EDS6344 - AI FOR ENGINEERS

SPRING 2023

GROUP 2 – PROJECT REPORT

G-RESEARCH CRYPTO FORECASTING USING DEEP LEARNING

PROFESSOR : RASIAH LOGANANTHARA

GROUP_2:

Rahul Krishna Gunneri	**– 2210264**
SriDivya Mandadapu	**– 2211703**
Geethika Sai Kannikanti	**– 2210616**
Sreeja Reddy Nannuri	**– 2205712**
Sai Goutham Reddy Komatireddy	**– 2150153**

Contents

ABSTRACT:

Cryptocurrencies, such as Bitcoin, have gained significant attention in recent years, and their volatile nature makes them an attractive subject for financial forecasting Cryptocurrency has the potential to streamline existing financial infrastructure, making it faster and less expensive. Their technology and architecture decentralize existing monetary systems, allowing transacting parties to exchange value and money without relying on intermediary institutions like banks. Prices of several cryptocurrencies are highly interrelated. Bitcoin has historically been a key driver of cryptocurrency price movements, but other coins also have an impact on the market.

Predicting how prices will behave in the future is critical in financial modeling. Given the extreme volatility of the assets, the non-stationary nature of the data, market, and meme manipulation, asset correlation, and the extremely fast-changing market conditions, forecasting cryptocurrency returns remains an open and extremely difficult task. Time series forecasting is a popular technique for predicting the future prices of cryptocurrencies.

INTRODUCTION:

The use of cryptocurrencies has gained widespread attention in recent years, with the emergence of Bitcoin and other digital currencies. As a result, many investors and traders are interested in predicting the price movements of these assets in order to make informed investment decisions. One way to approach this problem is through time series forecasting, which involves analyzing historical price data to make predictions about future prices.

In this report, we are exploring the use of time series forecasting techniques for predicting cryptocurrency prices. Specifically, we will focus on the application of the popular time series model Prophet.

DATA DESCRIPTION:

The dataset is obtained from Kaggle. This dataset includes statistics on previous trades for several crypto assets, including Bitcoin and Ethereum. Forecasting the assets' future returns is our challenge. This dataset contains 10 variables which are given below.

timestamp: All timestamps are returned as second Unix timestamps (the number of seconds elapsed since 1970-01-01 00:00:00.000 UTC). Timestamps in this dataset are multiple of 60, indicating minute-by-minute data.

Asset_ID: The asset ID corresponding to one of the cryptocurrencies (e.g. Asset_ID = 1 for Bitcoin). The mapping from Asset_ID to crypto asset is contained in asset_details.csv.

Count: Total number of trades in the time interval (last minute).

Open: Opening price of the time interval (in USD).

High: Highest price reached during the time interval (in USD).

Low: Lowest price reached during the time interval (in USD).

Close: The closing price of the time interval (in USD).

Volume: The number of crypto asset units traded during the minute.

VWAP: The asset's average price over the time interval, weighted by volume. VWAP is an aggregated form of trade data.

Target: Residual log-returns for the asset over a 15-minute horizon

Each asset ID refers to respective asset name which refers to different crypto currencies. There are 14 cryptocurrency data present in this exhausting dataset.

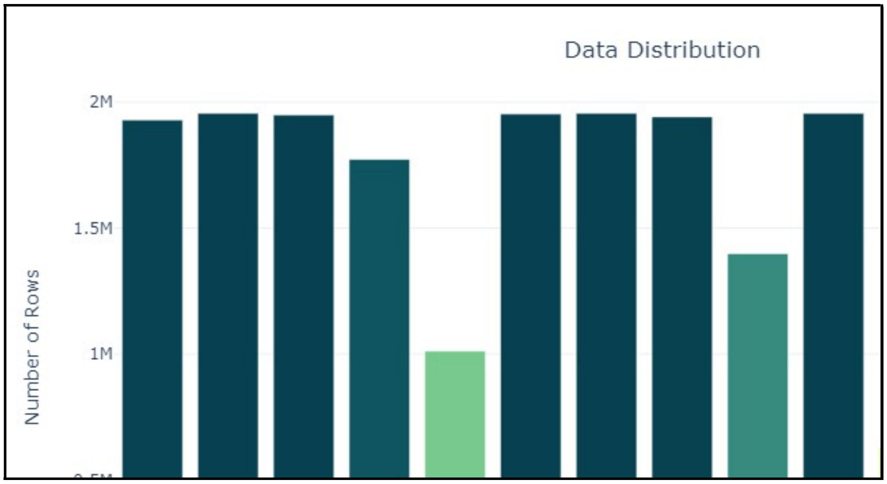
```
In [6]: asset_details
Out[6]:
```

	Asset_ID	Weight
0	2	2.397895
1	0	4.304065
2	1	6.779922
3	5	1.386294
4	7	2.079442
5	6	5.894403
6	9	2.397895
7	11	1.609438
8	13	1.791759

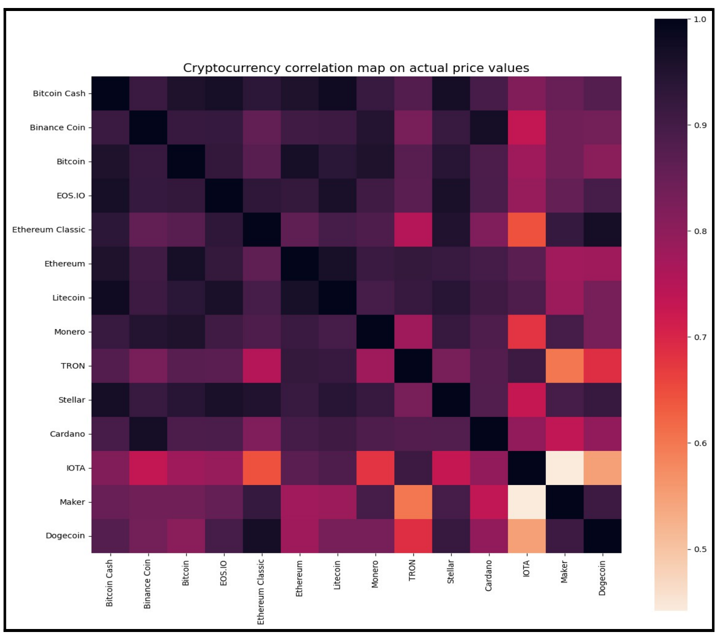
```
{1: 1955
 6: 1955
 9: 1955
 5: 1952
 2: 1948
 7: 1941
 0: 1929
 13: 185
 3: 1773
```

DATA VISUALIZATION:

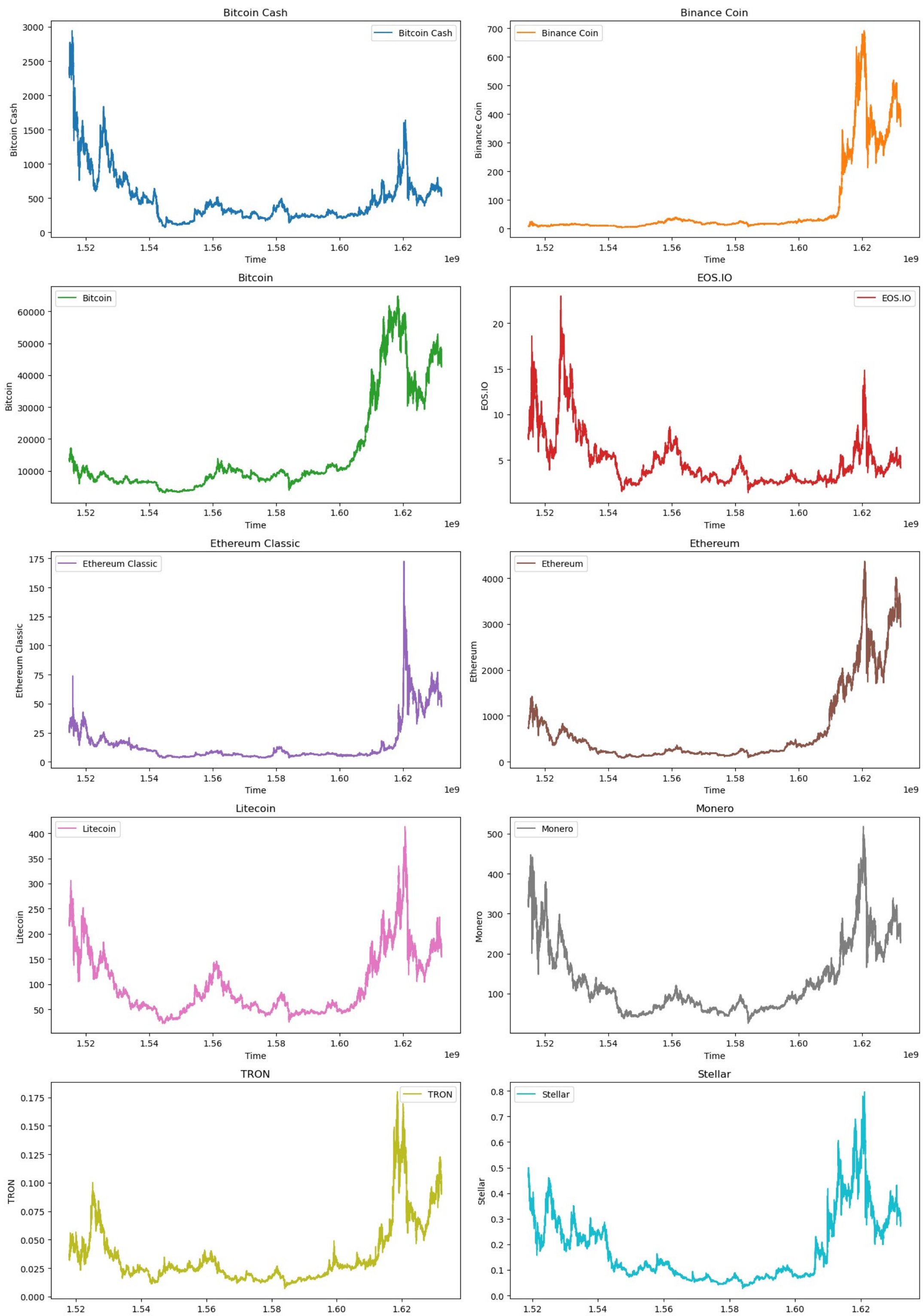
To understand this overwhelming dataset, data visualization is a best practice to learn the patterns in the dataset. There are some visualizations showing the trends and patterns of the data.

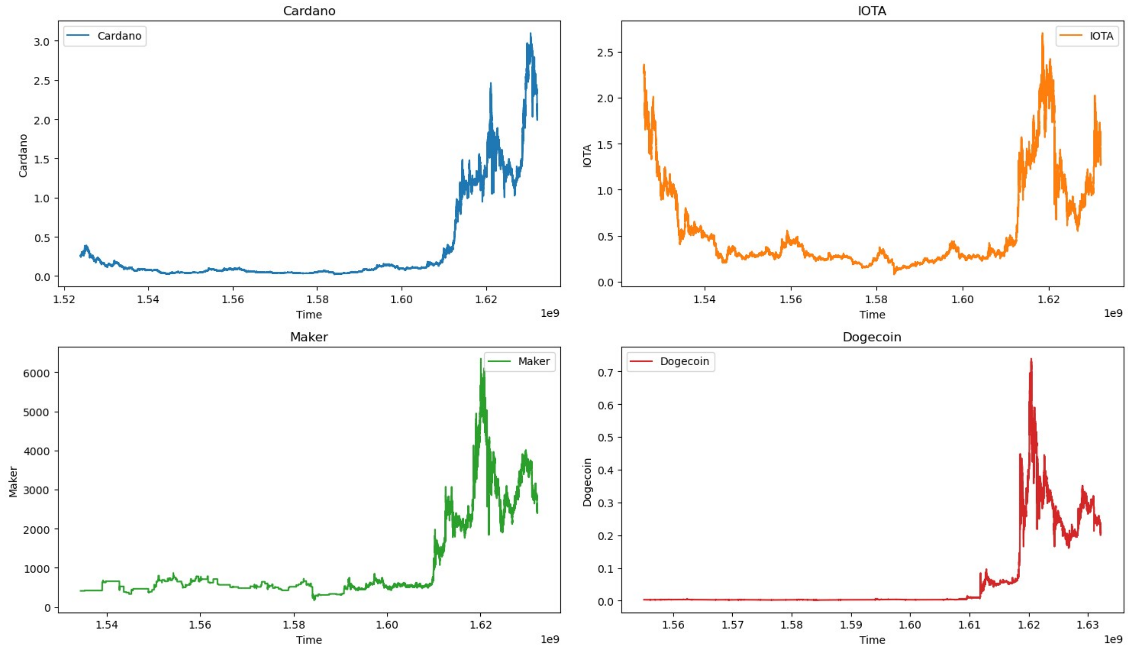


The above graphs show the number of records for each asset Id, from this graph one can say that higher the bar of respective asset Id implies the greater number of records and it's older crypto currency.



Above correlation heatmap can disclose the relation between one assets market with other one. High correlated assets tend to perform similarly at the market. From this we can say that one can implement same models for the correlated assets.


Bitcoin Cash
Binance Coin
Bitcoin
EOS.IO
Ethereum Classic
Ethereum
Litecoin
Monero
TRON
Stellar
Time
1e9



We are visualizing trends for each 'Asset ID' i.e, Bitcoin Cash, Binance Cash, Bitcoin, EOS.IO, Ethereum Classic, Ethereum, Litecoin, Monero, TRON, Stellar, Cardano, IOTA, Maker, Dogecoin with 'Time.'

DATA PREPROCESSING:

Handling Null Values:

- 3.10% null values are present in the original data which means we have in total 7,50,338 null value rows.

```
Total Null Target Rows =  750338
Percentage of NUll rows in Training Data = 3.10%
```

- Instead of removing null values if we plan to replace with mean or some calculated value then it consumes lot of computational cost. So, it's good idea to remove the null values.
- Even though after dropping null values from the original dataset we still have 2,34,86,465 values in the dataset, so we are removing null values doesn't impact.

Data Transformation:

One needs to convert the 'timestamp' from int format to datetime format, from this column we have extracted specific components e.g., year, month, day, hour, and minute for our analysis. There are 1440 unique dates present. Analysis of minutes and hours also doesn't help much because the data is still huge to handle. Therefore, we resampled the entire data based on unique dates in such a way to reduce the size of the dataset and dropped minute and hour attributes.

```
0     1514764860
1     1514764860
2     1514764860
3     1514764860
```

```
 0    2017-12-31 18:01:
1    2017-12-31 18:01:0
2    2017-12-31 18:01:0
3    2017-12-31 18:01:0
```

```
In [15]: unique_dates=len(list(
         unique_dates
```

Resampling :

We are resampling the data based on date taking the average of other attribute in such a way not to lose the statistical value of the data. And creating individual data frame for each asset ID then storing them in different excel files in a way to reduce reloading the entire dataset.

```
df_0 (1929204, 14)
df_1 (1955978, 14)
df_2 (1948676, 14)
df_3 (1773136, 14)
df_4 (1011892, 14)
df_5 (1952838, 14)
df_6 (1955860, 14)
df_7 (1941801, 14)
df_8 (1398965, 14)
df_9 (1955509, 14)
df_10 (630469, 14)
df_11 (1461658, 14)
df_12 (1716933, 14)
df_13 (1853546, 14)
```

```
df_0(1359, 11)
df_1(1360, 11)
df_2(1360, 11)
df_3(1253, 11)
df_4(893, 11)
df_5(1360, 11)
df_6(1360, 11)
df_7(1360, 11)
df_8(1231, 11)
df_9(1360, 11)
df_10(619, 11)
df_11(1360, 11)
df_12(1313, 11)
df_13(1323, 11)
```

MODEL BUILDING:

Prophet is a time series forecasting model to accurately forecast time-series data and this is an open-source library for Python that was developed by Facebook's Data Science team It uses a decomposable time series model with three main components: trend, seasonality, and holidays. Prophet is particularly useful for forecasting time series data that has strong seasonal patterns and can handle missing data and outlier values. With its user-friendly interface and automatic parameter selection, Prophet can be easily implemented by data analysts and business users alike. It has been widely adopted in industries such as retail, finance, and healthcare to improve forecasting accuracy and decision-making.

Prophet can manage missing data, outliers, and changes in trends over time, making it an effective tool for forecasting time-series data. The impacts of holidays and other one-time occurrences, which might significantly affect the statistics, can also be taken into consideration.

In Prophet, the input data should be structured with two columns: 'ds' for the timestamps and 'y' for the corresponding measurements or values here we are taking ‘Close’ values as ‘y’.

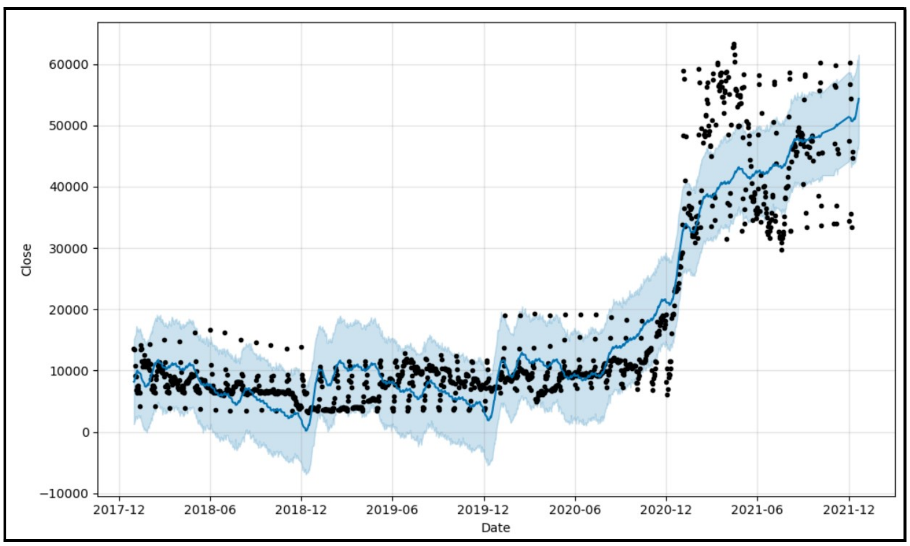
	date_	Asset_ID	Count	Open	High	Low	Close	
0	2017-12-31	0.0	13.132565	8.378154	8.388847	8.365258	8.376281	48
1	2018-01-01	0.0	15.528148	8.285865	8.296446	8.272346	8.283114	59
2	2018-01-02	0.0	54.082811	9.546879	9.569352	9.521549	9.545497	170

ds	
2017-12-31	1354
2018-01-01	1342
2018-01-02	923

- Predicted the data for the last 365 days and We are generating a forecast for the time series data using the trained Prophet model.

```
In [149]: future = m.make_future_dataframe(periods=365)
          forcast = m.predict(future)
          forcast.head()
```

- Generated a plot of the forecasted values for the dependent variable.



- The above figure displays the projected values as a blue line and the historical values of the dependent variable as black dots. There are also shaded sections on the plot to illustrate the forecast's uncertainty, with brighter shades denoting a broader range of uncertainty and darker shades denoting a smaller range of uncertainty.

- The plot's x-axis and y-axis are added with labels using the xlabel and ylabel options, respectively. The x-axis label in this instance is "Date," while the y-axis label is "Close."
- And generated a plot that shows the trend, weekly seasonality, and yearly seasonality components of the time series data.

RESULT:

We created prophet models based on two values, one is the date column which is represented as ds and acts as input and the other is the closed value column which is represented as y and acts as output.

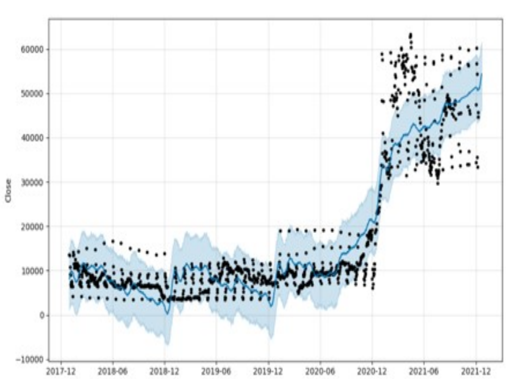
The dataset is divided into two sets: the training set and the test set. The last 50 dates in the dataset are considered as the test set. If you look at the graphs that are obtained, we are not only forecasting a single value but a probability band range. Where the line represents y hat value, and the probability band range is the difference between yhat upper and yhat lower.

Here the y-hat upper is the values above the line and the yhat lower are the values below the line. We built the model for the 14 assets, considering the same two values as input and output.

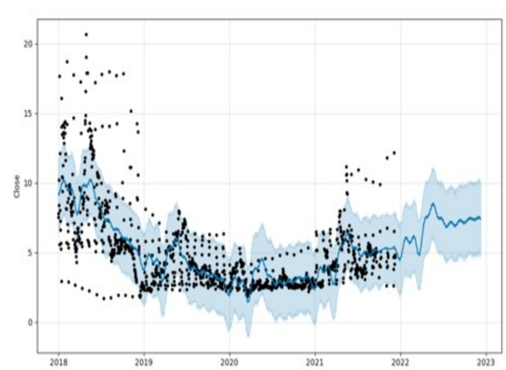
	ds	trend	yhat_lower	yhat_upper	yhat
0	2017-12-31	635.394249	185.604309	1136.227888	641.159783
1	2018-01-01	634.362785	191.527490	1143.540606	647.492712
2	2018-01-02	633.331320	242.360365	1169.546076	682.323339
3	2018-01-03	632.299856	216.058889	1088.503886	661.954469
4	2018-01-04	631.268392	243.759074	1144.082959	687.427891

Below are the prophet models of few assets:

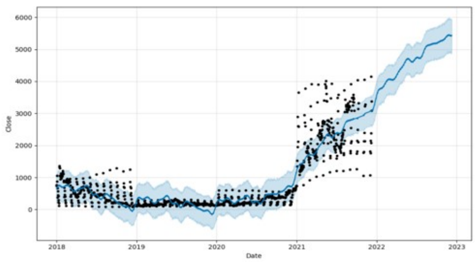
Bitcoin asset:



Etherium asset:



Litecoin asset:



CONCLUSION:

While the prophet model performs poorly for Litecoin, it performs better for Bitcoin Cash and Binance Coin.

- These MSE, MAE values are very large because of the range of values, this needs normalization or other scaling methods to improve the MSE, MAE scores.

```
MSE: 129974702.93662077
MAE: 9094.816487369208
MAPE: 0.2254800272786834
smape: nan
```

```
MSE: 3.666052836771796
MAE: 1.151116686674943
MAPE: 0.21328310108316337
smape: nan
```

```
MSE: 1138249.302428968
MAE: 839.7237951004342
MAPE: 0.45259909021547595
```

Future Scope:

- One needs to implement an encoder model with complex deep learning models to get more accuracy must be used to forecast Close value to maximize profit.
- By data integrating, and connecting the economy and other factors, we can improvise accurate predictions of the market values.

REFERENCES:

https://www.kaggle.com/code/vbmokin/g-research-crypto-forecasting-baseline-fe

https://www.kaggle.com/code/tahahabib/crypto-forecasting

https://www.kaggle.com/competitions/g-research-crypto-forecasting/data