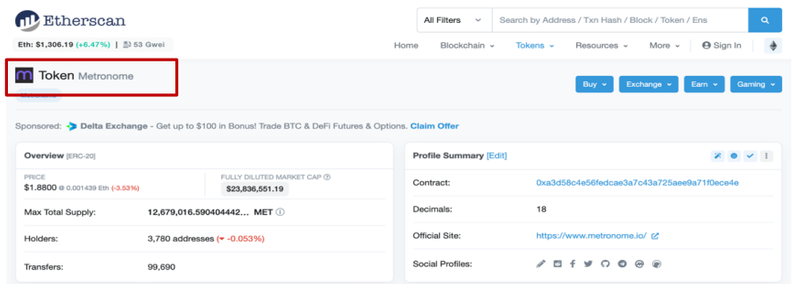
Smart contract Token Cluster 1 - Liquidity



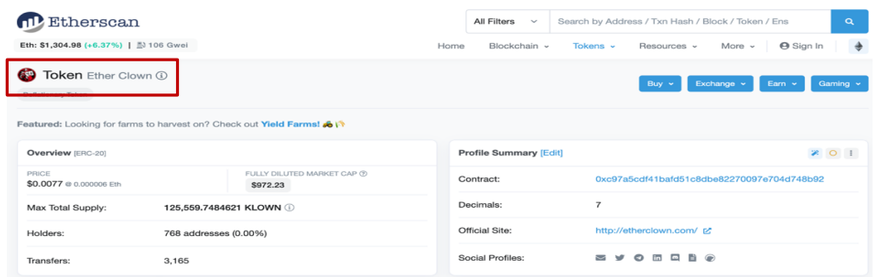
Smart contract Token Cluster 2 - ICO



Smart contract Token Cluster 3 - Exchange

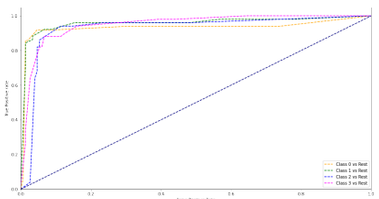


Smart contract Token Cluster 4 - Market Capital



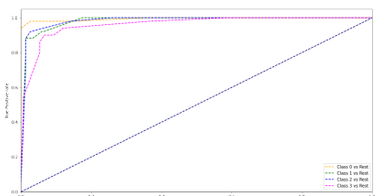
Analysis, Results and Discussions (Decision Tree)

Default parameter



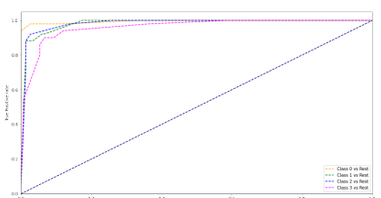
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9574 or True Positive Rate (TPR) and (1 - Specificity) 0.0453 or False Positive Rate (FPR) to choose an optimal threshold point.

Random Oversampling



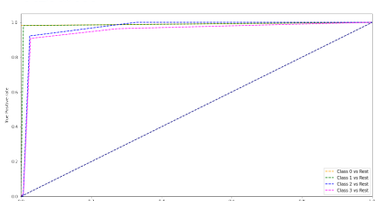
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9574 or True Positive Rate (TPR) and (1 - Specificity) 0.0426 or False Positive Rate (FPR) to choose an optimal threshold point.

SMOTE



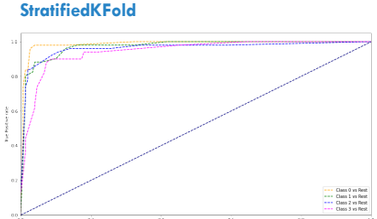
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9574 or True Positive Rate (TPR) and (1 - Specificity) 0.0426 or False Positive Rate (FPR) to choose an optimal threshold point.

ADASYN



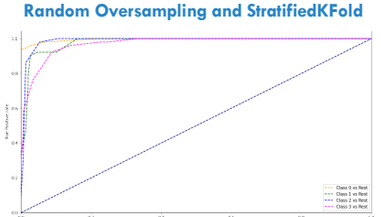
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9815 or True Positive Rate (TPR) and (1 - Specificity) 0.0185 or False Positive Rate (FPR) to choose an optimal threshold point.

StratifiedKFold



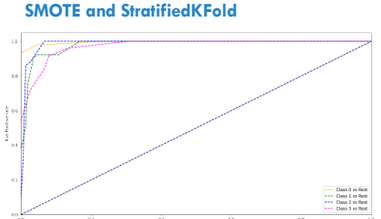
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9574 or True Positive Rate (TPR) and (1 - Specificity) 0.0453 or False Positive Rate (FPR) to choose an optimal threshold point.

Random Oversampling and StratifiedKFold



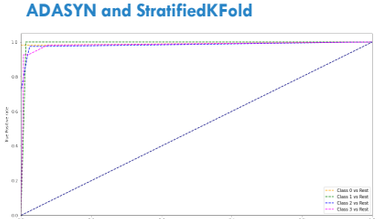
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9583 or True Positive Rate (TPR) and (1 - Specificity) 0.0417 or False Positive Rate (FPR) to choose an optimal threshold point.

SMOTE and StratifiedKFold



The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9787 or True Positive Rate (TPR) and (1 - Specificity) 0.0417 or False Positive Rate (FPR) to choose an optimal threshold point.

ADASYN and StratifiedKFold



The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9815 or True Positive Rate (TPR) and (1 - Specificity) 0.0185 or False Positive Rate (FPR) to choose an optimal threshold point.

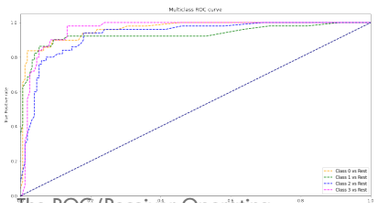
Evaluation Metrics

Class Imbalance / Hyperparameter Tunning	Accuracy	Precision	Recall	F-Score
Default	86.00%	85.99%	86.02%	86.00%
StratifiedKFold	87.05%	87.67%	87.55%	87.00%
Random Oversampling	91.00%	91.09%	91.05%	91.00%
Random Oversampling StartifiedKFold	90.00%	90.65%	90.01%	90.00%
SMOTE	91.00%	91.09%	91.05%	91.00%
SMOTE StratifiedKFOLD	89.00%	89.28%	89.02%	89.00%
ADASYN	95.00%	94.69%	94.83%	95.00%
ADASYN StratifiedKFOLD	96.00%	95.77%	95.76%	96.00%

Decision Tree machine learning algorithm, Class imbalance ADASYN, Hyperparameter tunning StratifiedKFold and Grid search CV with finetuned parameter combination using this dataset performed well with best accuracy of **96%**

Analysis, Results and Discussions (Random Forest)

Default parameter

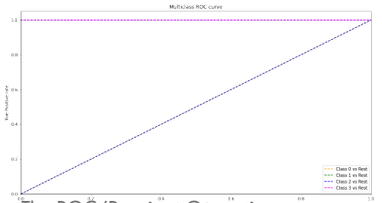


The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9535 or True Positive Rate (TPR) and (1 - Specificity) 0.0465 or False Positive Rate (FPR) to choose an optimal threshold point.

Random Oversampling

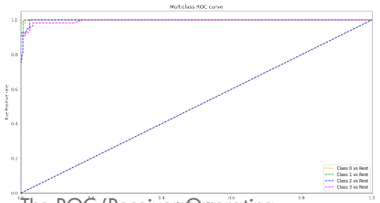
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

SMOTE



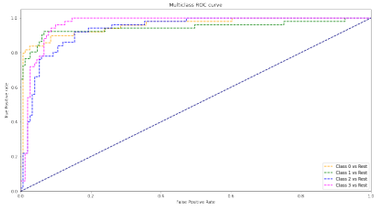
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

ADASYN



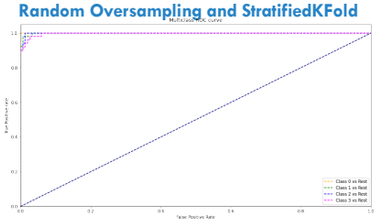
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9815 or True Positive Rate (TPR) and (1 - Specificity) 0.0185 or False Positive Rate (FPR) to choose an optimal threshold point.

StratifiedKFold



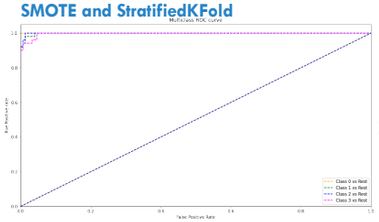
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9778 or True Positive Rate (TPR) and (1 - Specificity) 0.0222 or False Positive Rate (FPR) to choose an optimal threshold point.

Random Oversampling and StratifiedKFold



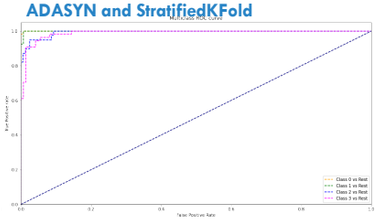
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

SMOTE and StratifiedKFold



The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

ADASYN and StratifiedKFold



The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9815 or True Positive Rate (TPR) and (1 - Specificity) 0.0185 or False Positive Rate (FPR) to choose an optimal threshold point.

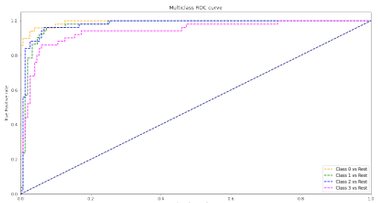
Evaluation Metrics

Class Imbalance / Hyperparameter Tunning	Accuracy	Precision	Recall	F-Score
Default	83.50%	83.90%	83.49%	84.00%
StratifiedKFold	82.00%	82.33%	81.99%	84.00%
Random Oversampling	100.00%	100.00%	100.00%	100.00%
Random Oversampling StartifiedKFold	97.00%	97.04%	97.00%	97.00%
SMOTE	100.00%	100.00%	100.00%	100.00%
SMOTE StratifiedKFOLD	97.05%	97.56%	97.50%	97.00%
ADASYN	95.00%	94.69%	94.83%	95.00%
ADASYN StratifiedKFOLD	96.00%	95.77%	95.76%	96.00%

Random Forest machine learning algorithm, Class imbalance SMOTE, Hyperparameter tunning StratifiedKFold and Grid search CV with finetuned parameter combination using this dataset performed well with best accuracy of **97.05%**

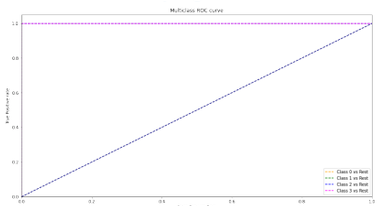
Analysis, Results and Discussions (XGBoost)

Default parameter



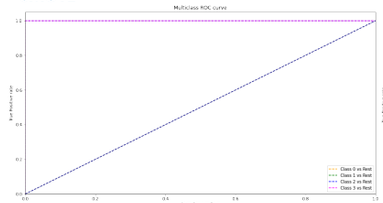
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9787 or True Positive Rate (TPR) and (1 - Specificity) 0.0213 or False Positive Rate (FPR) to choose an optimal threshold point.

Random Oversampling



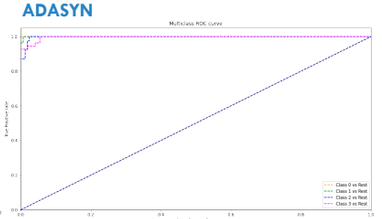
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

SMOTE



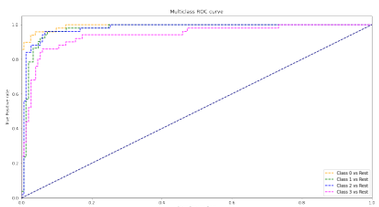
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

ADASYN



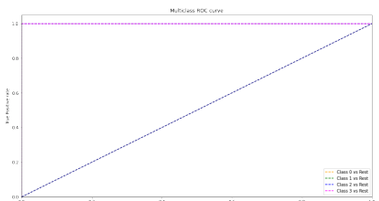
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9787 or True Positive Rate (TPR) and (1 - Specificity) 0.0417 or False Positive Rate (FPR) to choose an optimal threshold point.

StratifiedKFold



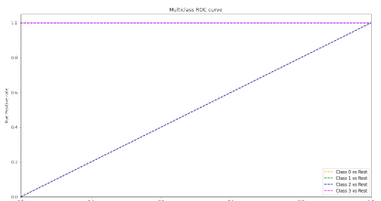
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9787 or True Positive Rate (TPR) and (1 - Specificity) 0.0213 or False Positive Rate (FPR) to choose an optimal threshold point.

Random Oversampling and StratifiedKFold



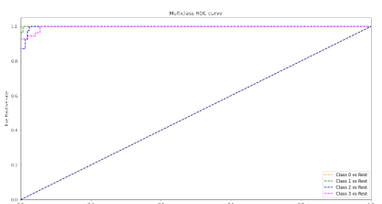
The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

SMOTE and StratifiedKFold



The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 1.0 or True Positive Rate (TPR) and (1 - Specificity) 0.0 or False Positive Rate (FPR) to choose an optimal threshold point.

ADASYN and StratifiedKFold



The ROC (Receiver Operating Characteristic) the trade-off between Sensitivity 0.9787 or True Positive Rate (TPR) and (1 - Specificity) 0.0417 or False Positive Rate (FPR) to choose an optimal threshold point.

Evaluation Metrics

Class Imbalance / Hyperparameter Tunning	Accuracy	Precision	Recall	F-Score
Default	88.00%	88.30%	88.02%	88.00%
StratifiedKFold	88.00%	88.30%	99.02%	88.00%
Random Oversampling	100.00%	100.00%	100.00%	100.00%
Random Oversampling StartifiedKFold	100.00%	100.00%	100.00%	100.00%
SMOTE	100.00%	100.00%	100.00%	100.00%
SMOTE StratifiedKFOLD	100.00%	100.00%	100.00%	100.00%
ADASYN	97.00%	96.77%	97.04%	97.00%
ADASYN StratifiedKFOLD	97.00%	96.77%	97.04%	97.00%

XGBoost machine learning algorithm, Class imbalance ADASYN, Hyperparameter tunning StratifiedKFold and Grid search CV with finetuned parameter combination using this dataset performed well with best accuracy of **97.00%**

Conclusions, Future Recommendations and Limitations

Conclusion

After performing Machine Learning Classification with combination of class imbalance and hyperparameter tunning and Metric evaluation it's been observed as listed below

- ❖All 4 Classes overfitting performed accuracy range between 96.05% and 100%
- ❖2 Classes overfitting out of 4 classes performed accuracy range between 95% and 96%
- ❖1 Class overfitting out of 4 classes performed accuracy range between 89% and 91%
- ❖No Overfitting of all 4 classes performed accuracy range between 80% and 88%.

XGBOOST with StratifiedKFold clustered all 4 classes performed well with the 88% accuracy out of 24 combinations.

We can conclude from this work that it is possible to perform Ethereum Decentralized Financial (DeFi) cluster profiling for Smart Contract Token transaction for those EOA's listed in Ethereum Name Service (ENS) using Google BigQuery Public dataset. Applying this approach, we can cluster Ethereum Smart Contract tokens. The outcome of this research paper would provide insight to build Money Lego's based on most Smart Contract Tokens used and provide recommendation to attract investors. It signifies that our model works well in grouping the similar Smart Contract Token and work well with an unbalanced dataset.

In addition, we observed EOA and ENS have provided no value-add to this study. Analysis token for all the year as 2019 has little token compare to overall 10,376 tokens.

Future Recommendations

- ❖Mining Gas price Prediction: as the transaction fees are based on gasPrice and times gasUsed, the users can control the gasUsed real time using Google BigQuery Public Ethereum Dataset to minimize the transaction fees charged by miners.
- ❖Identify Smart Contract Similarities: As there is chance of similar smart contract codes stored in encrypted format and the call of smart contracts. Code similarity evaluation to recommend reusing already available performing real time using Google BigQuery Public Ethereum Dataset Smart contract code similarity detection
- ❖Smart Contract Vulnerability real time detection: Smart contract real time using Google Ethereum Public Dataset vulnerability detection methods can motivated software vulnerability detection methods in Ethereum blockchains

Code and Dataset Path

Code Path:
https://github.com/raparama/ResearchPaper

Dataset Path:
https://drive.google.com/drive/folders/1D4SPUzV8SWjFmu7P0YQUln255vMeVY5o?usp=sharing

Thank you