Information Processing and Management 59 (2022) 102968

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Real-world model for bitcoin price prediction

Rajat Kumar Rathore [a], Deepti Mishra [a], Pawan Singh Mehra [b,*], Om Pal [c,*], AHMAD SOBRI HASHIM [d], Azrulhizam Shapi'i [e], T. Ciano [f], Meshal Shutaywi [g]

[a] G.L.Bajaj Institute of Technology and Management, Greater Noida, India
[b] Department of CSE, Delhi Technological University, New Delhi
[c] Ministry of Electronics & Information Technology, Govt of India, India
[d] Computer & Information Sciences Department, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia
[e] Center For Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Malaysia
[f] Faculty of Business and Law, University of Portsmouth, Richmond Building, Portland Street, Portsmouth, United Kingdom of Great Britain
[g] King Abdulaziz University, College of Science & Arts, Department of Mathematics, Rabigh, Saudi Arabia

ARTICLE INFO

Keywords:
Bitcoin
Cryptocurrency
Machine learning
Prediction
Time series analysis
Fbprophet model

ABSTRACT

Cryptocurrency is a new sort of digital asset that has evolved as a result of advances in financial technology, and it has provided a significant research opportunity. There are many algorithms for price prediction for crypto currencies like LSTM and ARIMA. However, the downside is that LSTM-based RNNs are difficult to comprehend, and gaining intuition into their behavior is tough. In order to produce decent outcomes, rigorous hyperparameter adjustment is also essential. Furthermore, crypto currencies do not precisely adhere to past data, and patterns change fast, reducing the accuracy of predictions. Cryptocurrency price forecasting is difficult due to price volatility and dynamism. Because the data is dynamic and heavily influenced by various seasons, the ARIMA model is unable to handle seasonal data. In order to provide better price predictions for crypto traders, a new model is required. The objective of the study is to apply Fbprophet model as the key model because it is superior in functionality as compared to LSTM and ARIMA additionally removing the pitfalls generated in LSTM and ARIMA model while analyzing the cryptocurrency data. This study provides a methodology for predicting the future price of bitcoin that does not rely solely on past data due to seasonality in historical data. So, after fitting the seasonality and smoothing, the model is constructed that can be useful for real-world use cases. In case of crypto currencies where less historical data is available and it is hard to find pattern, proposed method can easily deal this type of problems. Overall difference between predicted and actual values is low as compared to other model even after seasonal data was available.

1. Introduction

Cryptocurrencies are virtual or digital currencies that are used in today's financial systems. Because no government, central authority, or bank issues these virtual currencies, these are decentralized. All cryptocurrencies are based on Blockchain technology,

* Corresponding authors.
E-mail addresses: rajatatgio@gmail.com (R.K. Rathore), itsdeepti.s@gmail.com (D. Mishra), pawansinghmehra@gmail.com (P.S. Mehra), ompal.cdac@gmail.com (O. Pal), sobri.hashim@utp.edu.my (A.S. HASHIM), azrulhizam@ukm.edu.my (A. Shapi'i), mshutaywi@kau.edu.sa (M. Shutaywi).

https://doi.org/10.1016/j.ipm.2022.102968
Received 31 January 2022; Received in revised form 25 April 2022; Accepted 28 April 2022
Available online 19 May 2022

which is incredibly complicated and tries to store data in such a way that it is tough and beyond expectations to hack and modify. Cryptography further secures these currencies, making it hard to create fraudulent cryptocurrencies. Cryptocurrencies are still in their infancy, and it is impossible to say if these can ever be extensively adopted in global markets, but El Salvador became the first country to use bitcoin as legal money alongside the US dollar in November 2021.

There are a bunch of cryptocurrencies flowing in the digital market. In such cryptocurrencies Bitcoin is the most well-known, and it is impacted and influenced by external variables such as social sites, digital data, market analysis and so on. It was first introduced by Nakamoto in 2008 (Giudici, Milne & Vinogradov, 2020). Peer-to-peer transactions are possible thanks to blockchain technology. Blockchain, the technology that underpins the Bitcoin cryptocurrency system, is extensively critical for assuring enhanced security and privacy across a variety of sectors, comprising the IoT (Arias-Oliva, Pelegrín-Borondo & Matías-Clavero, 2019). It is basically a distributed digital record of transactions that spans the whole network of computer systems that make up the blockchain. The first component of the blockchain is a transaction, and the second is a block (Jaquart, Dann & Weinhardt, 2021; Golbabai & Ezazipour, 2019). The participant's activity is represented by the transaction, and the block is a data collection that stores the transaction as well as extra information like the right sequence and creation timestamp.

The cryptocurrency market is currently the most attractive domain for financial speculation. Many individuals have gained a lot of money by speculating in the digital markets, and also marketing investment process which are filled with concealed pitfalls still endowing in bitcoins, ethereum etc. (Giudici et al., 2020; Arias-Oliva et al., 2019). The number of machine learning algorithms to use to reduce these risks, as well as a comprehensive set of possible market-predictive characteristics (Jaquart et al., 2021). Due to the capacity to dynamically pick from a potentially enormous number of characteristics and understand complicated, high-dimensional correlations between features and targets, machine learning approaches have become more popular in this sector (Golbabai & Ezazipour, 2019; Ezazipour & Golbabai, 2020). Artificial Intelligence with machine learning techniques is attractive due to variances in forecasting capacity per coin Sebastião and Godinho (2021). Low-volatility cryptocurrencies are more predictable than high-volatility cryptocurrencies.

Because of price fluctuations and instability, cryptocurrency prices are tough to predict. This vacuum in the field is filled by comparing several machine learning models for forecasting market movements of the most relevant cryptocurrency—bitcoin. Investing in Bitcoin is comparable to investing in stocks. none of the risk factors that can identify negotiations and changes in stock price related to cryptocurrencies.

In case of stocks, there are various parameters available like P/E value, ROE(return on equity), ROCE, EBITDA, etc. to predict the prices of stocks but in case of crypto currencies there are only few parameters available, price movement is highly volatile and dynamic in nature, trends and pattern are complex and changes dynamically within a short period of time and there are multiple seasonality with hourly, daily and weekly data. Most of trading algorithms requires a lot of parameter, they highly depends on historical data,they cannot handle high volatility and strong multiple seasonality. Due to highly dynamic and volatile nature,there might have missing observations and large outliers. There are various algorithms available for them like ARIMA and LSTM-based recurrent neural network but all these requires many parameter, LSTM based RNNs are difficult to interpret and it is challenging to gain intuition into their behavior, these algorithm are good for long term pattern but in case of crypto pattern and trends changes within few hours. So, there is a need of algorithm which can handle all these problems and can even deal with holidays known in advance and missing observation and large outliers and seasonal effect cause by human behavior and can provide more accurate prediction

In this paper, the aims is to achieve the a model to predict closing price of Bitcoin along with Bitcoin Opening Price, Bitcoin Day High Price, Bitcoin Day Low Price, Bitcoin Day Volume and Market Capitalization of Bitcoin on particular Day by using deep leaning algorithms and various concepts of machine learning, which can find hidden patterns in data, combine them, and make considerably more accurate predictions. To achieve the aim the following task will be performed:

a) Descriptive Analysis
b) Exploratory Data Analysis Along with Data Pre-Processing
c) Statistical Test to remove seasonality and make it stationary {using Ad fuller Test}
d) Data Transformation
e) Preparing the data, smoothing data and try to adjust seasonality. The concept like Differencing is often used.
f) Building a Time-Series Model for which Fbprophet Library is being used.
g) And at last, cross-validation is performed on model.

After applying all these steps, the generated model is:

1. That can perform all features of previous models.
2. That doesn't require much prior knowledge or experience of forecasting time series data since it automatically finds seasonal trends beneath the data and offers a set of 'easy to understand' parameters.
3. That can Deal with holidays known in advance, missing observations, and large outliers.
4. That can encounter means hourly, daily, or weekly observations with strong multiple seasonality's.

The fundamental goal of these models is to create a trustworthy prediction model based on previous bitcoin prices that investors can trust. The article also attempts to address few issues like 1. Utilizing the machine learning techniques in prediction of cryptocurrency prices for investors and decision makers. 2. Selection of best fit model for predicting prices of cryptocurrency.

Next section discusses about the study related to crypto currency. Section 3 provides the details of the dataset. Detailed

methodology is discussed in detail in Section 4 which is followed by section 5 explaining the implementation and results. Further section 6 is related to discussion of the results followed by conclusion.

2. Literature survey

There is very less availability of price prediction models for Bitcoins as it a modern technology in current scenario (Hitam & Ismail, 2018). Time series models interacts with data from daily time series, 10-minute, and 10-second intervals. These models constructed three time series data sets for 30, 60, and 120 min, then used GLM/Random Forest to generate three linear models from the datasets. To estimate the price of Bitcoin, these three models are linearly integrated (FAUZI & PAIMAN , 2020) handles with data from daily time series, 10-minute, and 10-second intervals. To estimate the price of Bitcoin, these three models are linearly integrated. Instead of directly anticipating the stock's future price, the writers in anticipate the stock's trend. A pattern may be drawn from the trend. These can make both small (day or week-long) and large forecasts (months). It can be discovered that the long forecasts gave superior outcomes, with an accuracy rate of 79 percent. Another intriguing method reflected in the study is the network's performance evaluation criteria which functions on projected output. The authors applying machine learning approaches handles both deep learning technique and regression techniques for prediction of Bitcoin integrating with gradient descent and linear search.

According to a study published applied for high dimensional data which is related to Bitcoin daily price prediction, shows that logistic regression and linear discriminant analysis reach a 66 percent accuracy rate. Outpacing (a complex technique which is based on machine learning), on the other hand, outperforms the standard results for everyday prediction for price, with 66 percent and 65.3 percent accuracies for statistical approaches and machine learning algorithms, respectively. The study demonstrated two types of prediction models created using Bayesian optimized RNN and LSTM to forecast the price of BTC. (Jamali, Sadegheih, Lotfi, Wood & Ebadi, 2021). The study found that LSTM performed better, with a 52 percent accuracy and an RMSE of 8%.

Time series are a specific instance of LSTM-based recurrent neural networks, which are perhaps the most powerful way to learning from sequential data. When learning from large datasets with complex patterns, the potential of LSTM-based models is fully realized. They do not rely on certain assumptions about the data, such as time series stationarity or the availability of a Date field, like ARIMA or Prophet do. However, the downside is that LSTM-based RNNs are difficult to comprehend, and gaining intuition into their behavior is tough. In order to produce decent outcomes, rigorous hyperparameter adjustment is also essential. In the case of cryptocurrencies, substantial multiple seasonality has a significant impact on LSTM accuracy.

Most of the investment process is based on a cryptocurrency's previous pricing. Building Markov chains is one of the most essential tactics used by investors. This technique entails using numerous decision trees to select the cryptocurrency that is expected to produce a higher return when sold, as well as comparing the anticipated return to the actual amount. is an example. ANN is also very effective in optimizing problems (Ebadi, Hosseini & Hosseini, 2017; Tirandazi, Rahiminasab & Ebadi, 2022; Jamali et al., 2021). According Markov chain models, the transitioning from one state to another state solely depends on the current state,in these models we can see an ignorance of all previous trends and other than that in case of cryptocurrencies is might possible that the current state is a result of seasonality and volatility, at this conditions the efficiency of model might compromised. Whereas proposed prophet model manages the trends as well it able to predict the trends at the currents states. Prophet model is also able to deal with missing data values and outlier along with the seasonality affects or trends. From few years, in time series forecasting a ARIMA model is widely spread i.e. known as autoregressive integrated moving average (Farhath, Arputhamary & Arockiam, 2016). ANN also an option and substitute for forecasting (Tealab, 2018). The superiority of ARIMA models and ANNs in forecasting performance is frequently compared, with inconsistent results (Wang, Zou, Su, Li & Chaudhry, 2013). The study suggested a hybrid model applying ARIMA and ANN (Khandelwal, Adhikari & Verma, 2019). The combined model can be an effective technique to increase forecasting accuracy attained by any of the models used alone, according to experimental findings with real data sets. The results signifies that the Bitcoin prediction can be accurate by applying ML ensemble method (Jaquart et al., 2021). Decision making should be done at right time interval while reducing the risk. Over few decades Autoregressive integrated moving average (ARIMA) is one of the most widely used linear models in time series forecasting (Zhang, 2003). Recent research into artificial neural networks (ANNs) for forecasting suggests that ANNs might be a viable alternative to standard linear approaches. The superiority of ARIMA models and ANNs in forecasting performance is frequently compared, with inconsistent results. To take use of the distinctive strengths of linear and nonlinear concept in ARIMA and ANN, a hybrid technique including both ARIMA and ANN models is suggested in this study (Chen, Li & Sun, 2021). Experimental findings signify, for improving forecasting accuracy the integrated model is more efficient in comparison to other models if applied separated and isolated.

However, when it comes to cryptocurrency, where data is restricted in terms of characteristics, seasonality can occur on a weekly, daily, or even hourly basis, data has big outliers, and does not entirely rely on historical data, the market in cryptocurrencies moves dynamically within a short period. All of these may have a negative impact on the ARIMA model's performance. Further tuning is sometimes required for algorithms like ARIMA to generate respectable results, which is out of reach for many people who are not properly qualified specialists. To address the constraints of the ARIMA model in terms of crypto trading, a new model based on Facebook Prophet is developed in this study. Because Prophet is primarily built to find patterns in business time series, it requires minimal hyperparameter tweaking. Prophet is robust to missing data and trend shifts, and it can usually manage outliers and trend shifts caused by new items and market events. Unlike auto.arima, Prophet shows a realistic seasonal pattern, even if the absolute numbers are a little off from the actual data. Prophet is unique in that it requires no prior knowledge or expertise in forecasting time series data since it automatically detects seasonal trends underlying the data and provides a set of 'simple to comprehend' parameters. Prophet is also built to deal with holidays that are known ahead of time, missing data, and significant outliers. As a result, even non-statisticians may use it and achieve pretty decent results that are often on par with, if not better than, those generated by specialists.

The patterns of time series are complex and vary dynamically over time, but Prophet only pays attention to such changes when the trend shifts. The seasonality prior scale is ineffective, however the greater trend prior scale performs better. However, because Prophet, unlike other models, does not directly consider recent data points, if there are some seasonality patterns in the dataset and these patterns are not consistent or smooth (as in the case of Cryptocurrencies), this can severely hurt performance when prior assumptions do not fit. To overcome all of Facebook Prophet's limitations, first and foremost, data must be subjected to Quantitative forecasting, which will include trend projection, the Naive technique, assessing seasonality using the Adfuller Test, moving averages, and exponential smoothing. After all of this, the FbProphet model will be able to outperform all other models by a large margin.

Comparison of LSTM and ARIMA model

In Fig. 1, we had used data Bajaj Finserv Ltd., an Indian Financial Company in Order to compare the two models in Fig. 1, The Data spans the period from 2008 until end of 2021. From the Fig. 1, we can clearly see ARIMA model yields better performance than LSTM

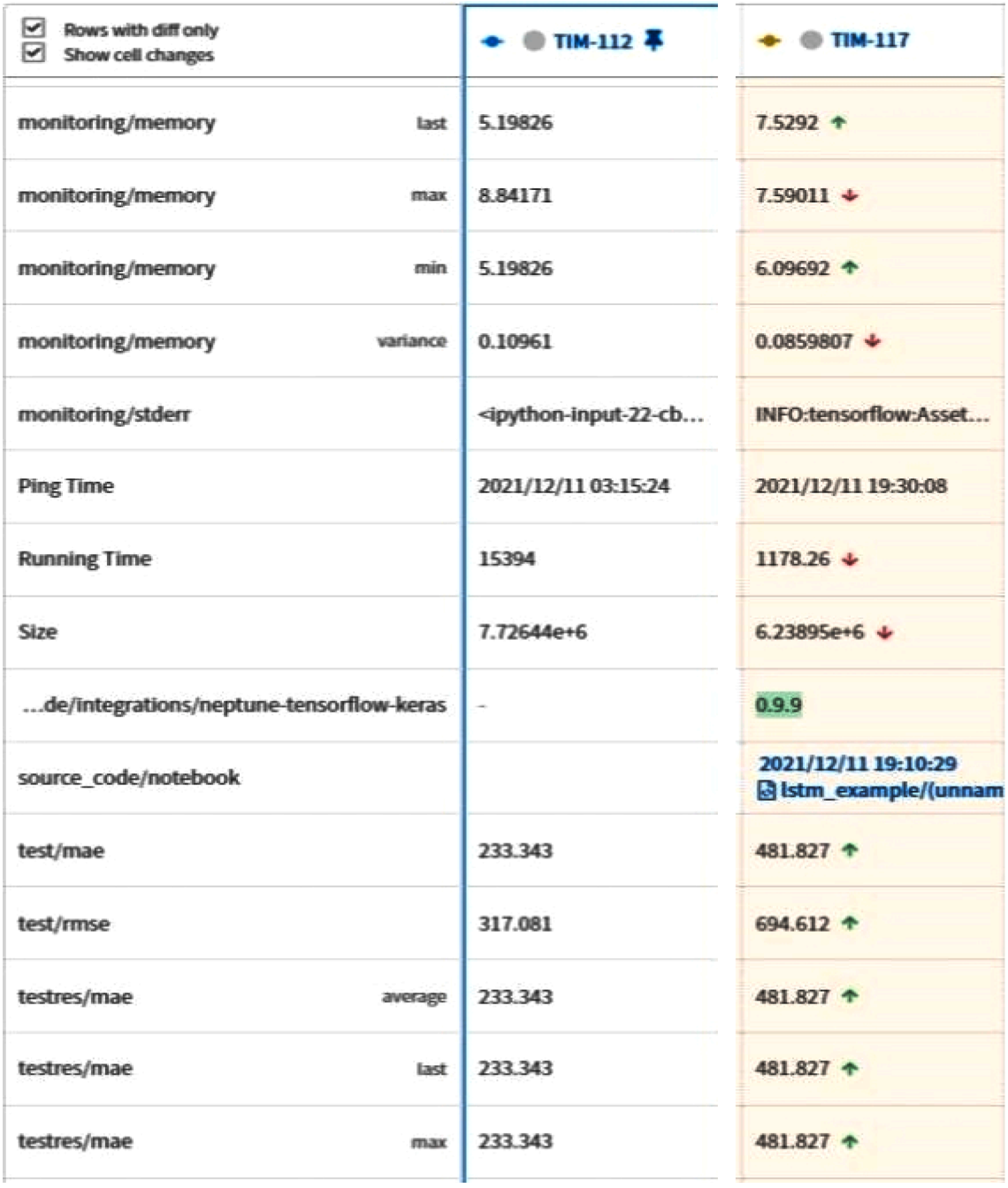

Fig. 1. The mean square error and the mean average error ARIMA and LSTM models can be seen next to each other.

model.

3. Datasets

The data for this study came from an open-access website - https://www.kaggle.com/team-ai/bitcoin-price-prediction/version/1. Used data is historical data to form Naïve model and to search for historical trends. Data can be use at real time using Big data concepts like Spark Streaming and Kafka. Data is made up of a single.csv file which consists of date, open, high,low, volume and market cap. Of Bitcoin. This .csv file contains record of 1556 days from 28th April 2013 to 31st July 2017. ~~Data can also be downloaded from https://in.tradingview.com/chart/?symbol=COINBASE%3ABTCUSD from where data can easily export data in form of .csv file. The dataset used in project do not have outliers, but if dataset have some outliers and missing values they can be removed using averaging, replacing null value with mean similar type of data,by adding and subtracting the variance in outliers etc.~~

~~The data for this study came from an open-access source. It is made up of a single.csv file which consists of date, open, high, low, volume and market cap. Of Bitcoin. This .csv file contains record of 1556 days from 28th April 2013 to 31st July 2017. This model can be applied on any other Data of similar type.~~ Table 1 shows sample of raw data.

~~Raw Data contains "Open" which represents opening price, "High" represents highest price, "Low" represents lowest price and "Close" represents closing price of Bitcoin on particular "Date". The available data is raw data which might have some outliers, so firstly data need to be prepared. For data preparation, libraries like Pandas(extremely used for data manipulation and data cleaning), Numpy (to perform numerical operation on data) and for data visualization libraries like matplotlib and seaborn are used.~~

~~Description of Data is as follows.~~

~~The datatype of Raw Data is As Follows~~

~~Table 1 shows a sample of the raw datasets,~~ Table 2 shows description of the dataset like mean, standard deviation, minimum,etc. and Table 3 shows the datatype of columns in the Dataset.

In above figure "Date" have object datatype, but "Date" feature must support something known as timestamp nature because it is the must condition for a Time Series case so firstly convert it to datetime format and then to maintain hierarchy, sort the data according to Date. For Time Series, it is must to make "Date" as index feature to avoid key Error.

4. Methodology

~~To achieve the aim of paper, firstly fetch raw data from third party API's or the data can be extracted from some big data bases like MongoDB or web Scrapping. Then a lots of data cleaning is performed on raw data and must perform Exploratory Data Analysis (EDA) on this data. Now from this data to build a model, in such case, Multiple algorithms can be used. One of them is Naive Model which can also be termed as Base Line Model, then there are some amazing models like auto regressive model, then there are some moving average model, then there is ARIMA model, so there are tons such models available there for data.~~ If there exist some seasonal data, then there is something known as SARIMAX. During data cleaning, once have cleaned data over here there is one more thing which is exactly known as feature engineering which will almost take 70% for this entire project. In this feature engineering it can be detected that whether data is Stationary or not. So to check, there are a lots of steps like some statistical test from this the trend of the feature can be understood by basically using Line plot function or Line plot curve, then there are some statistical test like Augmented Dickey Fuller Test(adfuller Test) one of the famous Test to detect whether data is Stationary or not because stationarity is crucial term relating to time series analysis that affects robustly the interpretation and analysis of data. Every data point is assumed independent in time series models for prediction and forecasting. Stationary data is the crucial concept, it should be consistent and not vary with time. It can be concluded that values may be different by avoiding the consistency in general. For sustaining the stationarity mean and variance are key values. Time graphs are generated for price of bitcoin

~~Very first there is to work with multiple libraries like Pandas which is extensively used in case of data manipulation or data cleaning then there is NumPy which is exactly Numerical python which will be used when there have to perform some numerical stuff on data, then to deal with data visualization stuff Matplotlib will be used even sometime Seaborn which will return some interactive visualization compare to matplotlib and there is one more library which is Sklearn which is extensively used in case of Data Modelling and with respect to time series use cases library known as stats models will be used. Fig. 2 shows raw sample data.~~

~~In above Fig. 3 date have object datatype, but date feature must support something known as timestamp nature because it is the must condition for a Time Series case so firstly convert it to datetime format and then will sort the data according to Date. Then make Date as index feature to avoid key Error.~~

Now, when it comes to predicting, there are two basic types of forecasting: quantitative and qualitative forecasting.

Table 1
Sample of Raw Data.

	Date	Open	High	Low	Close	Volume	Market Cap
0	Jul 31, 2017	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000
1	Jul 30, 2017	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000
2	Jul 29, 2017	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000
3	Jul 28, 2017	2679.73	2897.45	2679.73	2809.01	1380,100,000	44,144,400,000
4	Jul 27, 2017	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000

Table 2
Description of Raw Data.

	Open	High	Low	Close
count	1556.000000	1556.000000	1556.000000	1556.000000
mean	582.625328	597.992847	567.851446	584.239396
std	523.137312	542.992855	505.877401	525.904442
min	68.500000	74.560000	65.530000	68.430000
25%	254.287500	260.327500	248.835000	254.320000
50%	438.600000	447.560000	430.570000	438.855000
75%	662.437500	674.525000	646.735000	663.402500
max	2953.220000	2999.910000	2840.530000	2958.110000

Table 3
Datatype of Raw Data.

```
Date          object
Open         float64
High         float64
Low          float64
Close        float64
Volume        object
Market Cap    object
dtype: object
```

Fig. 2. Raw Sample Data.

1 Quantitative forecasting makes use of data that can be measured. It makes use of previous data that is both dependable and accurate, which removes the possibility of forecasting inaccuracy and bias.
2 Qualitative forecasting relies on non-quantifiable data. It is excellent for new firms that do not have any or much historical data because it is based on views and expert recommendations. Qualitative approaches are beneficial, but it's critical to consider the data in a nonjudgmental and unbiased manner.

Because historical data is accessible, this research will concentrate on quantitative forecasting.

The four techniques to quantitative forecasting are trend projection, the nave approach, moving averages, and exponential smoothing.We will use all of the ways to improve forecast accuracy and will continue to expand our model by using a few additional

Fig. 3. Datatype of different column of raw data.

approaches to improve performance.

Trend projection : One of the most common types of business forecasting is trend projection. It's quite straightforward to comprehend, since it analyses prior trends in a time-series and predicts that the same trends will occur in the future.

1. a) Now there must performed exploratory data analysis considering the 'Close' feature the end goal is to predict what is the closing price of Bitcoin, so a copy of data in 'Data' will be created and analyze the close feature deeply to understand it is trend. To understand it is trend line plot will be used and get Fig. 4

From Fig. 5-line plot there can be seen in 2014 there is a spike in closing price of Bitcoin and then in 2018 again there is a spike. b) Now, resample the data in a particular date range and after resampling the sum of closing price is found, for average closing price the mean function will be used and use various such functions and plot all these values with respect to date feature in Fig. 6. In Fig. 6 plot sum on yearly basis.

In Fig. 7 there can see the exact closing price with respect to different year using mean function. Similarly, it can be applied to distinct functions on yearly, monthly, quarterly bases etc.

3: a) Analyze average weekly closing price: for this group by 'Close' feature of data will be used and take mean according to week and plot them in Fig. 8, b) Analyze average closing price by day: From this, find the average closing price per day and plot in Fig. 9, similarly analyze the trend according to the quarter and plot it as in Fig. 10.

2. Analyzing the trend of closing price in Weekdays & weekends: for this create a function in which from weekofdays function if day<5, then it will be considered it as weekday else weekend as in Fig. 11 and plot in Fig. 12 and from this it can be seen a minor difference in two graphs.

5: Now build Baseline Model or Naive Model and using this model prediction will be performed:

The naive method takes into account what happened in the previous period and predicts that the same thing will happen again. The sole basis for Naive forecasting models is historical observation. They don't try to explain the underlying causal links that result in the forecasted variable.This model is all about "the previous value is the best reflector of the next Value" or we can say that next value completely depends on previous value.

A simple example of a naive type:

1. Use the actual sales of the current period as the forecast for the next period. Let us the symbol Y'_{t+1} as the forecast value and the symbol Y_t as the actual value. Then

$$Y'_{t+1} = Y_t$$

2. If you consider trends, then

$$Y'_{t+1} = Y_t + (Y_t - Y_{t-1})$$

This model adds the latest observed absolute period -to-period change to the most recent observed level of the variable.

BaseLine model or Naive Model: This model is all about "the previous value is the best reflector of the next Value" or it can be stated that next value completely depends on previous value. This is the basic summary behind this baseline model. For this it will use inbuilt function of pandas' library using Shift operations on data and store this data in 'prediction_naive' in Fig. 13.

Now plot the different between my actual value and my predictive trend value in Fig. 14.

From the plot Fig. 14 this can be seen that the prediction not much differ from actual values, so it almost overlaps the actual value graph.

Now to check how exact model is performing, use sklearn to find the mean_square_error (ignoring NaN values). An error rate of 37.23363264835875 will be got which means that there is a difference of approx. $+-37.2$ in prediction and actual values. This is a good model but still can't stay with this model as in real world use cases because there might have seasonality, data might not be smooth, that previous data is not true reflector of future data. So it can't considered as baseline model, even it have a good performance.

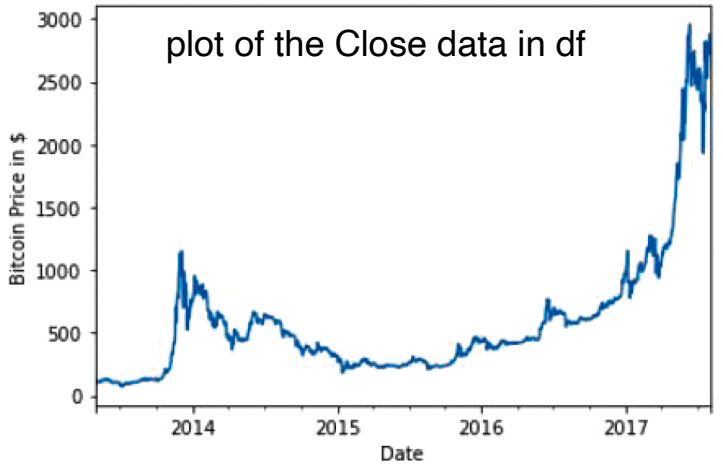

Fig. 4. Close Feature Plot.

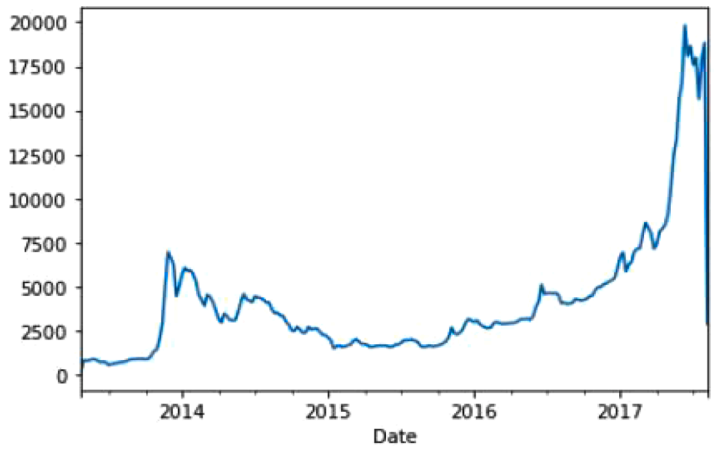

Fig. 5. Plot resample data sum on weekly basis.

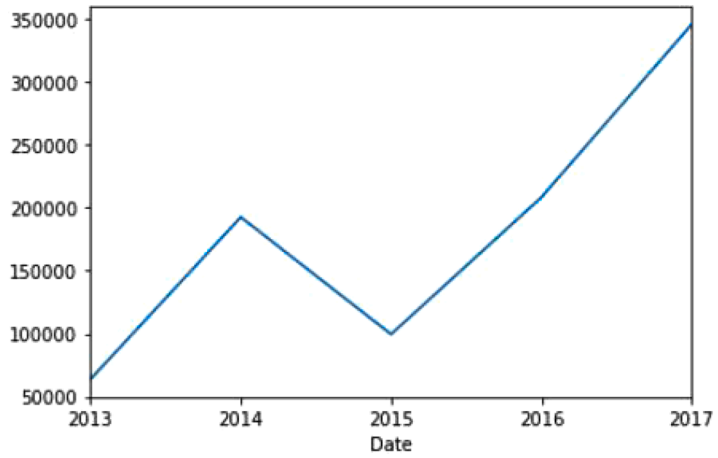

Fig. 6. Resample of Data on Yearly Basis.

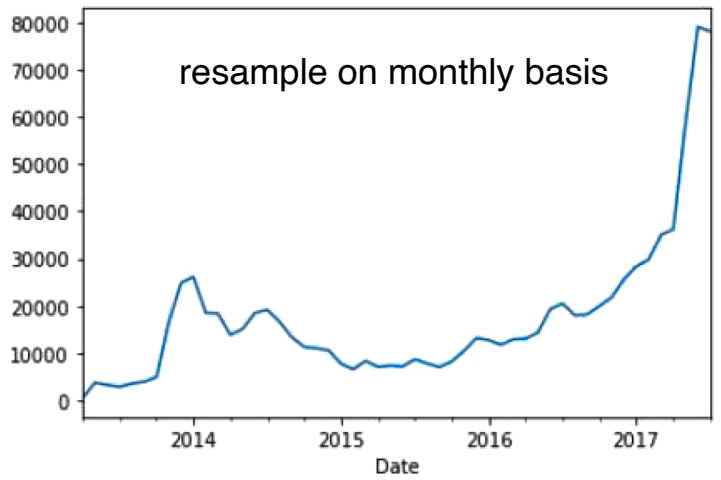

Fig. 7. Mean Closing Price in Different Year.

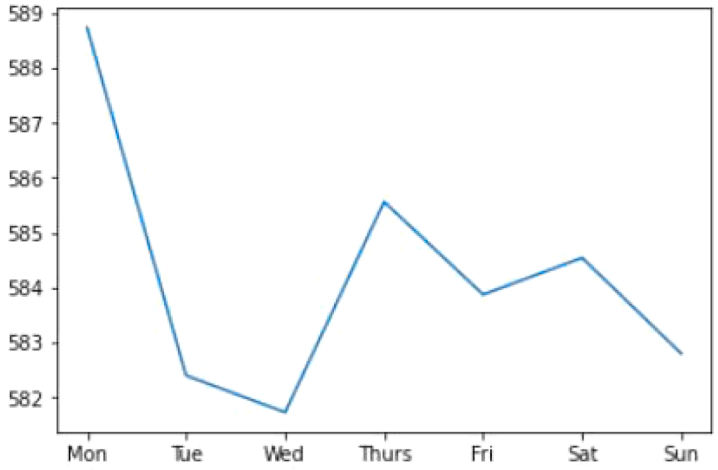

Fig. 8. Mean of 'Close' by ~~week~~. day of the week

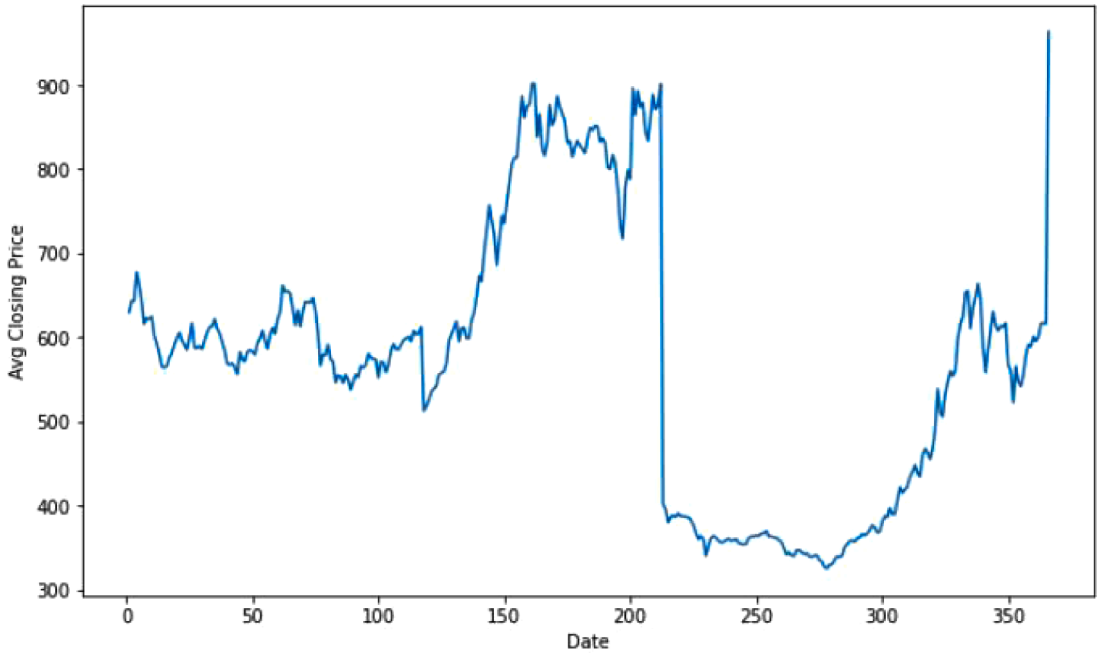

Fig. 9. Average Closing Price Each Day. of the year

6: Examine if there exists seasonality in the data or not : When the mean and the variance of the data is constant that is basically stationary nature of data and for a time series model data must be stationary and if it is not stationary then make it stationary so that can apply other advance algorithms like SARIMA which can give a better prediction result. So apply Rolling(moving Average) on the data at window period of 7 and compute mean and standard deviation to eliminate the seasonality of curve in Fig. 4 and plot the new curve as in Fig. 15.

In above graph the orange one is exactly the given series and blue which get overlapped here on the green curve is mean values. From this, it has been computed that rolling mean is not stationary and is varying with time. So now must remove its seasonality also and make it stationary. But firstly, lets prove that there is some seasonality in data for which use some statistical approach which is Adfuller Test.

The Dickey-Fuller test is a unit root test that examines the null hypothesis that α=1 in the following model equation. alpha is the coefficient of the first lag on Y. Null Hypothesis (H0): alpha=1

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta Y_{t-1} + e_t$$

Fig. 10. Average Closing Price Each Day.

Fig. 11. Weekdays and Weekends.

where, y(t-1) = lag 1 of time series delta Y(t-1) = first difference of the series at time (t-1)

Fundamentally, it has a similar null hypothesis as the unit root test.

That is, the coefficient of Y(t-1) is 1, implying the presence of a unit root.

If not rejected, the series is taken to be non-stationary.

The Augmented Dickey-Fuller test evolved based on the above equation and is one of the most common form of Unit Root test.

Adfuller test is mostly depends on one parameter, p-value. There are two types of Hypotheses:

1 Null Hypothesis is represented by H0

2 Alternate Hypothesis which denoted by H1

There will be 2 case either the p-value is greater than 0.05 (Strong evidence against null hypothesis or can reject null hypothesis and data is stationary) or less than 0.05 (weak evidence against null hypothesis, hence data is not stationary).

Lets consider an simple example: We have data which have mean as :$\mu = 120$ (H0 Null hypothesis),ie alternative hypothesis Hα: $\mu > 120$ or $\mu<120$,and assuming that α(level of sig.) = 0.05. The sample values that u took are as n(sample_size) =40, $\sigma = 32.17$ and $\bar{x}$ (sampling_mean) = 105.37.What is the conclusion for this hypothesis,ie what about P-value?

We know that,

$\bar{\sigma}_x = \sigma\sqrt{n}$

Now substitute the given values

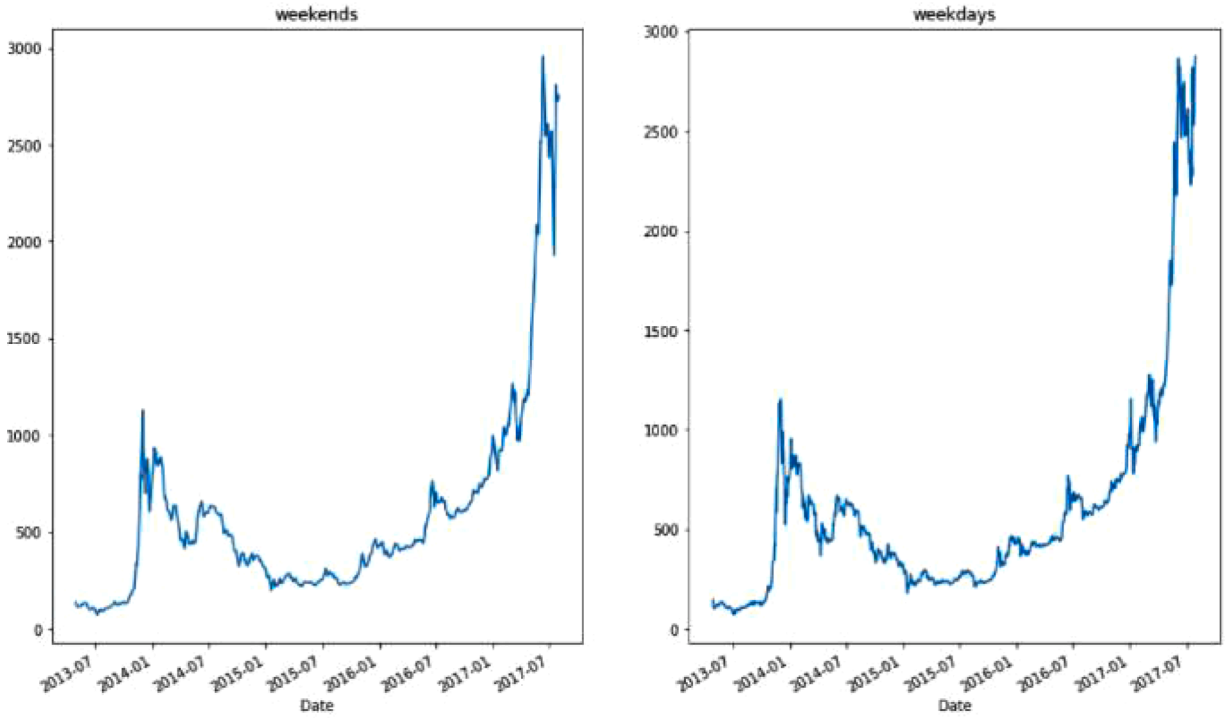

Fig. 12. Weekdays and Weekends Plot.

Fig. 13. Naïve Prediction.

$\sigma\bar{x} = 32.17\sqrt{40}$=5.08
As per the test static formula, we get
t = (105.37 – 120) / 5.0865 = −2.8762
Now we have to Use Z-Score table, finding the value of P($t > -2.8762$) we get,
P ($t < -2.8762$) = P($t > 2.8762$) = 0.003
Therefore,
If P($t > -2.8762$) =1 - 0.003 = 0.997
P- value =0.997 which is > 0.05
As value of $p > 0.05$, we are failed to reject Null hypothesis
Therefore, the null hypothesis is accepted, ie. The result is not statistically significant
The Null & Alternate hypothesis regarding this use-case is
Null Hypo->> closing price is stationary in nature
Alt. Hypo->> closing price is not stationary in nature

There will be two cases 1. P-value > 0.05 2. P-value < 0.05. First case signifies null hypothesis can be rejected as there is convincing evidence against it. Second case signifies the acceptance of hypothesis as there is unconvincing evidence in its against hence data is not stationary. From statsmodel library directly import Adfuller and use it in program on data of 'close'. From this, p-value=0.0002154535155876224 which is less than 0.05 so null hypothesis rejected, hence data is stationary. There is still some

Fig. 14. Naïve Prediction Vs Actual Value Plot.

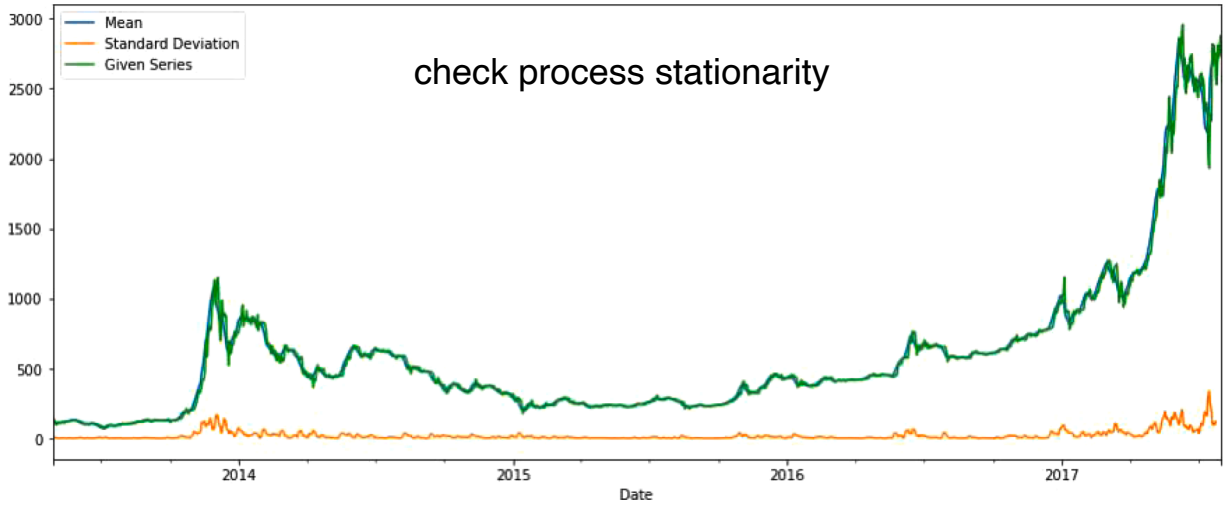

Fig. 15. Mean and Standard Deviation Plot.

seasonality and outliers in the data, which may be reduced by applying Log Transformation on the data. ~~The Log Transformation is used to remove outliers from data that are either very high or very low, There is still some seasonality in data now try to eliminate this by using Log Transformation on data. Log Transformation is used for removing Either extremely High or exceptionally Low outliers from the data~~, use log function which is available in NumPy for this and plot it as in Fig. 16 after removing seasonality from data.

~~Now, Smoothening and moving average(M.A) have to be performed on data. In Smoothening, rather than just getting an average and using it as the next forecast, it decreasingly weights exponents depending on outside factors, like season or age of a product. This is done to 'smooth' the averages and create a reliable forecast. Smoothening is done by using moving average. Moving Average is universally used in financial market. Rolling is a window that have considered for moving average (M. A)~~. Now plot this Rolling Transformation as Fig. 17.

Now can compute the difference between log_data and rolling_average and checks its stationarity also as in Fig. 18.

From Above Graph there are convincing results and proof against null hypothesis, so null hypothesis will get rejected hence data is Stationary. From the series, time series is approx. Stationary with constant interval. Now apply differencing using shift and plot it as in Fig. 19 below.

With Respect to seasonality there is the trend as shown in Fig. 20. Now if wants to check its stationarity and want to check whether have made some Seasonality or not, similarly check it for this and then plot it as in Fig. 20.

From this it can be concluded that dicky fuller Test is very much less than 1% critical Value.

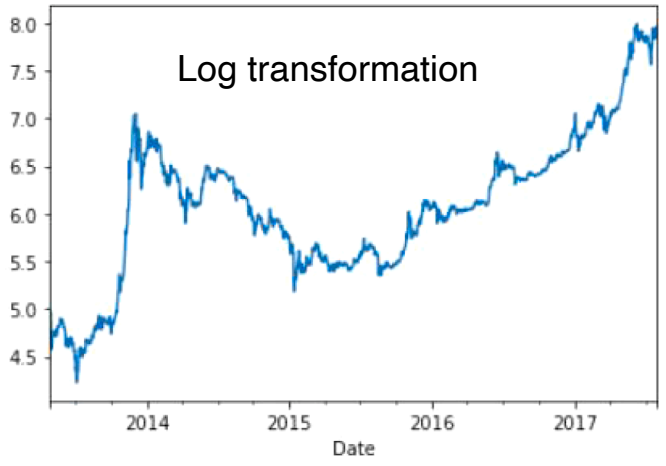

Fig. 16. After Removing Seasonality Factor.

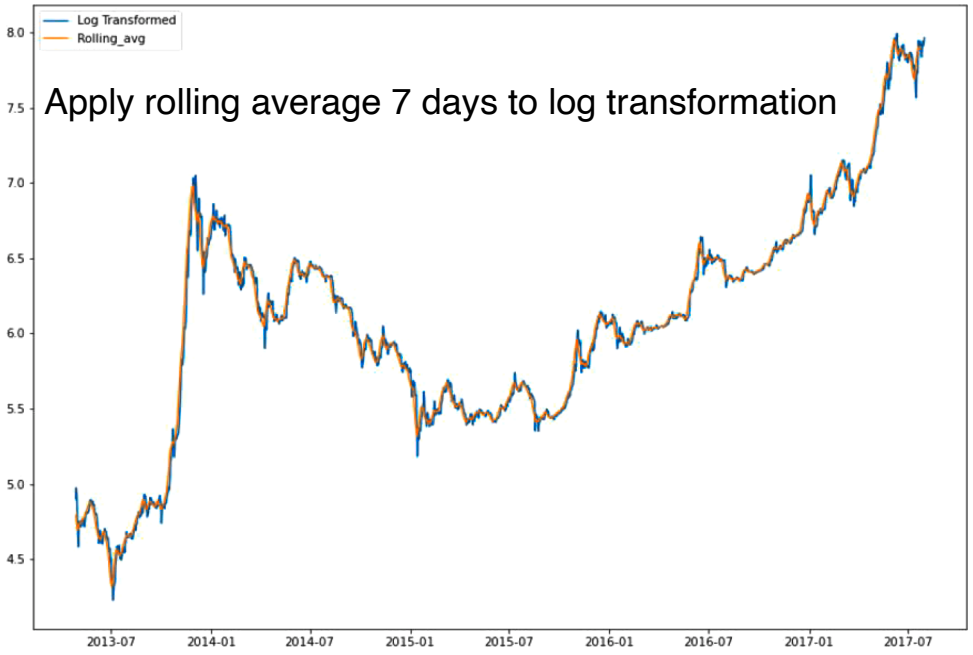

Fig. 17. Log Transformation and Moving Average.

Now, prepared data for forecasting using FbProphet

B. Forecast using prophet model and plot forecasting

"Facebook Prophet" is a Facebook-developed open-source toolkit for univariate time series forecasting that use a Bayesian-based curve fitting approach to forecast time series data. Prophet is unique in that it requires no prior knowledge or expertise in forecasting time series data since it automatically detects seasonal trends underlying the data and provides a set of 'simple to comprehend' parameters. Prophet is also built to deal with holidays that are known ahead of time, missing data, and significant outliers. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

The package employs an easily interpreted, three component additive model whose Bayesian posterior is sampled using *STAN*. In contrast to some other approaches, the user of Prophet might hope for good performance without tweaking a lot of parameters. Instead, hyper-parameters control how likely those parameters are a priori, and the Bayesian sampling tries to sort things out when data arrives. Prophet's default settings produce forecasts that are often [as] accurate as those produced by skilled forecasters, with much less effort.

Prophet is a model that is exactly available in FbProphet library. It manages irregular intervals or irregular holidays. If there are some noises in data or are some outliers in data, then that scenario also gets managed by this Fb prophet module. Adjusting with non-linear patterns may be historical or seasonal or weekly, a time series model that is prophet applies additive model. It has capability to

```
ADf Test Statistics : -7.188887202324554
P-value : 2.5340955586100355e-10
lags used : 22
No. of obs used : 1527
Strong Evidence Against Null Hypothesis & Reject this Null Hypo. and Data is Stationary
```

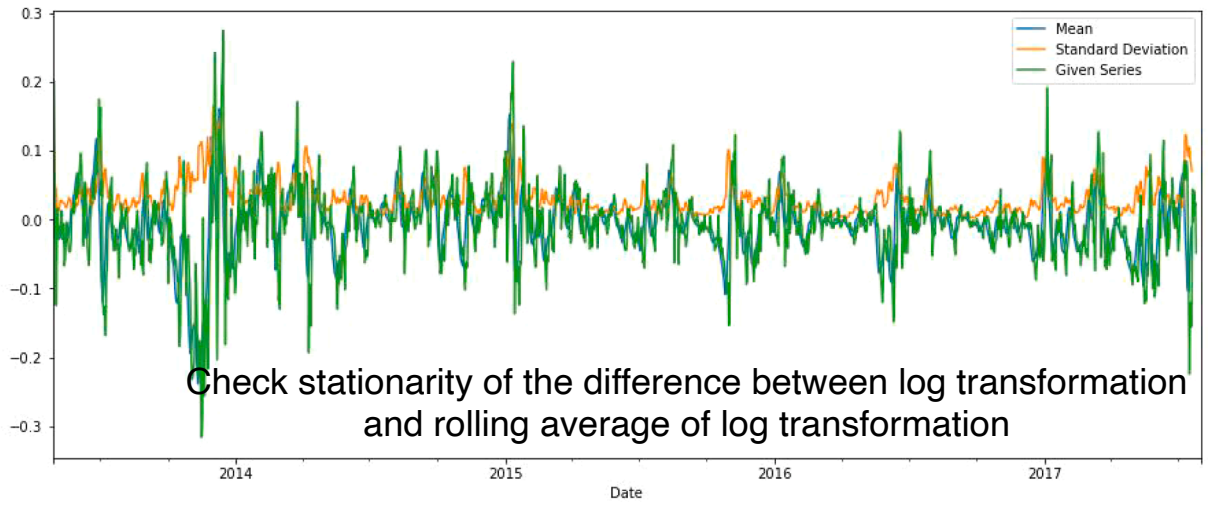

Fig. 18. Rolling And Moving Average Difference is Stationary.

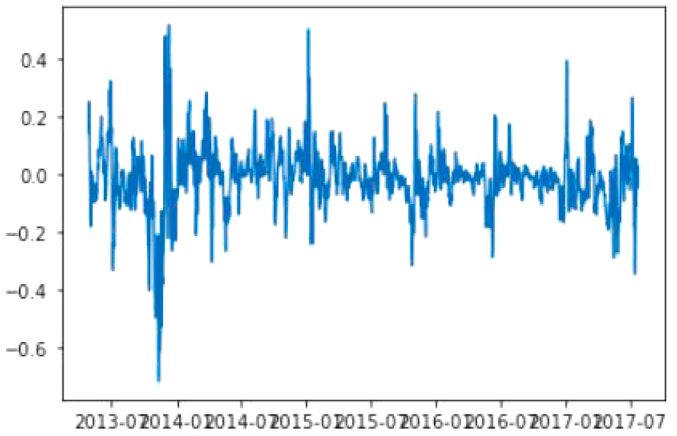

Fig. 19. Differencing.

~~deal with outliers and missing values.~~

Before fitting prophet model there is need to prepare data according to prophet model documentation. It must make sure that data will follows all its protocols, and its protocols are as follows. Make date as 'ds' and the output feature as 'y'

Now fit the model for a period of 500 Days with frequency of Day. The forecast value will look like as in Fig. 21.

Where yhat is actual prediction and yhat_upper is prediction for upper bound and yhat_lower is prediction for lower bound. Now to plot this forecast for which use inbuild feature of Fbprophet library which is forecast and plot it using plot function as in Fig. 22.

In the graph the black dot are actual data, blue line is prediction curve and light blue line is trend. Similarly, can plot for weekly basis, yearly basis, monthly basis, etc. as in Fig. 24.

Now to cross validate forecast model for which calculate forecast error. To compute forecast error will be compared to actual values to predicted values. In Fbprophet there is an in build cross validation techniques. Inside this cross-validation technique. There are some parameters which are Horizon parameter, size of Initial training period, spacing between cutoff period and plot it as in Fig. 23. In Fig. 23 plot for root-mean-square error, similarly, can plot for mean-square error and further check for errors.

```
ADf Test Statistics : -6.511722596316726
P-value : 1.0961860829579836e-08
lags used : 23
No. of obs used : 1525
Strong Evidence Against Null Hypothesis & Reject this Null Hypo. and Data is Stationary
```

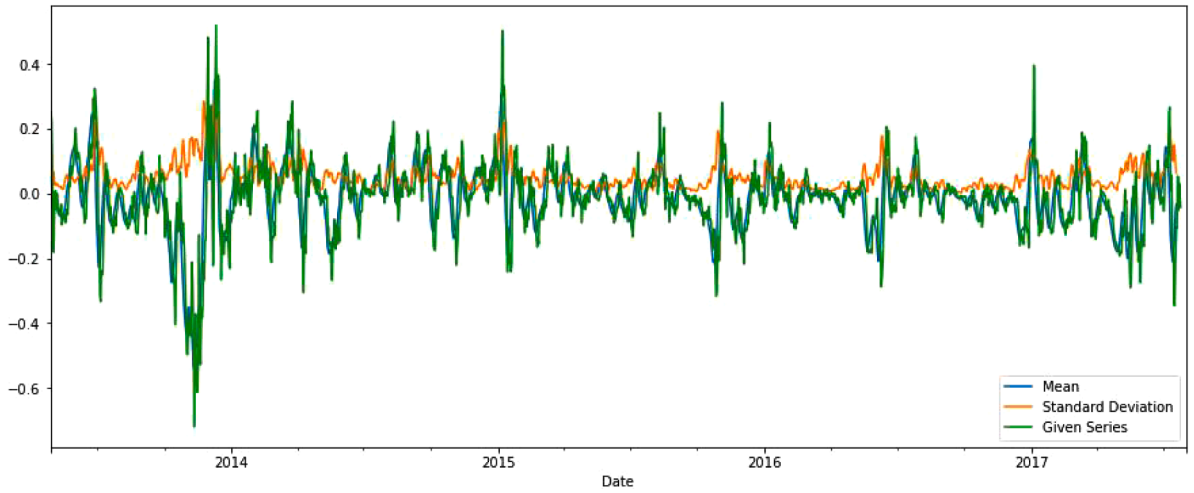

Fig. 20. Critical value of Dickey fuller test is less than (approx.) 1%.

	ds	yhat	yhat_lower	yhat_upper
2046	2018-12-04	5587.349564	3155.666046	8010.214200
2047	2018-12-05	5586.897199	3145.849320	8052.950010
2048	2018-12-06	5590.384867	3085.462484	8043.696604
2049	2018-12-07	5587.862758	3103.062760	8078.466456
2050	2018-12-08	5587.249932	3133.705219	8141.700373
2051	2018-12-09	5586.850155	3069.009790	8076.865174
2052	2018-12-10	5590.897150	3082.779889	8047.689207
2053	2018-12-11	5591.548155	3089.374667	8102.190623
2054	2018-12-12	5588.748898	3069.402777	8070.284320
2055	2018-12-13	5590.484927	3030.567550	8169.309468

Fig. 21. Forecast Values.

5. Results and analysis

This project is created on Jupyter Notebook (anaconda 3) using python. The data for this study came from an open-access website - https://www.kaggle.com/team-ai/bitcoin-price-prediction/version/1. Used data is real-world or actual historical data to form Naïve model and to search for historical trends. Data can be use at real time using Big data concepts like Spark Streaming and Kafka. Using these big data tools we can collect data at real time and analyze them and apply this model at real time. Data is made up of a single.csv file which consists of date, open, high,low, volume and market cap. Of Bitcoin. This .csv file contains record of 1556 days from 28th April 2013 to 31st July 2017. For real time collection of Data, we can collect data from https://in.tradingview.com/chart/?symbol=COINBASE%3ABTCUSD from where data can easily exported to big data tools and it will be easy to apply this model at real time. In this section the output of Naïve prediction is shown in Table 4 and then plot this Naïve model prediction in Fig. 25. As Naïve model was based on historical data, and it is not possible to predict the future on bases of historical data in real world. So created a new model with help of Fbprophet library, forecasted it in Table 5 and plot it in Fig. 26. After prediction cross validated model and

Fig. 22. Forecast using FbProphet.

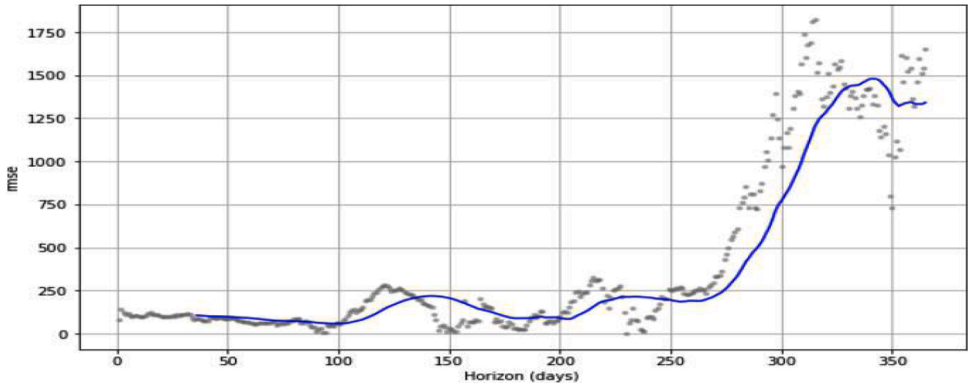

Fig. 23. Root_Mean_Square Error.

find the root_mean_square error in model and plot it in Fig. 27. Fig. 27 shows the difference between actual and predicted valued for datasets. In Fig. 26 the black dots show actual data, blue line shows the predicted curve and blue shaded region shows the trend of Bitcoin.

As Naïve model was based on historical data and it is not possible to predict the future on bases of historical data in real world. The historical data might have seasonality, So we checked for seasonality using AdFuller and remove seasonality. Fig. 27 shows seasonality present in data. Fig. 28 shows the naïve model after removing seasonality. After removing seasonality, there might be some outliers so to eliminate those outliers log transformation is applied on data and there is a need to make data smooth using Rolling transformation, Fig. 29 shows data after applying log transformation and rolling. Now we further check for seasonality in our processed data. From Fig. 28 we can say our data is Stationary. Now finally, we can create a time series model using Facebook Prophet. Forecasted values are in Table 4 and plot it in Fig. 26. After prediction there is a need of cross validate for errors, so we cross validate model and find the root_mean_square error in model and plot it in Fig. 27. Fig. 27 shows the difference between actual and predicted valued for datasets. In Fig. 26 the black dots shows actual data, blue line shows the predicted curve and blue shaded region shows the trend of Bitcoin.

6. Discussion

This study's proposed model for cryptocurrency prediction can be recognized a consistent and suitable model. Previous models such as BTC rates 43 percent. Whereas model based on LTC Multi-linear regression as R2 rates percent of 43. Whereas its precision for Bitcoin is 57 percent. The study shows Bitcoin logistic regression rates 67 whereas LDA have 64 percent accuracy. Linear discriminant analysis as same like bitcoin regression 67 percent. Models occurred in existence before such as LTC Multi-linear regression model R2

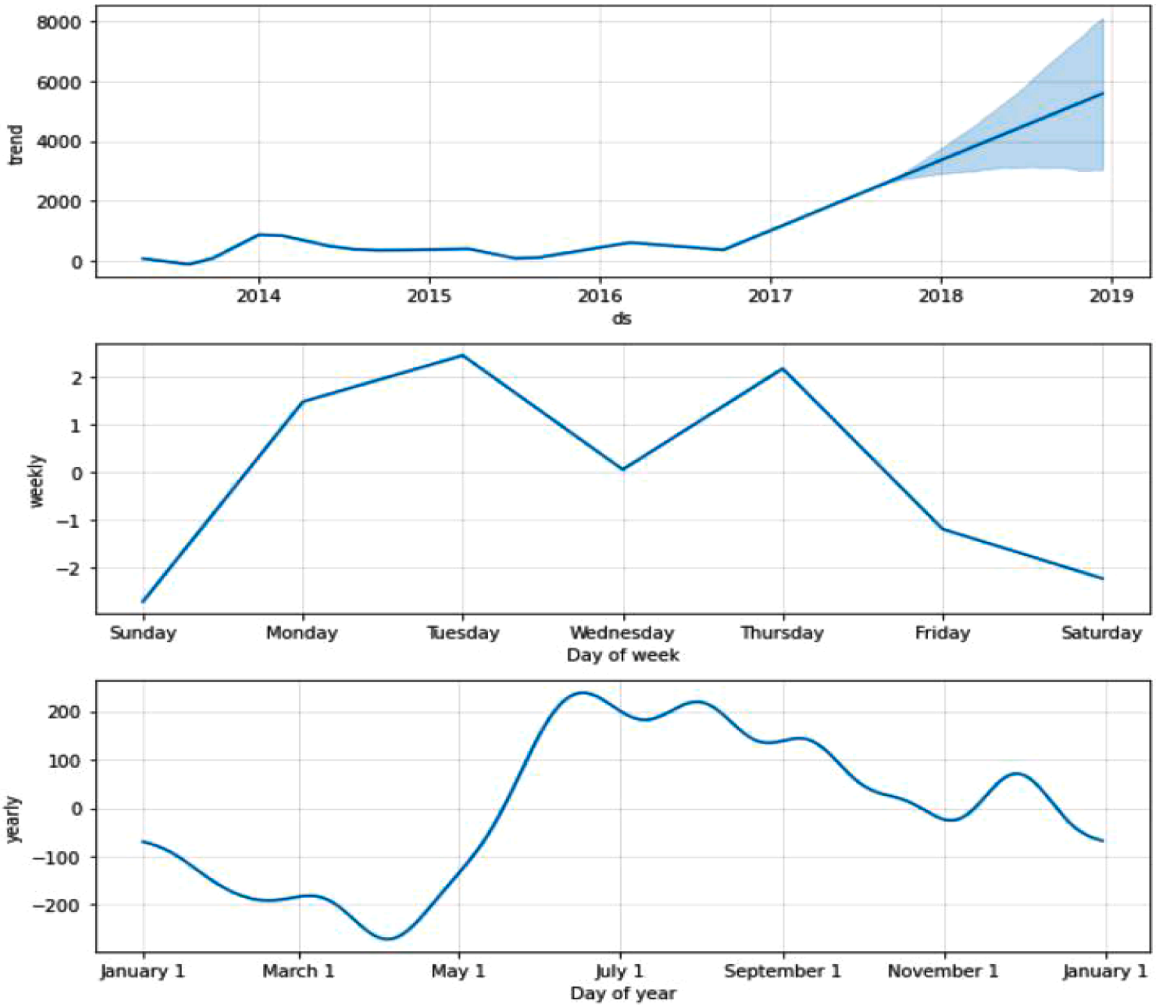

Fig. 24. Weekly, yearly and monthly plot.

Table 4
Naïve Prediction Table.

Date	Open	High	Low	Close	Volume	Market Cap	dayofweek	prediction_naive
2017–07–31	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000	weekdays	NaN
2017–07–30	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000	weekends	2875.34
2017–07–29	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000	weekends	2757.18
2017–07–28	2679.73	2897.45	2679.73	2809.01	1380,100,000	44,144,400,000	weekdays	2726.45
2017–07–27	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000	weekdays	2809.01

have 42 percent for LTC. But have accuracy of 56 percent for Bitcoin. Bitcoin Logistic regression and linear discriminant analysis rates LR have 65 percent. Methods are often based on previous data, and in real-world issues, future outcomes cannot be anticipated only on the basis of historical data. Seasonality in historical data may exist, or pattern accuracy in other models may be impaired. But in this model here created a function for removing seasonality and used advanced modules like Fbprophet which are one of the best modules for real time data and time series model and at last cross-validated the model in Fig. 27 and can clearly see accuracy of this model is far better than other models.

ARIMA is a strong model that, as we've seen, produced the greatest results for stock data. One difficulty is that it may need meticulous hyperparameter adjustment and a thorough comprehension of the data. When comparing ARIMA to LSTM and Fbprophet Models in the Stock Market, it can produce superior results. However, when it comes to cryptocurrencies, where there is severe seasonality, ARIMA can handle data with trend but does not support time series with trend and seasonality. Unlike ARIMA and Prophet, LSTM does not require on certain assumptions about the data, such as time series stationarity or the presence of a Date field. One problem is that LSTM-based RNNs are difficult to comprehend, and gaining intuition into their behavior might be difficult. In

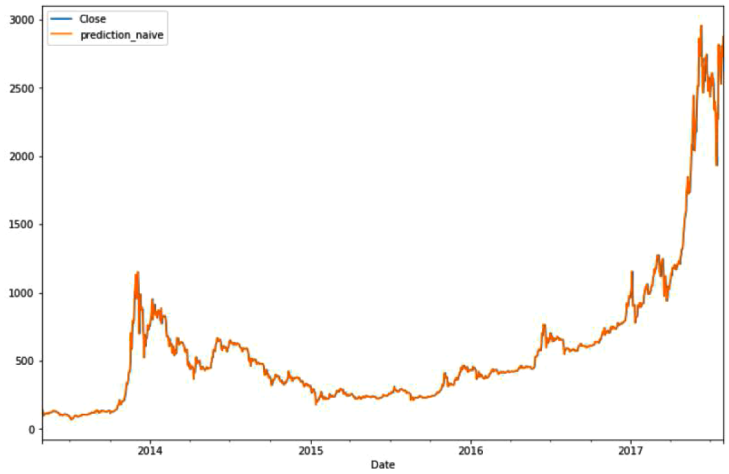

Fig. 25. Naïve Prediction Plot.

Table 5
FbProphet Forecast Table.

	ds	yhat	yhat_lower	yhat_upper
2046	2018–12–04	5587.349564	3155.666046	8010.214200
2047	2018–12–05	5586.897199	3145.849320	8052.950010
2048	2018–12–06	5590.384867	3085.462484	8043.696604
2049	2018–12–07	5587.862758	3103.062760	8078.466456
2050	2018–12–08	5587.249932	3133.705219	8141.700373
2051	2018–12–09	5586.850155	3069.009790	8076.865174
2052	2018–12–10	5590.897150	3082.779889	8047.689207
2053	2018–12–11	5591.548155	3089.374667	8102.190623
2054	2018–12–12	5588.748898	3069.402777	8070.284320
2055	2018–12–13	5590.484927	3030.567550	8169.309468

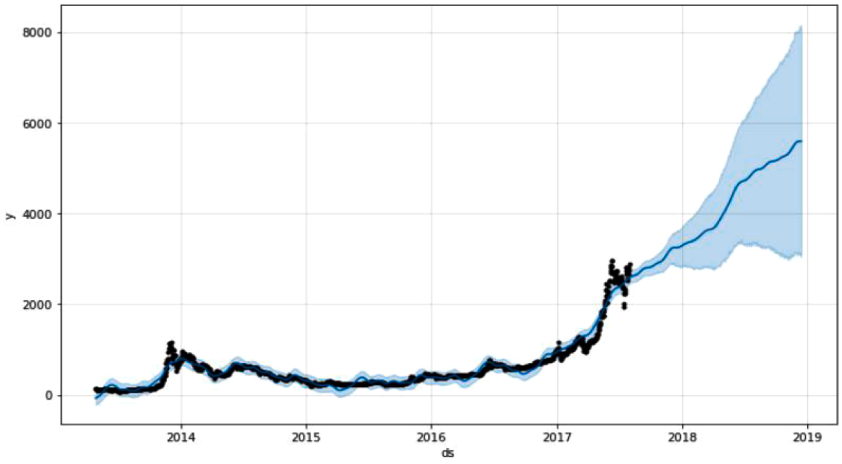

Fig. 26. Fbprophet prediction plot.

order to produce decent outcomes, rigorous hyperparameter adjustment is also essential. So there was a need in the case of bitcoin; a ~~model that could dynamically track changes in trends and data, manage severe seasonality, and handle a large number of outliers and changes in trends owing to market occurrences was required. Unlike some other techniques, the user of Prophet may expect decent results without adjusting a lot of settings. Instead, hyper-parameters determine how likely certain parameters are a priori, and Bayesian sampling attempts to sort things out once the data is collected. It works best with time series with substantial seasonal influences and historical data from several seasons.~~ However, when there is a lot of seasonality and seasonality patterns in a dataset, and these patterns aren't consistent or smooth, and trends change quickly, it can hurt the model's performance. To address this and other

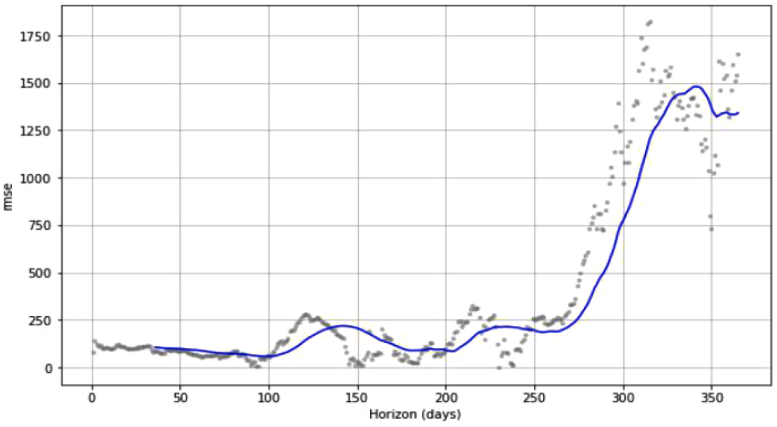

Fig. 27. Cross validation using Root_Mean_Square Error.

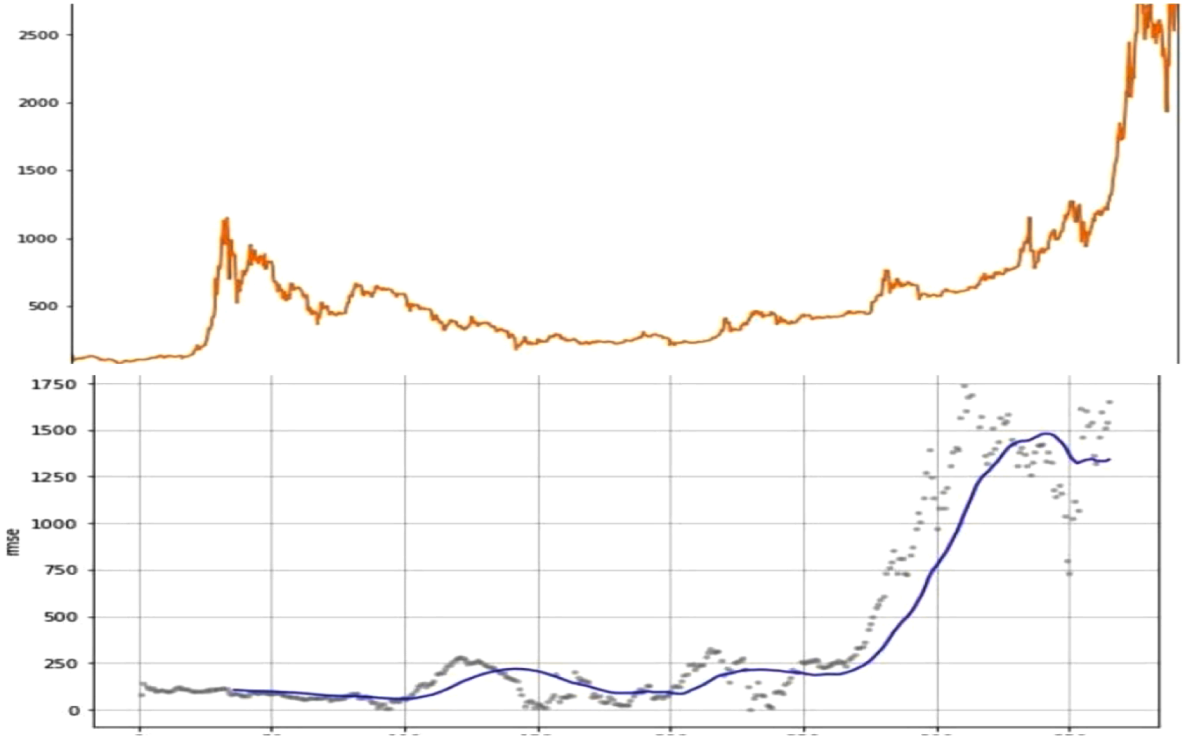

Fig. 28. Naïve and Fb Prophet model curve comparision.

limitations of Fbprophet, we used quantitative forecasting on the data to get stationary data that is free of outliers, consistent, and smooth. Now, data may be used to use the Prophet time series model to provide a reliable forecast.

7. Conclusion

Cryptocurrencies are volatile, trends are dynamic, data is neither consistent nor smooth, there is a lot of seasonality in data, and trends in cryptocurrencies can't be totally based on previous data since they might change dynamically. Majorly two machine learning algorithms are proposed and implemented for forecasting Bitcoin values in this article. The accuracy of several models was evaluated using performance metrics, as illustrated in Fig. 27. The methodology included in the paper majorly focuses on fbprophet model applied for cryptocurrency. The outcome of the methodology in the paper shows that in the case of crypto currencies, time series patterns are complex and vary dynamically over time, but Prophet only tracks such changes when the trend shifts. Prophet is forgiving of missing data and trend changes, and it usually handles outliers well. Prophet is built to deal with holidays that are known ahead of time, missing observations, and huge outliers. The key aspect of the outcome is that, because cryptocurrency patterns are not consistent

Fig. 29. Data After Log transformation.

and fluctuate over time, Facebook's Prophet is the best solution in this situation. ~~Prophet is unique in that it does not require much prior knowledge or expertise with time series data forecasting since it automatically detects seasonal trends underlying the data and provides a set of' simple to understand' parameters, therefore utilized Fbprophet to predict real-world outcomes for cryptocurrencies after eliminating the seasonality effect and smoothening of data.~~

The real and forecasted prices were then compared. The precision of the model, as shown in Fig. 27, is significantly superior to that of previous models. ~~Based on these findings, discovered that the Naive model is only useful with historical data, and while its accuracy is excellent, still cannot rely on past data completely for real-world scenarios, therefore utilized Fbprophet to predict real-world outcomes after eliminating the seasonality effect.~~

The results of the conducted experiments signify that according to the trial findings: 1. The AI algorithm is accurate and suitable for bitcoin prediction 2. Fbprophet Model is better than Naïve model for real world test cases.

Fbprophet Model can give better result when data is consistence, stationary and smooth therefore there is need for applying Quantitative forecasting on data before Fbprophet.

~~In future research, it can be looked into other factors that could influence cryptocurrency prices. Model can be further evolved by applying Qualitative forecasting on data. Unlike Quantitative methods, Qualitative forecasting uses data that can't be measured. It relies on opinions and expert advice and is useful for new companies that don't have any or much historical data. Qualitative methods are useful, but it's important to take the information into account in a nonjudgmental and unbiased manner. In future research, it can be investigated other factors that could influence cryptocurrency prices, with a particular focus on the impact that specifically on tweets by applying the concept of natural language processing and sentiment analysis. There are only few API's are available because storage and analysis the trading sites have to provide the real time data and generally companies do not provide and this data need to be processed in real time. For real time analysis, we can use Big data technologies like Kafka and Spark Streaming~~

~~CRediT authorship contribution statement~~

~~**Rajat Kumar Rathore:** Conceptualization, Methodology, Writing – original draft. **Deepti Mishra:** Conceptualization, Methodology, Writing – original draft. **Pawan Singh Mehra:** Conceptualization, Writing – review & editing. **Om Pal:** Conceptualization, Supervision, Project administration. **AHMAD SOBRI HASHIM:** Supervision, Validation. **Azrulhizam Shapi'i:** Data curation, Formal analysis. **T. Ciano:** Supervision, Formal analysis. **Meshal Shutaywi:** Visualization.~~

References

Arias-Oliva, M., Pelegrín-Borondo, J., & Matías-Clavero, G. (2019). Variables influencing cryptocurrency use: A technology acceptance model in Spain. *Frontiers In Psychology*.

Chen, Z., Li, C., & Sun, W. (2021). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*.

Ebadi, M. J., Hosseini, A., & Hosseini, M. M. (2017). A projection type steepest descent neural network for solving a class of nonsmooth optimization problems. *Neurocomputing, 235*, 164–181.

Ezazipour, S., & Golbabai, A. (2020). A globally convergent neurodynamics optimization model for mathematical programming with equilibrium constraints. *Kybernetika, 56*(3), 383–409.

Farhath, Z. A., Arputhamary, B., & Arockiam, L. (2016). A survey on arima forecasting using time series model. *International Journal of Computer Science and Mobile Computing, 5*(8), 104–109.

Fauzi, M. A., & Paiman, N. (2020). Bitcoin and cryptocurrency: challenges, opportunities and future works. *The Journal of Asian Finance, Economics and Business, 7*(8).

Giudici, G., Milne, A., & Vinogradov, D. (2020). Cryptocurrencies: Market analysis and perspectives. *Journal of Industrial and Business Economics*, 1–18.

Golbabai, A., & Ezazipour, S. (2019). A projection-based recurrent neural network and its application in solving convex quadratic bilevel optimization problems. *Neural Computing and Applications*, 1–14.

Hitam, N.A., & Ismail, A.R. (2018). "Comparative performance of machine learning algorithms for cryptocurrency forecasting," 11(3), pp. 1121–1128.

Jamali, N., Sadegheih, A., Lotfi, M. M., Wood, L. C., & Ebadi, M. J. (2021). Estimating the depth of anesthesia during the induction by a novel adaptive neuro-fuzzy inference system: A case study. *Neural Processing Letters, 53*, 131–175.

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science, 7*, 45–66.

Khandelwal, I., Adhikari, R., & Verma, G. (2019). Time Series Forecasting Using Hybrid ARIMA and ANN Models Based on DWT Decomposition. *Procedia Computer Science, 48*, 173–179.

Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*.

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal, 3*(2), 334–340.

Tirandazi, P., Rahiminasab, A., & Ebadi, M. J. (2022). An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*.

Wang, L., Zou, H., Su, J., Li, L., & Chaudhry, S. (2013). An ARIMA-ANN hybrid model for time series forecasting. *Systems Research and Behavioral Science, 30*(3).

Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. *Neurocomputing, 50*(17).