A Sample Project on Blockchain Network

Course No: 7570

Group-3 (Md. Ashraf Uddin, Seunghwan Youn, Mostafa Raad)

Contents

- Load network
- Load darknet address
- Feature extraction
- Process model
- Result

Load Network

def readNetworkData(month):

- This method performs all necessary operations for loading data and linking/processing the data
- It takes as parameter a month number, then read and process data of that month
- It stores address info of that month in a day by day basis
- Address info includes transaction timestamps, price of an address, hash of an address, no of output address of the corresponding transaction, average price of all output addresses of the corresponding transaction
- It stores these six data into six numpy arrays in a day by day basis under a folder for this month

addr_counts	12/15/2021 3:05 AM	File folder
addr_hashes	12/15/2021 3:05 AM	File folder
addr_prices	12/15/2021 3:05 AM	File folder
addr_tx_index	12/15/2021 3:05 AM	File folder
avg_prices	12/15/2021 3:05 AM	File folder
bad_prices	12/15/2021 3:43 PM	File folder
bad_times	12/15/2021 3:43 PM	File folder
tx_times	12/15/2021 3:05 AM	File folder

```
E:\Projects\blockchain\second part\extract_feature>python load_network.py
Loading month 1 data ...
Loading month 2 data ...
Loading month 3 data ...
Loading month 4 data ...
Loading month 5 data ...
Loading month 6 data ...
Loading month 7 data ...
Loading month 8 data ...
Loading month 9 data ...
Loading month 10 data ...
Loading month 11 data ...
Loading month 12 data ...
Network data loading finished.
Graph data loading and linking time: 911.8575859069824 seconds
```

Load Darknet Address

def readDarknetData(file_path):

- It takes the path of a darknet file, converts it into a pandas dataframe and reads all rows
- For each row, it parses the timestamp and extract year, month and date.
- It adds the timestamp and price of the row into the dictionary of the corresponding day
- The purpose is to store all addresses of this file into a dictionary against transaction dates

addr_counts	12/15/2021 3:05 AM	File folder
addr_hashes	12/15/2021 3:05 AM	File folder
addr_prices	12/15/2021 3:05 AM	File folder
addr_tx_index	12/15/2021 3:05 AM	File folder
avg_prices	12/15/2021 3:05 AM	File folder
bad_prices	12/15/2021 3:43 PM	File folder
bad_times	12/15/2021 3:43 PM	File folder
tx_times	12/15/2021 3:05 AM	File folder

```
E:\Projects\blockchain\second part\extract_feature>python load_darknet.py
Reading Darknet market data...
Darknet data reading finished.
Darknet data loading time: 246.4784300327301 seconds
```

Feature Extraction

def getFeaturesOfDay(month, day, isBadAddrAvailable, csvWriter):

- It takes as input a particular day of a month. Then, opens data numpy files for that day of the given month.
- It opens both graph data and darknet data of the given day.
- After reading all darknet/bad address of that day, we read 1k good address for the same day

```
E:\Projects\blockchain\second part\extract_feature>python get_features.py
Extracting features...
Feature extraction finished.
Feature extraction time: 373.78524231910706 seconds
```

Feature Extraction

	A	B	C	D	E	F	G	H	I	J
1	address_hash	price	no_transaction	no_neighbor	avg_price_neighb	month	day	is_bad_address		
2	16MnndGE4A8p2Xxm	25000	115786	1	1711727	4	1	FALSE		
3	1EoCZFACXKLscXRwcV	3398455	115786	1	1711727	4	1	FALSE		
4	1Q63T5fpJtdsUiKuRde	205846000	115786	1	214792232	4	1	FALSE		
5	1PaFXW3KogmLNd6S	223738464	115786	1	214792232	4	1	FALSE		
6	1MsvVkED9SsXK6FiLu	1590000	115786	1	3995000	4	1	FALSE		
7	1NxaBCFQwejSZbQfW	6400000	115786	1	3995000	4	1	FALSE		
8	12XZS4Q9neY4B9Wt	37841000	115786	1	23411047	4	1	FALSE		
9	1bBu1Wzp4Nm14FC	8981094	115786	1	23411047	4	1	FALSE		
10	16DE4ifyng1VotK7qnf	582947	115786	1	955586	4	1	FALSE		
11	1A9HNFUEh6nev4k3f	1328225	115786	1	955586	4	1	FALSE		
12	14MdorHwrLeqEYMYG	30000	115786	1	482861	4	1	FALSE		
13	1D5P8HKDxZs1yryKfaf	935723	115786	1	482861	4	1	FALSE		
14	1LzYT6TmnsTMxuwvx	100000	115786	1	57494	4	1	FALSE		
15	1A1edBPJSugXe5eipq	14989	115786	1	57494	4	1	FALSE		
16	1LuckyR1fFHEsXYyx5C	8000000	115786	1	5550000	4	1	FALSE		
17	1MsvVkED9SsXK6FiLu	3100000	115786	1	5550000	4	1	FALSE		
18	1L2uq5tG9wqJvP6BGv	49116000	115786	2	16639316	4	1	FALSE		
19	1876GgA6cwKJmVRyD	741127	115786	2	16639316	4	1	FALSE		
20	19C6jn36dVQhT3swK	60822	115786	2	16639316	4	1	FALSE		
21	1Jz9Btx31H7BAC3Tqjq	1000000	115786	1	559286	4	1	FALSE		
22	19QzfrDbJGYLUvWW	118573	115786	1	559286	4	1	FALSE		
23	1A3N6QtAZVwNxUGf	8901	115786	1	2369450	4	1	FALSE		

Features:

1. address_hash: Unique ID of the address
2. price: Bitcoin amount for the address
3. no_transaction: No of transactions in the 24 hour window of the address
4. no_neighbor: No of extra output addresses of the transaction where this address is also an output
5. avg_price_neighbors: Average price/bitcoin amount of the neighbors of this address
6. month: month when the transaction of this address happened
7. day: day when the transaction of this address happened
8. is_bad_address: True if this address is a darknet address, False otherwise

Process Model

def loadAndProcessData(data_path):

- It splits the total data into 80% training data and 20% test data.
- It first splits rows with good address and bad address.
- Then splits both good and bad data into train and test set with 80% x 20% ratio
- It generates training data by concatenating good and bad training data.
- After shuffling train and test data, it separates features and labels of training data and test data.
- 1 indicates bad address and 0 indicates good address.
- Train feature, train labels, test features and test labels are saved as numpy arrays.

predictions_rf.npy	12/15/2021 5:35 PM	NPY File	116 KB
predictions_xgb.npy	12/15/2021 5:39 PM	NPY File	116 KB
testX.npy	12/15/2021 5:31 PM	NPY File	695 KB
testY.npy	12/15/2021 5:31 PM	NPY File	116 KB
trainX.npy	12/15/2021 5:31 PM	NPY File	2,778 KB
trainY.npy	12/15/2021 5:31 PM	NPY File	464 KB

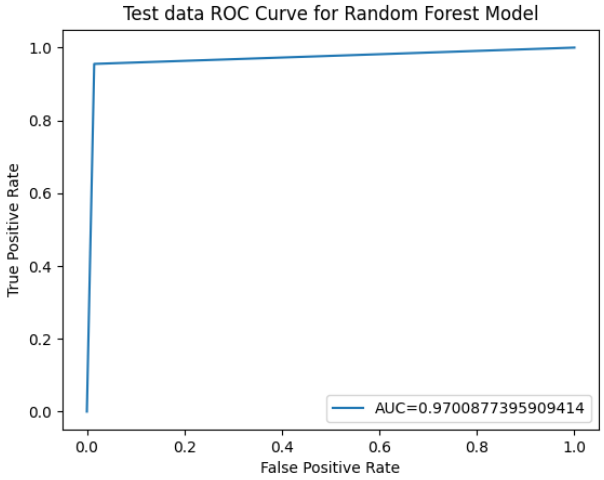
Results

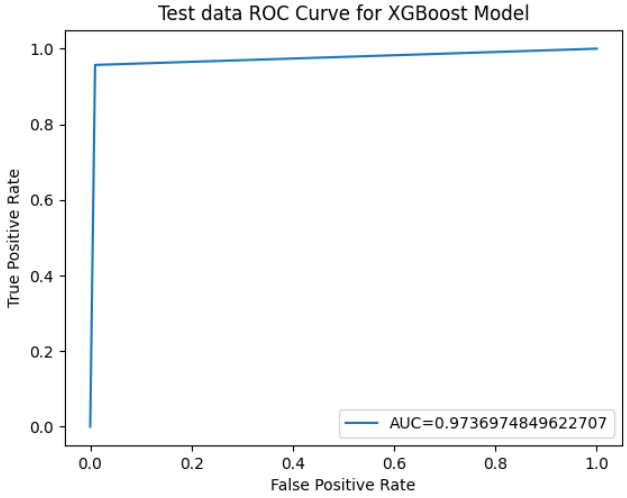
Model Name	Accuracy	Precision	Recall	F1 Score
Random Forest	0.97	0.97	0.95	0.96
XGBoost	0.97	0.98	0.95	0.97

```
Total training time for random forest: 241.53944611549377 seconds
Accuracy: 0.9735056869958486
Precision: 0.9754765953638235
Recall: 0.9552147239263804
F1 Score: 0.9652393393260417
Total evaluation time for random forest: 4.086791038513184 seconds
```

```
Total training time for XGBoost: 116.41383838653564 seconds
Accuracy: 0.9775220223429748
Precision: 0.9841384282624369
Recall: 0.9570552147239264
F1 Score: 0.9704078912290055
Total evaluation time for XGBoost: 2.1157400608062744 seconds
```

Results


Test data ROC Curve for Random Forest Model
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.9700877395909414


Test data ROC Curve for XGBoost Model
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC=0.9736974849622707

Thank You

Any Questions?