Stock Price Analysis, Prediction & Benchmarking of different Machine Learning Algorithms.

Amandeep Singh Oberoi (A20466752)
Amit Nikam (A20470263)

Abstract

Prediction and analysis of returns from the stock market is a very challenging task: a result of the volatile as well as non-linear nature of the financial stock markets. Traditionally, to predict financial market movements - investors would analyze the stock prices as well as technical indicators. In addition to those, the news related to these stocks also served as a useful source of information. The stock market indicators such as: High, Low and Open prices of stock are used for creating new variables which are used as inputs to the prediction models with target as Closing Price for the day. The models are evaluated using standard strategic indicators and the market news with respect to market sentiment.

Bitcoin data for the last 2 years from yahoo finance has been used as our data resource for:

1) Benchmarking different machine learning algorithms and using test statistics to check for reliability in their predictions.
2) Find out the significance of financial indicators and their accuracy in forecasting the trend.
3) Figure out how sentiment of the market affects the price of data: Use a sentimental analysis model and compare the results with ongoing trends.

Introduction

Stock Market and Cryptocurrencies are both volatile, dynamic and unpredictable. Predicting trends in data is a challenging task which depends on various parameters which includes: Global economic conditions, Market sentiment, Private sector growth or an unprecedented calamity or recession; as a recent example: Covid-19. People can buy and sell currencies, commodities, digital currencies or any equities in exchange for a stake in the market. Thus, reducing losses and maximizing profits by accurately predicting the price of a stock in near future is the aim of traders, brokers and companies all over the world. The data depends on inflation, commodities, real estate and foreign equities, incidental Transactions, Demographics, Trends, Market Sentiment, news, Earnings per share (EPS), supply and demand [1]. Predicting stock prices is a challenging task even after using deep learning. To extract data from and to form patterns and relationships from such huge and non-linear data is quite difficult. Firstly, In order to check the reliability of different machine learning algorithms we benchmark and determine using statistical methods to verify the validity of the particular model. Secondly, we determine the accuracy of technical indicators in predicting the stock value: some of them include- moving averages, moving average convergence divergence (MACD), relative strength index (RSI) and Bollinder band. Lastly, we want to determine the influence of market mentality and emotion on stock prices and the global market. Sentiment analysis using NLP and textblob makes use of the current financial news to determine the change in trend.

Related Work

Most of the previous work in this area use classical algorithms like linear regression (LR) [3], K-Nearest neighbour (KNN), relative strength index(RSI), Moving average (MA), Moving Average Convergence / Divergence (MACD) [5] and also using some linear models like Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) [6], for predicting stock prices. Recent work shows that stock market prediction can be enhanced using machine learning. Techniques such as Support Vector Machine (SVM), Random Forest (RF) and some techniques based on neural networks such as Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and deep neural networks like Long Short Term Memory (LSTM) also have shown promising results [2] [7].

These are good approximators and are able to find the input and output relationship of a very large complex dataset.

Data

The historical data for the Bitcoin market value has been collected from Yahoo Finance [8]. The dataset currently includes 5 years of data.

The data contains information about the stock such as High, Low, Open, Close, Adjacent close and Volume. Only the day-wise closing price of the stock has been extracted as a target variable.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-05-02	451.933014	452.445007	441.776001	444.669006	444.669006	9.212700e+07
1	2016-05-03	444.726990	451.096985	442.617004	450.303986	450.303986	5.936640e+07
2	2016-05-04	450.183014	450.377991	445.630005	446.721985	446.721985	5.040730e+07
3	2016-05-05	446.710999	448.506012	445.882996	447.976013	447.976013	5.044080e+07
4	2016-05-06	447.941986	461.375000	447.067993	459.602997	459.602997	7.279680e+07
...	...	...	...	...	...	...	...
1822	2021-04-28	55036.636719	56227.207031	53887.917969	54824.703125	54824.703125	4.800057e+10
1823	2021-04-29	54858.089844	55115.843750	52418.027344	53555.109375	53555.109375	4.608893e+10
1824	2021-04-30	53568.664063	57900.718750	53129.601563	57750.175781	57750.175781	5.239593e+10
1825	2021-05-01	57714.664063	58448.339844	57052.273438	57828.050781	57828.050781	4.283643e+10
1826	2021-05-02	NaN	NaN	NaN	NaN	NaN	NaN

1827 rows × 7 columns

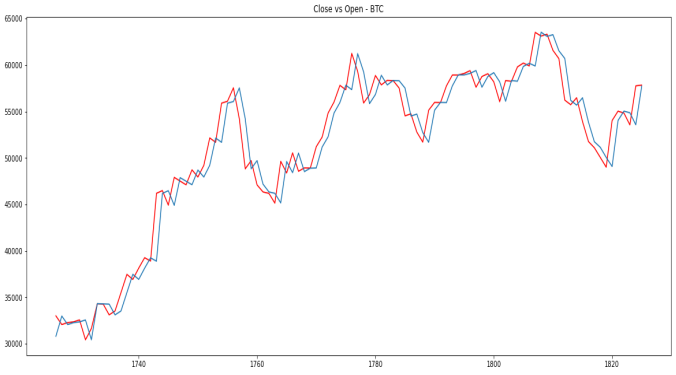
Statistics of the dataset that is used for training and testing are as follows: There is a total of 5 Years of data available. For Testing, we used 20% of total data available.

Description of Data

Date is a categorical data while open, close, high, low, adj close and volume are all real numbers and discrete.

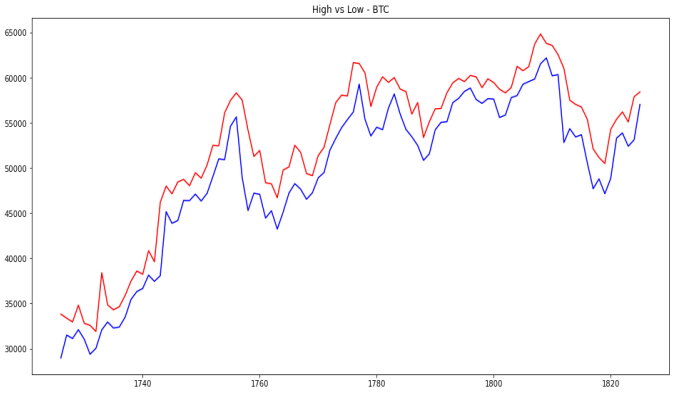
Open data gives us the opening value of the bitcoin on a certain day. Close data gives us the closing value of the bitcoin on a certain day. Comparing the opening and closing data gives us valuable information about the rise or fall in the value of the Bitcoin for a certain day. This data is visualized in the following graph.

Close vs Open stock prices with respect to Date:



High gives us the value of the highest rate the stock had during a particular day. Similarly, low gives us the lower value of the stock during a particular day. Visualization of these two features gives us an almost similar plot to that of 'Close vs Open' plot. This is because high and low are the max bounds of the day, thus open and close will always be between these two ranges for any specific day. Although the two look similar, it can easily be seen that the prices fluctuate a lot during any given day.

High vs Low stock prices with respect to Date:



Candlestick Visualization for each day trends:

Green marker represents that the close price was higher than open price. Red marker represents that the close price was lower than the open price. Candlestick represents the range of price of that day as well as the general trend followed. This is a highly useful representation as it integrates high, low as well as open, close with respect to the date into one single representation.

Candlestick Visualization:

Download the complete data description at Link.

Methodology

Moving Average:

MA is a simple technical analysis tool to smooth out the price and remove the noise (short term price fluctuations). They can be tailored to any short or long term time frame. The advantage of MA is that it helps traders identify trend directions and the tentative formation of resistance and support lines.

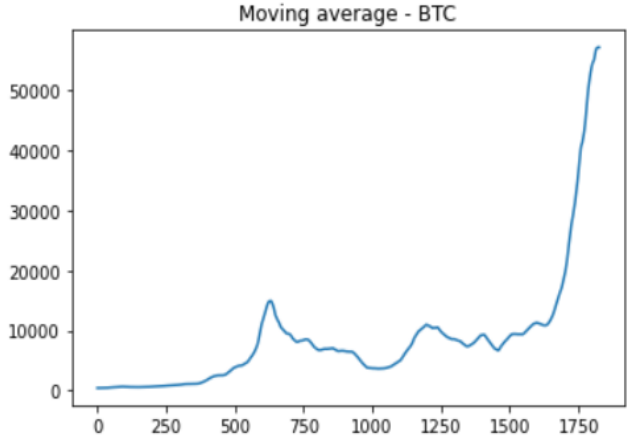
We calculated the Simple moving average using rolling means

in python pandas.[9]

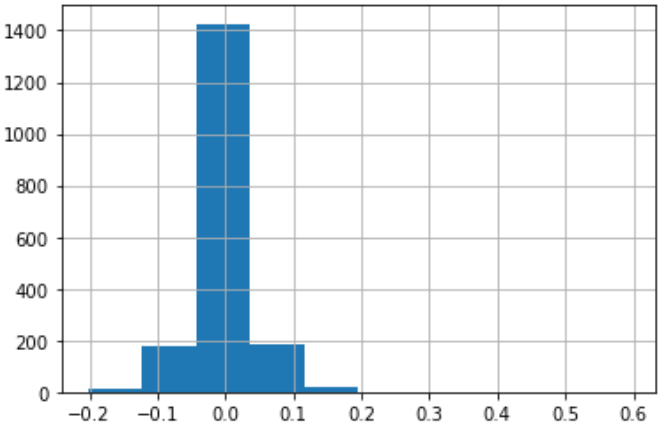
The graph shows a rising trend in the closing price after some fluctuations.

Moving Average:

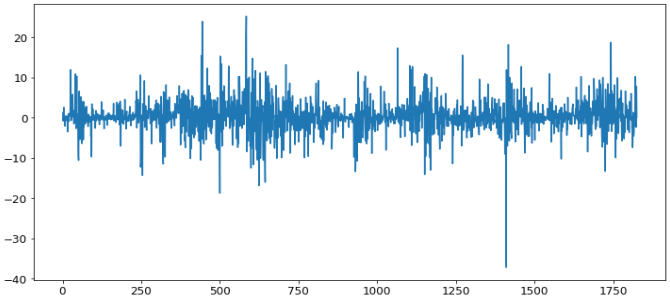
Exploratory Data Analysis:



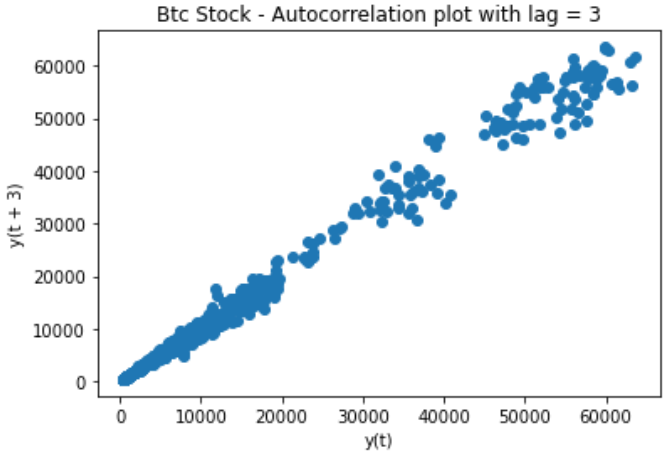
Daily return: It is calculated by shifting the price back by 1 day(daily lag). If we buy and sell a Bitcoin on the same day we will make a loss. As data is more inclined to the negative side.



Percentage change: denotes the percentage of difference in two consecutive days. We see that average is from 10 to -5 with few spikes.



Autocorrelation: tells us about how the past data(with 3 days lags) correlates with the present day close price. This data shows linearity and strong autocorrelation with closing price. Therefore, we can apply time series models for predicting the price with good accuracy.



Linear Regression:

Linear regression is the analysis of separate variables to define a single relationship and is a useful measure for technical and quantitative analysis in financial markets.

Investors and traders who use charts recognize the ups and downs of price printed horizontally from day-to-day or week-to-week, depending on the evaluated time frame. The different market approaches are what make linear regression analysis so attractive.

Plotting stock prices along a normal distribution—bell curve—can allow traders to see when a stock is overbought or oversold. Using linear regression, a trader can identify key price points—entry price, stop-loss price, and exit prices. A stock's price and time period determine the system parameters for linear regression, making the method universally applicable. [10]

For finding the Linear Regression, we have used the following package:

```
from sklearn.linear_model import LinearRegression
```

Mathematically linear regression is expressed as follows

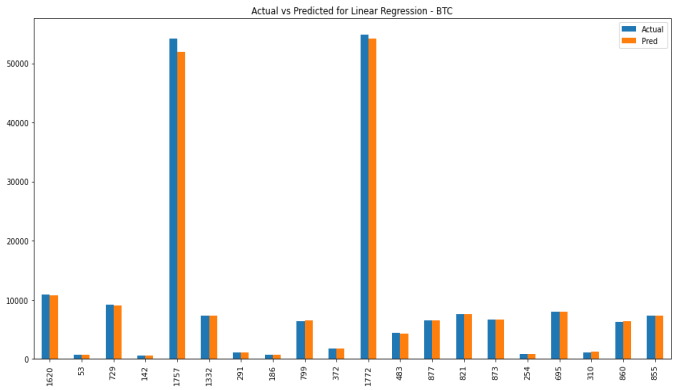
```
Prediction(w, x) = w0 + w1.x1 + ... + wp.xp
```

where the weights (w0 to wp) of the features are such values that they give us the closest prediction to the actual output i.e. have least error. We do this by getting the arg-min of error.

In our case, the closing value of the market is going to be the feature to be predicted. While the rest of the features are fitted as inputs to the regression model.

Hyperparameters we used for linear regression were the following: fit_intercept=True, normalize=False, copy_X=True, n_jobs=None, positive=False.

Actual observations vs Predictions of Linear Regression:



KNN

The kNN algorithm is a non-parametric algorithm that can be used for either classification or regression. Non-parametric means that it makes no assumption about the underlying data or its distribution. It is one of the simplest Machine Learning algorithms, and has applications in a variety of fields, ranging from the healthcare industry, to the finance industry. [12]

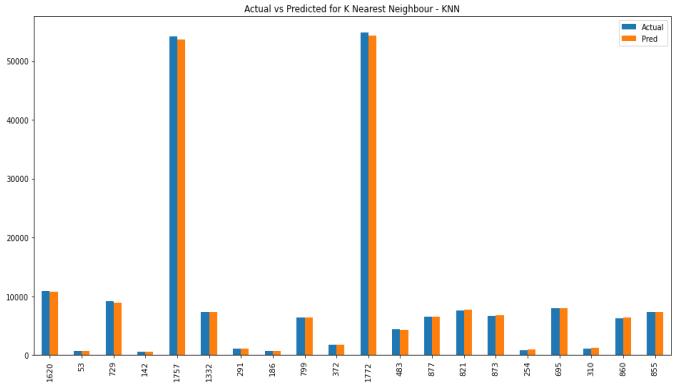
For each data point, the algorithm finds the k closest observations, and then classifies the data point to the majority. Usually, the k closest observations are defined as the ones with the smallest Euclidean distance to the data point under consideration.

For finding the nearest neighbor, we have used the following package:

`from sklearn.neighbors import KNeighborsRegressor`

In our case we have used the following hyperparameters: n_neighbors=3, weights='uniform', algorithm='auto'.

Actual observations vs Predicted for KNN:



SVM:

Support Vector Machines are one of the best binary classifiers. They create a decision boundary such that most points in one category fall on one side of the boundary while most points in the other category fall on the other side of the boundary. Consider an n-dimensional feature vector x = (X1,...,Xn) [14]. We can define a linear boundary (hyperplane) as,

$\beta 0 + \beta 1X1 + ... + \beta nXn = \beta 0 + {}^{n}\sum_{i=1} \beta iXi = 0$

Then elements in one category will be such that the sum is greater than 0, while elements in the other category will have the sum be less than 0. With labeled,

$\beta 0 + {}^{n}\sum_{i=1} \beta iXi = y$, where y is the label.

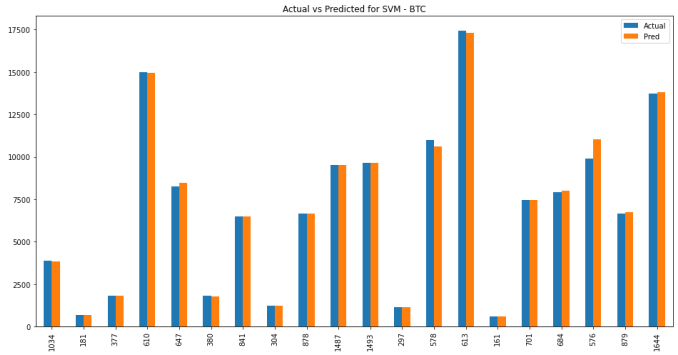
The closing value of the market is going to be the feature to be predicted. While the rest of the features are fitted as inputs to the SVM model.

For finding the nearest neighbor, we have used the following package:

`from sklearn.svm import SVR`

We have used the following hyperparameters since they were giving us the best results: kernel='linear', degree=3, gamma='scale'.

Actual observations vs Predicted values for SVM:



Gradient Boosting Regressor:

Decision trees are used as the weak learners in gradient boosting. Decision Tree solves the problem of machine learning by transforming the data into tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label. The loss function is generally the squared error (particularly for regression problems). The loss function needs to be differentiable. [18]

Also like linear regression we have concepts of residuals in Gradient Boosting Regression as well. Gradient boosting Regression calculates the difference between the current prediction and the known correct target value.

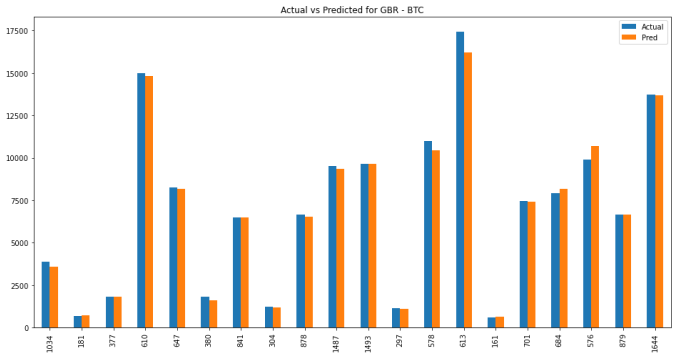
This difference is called residual. After that Gradient boosting Regression trains a weak model that maps features to that residual. This residual predicted by a weak model is added to the existing model input and thus this process nudges the model towards the correct target. Repeating this step again and again improves the overall model prediction. Also it should be noted that Gradient boosting regression is used to predict continuous values. [18]

For our GBR case, we have used the following package:

`sklearn.ensemble.GradientBoostingRegressor`

We have used the following hyperparameters since they were giving us the best results: n_estimators=100, learning_rate=0.1, max_depth=2, random_state=2.

Actual observations vs Predicted values for GBR:



Extreme Gradient Boosting:

Gradient boosting is a process to convert weak learners to strong learners, in an iterative fashion. The name XGBoost refers to the engineering goal to push the limit of computational resources for boosted tree algorithms. [16]

Boosting is an ensemble technique in which new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. The ensemble technique uses the tree ensemble model which is a set of classification and regression trees (CART). The ensemble approach is used because a single CART, usually, does not have a strong predictive power. By using a set of CART (i.e. a tree ensemble model) a sum of the predictions of multiple trees is considered. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction.

The objective of the XGBoost model is given as:

$Obj = L + \Omega$,

Where, L is the loss function which controls the predictive power, and Ω is regularization component which controls simplicity and overfitting. [17]

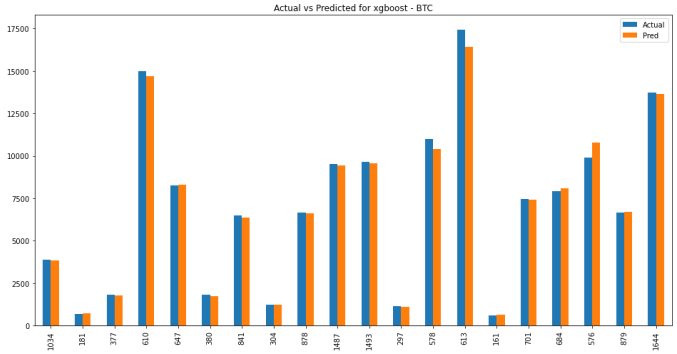
The loss function (L) which needs to be optimized can be Root Mean Squared Error for regression, Logloss for binary classification, or mlogloss for multi-class classification. The regularization component (Ω) is dependent on the number of leaves and the prediction score assigned to the leaves in the tree ensemble model. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. The Gradient boosting algorithm supports both regression and classification predictive modeling problems. [17]

For Extreme Gradient Boosting, we have used the following library:

```
from xgboost import XGBRegressor
```

The hyperparameters we set to get optimal outcomes are as follows: objective="reg:linear", random_state=42.

Actual observations vs Predicted values for XBG:



Long Short-Term Memory:

Long-Short-Term Memory Recurrent Neural Network belongs to the family of deep learning algorithms. It is a recurrent network because of the feedback connections in its architecture. It has an advantage over traditional neural networks due to its capability to process the entire sequence of data. Its architecture comprises the cell, input gate, output gate and forget gate.

The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. The cell of the model is responsible for keeping track of the dependencies between the elements in the input sequence. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell, and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit.

LSTM Networks are popularly used on time-series data for classification, processing, and making predictions. The reason for its popularity in time-series application is that there can be several lags of unknown duration between important events in a time series. [19]

For Extreme Gradient Boosting, we have used the following library: Keras

Hyperparameters used to get optimal outcome: Layer 1 - LSTM with 50 units, Layer 2 - LSTM with 50 units, Layer 3 - Dense with 1 unit.

Final loss we got after implementing LSTM was: 0.0102.

Accuracy Metric:

K Fold Cross Validation: We have calculated the accuracy of our predictions with respect to the test data using the K-Folds cross-validation from sklearn library with parameters: n_splits=10 and random_state=100. K-Folds divides the dataset into k consecutive folds where each of the fold is used

as validation set and other k-1 as training set.[11]

Model	Accuracy
Linear Regression	99.938%
K Nearest Neighbour	99.763%
SVM	99.933%
GBR	99.606%
XGB	99.917%
LSTM	98.988%

Test Measures:

Model	RMSE (L2)	R2
Linear Regression	205.551	0.9996
K Nearest Neighbour	278.598	0.9993
SVM	218.535	0.9995
GBR	290.232	0.9992
XGB	306.705	0.9991
LSTM	1968.756	0.9901

R2: This provides indication of goodness of fit and how well the model predicted the outcome on test data. Also, known as coefficient of determination it is used to calculate the percentage of the variance of dependent variable which is explained by independent variable.

Here, n= Total samples, Yi = individual observed values

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

then the variability of the data set can be measured with two sums of squares formulas:

- The total sum of squares (proportional to the variance of the data):

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

- The sum of squares of residuals, also called the residual sum of squares:

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

The most general definition of the coefficient of determination is

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

[14]

MSE: It is the average squared difference between the estimated values and the actual value. It is used to determine the quality of prediction of our model.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

[15]

n = Total samples, Yi = individual observed values and $\hat{Y}i$ = predicted values

Sentimental Analysis:

To understand how market sentiment and emotion are related to the price of Bitcoin. We created a sentiment analysis model to extract the information about outgoing sentiment and verify if the sentiment correlates with market price.

We extracted the daily news on BTC using Google news API. We used data for the past 1 month.

```
df_sentiment                    #title is basically headline with link of news and time of publication
```

	title	link	published
0	Cryptocurrency Price Check: Bitcoin Falls, DeF...	https://www.thestreet.com/investing/cryptocurr...	Sat, 27 Mar 2021 07:00:00 GMT
1	Crypto Shadow Banking Explained and Why 12% Yi...	https://www.bloomberg.com/news/articles/2021-0...	Sat, 27 Mar 2021 07:00:00 GMT
2	Crypto Markets Rebound, Bitcoin Price Consolid...	https://news.bitcoin.com/crypto-markets-reboun...	Sat, 27 Mar 2021 07:00:00 GMT
3	A "disastrous direction of travel": Why bitcoi...	https://www.hedgeweek.com/2021/03/27/297851/di...	Sat, 27 Mar 2021 07:00:00 GMT
4	Wharton Professor Explains All the Buzz About ...	https://scitechdaily.com/wharton-professor-exp...	Sat, 27 Mar 2021 07:00:00 GMT
...	...	...	...
2864	Bitcoin Reclaims $52K After Clawing Back Losse...	https://dailyhodl.com/2021/04/26/bitcoin-recla...	Mon, 26 Apr 2021 07:00:00 GMT
2865	Why Time sees opportunity in Bitcoin for adver...	https://digiday.com/media/why-time-sees-opport...	Tue, 27 Apr 2021 07:00:00 GMT
2866	How to Buy Enjin (ENJ) Crypto Right Now • Benz...	https://www.benzinga.com/money/how-to-buy-enji...	Mon, 26 Apr 2021 07:00:00 GMT
2867	CI GAM Launches Ethereum Mutual Fund - Finance...	https://www.financemagnates.com/cryptocurrency...	Mon, 26 Apr 2021 07:00:00 GMT
2868	Palantir Co-Founder Joe Lonsdale Says Bitcoin ...	https://www.benzinga.com/markets/cryptocurrenc...	Mon, 26 Apr 2021 07:00:00 GMT

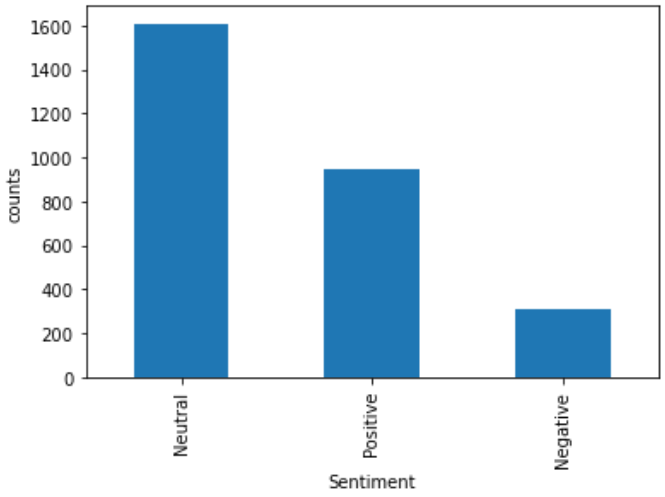
2869 rows × 3 columns

NLP: TextBlob library

TextBlob is a NLP based library. We are using it to calculate the polarity or sentiment of our data to understand market emotion as most of the market buys or sells on the basis of market sentiment.

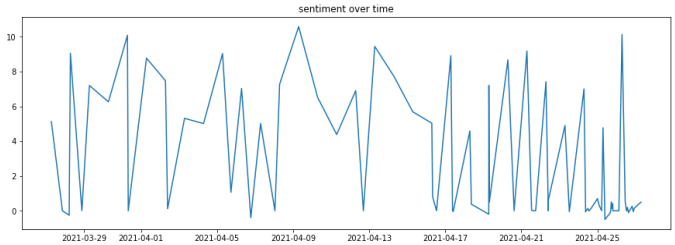
Textblob provides an estimate of polarity, subjectivity, intensity and confidence using NaiveBayesAnalyzer.

Total count sentiment in the data segregated into positive, negative and neutral comments.

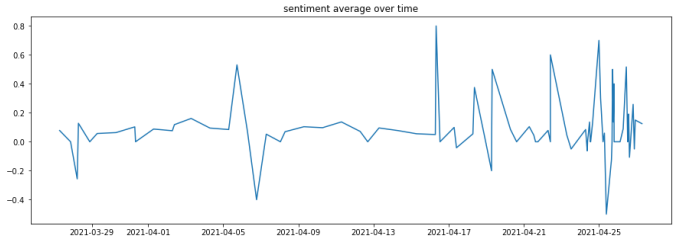


We observed positive sentiment in a greater quantity which correlates to the growing price trend of Bitcoin.

Average sentiment over the period of 1 month:



The average consolidates between 0.5 to -0.25 with a lot of fluctuations in the last few days. This may be due to the recent hike of Bitcoin prices in the last few days.



Technical Indicators:

MACD:

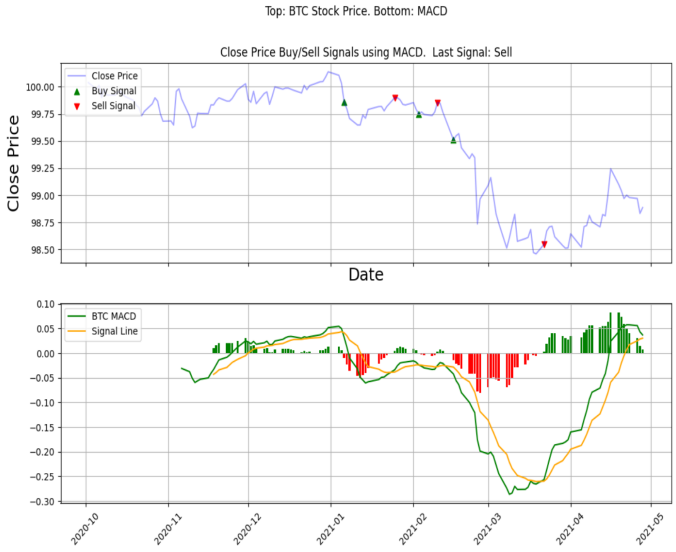
The MACD indicator is one of the most popular technical oscillator indicators

MACD helps us understand the relationship between the moving averages. Convergent is when the lines move closer to each other and divergence is when the lines move away from each other. The lines here are the moving averages.

MACD is a trend-following momentum indicator. It can help us assess the relationship between two moving averages of prices

Sell Signal: The cross over: When the MACD line is below the signal line.

Buy Signal: The cross over: When the MACD line is above the signal line



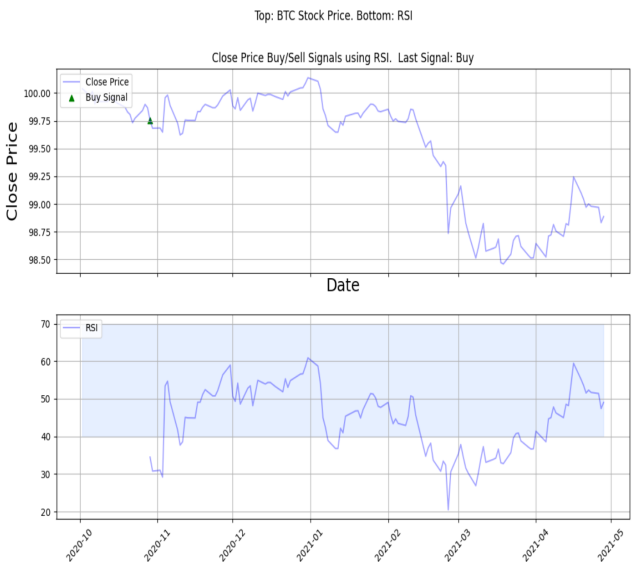
RSI:

RSI stands for Relative Strength Index. It's a widely used technical indicator and this is mainly due to its simplicity.

RSI indicator to measure the speed and change of price movements.

Essentially, overbought is when the price of a stock has increased quickly over a small period of time, implying that it is overbought.

The price of an overbought stock usually decreases in price.



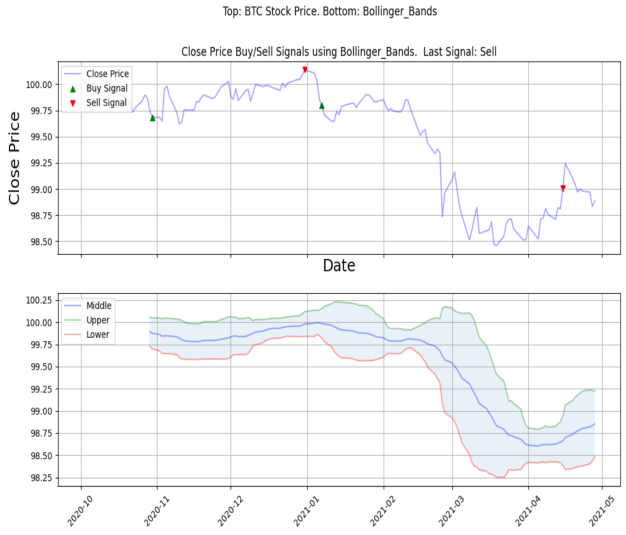
Bollinger band indicator:

The more volatile the stock prices, the wider the bands from the moving average.

Sell: As soon as the market price touches the upper Bollinger band

Buy: As soon as the market price touches the lower Bollinger band

Bollinger Band Indicator signals us to buy a stock but an external market event such as negative news can change the price of the stock.



Reference

[1]:https://www.investopedia.com/articles/basics/04/100804.asp

[2]:Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar. "Stock Closing Price Prediction using Machine Learning Techniques", Procedia Computer Science, Volume 167, 2020, Pages 599-606, ISSN 1877-0509.

[3]: Seber, George AF and Lee, Alan J. (2012) "Linear regression analysis." John Wiley & Sons 329.

[4]: Reichek, Nathaniel, and Richard B. Devereux. (1982) "Reliable estimation of peak left ventricular systolic pressure by M-mode echo graphic determined end-diastolic relative wall thickness: identification of severe valvular aortic stenosis in adult patients." American heart journal 103 (2) : 202-209.

[5]: Chong, Terence Tai-Leung, and Wing-Kam Ng. (2008) "Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30." Applied Economics Letters 15 (14) : 1111-1114.

[6]: Zhang, G. Peter. (2003) "Time series forecasting using a hybrid ARIMA and neural network mode." Neurocomputing 50 : 159-175.

[7]: Li, Lei, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. (2017) "Research on machine learning algorithms and feature extraction

for time series." IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC): 1-5.

[8]: Yahoo Finance - Business Finance Stock Market News, [Accessed on August 16,2018]

[9]:https://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-stocks.asp

[10]:https://www.investopedia.com/articles/trading/09/linear-regression-time-price.asp#:~:text=Key%20Takeaways,stock%20is%20overbought%20or%20oversold.

[11]:https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

[12]:https://datascienceplus.com/knn-classifier-to-predict-price-of-stock/

[13]:Predicting Stock Price Direction using Support Vector Machines. Author: Saahil Madge Advisor: Professor Swati Bhatt

[14]:https://en.wikipedia.org/wiki/Coefficient_of_determination

[15]:https://en.wikipedia.org/wiki/Mean_squared_error

[16]:https://towardsdatascience.com/forecasting-stock-prices-using-xgboost-a-detailed-walk-through-7817c1ff536a

[17]:https://blog.quantinsti.com/forecasting-markets-using-extreme-gradient-boosting-xgboost/

[18]:https://blog.paperspace.com/implementing-gradient-boosting-regression-python/

[19]:https://analyticsindiamag.com/hands-on-guide-to-lstm-recurrent-neural-network-for-stock-market-prediction/