

**Predictive Analytics**

***Reddit Sentiment Analysis for***

***Cryptocurrency Price Prediction***

***Team 8***

NAME CENSORED

*Raditya Fahritama*

NAME CENSORED

**School of Graduate Professional Studies**

Data Analytics

IE-575 – Predictive Analytics

Spring II, 2022

**Document Control**

**Work carried out by**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email Address** | **Task description** |
| NAME CENSORED |  | Data Collection, Data Preprocessing, EDA of Reddit comments and Bitcoin price Data, VADER Implementation, Exploring and experimenting time series models that can be implemented to forecast the prices of BTC and validation metrics, Implementation of SARIMAX and XGBoost(experimented but did not provide results as results have no significance), Documentation |
| Raditya Fahritama | rkf5230@psu.edu | Data Collection, Data Preprocessing, EDA, Random Forest Implementation, Documentation |
| NAME CENSORED |  | Data Collection, Data Preprocessing, Sentiment Analysis using Textblob, Exploring and experimenting time series models that can be implemented to forecast the prices of BTC and validation metrics, Documentation |

**Revision Sheet**

|  |  |
| --- | --- |
| **Date** | **Revision Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**TABLE OF CONTENTS**

[**1**](#_heading=h.2et92p0) **Introduction 3**

[**2**](#_heading=h.tyjcwt) **Problem Statement 3**

[2.1](#_heading=h.3dy6vkm) Challenges 3

[2.2](#_heading=h.1t3h5sf) Related Works 4

[**3**](#_heading=h.2s8eyo1) **Data Collection** 4

[**4**](#_heading=h.17dp8vu) **Data Preprocessing** 7

[**5**](#_heading=h.3rdcrjn) **Methodology** 12

[**6**](#_heading=h.26in1rg) **Results** 13

[**7**](#_heading=h.lnxbz9) **Discussion of Results** 30

[**8**](#_heading=h.35nkun2) **References** 31

# Introduction

A cryptocurrency is a digital asset that employs cryptography to encrypt transactions, control the generation of new cryptocurrencies, and verify the secure transfer of assets. Cryptocurrencies are digital currencies that differ from traditional currencies in that they are based on the notion of decentralized control, as opposed to traditional currencies' reliance on central banking institutions. The first cryptocurrency was created in 2008 when an unknown individual using the pseudonym Satoshi Nakamoto published a paper titled Bitcoin: A Peer-to-Peer Electronic Cash System in the public domain. In January 2009, Nakamoto released the bitcoin program as open-source code on SourceForge. The contributions of Satoshi Nakamoto sparked a wave of public interest, prompting others to build rival cryptocurrencies based on the same underlying technology but with different purposes.

There is a massive volume of unstructured data produced in the form of tweets, Reddit posts, internet articles, text messages, emails, and other formats. As a result, "natural language processing" (NLP) has emerged as a field of research or development. Web data beyond Twitter and social media has been an affluent area of research. Our project aims to integrate Reddit posts and cryptocurrency prices to understand if price fluctuations can be studied using social media behavior.

# Problem Statement

The objective of our project is to perform sentiment analysis on Reddit posts and anticipate price fluctuations in cryptocurrency (Bitcoin) over the year 2019 through time series analysis using various machine learning methods. With this project, we are trying to determine if there is a relationship between posts on Reddit and the price fluctuation in Bitcoin.

## Challenges

* No validation for sentiment analysis algorithms: There are no predefined metrics to validate the polarity scores given by the algorithms. Hence, we are using two methods for sentiment analysis in order to roughly compare the sentiment of posts predicted by each. However, due to the absence of any metric, it is difficult to conclude if the predicted sentiment score is accurate.
* Neutral sentiment in the collected Reddit dataset: It needs to be assessed how many Reddit posts have any sentiment at all. When the majority of the Reddit posts aren't objective, sentiment analysis of them has little value for the model. Furthermore, it must be demonstrated that there is a correlation between bitcoin Reddit posts sentiment and changes in cryptocurrency prices. The model will be filled with noise if this is not done.

## Related Works

* **Abraham et al.** present a solution to predicting cryptocurrency price changes using sentiment analysis using data collected from Twitter. To accomplish this, methods utilizing sentiment analysis of tweets were reviewed. With the data collected, cleaned, and adjusted where needed, the data was analysed to determine if it would be a valuable input to the final model. They were then analysed to create a sentiment score by day using VADER and compared to the price changes to that day to determine if a relationship between Twitter sentiment and cryptocurrency price changes could be determined. They also tried to establish a relationship between Google Trends and crypto currency price change.
* **Li et al.** have performed prediction analysis for ZClassic (ZCL), a private, decentralized, fast, open-source community driven virtual currency, which lends itself to a high level of predictability via tweet analysis. After retrieving data using **rtweet** package, an algorithm was created to classify each tweet as positive, negative, or neutral sentiment using natural language processing using **TextBlob**. They used Extreme Gradient Boosting Regression as it exhibited the smallest loss, or inaccuracy, and was thus chosen to train the model on the data.

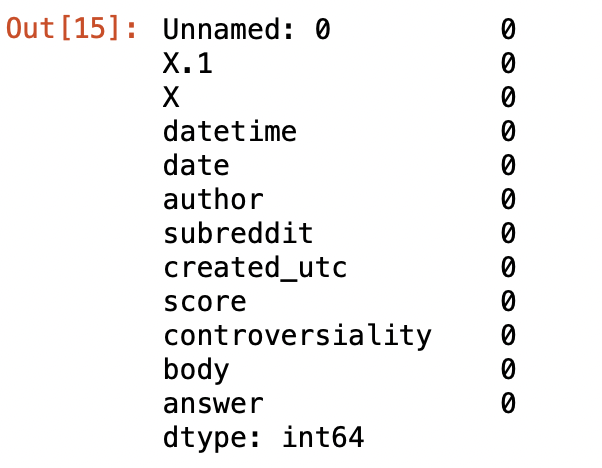
# Data Collection

* Data for sentiment analysis: Initially, the data provided to us wasn’t in the required timeframe. Hence, we tried scraping data from Reddit however retrieving dates for posts was a tricky part. On further exploration, the dataset was available on [Kaggle](https://www.kaggle.com/datasets/jerryfanelli/reddit-comments-containing-bitcoin-2009-to-2019). The dataset contains date-time, author, subreddit as well as the body of the posts



## 

The reddit posts data set consists of 10 columns and 493310 rows that contain the subject of ‘Bitcoin’ or ‘Crypto Currency’. The data has no missing values in the dataset i.e. posts corresponding to Crypto and bitcoin are present throughout 365 days (we considered data from 2019-01-01 to 2019-12-31)

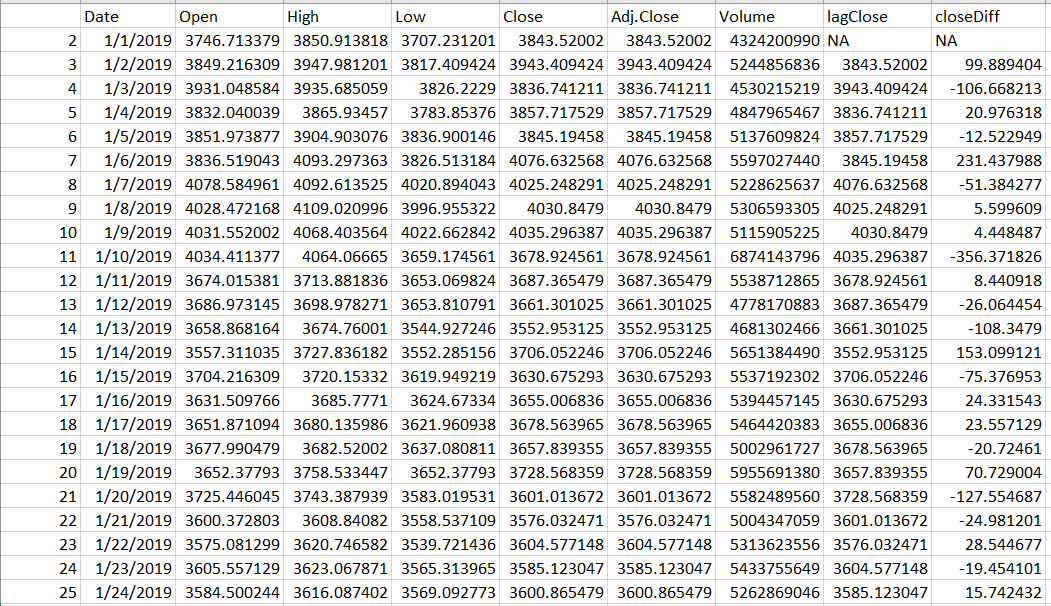
****

From this dataset we mainly use the ‘body’ column to get a corresponding Sentiment Score for each comment. The other columns like score, controversiality can be used as input features for machine learning models while predicting the prices.

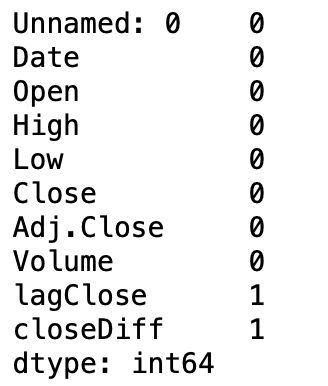
Data Dictionary:

1. datetime: Date and timestamp when the comment has been made
2. date: day when comment is posted/ the date of prices of BTC
3. author: reddit username/ author of the comment
4. subreddit: subreddit where the comment has been posted
5. created\_utc: Unix timestamp of when the comment has been created
6. score: Upvotes or Downvotes. Score is positive if there is an upvote and negative when there is downvote
7. controversiality: If there is controversiality for the comment posted
8. body: text of the comment posted

* Data for Price Prediction:



The bitcoin price data is also available for a time period of 1 year (2019-01-01 to 2019-12-31) with no missing values. The data set has 10 columns and 365 rows with 2 missing values (the missing values are present in the columns that are calculated using lag)



We use the Close Price as our target variable. Volume and closing difference will also be considered as input variables for machine learning models.

Data Dictionary:

1. Date: Date of BTC
2. Open: Opening value of BTC on the given date
3. High: Highest value of BTC on the given date
4. Low: Lowest value of BTC on the given date
5. Close: Closing value of BTC on the given date
6. Adj. Close: Adjusted close is the closing price after adjustments for all applicable splits and dividend distributions
7. Volume: Volume of transactions in USD on a given day
8. lagClose: Difference between current value and previous value (lag of 1 day) of BTC
9. closeDiff: Difference between opening amount and closing amount of BTC

# Data Preprocessing

* Data Preprocessing (Sentiment data):

## Removal of punctuations and Stopwords: Stop words and punctuations provide minimal semantic information. Removing these words and punctuation makes it easier to isolate and find the words required to differentiate the topics.

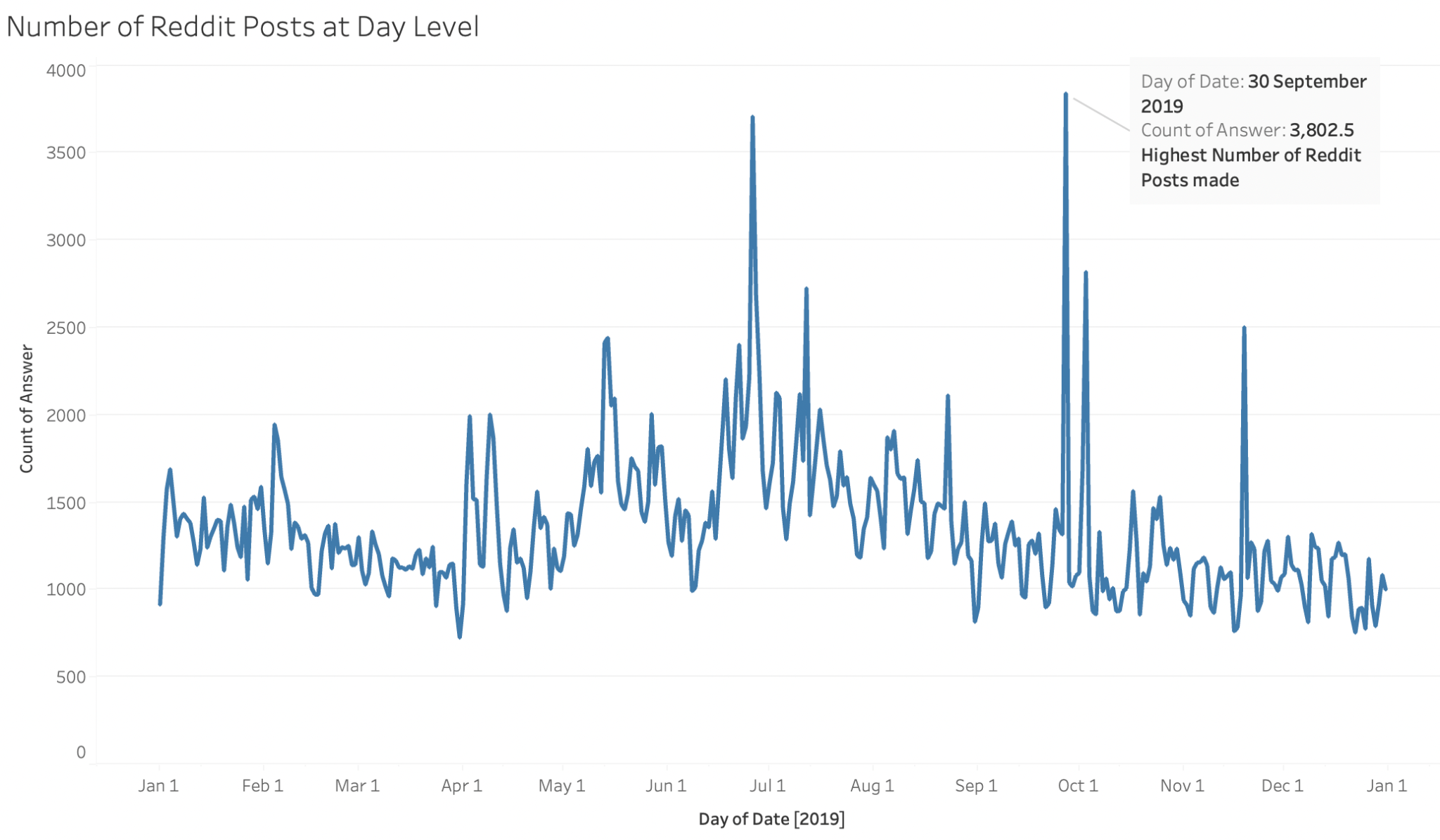
## Lemmatization and Tokenization: Lemmatization is the task of grouping up different forms of a word so that they can be computed as a singular item.Tokenization is the process of converting a string into pieces or “tokens”. For example, converting a sentence into words or combination of words.

## Creating Bigrams and Trigrams: Bigrams and Trigrams plays a significant role in text preprocessing to create relevant pairs of words.

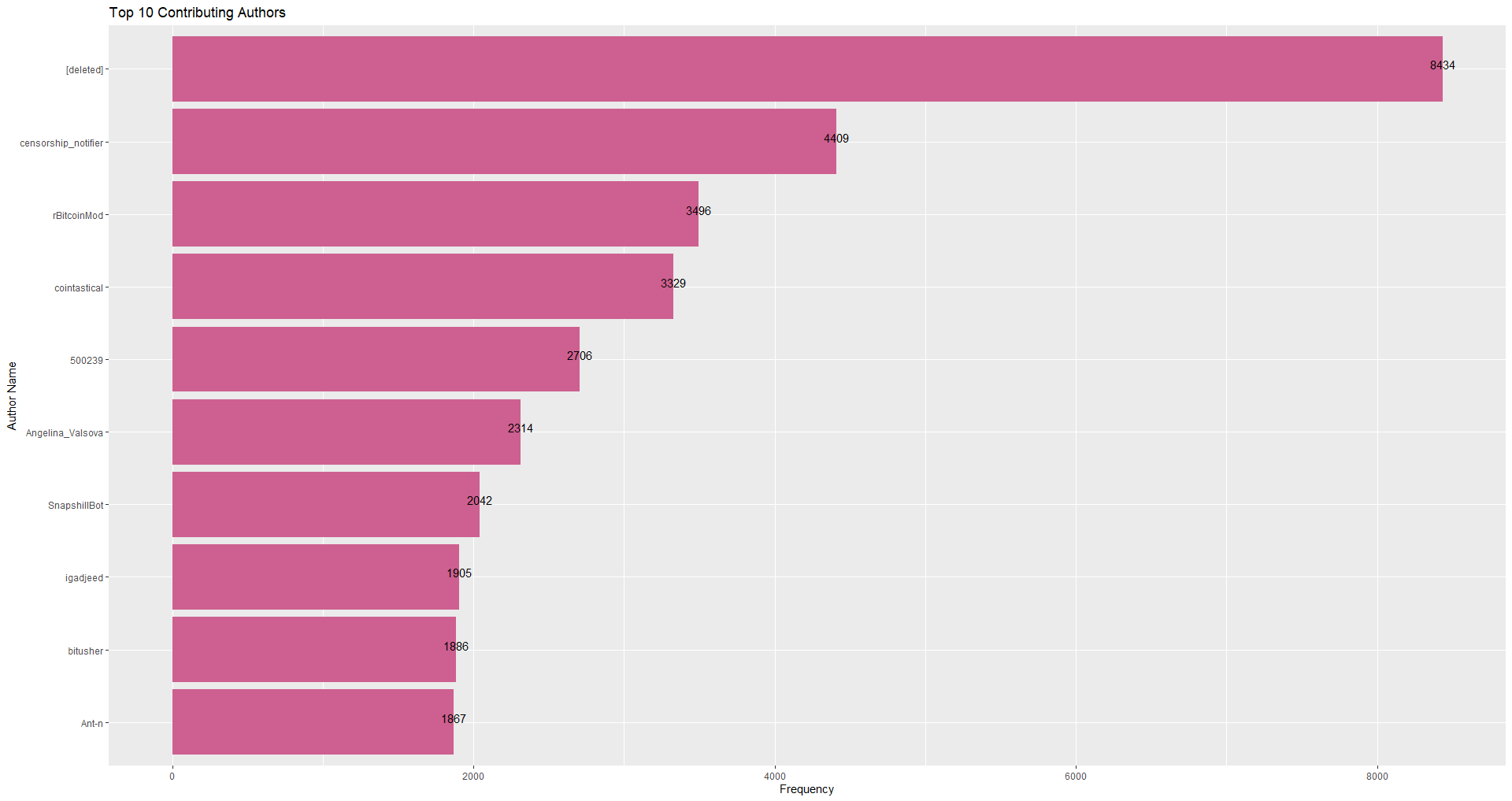
## Results:

## 

We can observe that special characters, punctuations and stop words are removed and the text has been lemmatized (for the answer column).



The above graph depicts the number of Reddit comments/ posts per day. Most posts are made on 30 September 2019, followed by July 1, 2019.

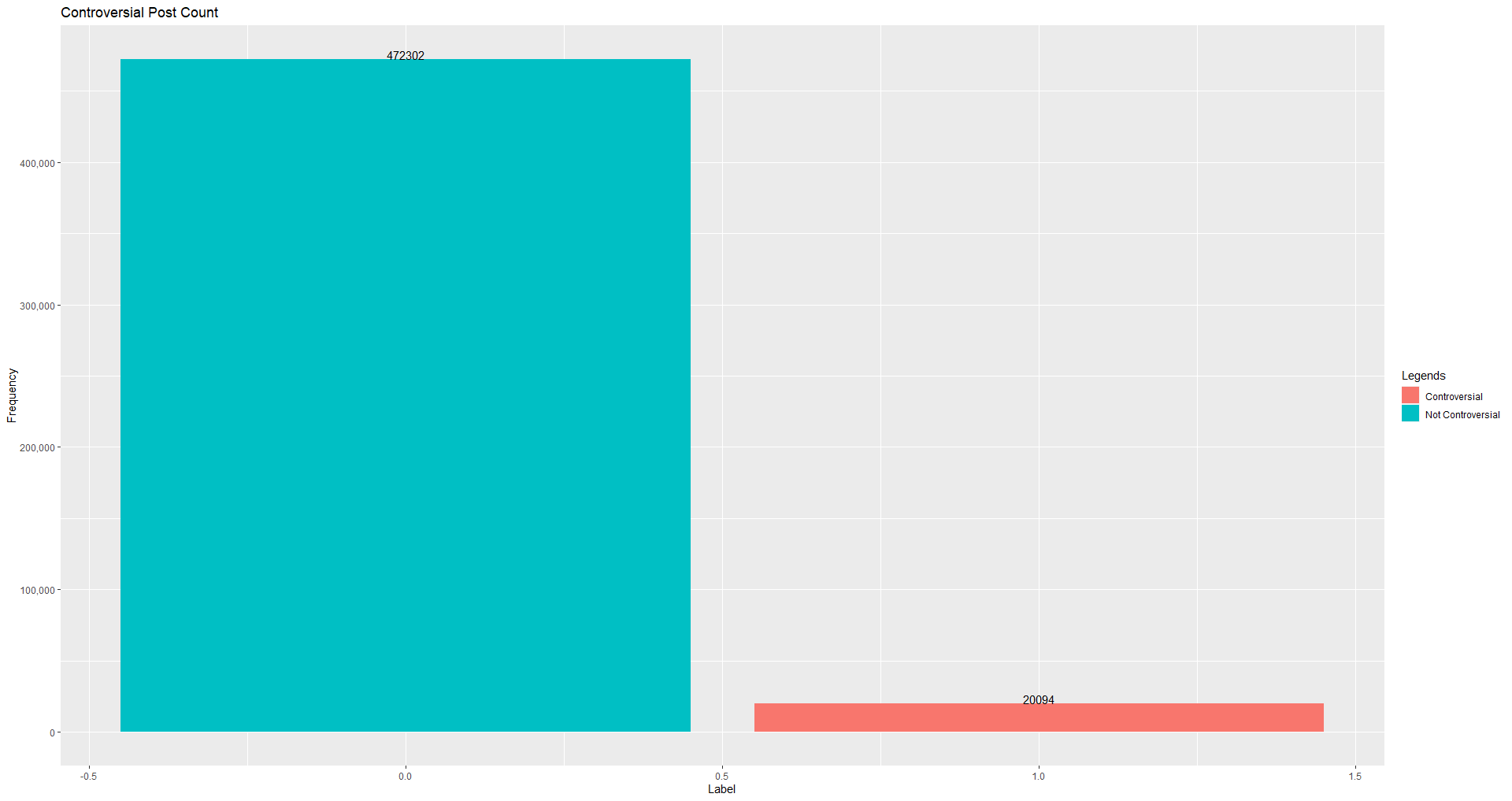


Most of the comments and posts have been deleted/ been made by anonymous users.

## 

## 

‘Bitcoin’ subreddit has the highest number of posts that correspond to BTC prices and discussions.

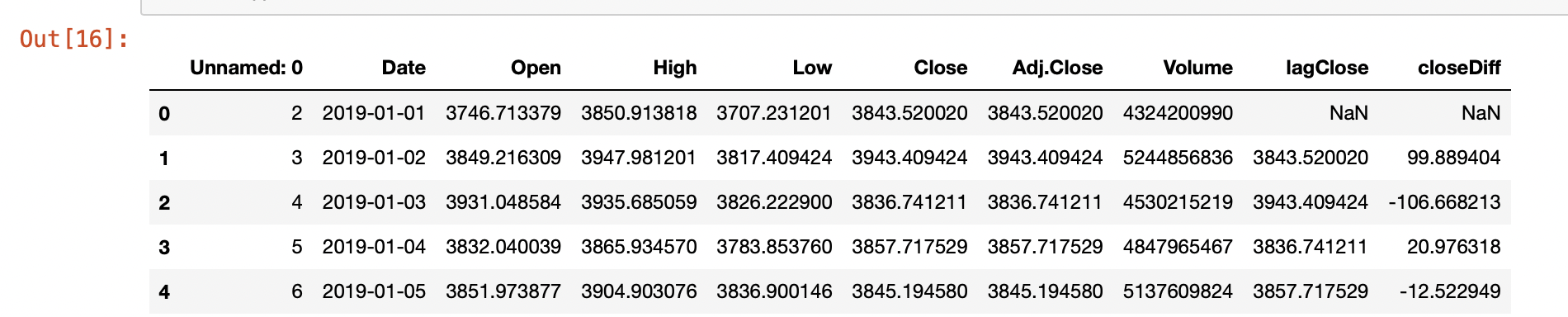


The number of posts characterized as Controversial is far less than the number of posts that are not controversial.

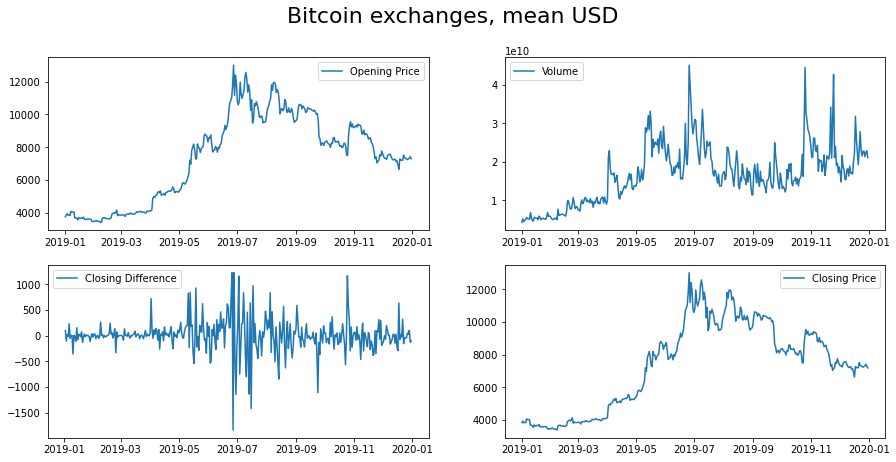
## Data Preprocessing (bitcoin data): Apart from eliminating null and missing values from the dataset, we require a constant mean and standard deviation for time series analysis, which can be verified by visualizing the mean and standard deviation for a rolling window of the bitcoin dataset. We will use the Self Lag Differencing strategy to get a constant rolling mean. It is the distinction between the current series and a delayed version of it. The shift might be in the order 1,2,3,4, etc., or NULL (for items for which no lagged version exists).

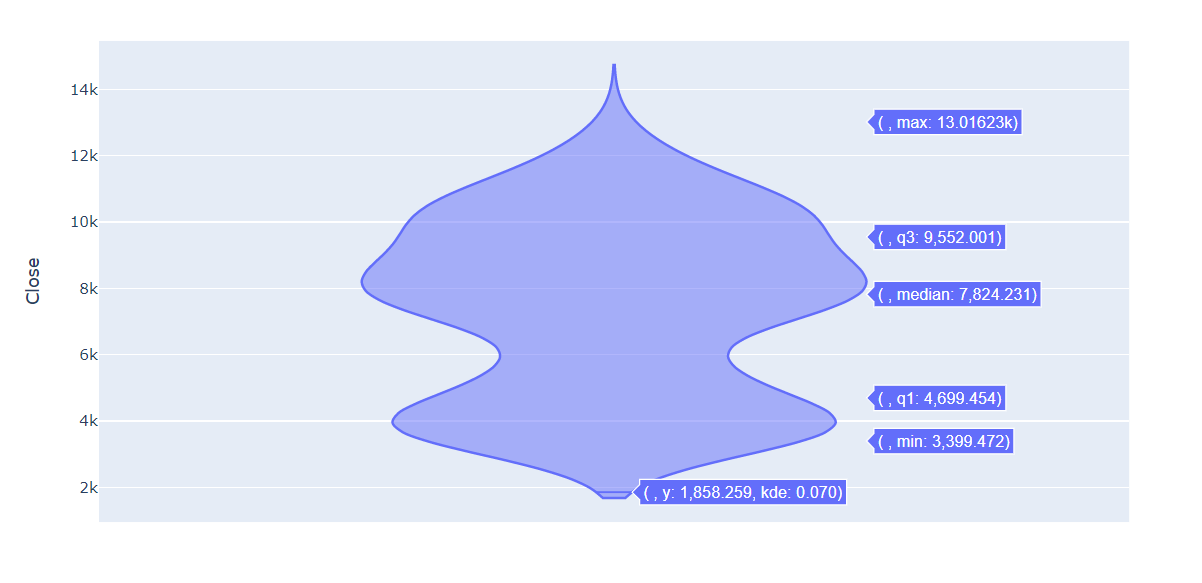
## Time[Price\_difference]=Time[Price]-Time[Price].shift(x)

## where x=lagged version(1,2,3,4, etc., or NULL)



**Results:**

****

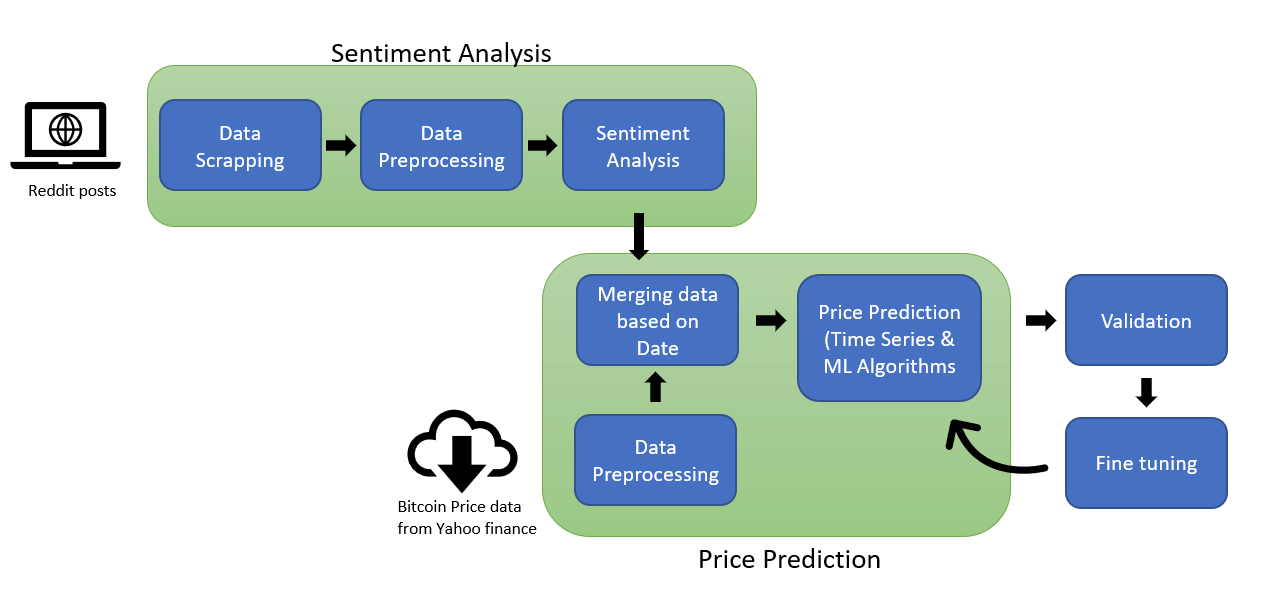


The opening and closing prices of BTC are almost similar. We can see that the price of BTC (both opening and closing price) shot up in July and had a continued trend of higher prices through the next half of the year. Along with the price the volume of the BTC that were transacted also shot up during July 2019. Later during November 2019, the volume of BTC transactions spiked up but the opening and closing price did not (compared to July).

The closing difference (difference of closing price between consecutive days) had much fluctuation during July.

Comparing the number of posts made on Reddit and the close price of BTC, during July we can find some kind of correlation. During October when the price observed a dip, the number of posts increased. During other parts of the year there is no much similarity in the number of posts and the price.

# Methodology



* **Sentiment Analysis:** 
  + **TextBlob:** TextBlob is a Python library for text processing. It offers a simple API for doing standard natural language processing (NLP) activities like part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, and translation, among others. TextBlob is useful since it behaves similarly to Python strings. As a result, you can transform and work with it in the same way for Python. It is a free and open-source Python library based on NLTK. It's simple to use, and it can process text in just a few lines of code. The lexicon it refers to is in [en-sentiment.xml](https://github.com/sloria/TextBlob/blob/eb08c120d364e908646731d60b4e4c6c1712ff63/textblob/en/en-sentiment.xml)

Textblob produces two outputs when a sentence is supplied to it: polarity and subjectivity. Polarity is the output that falls between [-1,1], with -1 denoting negative emotion and +1 denoting positive sentiment. Subjectivity is the output of [0,1], and it refers to personal judgements and views. Personal opinion, emotion, or judgment is often referred to be subjective, whereas objective relates to factual knowledge.

* + **VADER for Sentiment Analysis:** Sentiment analysis is a text analysis technique that finds polarity (e.g., a positive or negative opinion) in a text/ corpus. It tries to quantify an author's attitude, sentiments, assessments, and emotions using a computational approach of subjectivity in a text. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a text sentiment analysis model that takes into account both the polarity (positive/negative) and the intensity (strong) of emotion. The sentimental analysis of VADER is based on a lexicon that maps lexical elements to emotion intensities, which are referred to as sentiment scores. A text's sentiment score can be calculated by adding the intensity of each word in the text.

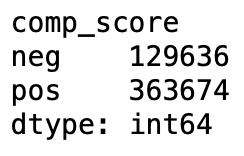
We are using two methods to perform sentiment analysis so that we can validate the output obtained the above two methods. On further validation, we found that the positive or negative sentiment scoring is similar for both methodogies used.

* **Price Prediction:** 
  + **Prophet:** Prophet is a time-series data forecasting process based on an additive model that fits non-linear trends with annual, weekly, and daily seasonality, as well as holiday impacts. It's an open-source software from Facebook's Core Data Science team that works best with time series with substantial seasonal effects and data from several seasons. Prophet is forgiving of missing data and trend changes, and it usually handles outliers well.
  + **SARIMAX:** SARIMAX stands for Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors which is an advanced version of the ARIMA model. It is a seasonal equivalent model like SARIMA and Auto ARIMA which deals with external effects. Time-series data is the data that is indexed in such a way that the data points represent the magnitude of changes occurring over time. Another advantage of using this model is that it can handle external effects as well as the seasonal pattern. This aspect of the model sets it apart from others. In the SARIMAX models, we need to provide two kinds of orders. The first one is similar to the ARIMAX model (p, d, q). The other is to specify the effect of the seasonality: Seasonal AR specification, Seasonal Integration order, Seasonal MA and Seasonal periodicity. We need to find the optimal parameters for the SARIMAX model that best fit the data and provide better predictions.
  + **Random Forest:** Random Forest is an ensemble machine learning technique that uses numerous decision trees and a statistical technique called bagging to perform both regression and classification tasks. Bagging and boosting are two of the most often used ensemble strategies for dealing with excessive variation and bias. In a random forest, each tree learns from a sample of the training observations. Because the samples are drawn via replacement, or bootstrapping, certain samples will be utilized numerous times in a single tree. The notion is that by training each tree on diverse samples, even if each tree has a high variance with regard to a specific set of training data, each tree will have a lower variance overall. The variance of the entire forest will be reduced, but not at the expense of increasing the bias.

# Results

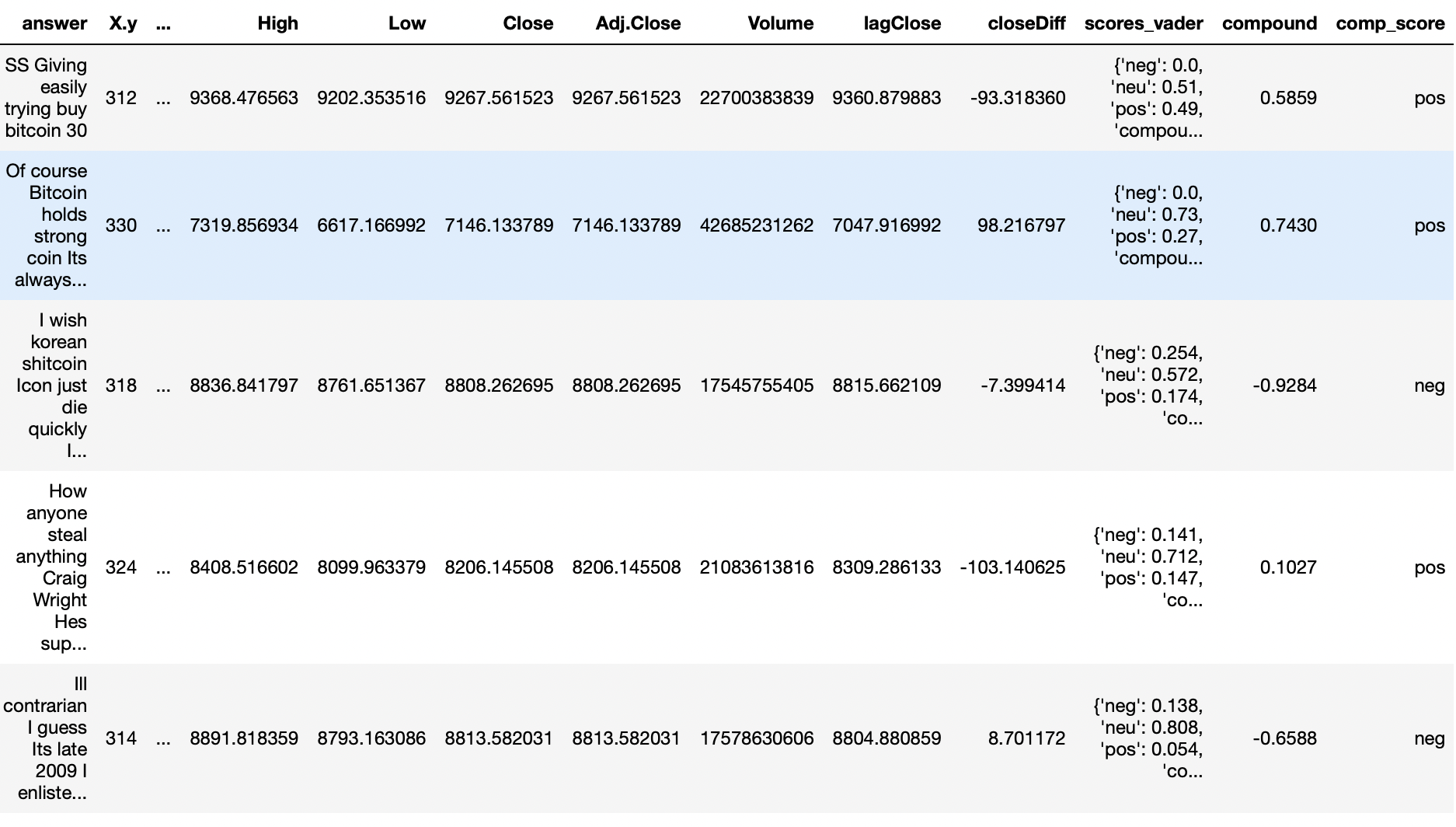
* **VADER Results:** By creating a lexicon or a "dictionary of emotions/ sentiments," lexical techniques try to relate words to emotion. This dictionary may be used to determine the sentiment of phrases and sentences without having to look at anything else. Sentiment can be categorical — negative, neutral, or positive — or numerical — a score for intensity. Lexical techniques consider the sentiment category or score of each word in the phrase to get the overall sentiment category or score. The value of lexical techniques rests in the fact that we don't need to train a model using labeled data because the lexicon of emotions has all we need to assess the sentiment of phrases.

Using VADER we generated the polarity score and thus category of emotions from the body of the post. The whole posts for the year 2019 can be summarized as:

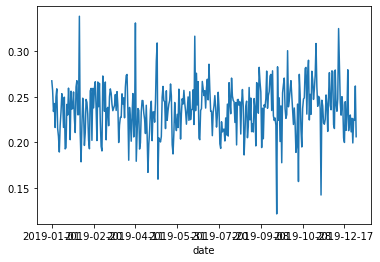


There exists more positive posts compared to negative that are related to BTC and CryptoCurrency on Reddit.

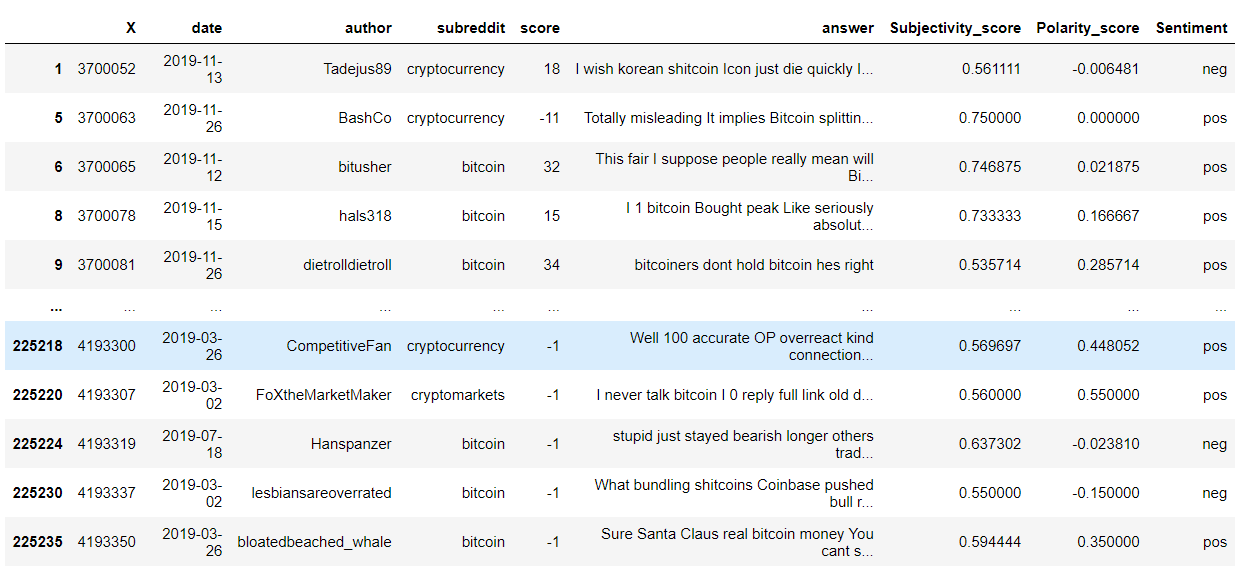
The sample output of VADER is shown below:



Average score of sentiment output from VADER:



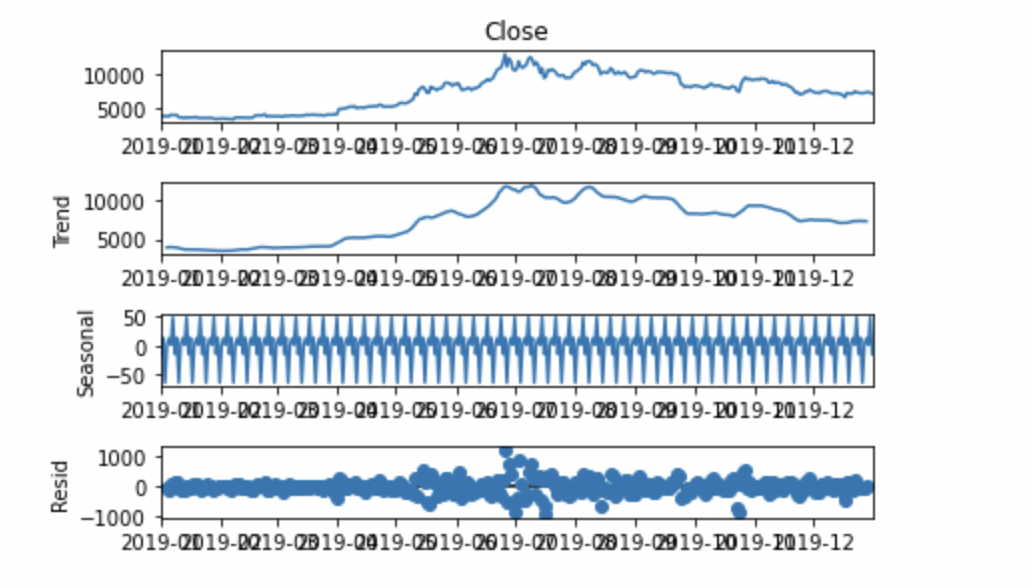
* **TextBlob Results:** Textlob uses en-sentiment lexicon to score the input corpus.



* **SARIMAX**: The first step of implementing SARIMAX includes eliminating columns that are no longer needed, checking for missing information, aggregating the values at day level (in our case no need of aggregation as data is present at day level) and indexing the date column.



We visualize our data using time-series decomposition, which divides our time series into three components: trend, seasonality, and noise.



Trend is the increasing or decreasing value in the time series. We can observe that there is an increasing trend after May 2019 and a slightly decreasing trend after October 2019. Seasonality is the repeating short-term cycle in the series. We can observe that there exists seasonality in the data. We can also see that the residual plot is distributed around 0.

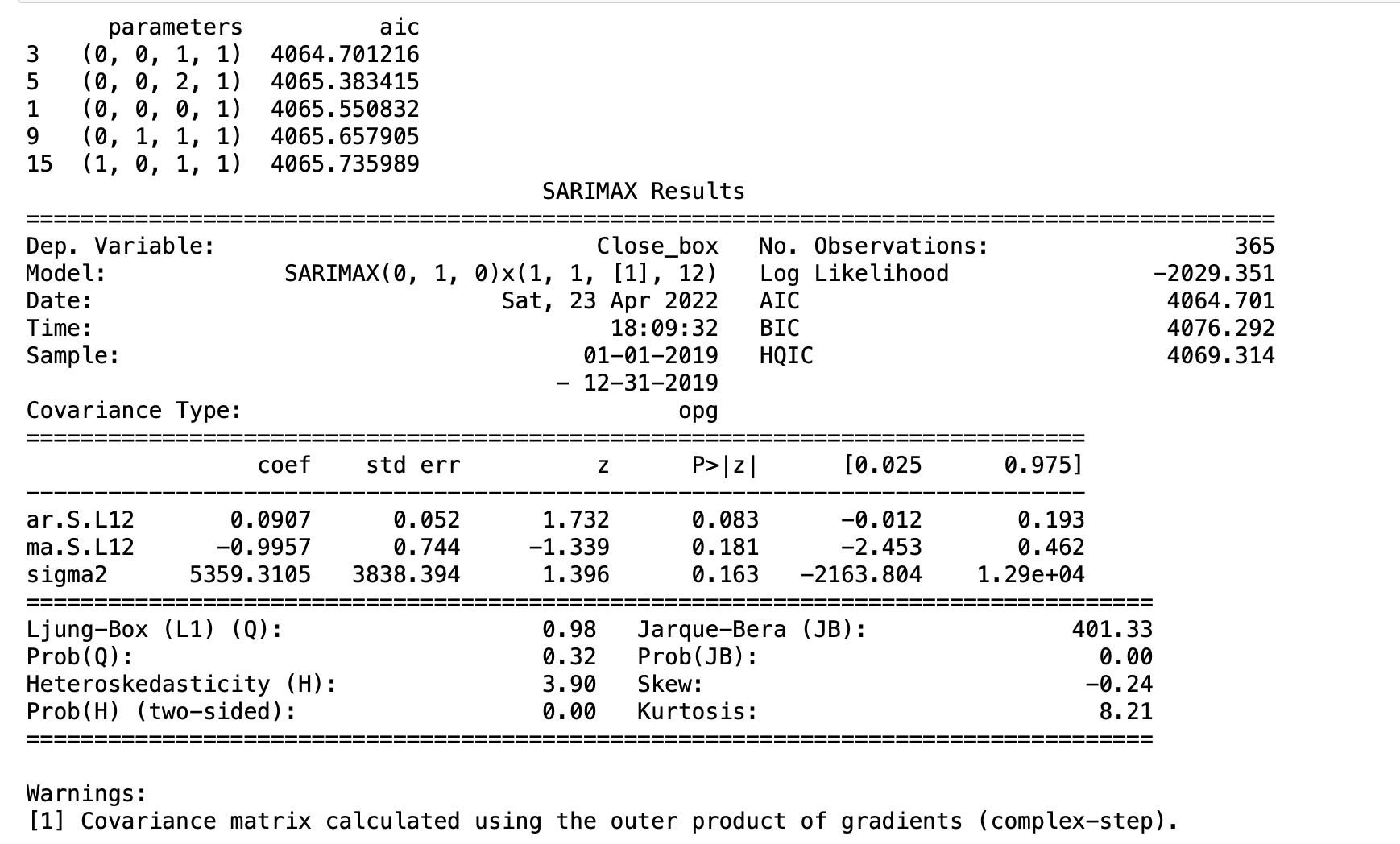


A stationary time series is required for improved prediction when creating a model for forecasting purposes in time series analysis. As a result, making a time series stationary is the first step in modeling. In autoregressive modeling, testing for stationarity is a common task. Performing the Dickey-Fuller test to check the stationarity of the Close Price column, we get p value to be 0.5 which is > 0.05; we fail to reject Null Hypothesis. We need to perform transformation to convert the values to stationary series.

Computing the differences between consecutive observations to make a non-stationary time series stationary is referred to as differencing. We performed differencing to generate stationary series.

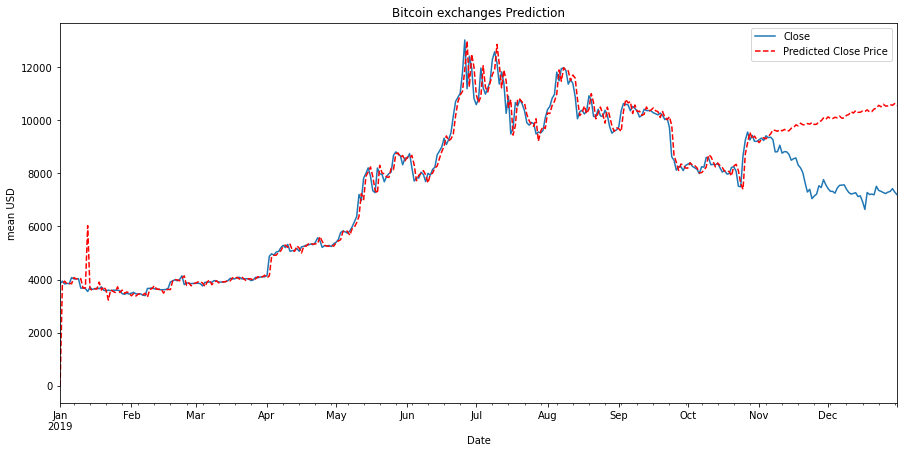
The Box-Cox transformation is a set of power transformations that are indexed by the lambda parameter. Whenever you use it, you must estimate the parameter from the data. The process in a time series could have a non-constant variance. The process is nonstationary if the variance fluctuates with time. A time series is frequently transformed to make it stationary. The process may appear stationary after applying Box-Cox with a specific lambda value.



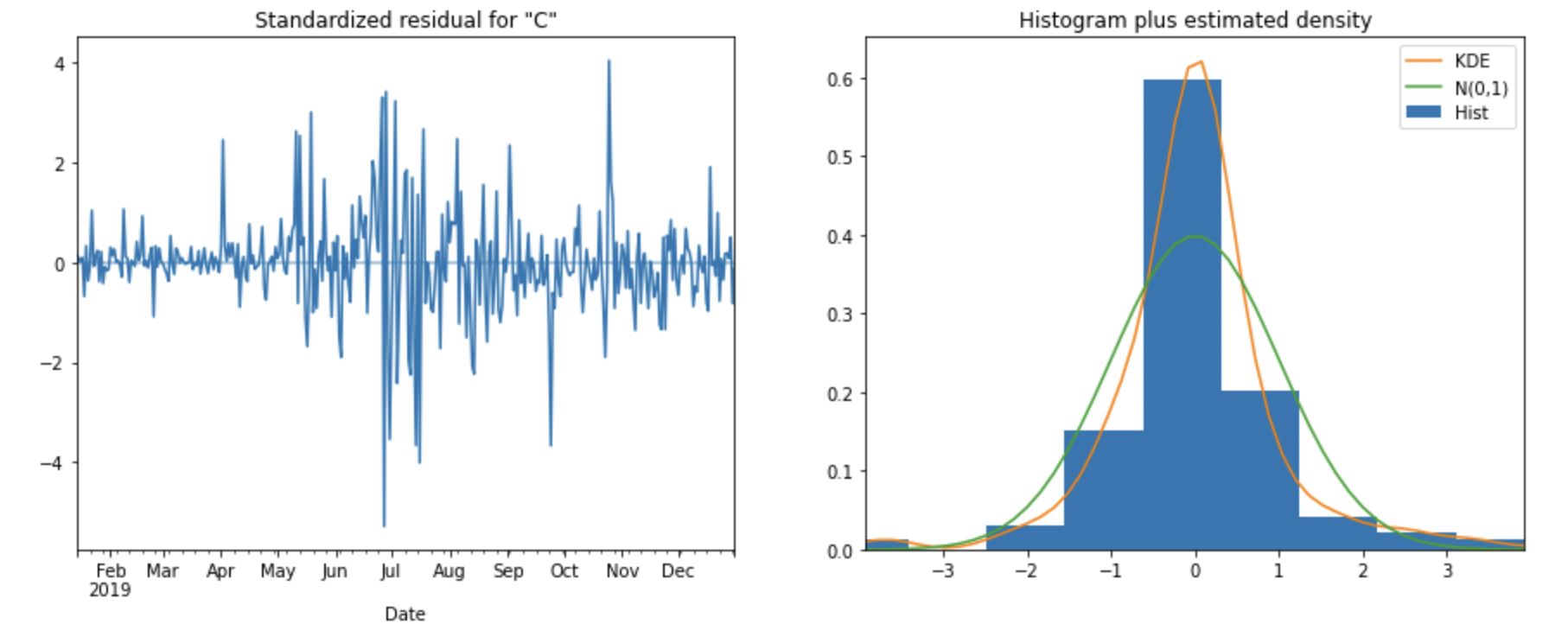


The best model that is fit to the given data is : (p,d,q) = (0,1,0) and (Seasonal AR specification, Seasonal Integration order, Seasonal MA and Seasonal periodicity) = (1,1,1,12). The other statistics of the model are shown in the figure.

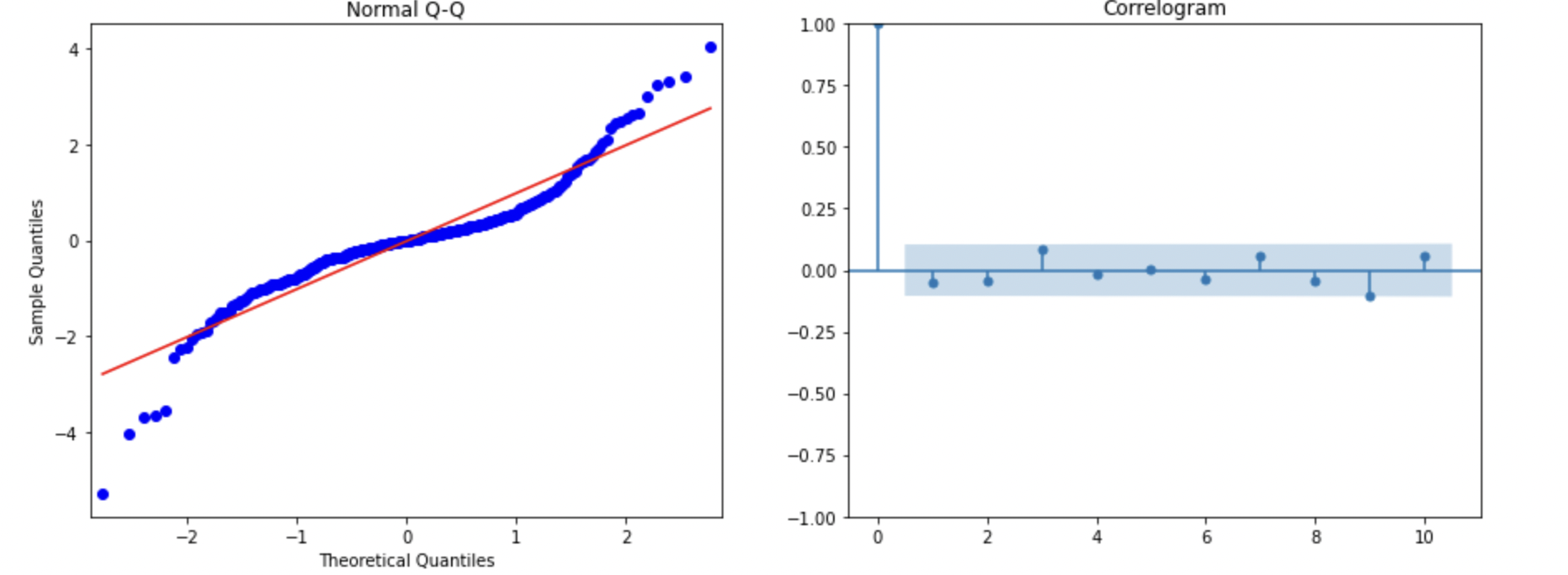
Prediction:



The above figure shows the output/ prediction of closing price using SARIMAX.

****

We can see that the predictions’ residual is normally distributed with a mean around 0.



The Normal Q-Q plot shows that residuals are slightly tailed/ almost normal and there is no correlation between the lag for the output.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | MSE | RMSE | MAPE |
| SARIMAX | 1169953.05 | 1081.64 | 7.61 |

**Random Forest**: For Random Forest, we need 3 data sources which contain the predictors needed for our model. That is the VADER dataset, TEXTBLOB dataset, and the 2019 bitcoin dataset.

We need to extract the needed column for each dataset. Like the compound score of the VADER sentiment score. And the Polarity score of the TEXTBLOB sentiment score. After getting the needed columns, because the sentiment score that we have is on a timely basis, we have to find a way to aggregate the scores into one day. So, we take the mean of the columns on a daily basis. After getting the aggregated scores, we merge it with the bitcoin dataset. After that, we take needed columns, that are:

Vader Input: compound score, negativity score, neutrality score, positivity score, positive negative score, vote score, post count, open price, close price.

TextBlob input: subjectivity score, polarity score, positive negative score, vote score, post count, open price, close price.

Then, we split the data into train and test. The partition used for this model is 80:20 for train and test. After that, we model the random forest regressor with close price as the target variable and others as the predictors. We also tune some parameters in order to get the best results. the parameters that are tuned:

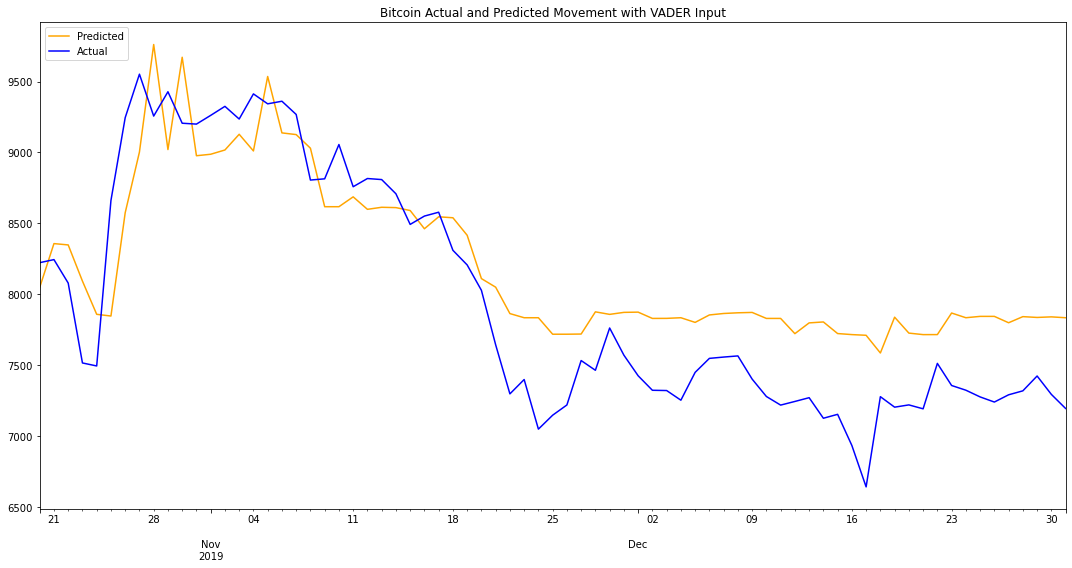
n\_estimators : The number of trees in the forest.

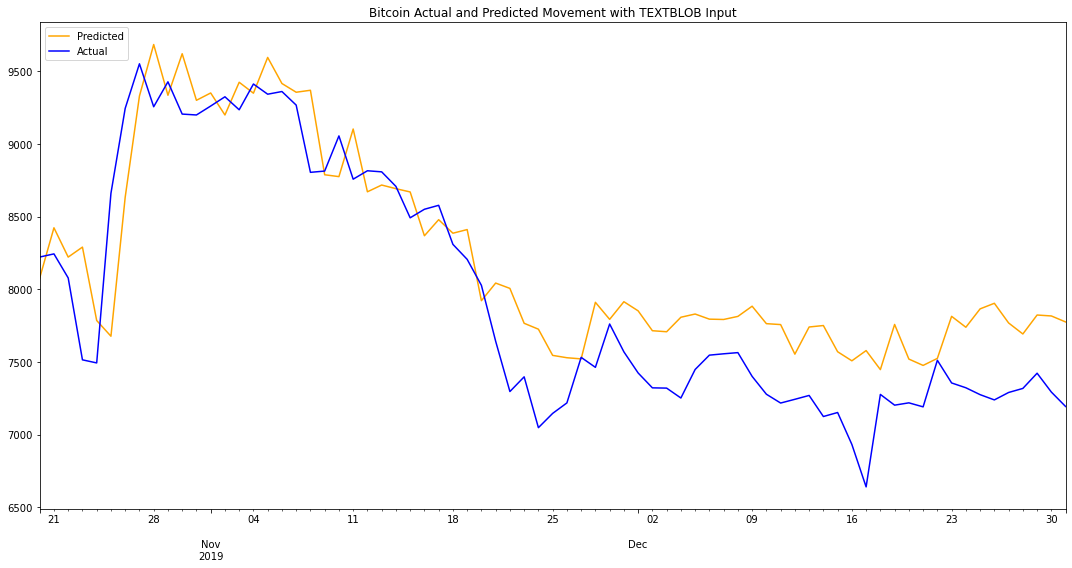
max\_depth : The maximum depth of the tree.

min\_sample\_split : The minimum number of samples required to split an internal node.

min\_sample\_leaf : The minimum number of samples required to be at a leaf node.

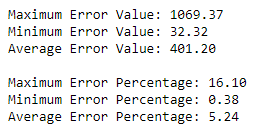
After fitting and testing the test predictors. We plot the actual price and the predicted price. the results are following:



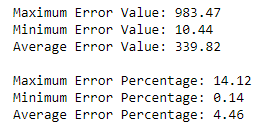


Judging from the plot, we can see from both of the graphs, the prediction from the TEXTBLOB input. seems to be closer with the actual prediction than the one with VADER input. This is maybe due to the fact that the predictors count is different between the two inputs.

To check the precise value, We calculate the difference between the prediction results and the respective closing price of Bitcoin by subtracting them and taking the absolute value of the difference. We also try to know the error value percentage of the actual value. The results are following:



VADER Statistics of Error



TEXTBLOB Statistics of Error

From the statistics perspective, we can see that with TEXTBLOB input, we got less error in prediction than with VADER input.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | MSE | RMSE | MAPE |
| VADER | 204284.208 | 401.2081 | 5.24 |
| TEXTBLOB | 150446.5484 | 339.8244 | 4.46 |

Feature Importance of VADER input

|  |  |
| --- | --- |
| Attribute | Importance |
| Open | 0.998867 |
| Post Count | 0.000346 |
| Mean of Positivity Score | 0.000270 |
| Mean of Vote Score | 0.000148 |
| Mean of Neutrality Score | 0.000140 |
| Mean of Negativity Score | 0.000136 |
| Mean of Positive Negative count | 0.000051 |
| Mean of Compound score | 0.000042 |

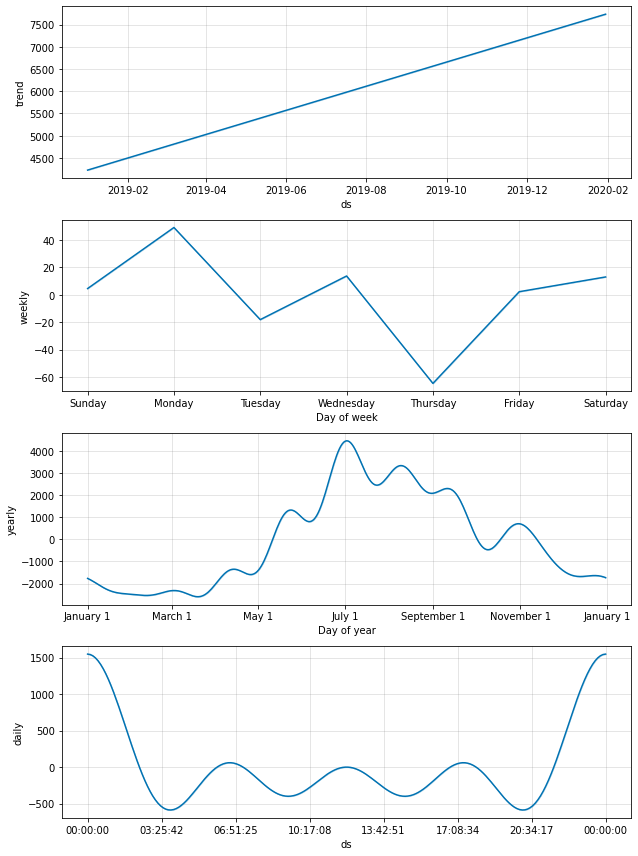
Feature Importance of TEXTBLOB input

|  |  |
| --- | --- |
| Attribute | Importance |
| Open | 0.995324 |
| Post Count | 0.001904 |
| Mean of Positive Negative count | 0.001136 |
| Mean of Vote Score | 0.000677 |
| Mean of Polarity Score | 0.000497 |
| Mean of Subjectivity Score | 0.000462 |

**Prophet:** For Prophet, eliminating columns that are no longer needed, verifying for missing information, aggregating values at the day level (no need for aggregation in our instance because data is present at the day level), and indexing the date column are the initial steps in implementing.

The trend in a time series is the value that is rising till the year end. After May 2019, we may see a growing tendency and a little declining trend after July 2019. For the weekly plot, we can there is a rise on Monday and a decline till Thursday and a further rise till Saturday. For the daily plot, the graph is below zero for most part of the day apart from late night and early morning (between 10PM and 3AM)

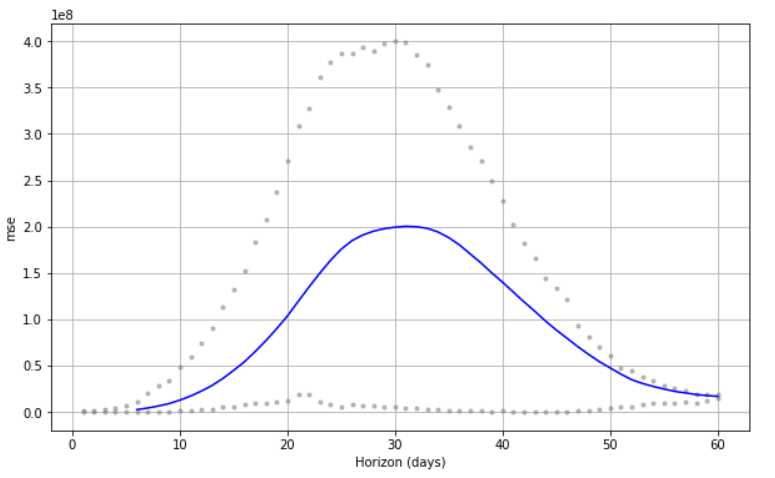
We first start with univariate implementation of Prophet with Date as input and ‘Close’ value as output. For running the model on different values of internal\_width, the best output was obtained for the value of 0.95. Hence, we are keeping it consistent for the later runs as we move to multivariate model.



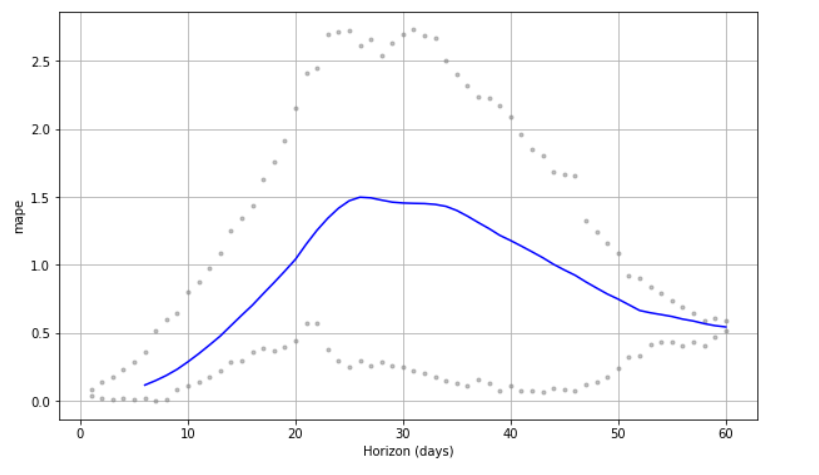
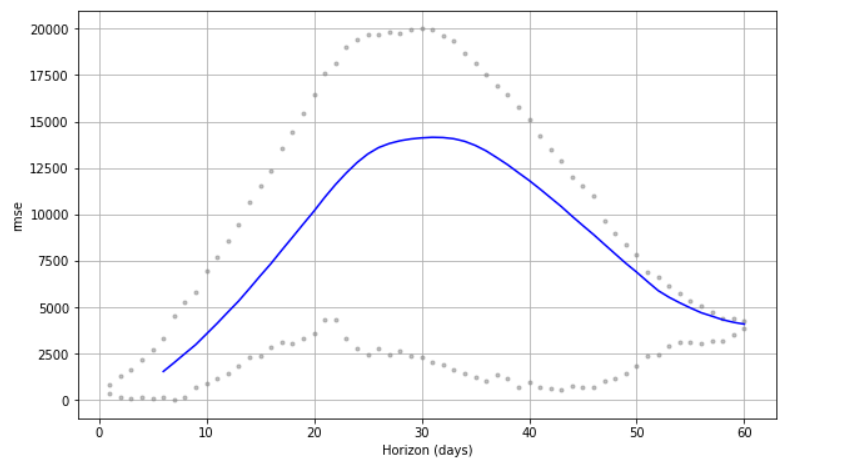
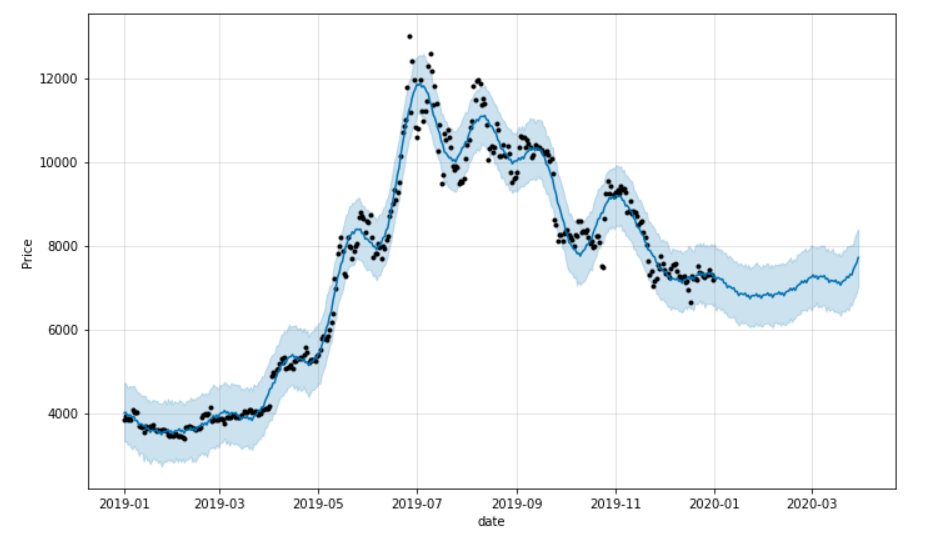
For multivariate implementation of Prophet, we merge the data with reddit sentiment analysis scored obtained from VADER and TextBlob respectively. The data is merged on the basis of data. We have also taken average compound score and post\_count of the reddit posts on a per-day basis. Then, we add the regressors in the model one by one to see the impact of each variable.

Prophet with VADER sentiment score:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Variables | MSE | RMSE | MAPE |
| 1 | Univariate | 2 | ~14000 | 1.5 |
| 2 | Compund,Post\_Count | 1.75 | ~4200 | ~0.5 |
| 3 | +Volume | ~2.2 | ~4500 | ~0.55 |
| 4 | -Volume, +Open | <1 | ~1000 | ~0.11 |

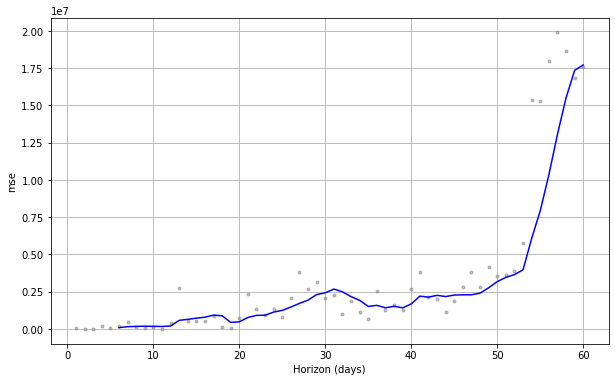


1. **Model 1:**

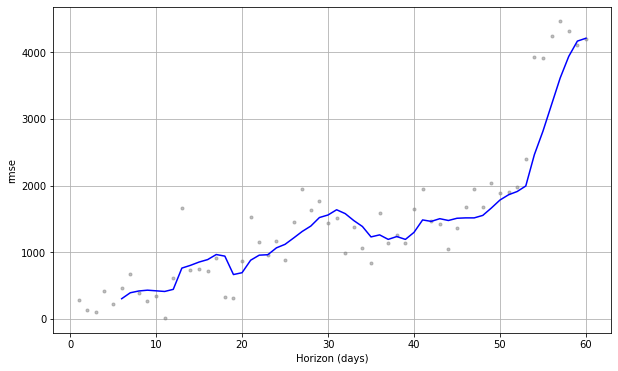
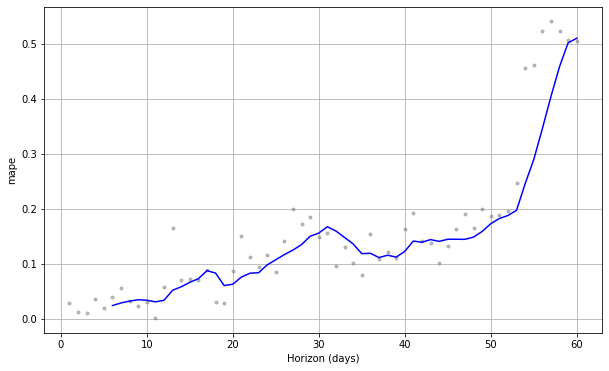
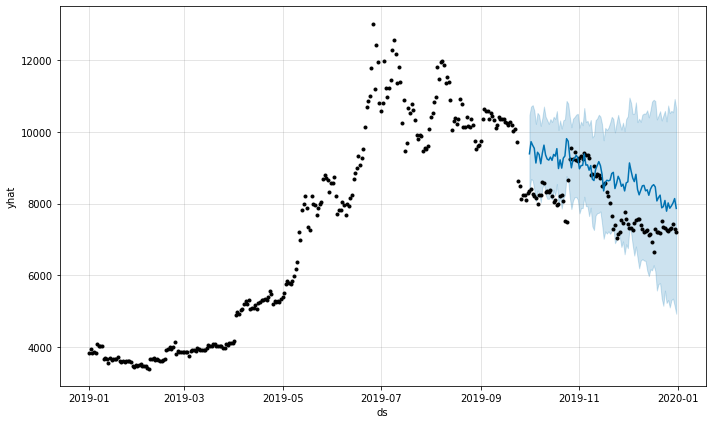


**Fig: Date vs Predicted Price**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

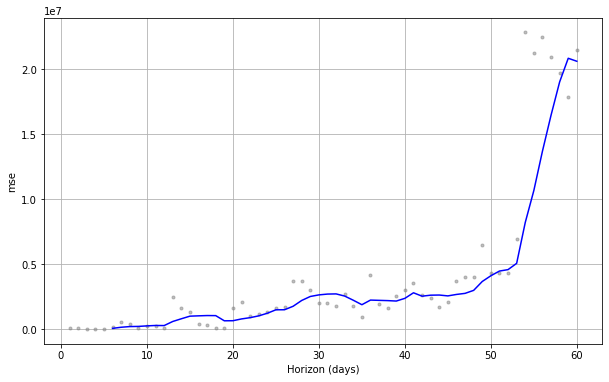


1. **Model 2:**

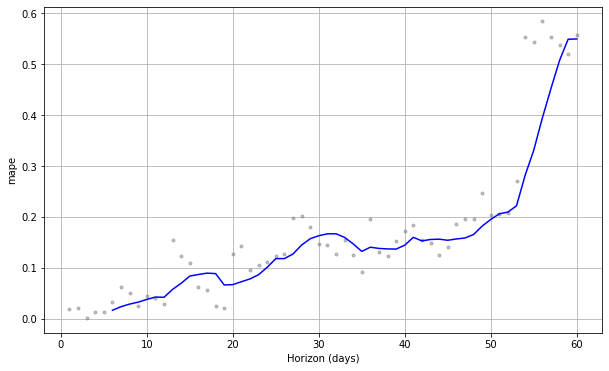
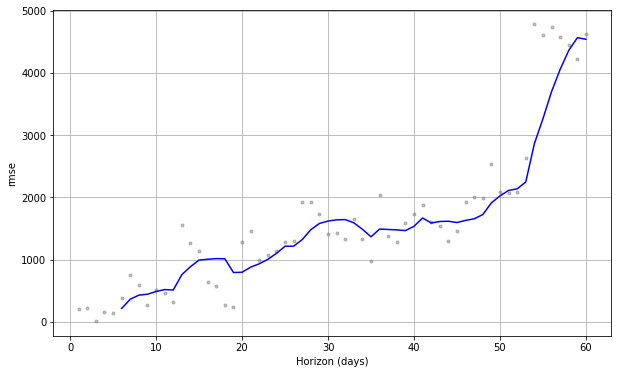
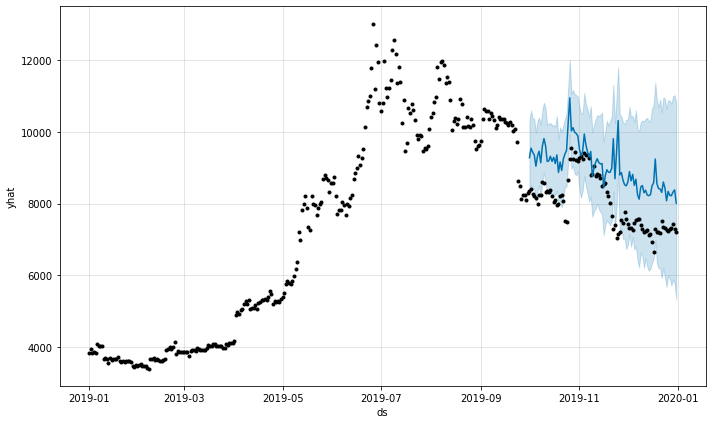


**Fig: Date vs Predicted value**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

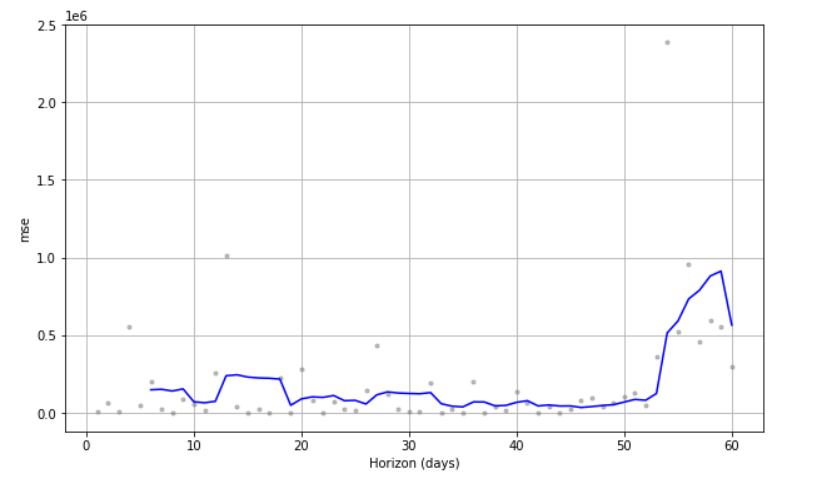


1. **Model 3:**

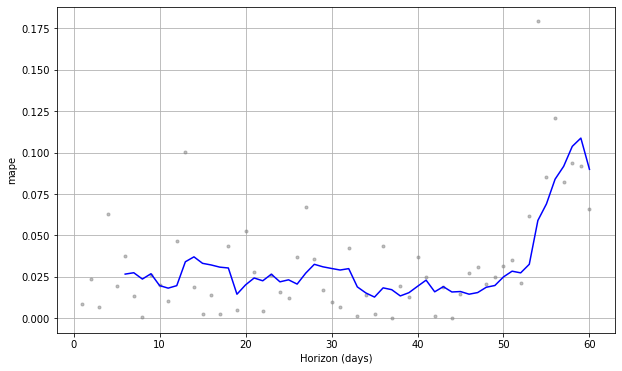
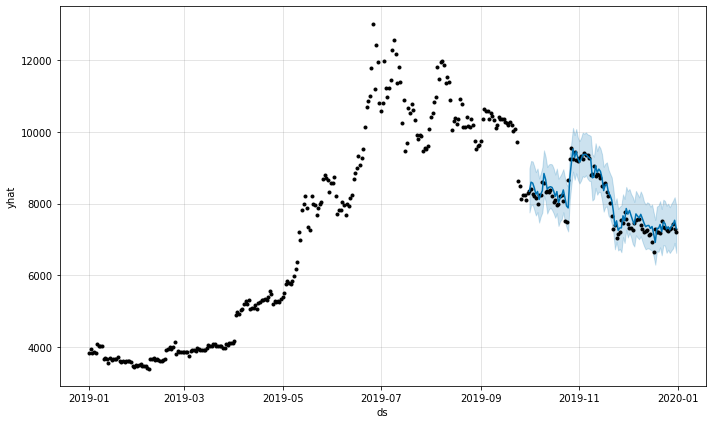


**Fig: Date vs Predicted value**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**



1. **Model 4:**



**Fig: Date vs Predicted value**

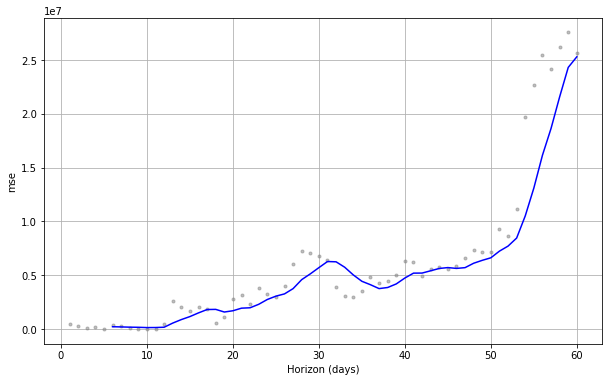
**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

Based on the result obtained from Prophet for VADER input, we can say that with the use of just sentiment scores and reddit post count, we are able to predict the close value for the predicted time period however it is not consistent. With the use of ‘Open’ value as input, the output obtained is more precise and the Mean Average Percentage Value is also very less.

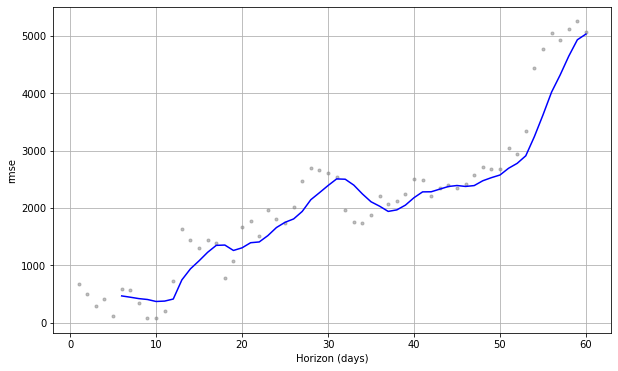
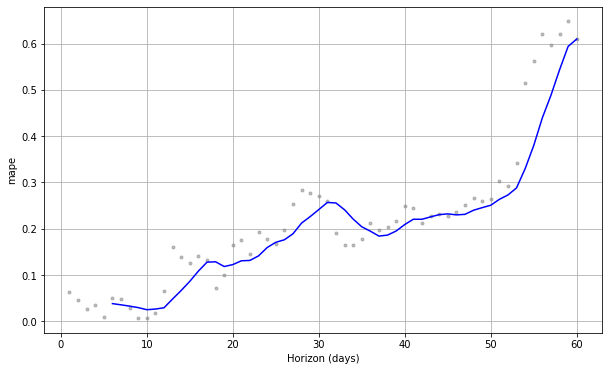
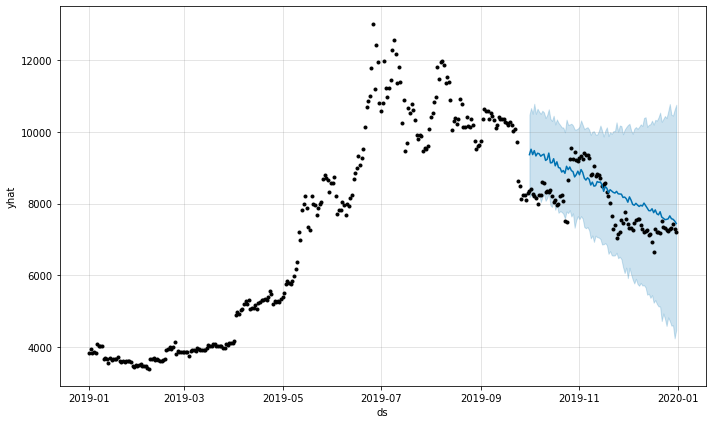
**Prophet with Textblob sentiment score**

For implementation with TextBlob Sentiment Analysis scores, we first filtered the reddit posts with Subjectivity\_score greater than 0.5. After that, we performed the similar averaging for Subjectivity\_score as well as Polarity\_Score on a per day basis with the Post\_count.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Variables | MSE | RMSE | MAPE |
| 1 | Polarity\_Score | ~2.5 | 5000 | ~0.6 |
| 2 | Polarity\_Score,Subjectivity\_score | ~2.5 | 5000 | ~0.6 |
| 3 | Polarity\_Score,Subjectivity\_score, Post\_count | ~1.4 | ~3700 | ~0.45 |
| 4 | Polarity\_Score,Subjectivity\_score, Post\_count,Volume | <2 | ~4200 | <0.55 |
| 5 | Polarity\_Score,Subjectivity\_score, Post\_count, Open | ~0.8 | ~950 | ~0.1 |

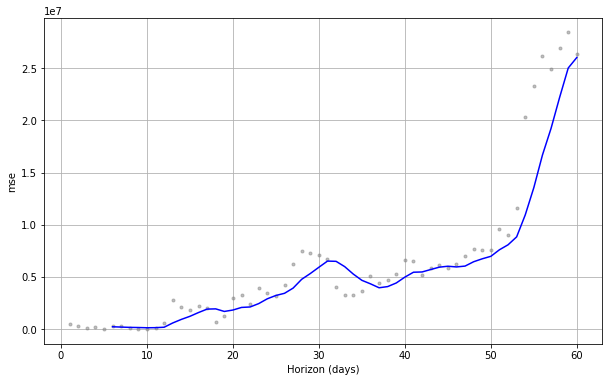


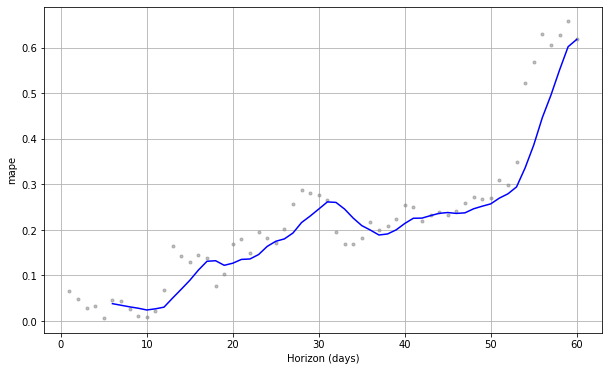
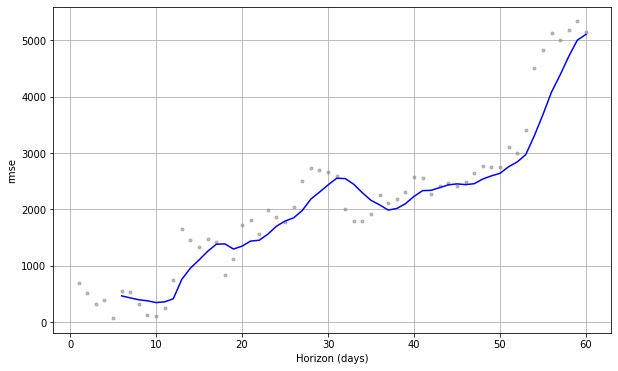
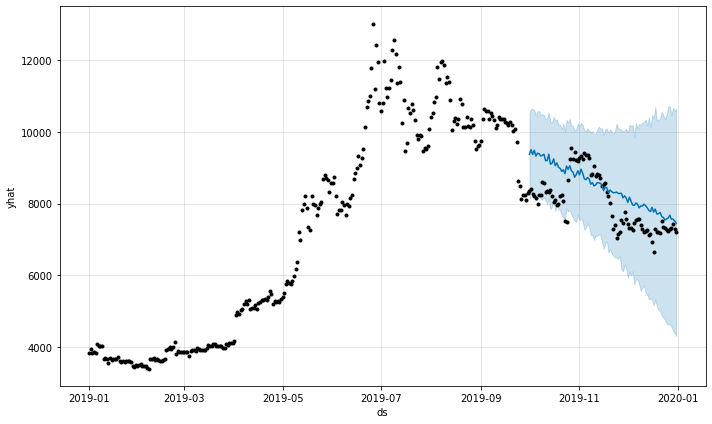
1. **Model 1:**



**Fig: Date vs Predicted value**

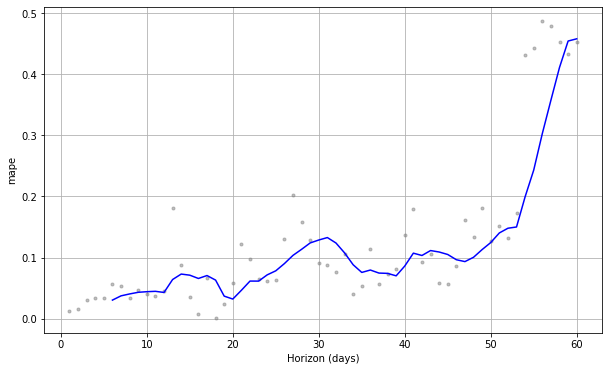
**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

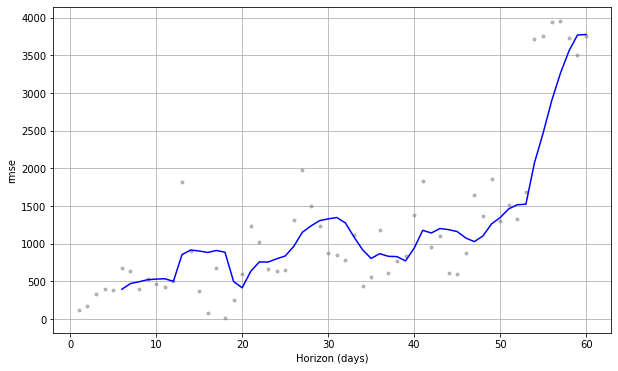
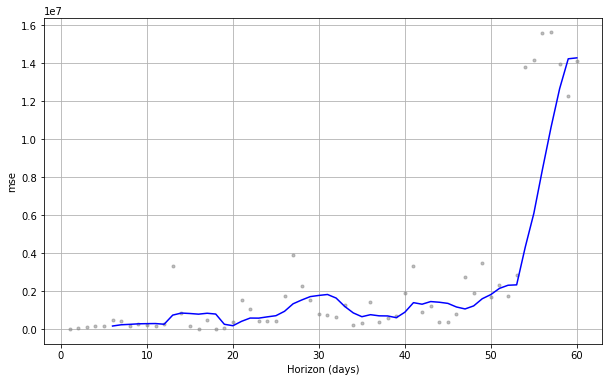
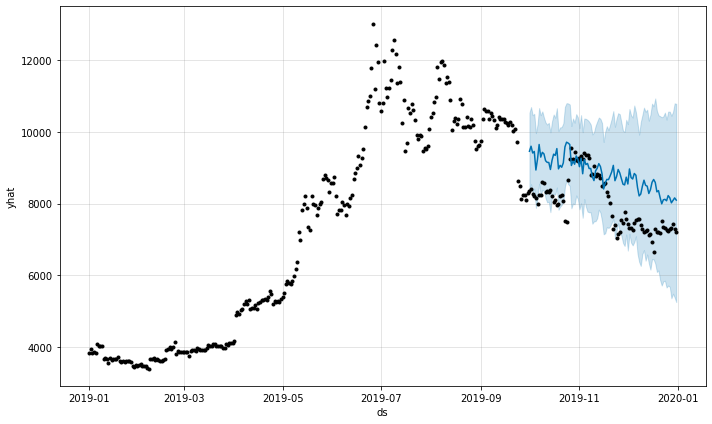
1. **Model 2:** 



**Fig: Date vs Predicted value**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

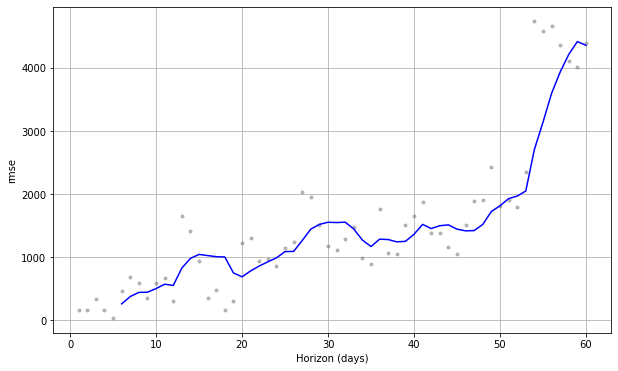
1. **Model 3:** 

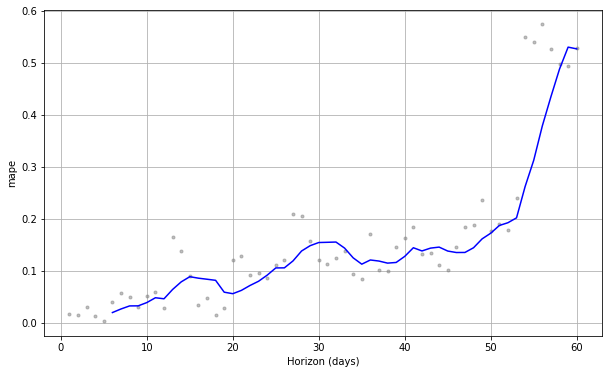
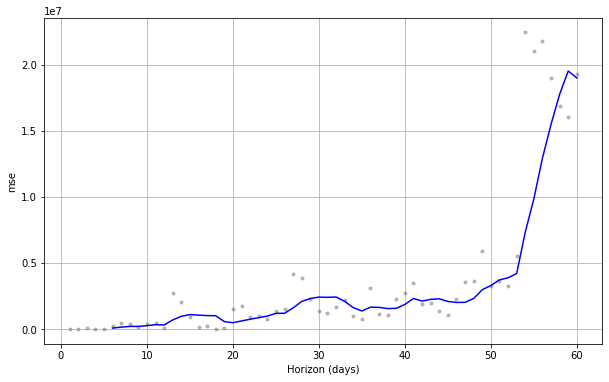
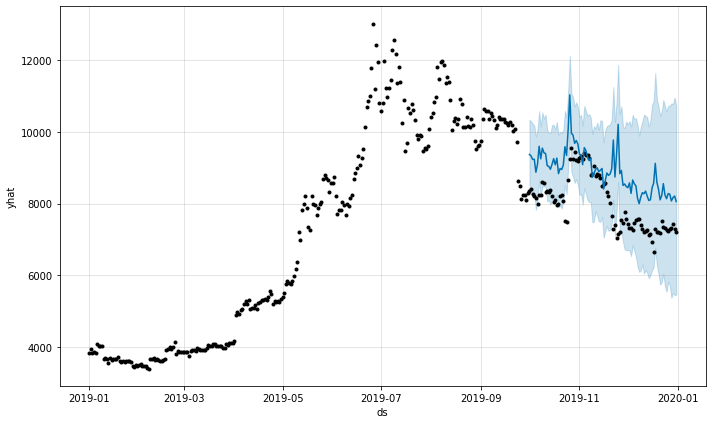


**Fig: Date vs Predicted value**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

1. **Model 4:**

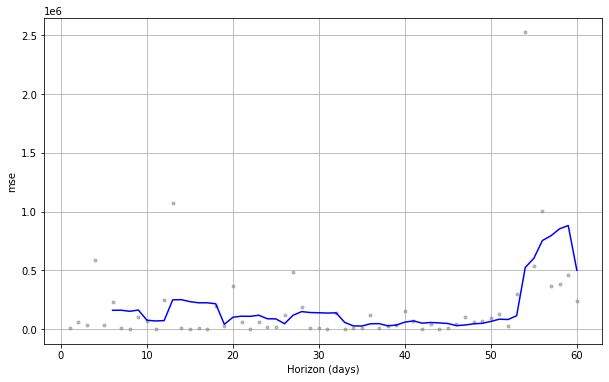


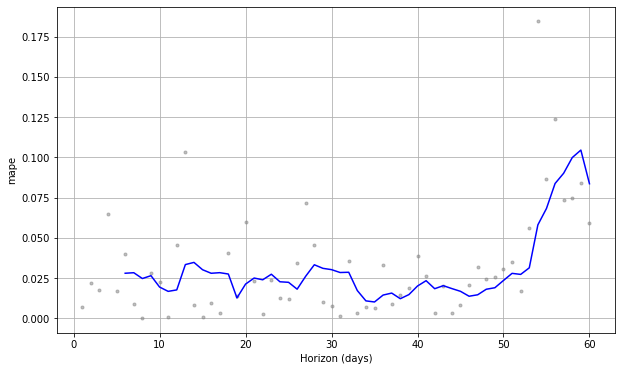
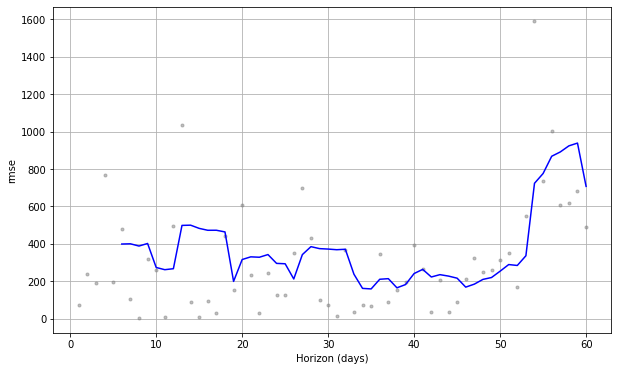
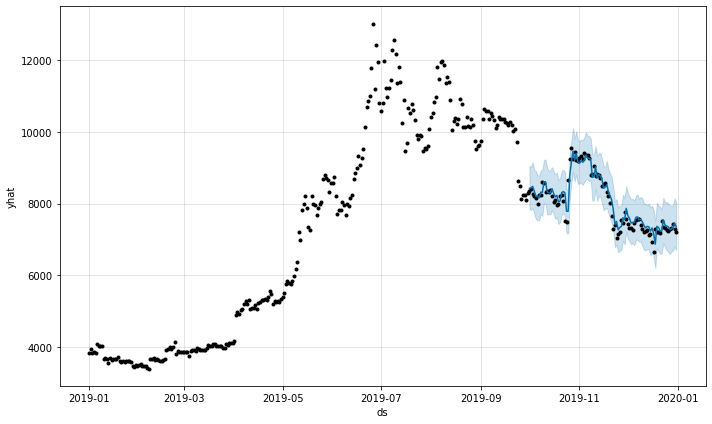


**Fig: Date vs Predicted value**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

1. **Model 5:**





**Fig: Date vs Predicted value**

**[Top-Down: MSE, RMSE, MAPE graphs w.r.t. Horizon days]**

Based on the output of model with inputs of sentiment scores from textBlob, the model performance is better compared to the models with sentiment scores from VADER. However, we see a similar trend for the multivariate models. The input features from sentiment scores provide good results however after including the ‘Open’ value of bitcoin prices,

# Discussion of Results

**Usefulness of work:**

Stock sentiment research can be used to figure out how investors feel about a certain stock or asset. Sentiment can sometimes be a good indicator of future price behavior. This is also an example of how trading psychology can influence a market, serving as a forecasting tool for future price fluctuations in a certain asset. Stock sentiment is influenced by a variety of factors, including news (economic, political, and industry-related) and social media. Stock values can be far more susceptible to quick adjustments during periods of extreme volatility. Certain factual and emotional events, such as bad comments on Twitter/social media and the news, can induce anxiety in the market, causing investors to sell a specific stock or firm in large numbers. When favorable news is provided, it can lead to optimism, which can lead to an increase in the price of a particular stock.

**Limitations of the work:**

Our analysis is only as good as the information available right now, no matter how good it is. We have no way of knowing what will happen tomorrow. When analyzing future movement, "everything else being equal" is used as a guiding principle. This suggests that if things stay the same, we expect a stock will rise as a result of a trend. Predictions are frequently based on strong emotional feelings—the higher the emotion, the stronger the trader may expect the price reaction. As a result, the trader expects that the stock will move in a straight line in the expected direction, resulting in significant winnings. When we consider all of the world's securities and then add in time variables, owning a position immediately before a large move is statistically exceedingly rare.

**Conclusion:**

Based on the value of Mean Square Error, Root Mean Square Error and Mean Absolute Percentage Error for the sentiment analysis methodologies, we can state that TextBlob is preferable for scoring reddit posts. As we know that textBlob also provides subjectivity score, it acts as a helpful feature in model building for price prediction. For the price prediction models, we can say that Prophet (with textBlob sentiment scores) has the best output compared to other combination of proce prediction and sentiment score methods. The MSE for Prophet was about 0.8 with RMSE around 950 and MAPE of around 0.1.

The algorithms used also are providing that the feature importance of Sentiment is negligible or very small. However, using multiple sources might help us have a better view of the impact of emotions/ sentiments on the price of bitcoin.

Additionally, factors like limited dataset, chain effect, fake posts also affect the performance of the model which is difficult to address in the given time-span. Considering other factors related to the posts or using/ deriving KPIs can help improve the results of prediction.

# References

* Abraham, Jethin; Higdon, Daniel; Nelson, John; and Ibarra, Juan (2018) "Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis," SMU Data Science Review: Vol. 1 : No. 3 , Article 1
* Li TR, Chamrajnagar AS, Fong XR, Rizik NR and Fu F (2019) Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model. Front. Phys. 7:98. doi: 10.3389/fphy.2019.00098