PREDICTION OF CRYPTOCURRENCY PRICES USING HISTORICAL DATA AND TWITTER DATA

This project and report are solely meant for submission to the Preliminary Round of the **MLDA DLW Hackathon 2021**, organized by the MLDA Club, Nanyang Technological University, Singapore, from Friday, October 15, 2021, to Sunday, October 17, 2021.

Team Number: 17

Team Name: Data Pirates

Team Members: Arora Kanupriya, Bansal Arushi, Bhatia Nipun, Pathak Siddhant

Abstract

A cryptocurrency (or crypto currency) is a digital asset designed to work as a medium of exchange that uses cryptography to secure its transactions, control the creation of additional cryptocurrencies, and verify the secure transfer of assets. Just like other assets, the price of a cryptocurrency is directly related to market supply and demand. These market forces can change for several reasons, including public opinion, the press, and social media. Market sentiment analysis is an essential part of many trading strategies. However, most cryptocurrencies suffer from a key drawback - high price fluctuations. Bitcoin, for example, increased its value by 1900% in 2017 before plunging below USD 8,000 per coin up from its high of USD 19,000 in the 2018 Jan/Feb crash. Since cryptocurrency price research is still a relatively unexplored area and equilibrium market prices are poorly understood, this project seeks to conduct sentiment analysis to explore possible drivers such as social media and news reports that lead to these price fluctuations.

PREDICTION OF CRYPTOCURRENCY PRICES USING HISTORICAL DATA AND TWITTER DATA

We aim to analyze the historical data, i.e., the open, high, low, and close prices of the Bitcoin cryptocurrency, along with real-time sentimental analysis of the tweets centered around the same for a similar time to predict the weighted price of the token by the end of the trading market.

Problem Statement

We believe that AI has endless possibilities, and it should not be limited to a certain theme. Therefore, the task is to build an AI-based solution that we are passionate about. The solution should be meaningful, have a positive impact on society or assist industries in their respective domain (software and hardware).

Our approach

We will scour through the Twitter data, i.e., the number of tweets within a day's timeframe (concerning the current time), and then match it with the real-time pricing to predict closing adjusted prices along with any anomaly (if possible).

Non-Technical Executive Summary

We wish to analyze and find if the Twitter sentiments of the public play any role in influencing the market value of cryptocurrency tokens. Similar strategies have been deployed to make predictions for stocks in the stock market. However, due to the steady nature of the market, the prices no longer depend on public sentiment. On the contrary, the prices of cryptocurrencies do depend on sentiment. Upon performing standard exploratory data analysis techniques, we realized that it is difficult to observe a correlation among the given predictor variables in the discussion. The correlation matrix of the variables is added to Appendix 1. The 0s and 1s do not present any clear inference. So, we need to dig deeper to make predictions. Thus, we try making use of high-level machine learning algorithms such as Multi-variate Linear Regression, Support Vector Machine, Random Forest Regressor, and Long-Short Term Memory (LSTM) model.

Technical Exposition

Collection and curation of the dataset.

We propose to involve the Twitter-based Python API called Tweepy for scraping tweets and perform basic data preprocessing with the help of the Natural Language Toolkit (NLTK) library to execute the VADER Sentimental Analysis upon the same, to obtain what we define as the Compound Score of a text. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It is used for sentiment analysis of text which has both polarities i.e., positive/negative. VADER is used to quantify how much positive or negative emotion the text has and the intensity of emotion. The compound score is the sum of positive, negative & neutral scores which is then normalized between -1(most extreme negative) and +1 (most extreme positive). **The more Compound score closer to +1, the higher the positivity of the text.**

We suggest involving the Alpha Vantage API to retrieve prices for the cryptocurrency in real-time. Alpha Vantage provides its services for free. From there, users can manipulate the data or store it for later use too. In addition to the historical price data for cryptocurrencies and stocks, there are more than 50 technical indicators available as well as performance data for 10 United States Equity Sectors.

With the help of our trained machine learning model, we shall pass these values to predict the coin's Weighted Price at the end of the day. Weighted Price, is the average price of an asset over a set period, weighted by volume. Just like OHLCV data, Weighted Price is an aggregated form of trade data. It's a useful indicator to eliminate noise in price movements for a given period.

Since scraping data from Twitter and Alpha Vantage API for training the model was computationally expensive, we found and utilized datasets found from Kaggle. we have taken two datasets available publicly on the Kaggle website under the following headers:

1. *Bitcoin Historical Data: Bitcoin data at 1-min intervals from select exchanges, Jan 2012 to March 2021*
2. *Bitcoin 17.7M Tweets and Price: Bitcoin Tweets sentiment analysis and the price.*

Since the size of the data increases exponentially with text, we have decided to only keep the numbers in the CSV(comma-separated value) files. Thus, the tweets meant for training purposes are not available for viewing purposes. To ensure consistency, it was made sure that the

data scraping processes for those datasets are the same as ours as stated above (cross verified with the information provided by the curators of the pre-processed datasets).

Data Manipulation and Exploration Process

The initial states of the collected datasets were not usable for our model training. There are a lot of unnecessary columns and missing rows in the data-frames loaded. For the unnecessary columns, we dropped them for the sake of simplicity. We utilized the Pandas library under Python for this purpose.

For formatting and manipulation purposes, we had to change the data types of certain columns, mainly those containing dates and timestamps. We aim to create a clean dataset that consists of columns “Open”, “High”, “Low”, “Close”, “Total volume of tweets”, “Compound_Score”. A code snippet, showcasing the same purpose, has been added to the Appendix.

Analytical and Modelling Steps

We made use of various machine learning models, namely:

1. Multi-variate Linear Regression
2. Random Forest Regressor
3. Support Vector Machine (SVM)
4. Multilayer Perceptron (MLP)
5. Long-Short Term Memory (LSTM) model

We shall discuss each model's incompleteness along with its results on our data. We use common performance metrics (Root Mean Squared Error and Mean Absolute Error) to make the judgment call which model serves the best in our situation.

1. ***Multi-variate Linear Regression:***

This is quite like the simple linear regression model we have discussed previously, but with multiple independent variables contributing to the dependent variable and hence multiple coefficients to determine and complex computation due to the added variables. Jumping straight into the equation of multivariate linear regression,

$$Y_i = \alpha + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + \ldots + \beta_n x_i^{(n)}$$

Equation 1: Multi-variate Linear Regression

where, Y_i = the estimate of i^{th} component of dependent variable y, where we have n independent variables

$x_i^{(j)}$ = the i^{th} component of the j^{th} independent variable/feature.

2. Random Forest Regression:

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. For each decision tree, Scikit-learn calculates the importance of a node using Gini Importance, assuming only two child nodes (binary tree):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

Equation 2: Calculation of importance of node using Gini Index

where, ni_j = the importance of node j

w_j = weighted number of samples reaching node j

C_j = the impurity value of node j

left(j) = child node from left split on node j

right(j) = child node from right split on node j

3. Support Vector Machine (SVM):

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Simply put, it does some extremely complex data transformations, then figures out how to separate your data based on the labels or outputs you've defined. Unlike other Regression models that try to minimize the error between the real and predicted value, the Support Vector Regressor (SVR) tries to fit the best line within a threshold value. The threshold value is the distance between the hyperplane and the boundary line.

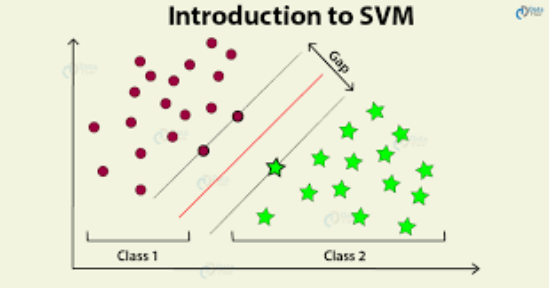

Figure 1: SVM Algorithm

4. *Multilayer Perceptron (MLP):*

A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses back-propagation for training the network. MLP is widely used for solving problems that require supervised learning as well as research into computational neuroscience and parallel distributed processing. Applications include speech recognition, image recognition, and machine translation. MLPs are designed to approximate any continuous function and can solve problems that are not linearly separable. The major use cases of MLP are pattern classification, recognition, prediction, and approximation.

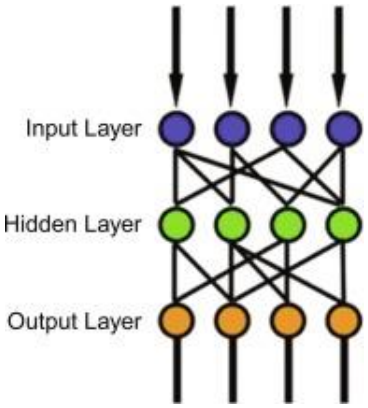

Figure 2:Schematic representation of an MLP with a single hidden layer.

5. *Long-Short Term Memory (LSTM):*

LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Long Short-Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It can handle the vanishing gradient problem faced by RNN. Unlike standard feedforward neural networks, LSTM has feedback connections, giving it the advantage over others. LSTMs make small modifications to the information by multiplications

and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things.

Performance Metrics

Root Mean Squared Error (RMSE) is a good measure of accuracy, but only to compare prediction errors of different models or model configurations for a particular variable and not between variables, as it is scale-dependent. The lower the RMSE, the better a given model can "fit" a dataset. A perfect RMSE value is 0.0, which means that all predictions matched the expected values exactly, which is rarely the case, and if it happens, it suggests your predictive modeling problem is trivial.

$$\mathrm{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Equation 3: Root Mean Squared Error

Mean Absolute Error (MAE) takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group. RMSE punishes larger errors more than smaller errors, inflating or magnifying the mean error score. This is due to the square of the error value. The MAE does not give weight to different types of errors and instead the scores increase linearly with error increases.

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |x_i - x|$$

Equation 4: Mean Absolute Error

Machine Learning Model/ Error Metrics	Root Mean Squared Error	Mean Absolute Error
Multi-variate Linear Regression	4.731893281133085	2.6001870657470194
Random Forest Regression	0.0	813.6578657865787
Support Vector Machine	2631.633668587859	1669.4916573941057
Multilayer Perceptron	340.324490855283	170.01980198019803
Long-Short Term Memory	0.0041880845267927985	0.003205464143801855

Table 1: Performance Metrics for various ML models

As we can observe from Table 1: Performance Metrics for various ML models, the 0.0 RMSE of Random Forest and absurdly high value of MAE imply a good probability of overfitting, i.e., a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. The small values of RMSE and MAE suggest that Multi-variate Linear Regression serves as a better model than Multilayer Perceptron, a Deep Learning model. This may be due to the size of the training data. We can mess around with the hyperparameters within to see what influences the final errors. However, the exceptionally low values of RMSE and MAE from Long-Short Term Memory (LSTM) make it by far the best model for our discussion undeniably.

For assumptions made and areas for further exploration, please refer to the Footnotes section.

Appendix

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12120 entries, 0 to 12119
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Open                   12120 non-null  float64
 1   High                   12120 non-null  float64
 2   Low                    12120 non-null  float64
 3   Close                  12120 non-null  float64
 4   Compound_Score         12120 non-null  float64
 5   Total Volume of Tweets 12120 non-null  float64
 6   Weighted_Price         12120 non-null  float64
dtypes: float64(7)
memory usage: 662.9 KB
```

Figure 1: Information about the Variables

	Open	High	Low	Close	Compound_Score	Total Volume of Tweets	Weighted_Price
count	12120.00	12120.00	12120.00	12120.00	12120.00	12120.00	12120.00
mean	7280.85	7287.59	7273.50	7281.09	0.10	1444.43	7280.52
std	3108.35	3113.74	3102.33	3108.57	0.04	747.49	3108.07
min	2665.21	2669.38	2655.79	2669.38	-0.18	3.00	2667.66
25%	5087.70	5099.93	5081.76	5088.41	0.08	961.00	5091.88
50%	6624.06	6627.92	6619.44	6622.62	0.10	1249.00	6624.01
75%	8400.24	8413.09	8396.19	8402.33	0.13	1690.25	8403.39
max	19546.88	19546.89	19526.18	19546.89	0.49	12696.00	19546.88

Figure 2: Summary Statistics about the Variables

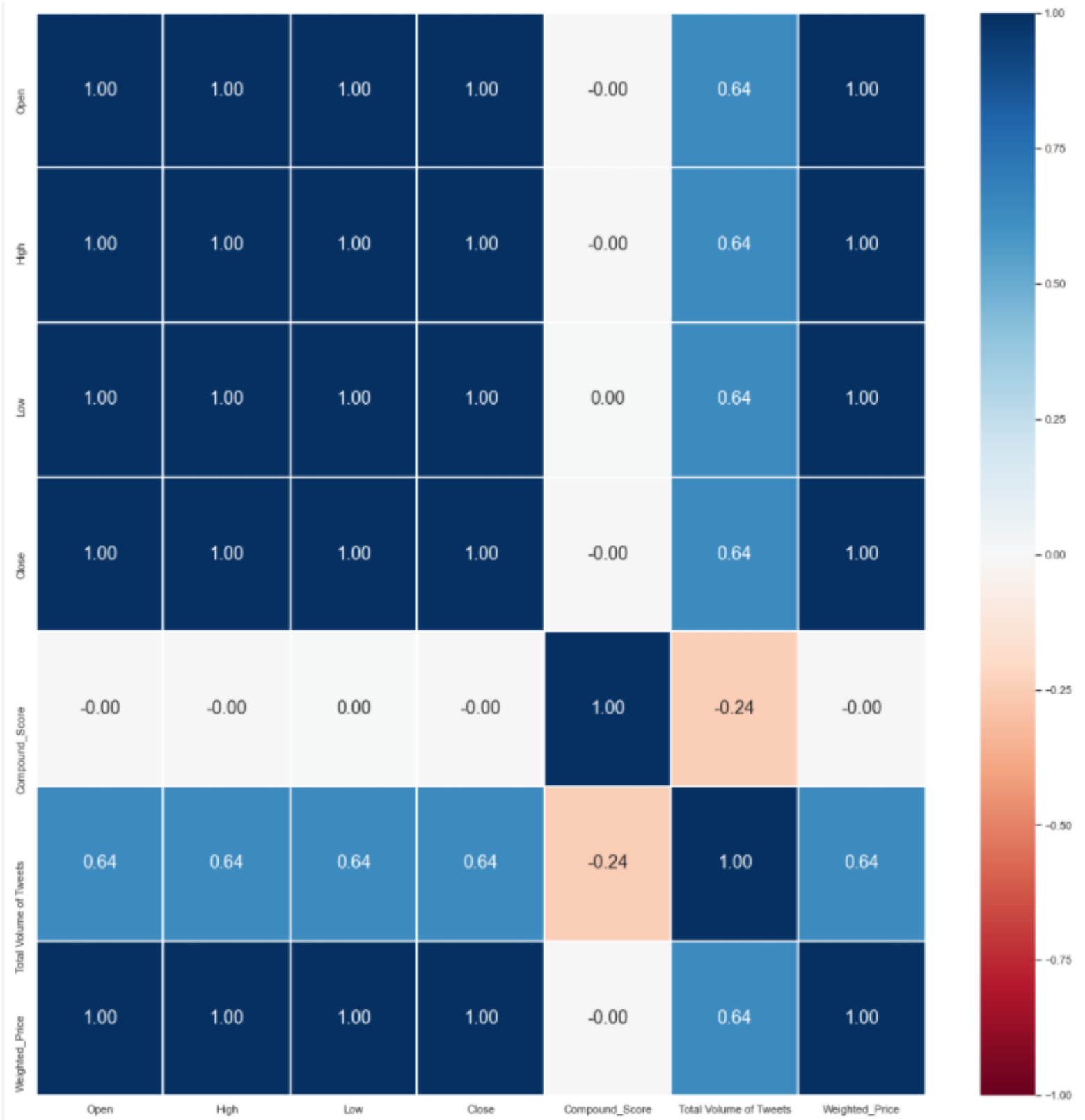

Figure 3: Heatmap of the correlation matrix

```
In [18]: y = pd.DataFrame(clean_df['Weighted_Price'])  # Response

In [15]: ohlc_predictors= ["Open", "High", "Low","Close"]
         d1 = clean_df[ohlc_predictors].reset_index()
         d1.drop(columns=["index"],inplace=True)

In [16]: sent_predictors= ['Compound_Score', 'Total Volume of Tweets']
         d2 = df[sent_predictors].reset_index()
         d2.drop(columns=["index"],inplace=True)

In [19]: clean = pd.concat([d1, d2, y], axis=1)

In [20]: clean.dropna(inplace=True)

In [21]: clean.head()
```

Out[21]:

	Open	High	Low	Close	Compound_Score	Total Volume of Tweets	Weighted_Price
0	2763.23	2763.24	2761.41	2762.00	0.082893	1027.0	2761.710702
1	2768.07	2772.97	2768.07	2768.07	0.053160	778.0	2772.411512
2	2779.77	2779.78	2779.77	2779.78	0.124251	836.0	2779.774992
3	2790.55	2793.25	2790.55	2790.55	-0.021037	984.0	2792.693685
4	2837.44	2837.44	2831.40	2831.40	0.055437	751.0	2832.734750

Figure 4: Data Processing and Manipulation

References

Alpha Vantage API Documentation. API Documentation | Alpha Vantage. (n.d.). Retrieved October 16, 2021, from https://www.alphavantage.co/documentation/.

Badiola, J. (2019, June 15). *Bitcoin 17.7 million tweets and Price*. Kaggle. Retrieved October 16, 2021, from https://www.kaggle.com/jaimebadiola/bitcoin-tweets-and-price.

Binance Academy. (2021, August 30). *What is crypto market sentiment?* Binance Academy. Retrieved October 16, 2021, from https://academy.binance.com/en/articles/what-is-crypto-market-sentiment.

Brownlee, J. (2021, February 15). *Regression metrics for machine learning*. Machine Learning Mastery. Retrieved October 16, 2021, from https://machinelearningmastery.com/regression-metrics-for-machine-learning/#:~:text=There%20are%20three%20error%20metrics,Mean%20Absolute%20Error%20(MAE).

How does sentiment analysis work? Keyhole. (2020, September 15). Retrieved October 16, 2021, from https://keyhole.co/blog/how-does-sentiment-analysis-work/.

Inamdar, A., & Bhagtani, A. (n.d.). *Predicting Cryptocurrency Value using Sentiment Analysis*. IEEE Xplore. Retrieved October 16, 2021, from https://ieeexplore.ieee.org/document/9065838.

Multilayer Perceptron. Multilayer Perceptron - an overview | ScienceDirect Topics. (n.d.). Retrieved October 16, 2021, from https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron.

Multivariate linear regression tutorials & notes: Machine learning. HackerEarth. (n.d.). Retrieved October 16, 2021, from https://www.hackerearth.com/practice/machine-learning/linear-regression/multivariate-linear-regression-1/tutorial/.

Ronaghan, S. (2019, November 1). *The mathematics of decision trees, random forest, and feature importance in Scikit-learn and Spark.* Medium. Retrieved October 16, 2021, from https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3.

Sharma, M. (2020, March 21). *Grid search for hyperparameter tuning*. Medium. Retrieved October 16, 2021, from https://towardsdatascience.com/grid-search-for-hyperparameter-tuning-9f63945e8fec.

Techopedia. (2017, March 30). *What is a Multilayer Perceptron (MLP)? - definition from Techopedia*. Techopedia.com. Retrieved October 16, 2021, from https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp.

Twitter. (n.d.). *API reference index | docs | Twitter developer platform*. Twitter. Retrieved October 16, 2021, from https://developer.twitter.com/en/docs/api-reference-index.

Yeoh, C. F. S. (1970, January 1). *Time Series & sentiment analysis of top cryptocurrencies*. Research Directory. Retrieved October 16, 2021, from https://dr.ntu.edu.sg/handle/10356/148200.

Zielak. (2021, April 11). *Bitcoin historical data*. Kaggle. Retrieved October 16, 2021, from https://www.kaggle.com/mczielinski/bitcoin-historical-data.

Footnotes

Assumptions made

We are assuming the cryptocurrency market is affected by only these 5-6 variables, which is however untrue because the market is extremely volatile. special events such as Election Results Day, UN global sessions, Independence Day, and various other occasions where the market doesn't behave rationally.

Areas of further exploration

Due to time constraints, a lot of intertwined fields were not explored. This project currently only considers Twitter data for sentimental analysis. However, with the GameStop (GME) stock market anomaly, we can assume Reddit is also a reliable source of information regarding the same. Thus, one area of further exploration is involving the Reddit API (PRAW) to also consider famous cryptocurrency-specific subreddit threads and discussion forums.

Also, this project currently focuses solely on Bitcoin, due to its centrality in the cryptocurrency market. However, we can investigate other tokens as well such as Ethereum, etc.

Alternatively, for easier user convenience we can inculcate this backend process onto a hosted website, or social media platform bots such as Telegram bots, Discord bots, WhatsApp bots, etc.

We can use GridSearchCV (a Python package under Scikit-learn) for hyperparameter tuning within a Random Forest Regressor or other models as well to fine-tune parameters to achieve optimal results.

The entire code, along with the datasets utilized shall be found in our GitHub Repository.