CS440 PROJECT REPORT

Emir Arda GÜN
Department of Electrical And Electronics Engineering
University of Ozyegin
Istanbul, Turkiye
arda.gun.24752@ozu.edu.tr

Seyit Kubilay Uluçay
Department of Electrical And Electronics Engineering
University of Ozyegin
Istanbul, Turkiye
kubilay.ulucay@ozu.edu.tr

***Abstract*—Social media is one of the biggest and most important communication tools of our time, but can it be manipulative if used correctly? Hundreds of thousands of tweets are tweeted every day on Twitter, the largest and most interactive social media platform, and these tweets may be manipulating people in some way, influencing them at decision-making points when making choices. The focus of our project is to investigate whether tweets can have a manipulative effect on the price of Bitcoin. To do this research, we aimed to develop a model using Multi-Layer Perceptron and BERT Transform. This report will be based on this model that we have developed.**

***Index Terms*—Machine Learning,Neural Network, Natural Language Processing, BERT Transform, Multi-layer Perceptron, Tweet, Dataset, Blockchain, Cryptocurrency, Bitcoin, Imputation Methods, Data Cleaning, Filtering**

I. Introduction

Initial goal was to investigate the relationship between Google Trends data and the number of daily tweets about Bitcoin and Bitcoin price using Machine Learning, but even the most recent and high quality twitter datasets[1] we could find had serious gaps and omissions.

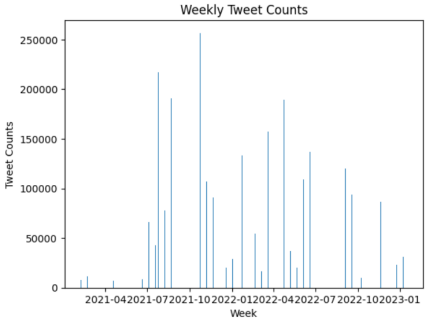

Fig. 1. Weekly Distribution Graph of the Data Set Showing the Number of Tweets Sent

As the figure shows, the distribution of the number of tweets was insufficient and incomplete to train a model. 40 out of 102 weeks had no data. So we wanted to see if we could fill these empty datasets or not. We used several imputation methods which are Spline Interpolation, Linear Interpolation, Moving Average Imputation and after these three imputations we tried Seasonal-Trend decomposition using LOESS Imputation because Interpolation of the decomposed trend component is done linearly. [2].

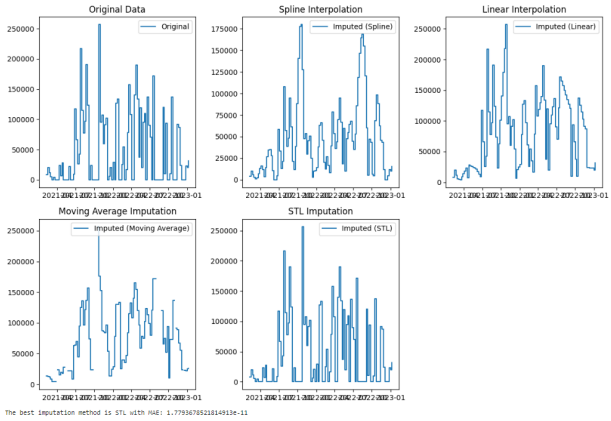

Fig. 2. The Imputation Methods That We Used

With Mean Absolute Error method we tried to see how implementation techniques results over our dataset and we got followed results:

- MAE for Spline Interpolation: 46,317.69222882127
- MAE for Linear Interpolation: 31,525.69306930693
- MAE for MA Imputation: 39,077.99438202247

and we got an (MAE) score of 1.7793678521814913e-11 in STL, and we were very satisfied at first because it was a very successful result, but when we examined the graph again, we realized that the dataset had not changed much at all, and some of the tweets sent were even below 0, which was impossible. because a negative number of tweets could not be sent in a day. Since other graphics could not bring the dataset to the desired level, we realized that we could not progress in this direction and decided to change the purpose of the project. Using the tweet dataset we had, we started to examine the relationship between tweets and bitcoin price. What we needed to build this model was to use BERT transform, which is one of the NLP methods and converts each tweet into an array.

II. The Project

A. Prepare The Dataset

The dataset we received via Kaggle had 13 different columns and 4689288 different rows, we only needed user

Name, Followers number of that user, tweet and date informations, for this reason we filtered the data while reading[3] it and in the current state The shape of our dataset became (4689288,3). After this imputation we add one more column as "1 week later" column that shows 1 week later's date.

B. Filtering The Dataset

Not every tweet in the dataset could be trusted and suitable for train the model, so we decided to add some filtering methods. Our initial focus was to get rid of Crypto Exchange Companies accounts because their tweets generally did not make sense, then we focused on accounts' follower numbers, realizing that tweets need to influence readers to affect the price of bitcoin and if the account has small followers, those tweets won't make sense. That's why we added a minimum 10,000 follower filter due to the influence factor. We then remove the telegram advertisements and add certain words that contain predictions, such as "buy" and "sell". After the filtering our dataset shape became (54460, 4).

C. Creating Bitcoin Price Dataset

We took start and end time datas from the tweet dataset and use them to create a new dataset for Bitcoin price. We use YahooFinance API[4] for getting the dataset and create another columns, the most important columns was "result" column that shows the price direction between the 1 week later. If the price change is positive, result was 1, else result was 0.

D. Merging 2 Datasets and Cleaning

After the Bitcoin Price dataset we merged Tweet Dataset and Bitcoin Price Dataset and we filtered the dataset for 2 columns which are "text" that shows tweets and "result" that shows price direction, with this merge, now we have filtered tweets and the bitcoin price change after 1 week. Current shape of dataset is (54460, 2).

	text	result
2021-02-05	#Bitcoin close to major breakout over @elonmus...	1
2021-02-06	#Bitcoin new ATH before the #KansasCityChiefs ...	1
2021-02-06	#BTC Never forget. The only thing #Bitcoin ha...	1
2021-02-06	#Bitcoin is back above $40,000, recouping near...	1
2021-02-06	Buying on the retrace has been a profitable in...	1
...	...	...
2023-01-09	Strongest Movers in #USDT 1 #Zilliqa $zil 🚀 ...	1
2023-01-09	For all of you that say 100B supply of #Pi is ...	1
2023-01-09	What you bullish on? . #altcoins #altcoin #m...	1
2023-01-09	JUST IN: Metropolitan Commercial Bank has anno...	1
2023-01-09	Update #cryptocurrency #BTC : $17,250 USD #C...	1

54460 rows × 2 columns

Fig. 3. Merged Dataset with Filtered

E. Grouping the Dataset For BERT

After merge, the distribution of result was

- Bitcoin prices decreased after 22,927 tweets
- Bitcoin prices increased after 31,533 tweets

We assume that the distribution is acceptable. Size of the data was still cannot be useful and we have to reduce the rows because BERT[5] transformation costs time, so we decided to combine tweets that has been sent same day, with this grouping method our dataset shape became (219,2).

F. BERT Process

First, we split the dataset as X and y as input and output array, then cut 80 percent as train set and 20 percent as test set. After the splitting, we apply BERT Transform for input arrays and create a new variable for outputs. After 2 minutes, the transformation is finished.

```
  0%|          | 0/175 [00:00<?, ?it/s]Token indices sequence length is longer than the specified maximum sequence length for t
his model (34008 > 512). Running this sequence through the model will result in indexing errors
  7%|▋         | 12/175 [00:06<01:16,  2.12it/s]We strongly recommend passing in an `attention_mask` since your input_ids may b
e padded. See https://huggingface.co/docs/transformers/troubleshooting#incorrect-output-when-padding-tokens-arent-masked.
100%|██████████| 175/175 [01:35<00:00,  1.82it/s]
100%|██████████| 44/44 [00:20<00:00,  2.10it/s]
```

Fig. 4. BERT Process's output at console

G. MLP Process

For Multi-layer Perceptron, we use MLPClassifier model from Sklearn[6] library. As train inputs we used the BERT transformed dataset and output dataset that has already been splitted. We trained the the model with several hidden layer sizes and several maximum iteration numbers, at each combination we calculate the accuracy score and saved all of the parameters and score at "result dataframe".

	HiddenLayer1	HiddenLayer2	MaxIter	Accuracy
0	1.0	1.0	1000.0	0.568182
1	1.0	1.0	2000.0	0.431818
2	1.0	1.0	3000.0	0.568182
3	1.0	1.0	4000.0	0.568182
4	1.0	1.0	5000.0	0.431818
...	...	...	...	...
2396	7.0	7.0	45000.0	0.590909
2397	7.0	7.0	46000.0	0.613636
2398	7.0	7.0	47000.0	0.477273
2399	7.0	7.0	48000.0	0.522727
2400	7.0	7.0	49000.0	0.568182

2401 rows × 4 columns

Fig. 5. Results Dataset

Then we sort to see the best score:

III. Conclusion

As shown in Figure 6, our best model has 0.727273 accuracy score and we assumed that model works fine. As told in Introduction part our first aim was see the relationship with number of tweets and Google Trends dataset, but after some

	HiddenLayer1	HiddenLayer2	MaxIter	Accuracy
1617	5.0	6.0	1000.0	0.727273
2242	7.0	4.0	38000.0	0.681818
2392	7.0	7.0	41000.0	0.681818
1574	5.0	5.0	7000.0	0.681818
2033	6.0	7.0	25000.0	0.681818
...	...	...	...	...
1951	6.0	5.0	41000.0	0.409091
453	2.0	3.0	13000.0	0.409091
1990	6.0	6.0	31000.0	0.409091
1431	5.0	2.0	11000.0	0.409091
1641	5.0	6.0	25000.0	0.409091

2401 rows × 4 columns

Fig. 6. Results Dataset

issues we changed our aim and decided to create a model that understand the manipulative way of the tweet on the Bitcoin price with almost 73 percent accuracy score. With this model from now on we can check whether the tweet with proper filtering, triggers the Bitcoin price positively or negatively.

REFERENCES

[1] Bitcoin Tweets Dataset: https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets

[2] Chandrasekaran, S. et al. (2016) "*Data Preprocessing: A New Algorithm for Univariate Imputation Designed Specifically for Industrial Needs*". CIplus. 5-8.

[3] Panda Dataframe Library : https://pandas.pydata.org/docs/index.html

[4] Yahoo Finance API: https://developer.yahoo.com/api/

[5] BERT Model: https://huggingface.co/

[6] SKlearn Library: https://scikit-learn.org/stable/index.html