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

Group 22:
Amandeep Singh Oberoi
Amit Nikam

About Project

1. Benchmarking different machine learning algorithms and using test statistics to check for reliability in their predictions.
2. Find out the significance of financial standard strategic indicator like MACD,RSI, BOLLINGERS.
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 using NLP.

About DATA

- The historical data for the Bitcoin market value has been collected from Yahoo Finance.
- The data contains information about the stock such as Date, High, Low, Open, Close, Adjacent close and Volume.
- ‘Date’ is a categorical data.
- Open, close, high, low, adj. close and volume are all real numbers.
- Data Statistics:
    - A total of 5 year of data is used.
    - For Training, first 80% data is used.
    - For Testing, data for remaining days after training data days is used.

	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


Close vs Open - BTC
65000
60000
55000
50000
45000
40000
35000
30000
1740
1760
1780
1800
1820


High vs Low - BTC
65000
60000
55000
50000
45000
40000
35000
30000
1740
1760
1780
1800
1820

Candlestick: one of the most important tool used in finance market. Red bar shows that open price was more than close price and green bar shows that open price was less that close price.

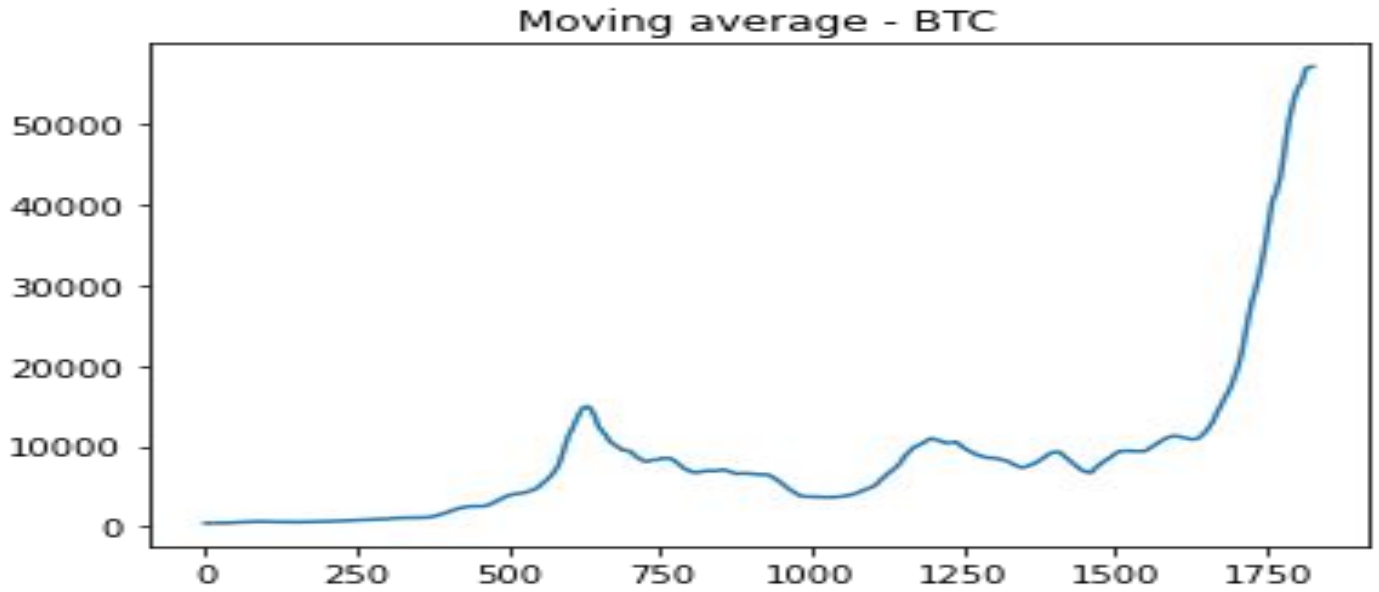
Libraries Used

Machine Learning:

- Sklearn
- Xgboost
- Keras
- Tensorflow
- Textblob
- Statsmodels
- Ta

Visualization and resources:

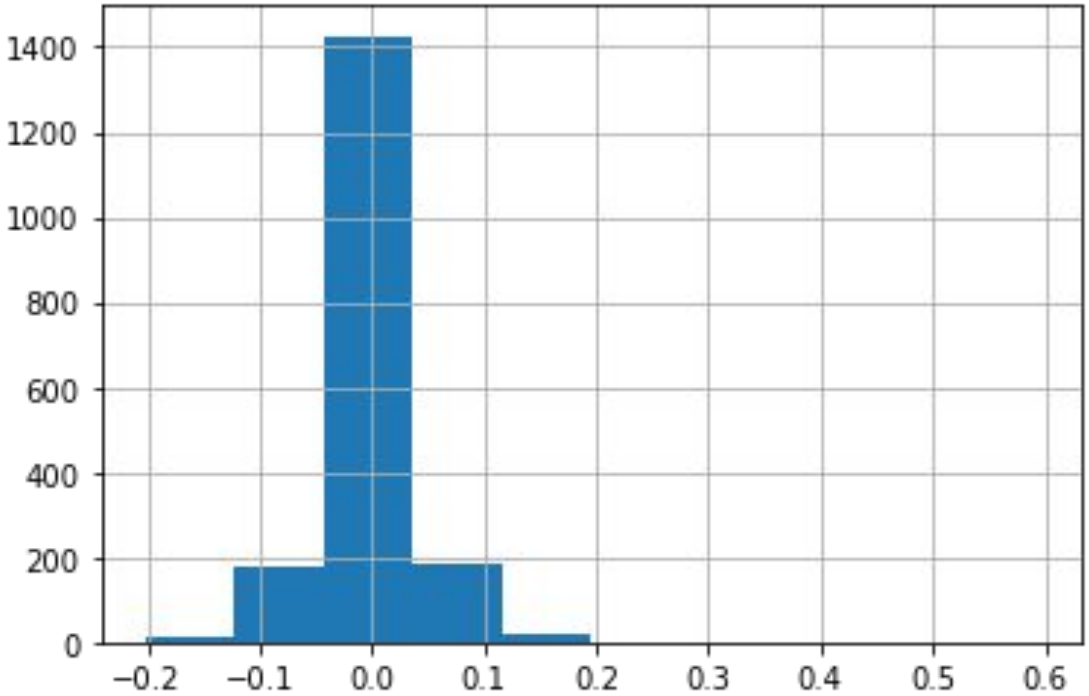
- Pandas
- Matplotlib
- Numpy
- Pygooglenews
- Math



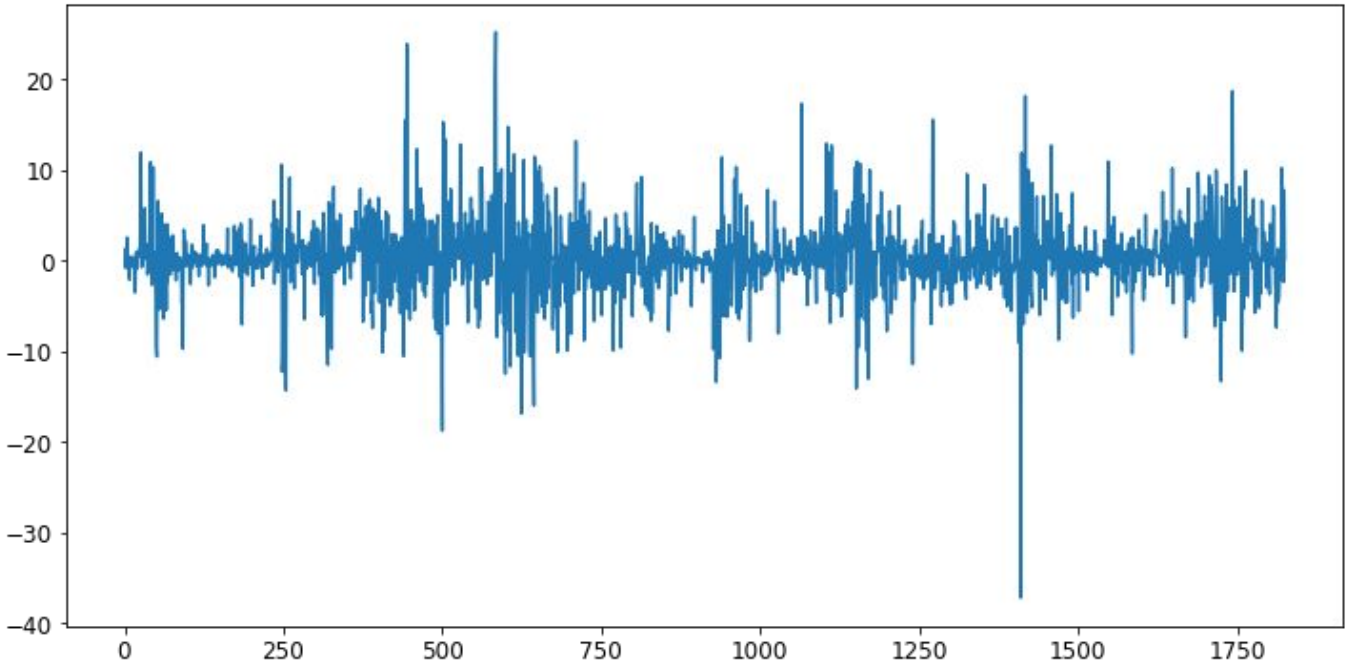
The **moving average** (MA) is a simple technical analysis tool that smooths out price data by creating a constantly updated average price.

It helps cut down the amount of "noise" on a price chart. Look at the direction of the moving average to get a basic idea of which way the price is moving.

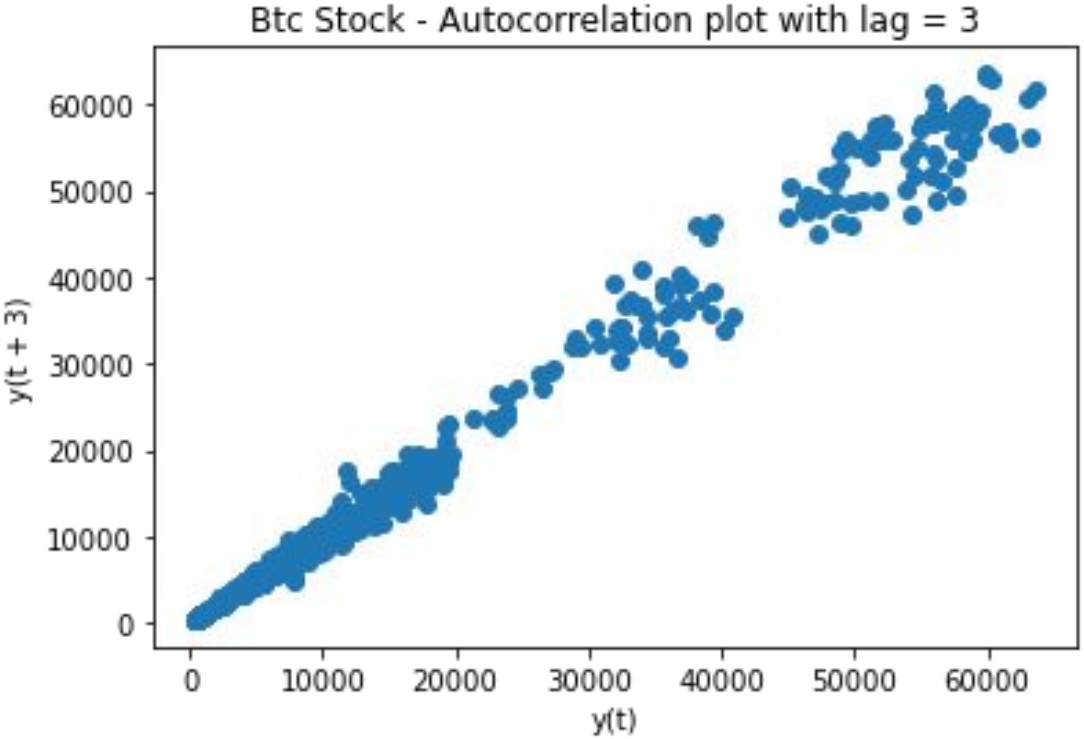
The data shows a positive trend with time.



Daily return is calculated by shifting the price back by 1 day(daily lag). We see that if we buy and sell a Bitcoin on same day we will make a loss. As data is more inclined to 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.



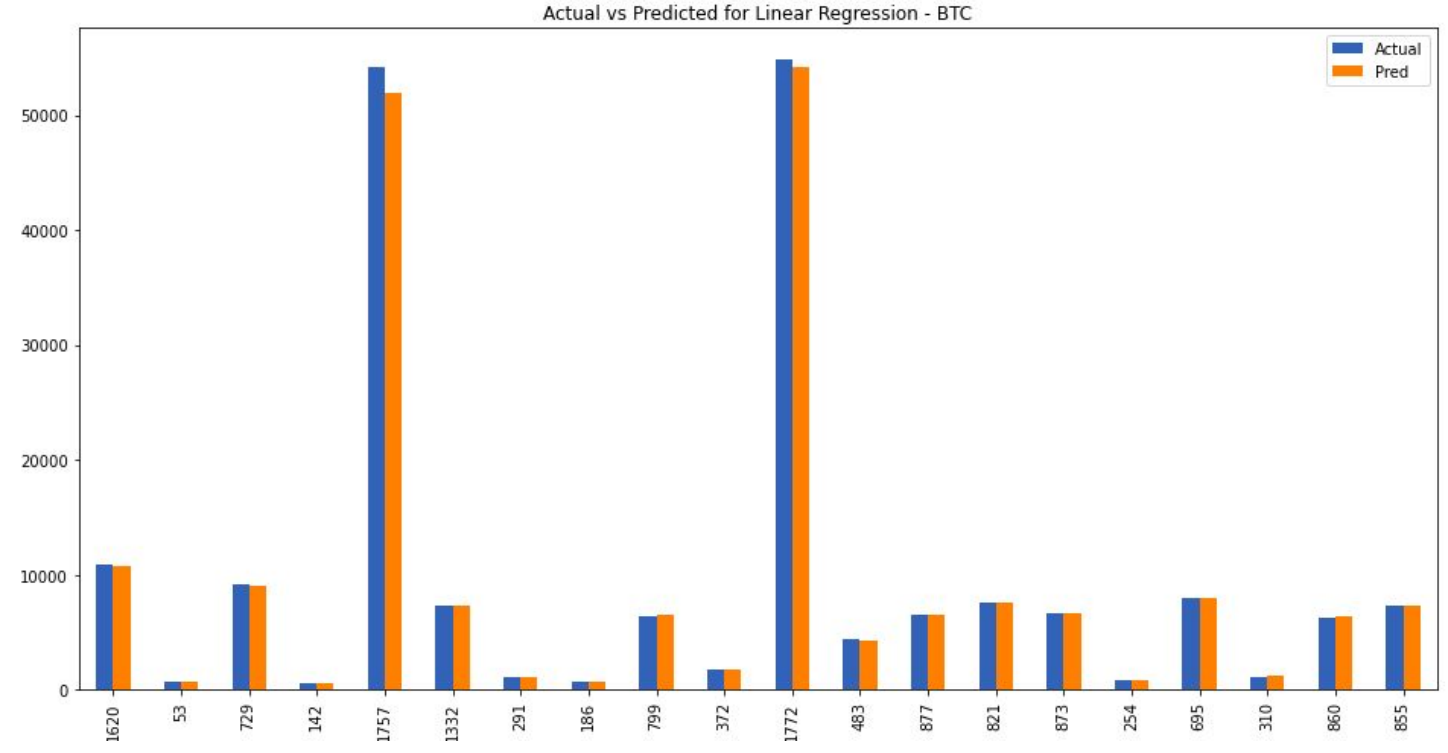
Autocorrelation tells us about how the past data(with 3 days lags) correlates with the present day close price. This data shows linearity and strong autocorrelation.

Benchmarks

Linear Regression

```
cross validation score mean:  99.95262764543614
```

```
plot_df=pd.DataFrame({'Actual':y_test,'Pred':y_pred})                    #last 20 days
plot_df.tail(20).plot(kind='bar',figsize=(16,8))
plt.title('Actual vs Predicted for Linear Regression - BTC')
plt.show()
```



Linear Regression:

Linear Regression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation

The equation has the form Y= a + bX, where Y is the dependent variable (that's the variable that goes on the Y axis), X is the independent variable (i.e. it is plotted on the X axis), b is the slope of the line and a is the y-intercept.

Linear Regression is susceptible to over-fitting but it can be avoided using some dimensionality reduction techniques, regularization (L1 and L2) techniques and cross-validation.

In our case, closing price of the market is the Y that is to be predicted. While other features (Open, High and Low) add to X.

We used the `sklearn.linear_model.LinearRegression` package to build our model.

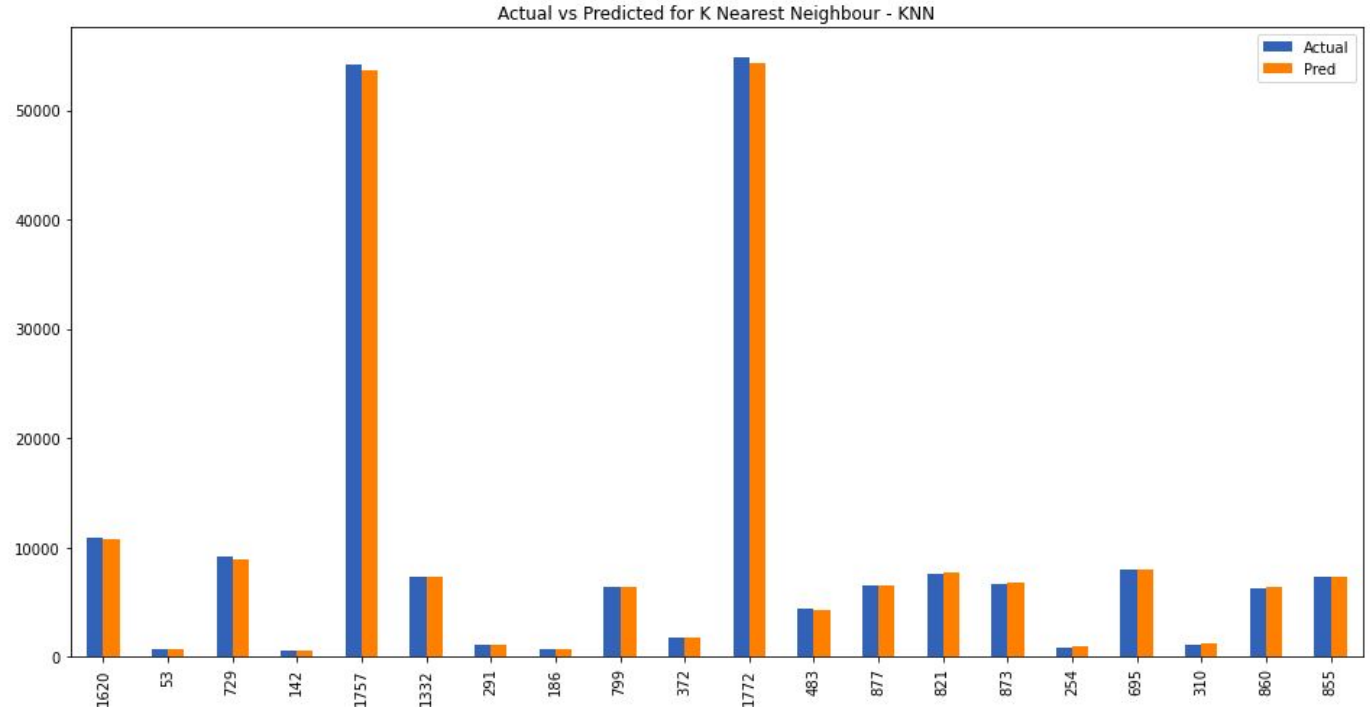
K Nearest Neighbour

```
cross validation score mean :  99.82805826817389
```

```
plot_knn_df=pd.DataFrame({'Actual':y_test,'Pred':y_knn_pred})
plot_knn_df.tail(20).plot(kind='bar',figsize=(16,8))

plt.title('Actual vs Predicted for K Nearest Neighbour - KNN')

plt.show()
```



K Nearest Regressor:

KNeighborsRegressor implements learning based on the nearest neighbors of each query point, where it is an integer value specified by the user

Nearest neighbors regression uses uniform weights and so can be advantageous to weight points such that nearby points contribute more to the regression than faraway points

It's biggest disadvantage the difficult for the algorithm to calculate distance with high dimensional data.

We used the `sklearn.neighbors.KNeighborsRegressor` library to implement this model.

For our problem, following hyper-parameters were used:

n_neighbors = 3

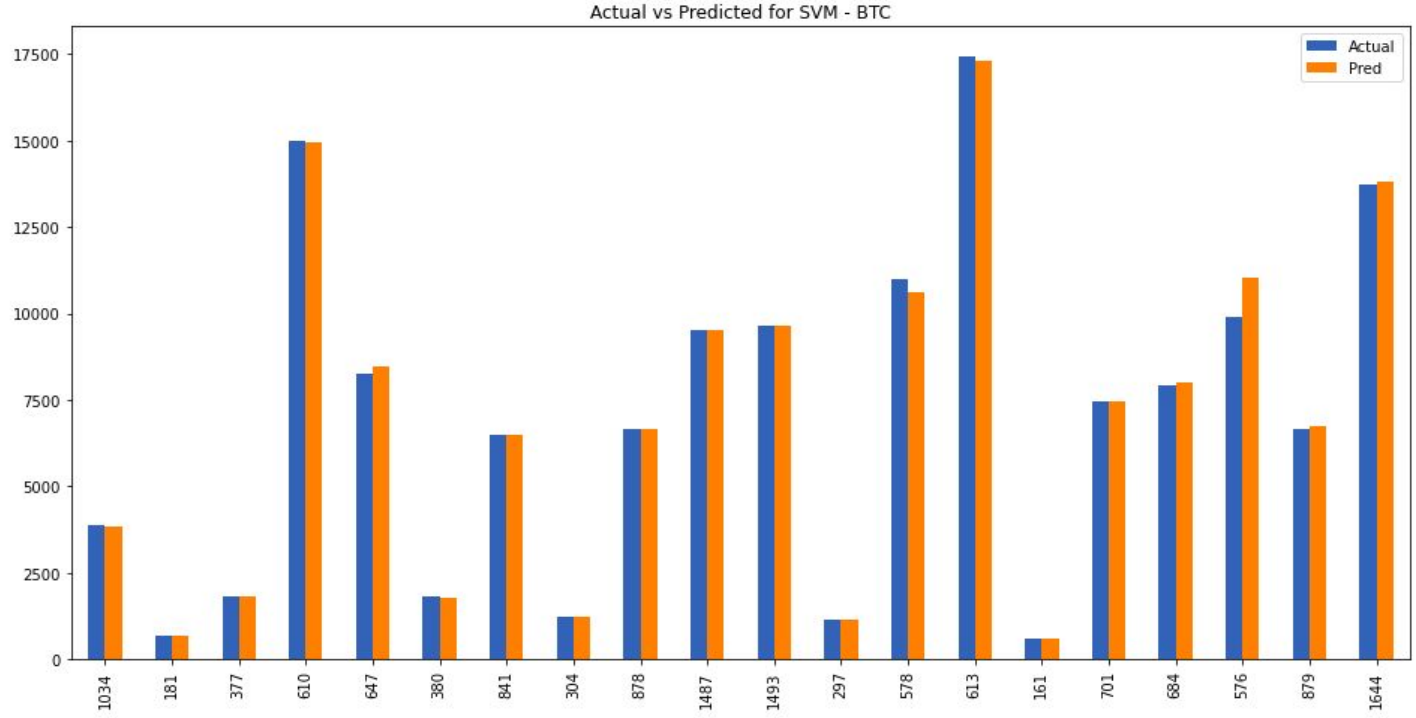
Support Vector Machine (SVM)

```
cross validation score mean :  99.94870990201272
```

```
plot_svm_df=pd.DataFrame({'Actual':y_test,'Pred':y_svm_pred})
plot_svm_df.head(20).plot(kind='bar',figsize=(16,8))

plt.title('Actual vs Predicted for SVM - BTC')

plt.show()
```



Support Vector Machines:

SVMs can 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.

A linear boundary (hyperplane) is defined as (Xi are features),
$\beta 0 + \beta 1 X1 + ... + \beta n Xn = \beta 0 + {}^{n}\sum_{i=1} \beta i Xi = 0$

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.

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.

With label ‘y’ -> $\beta 0 + {}^{n}\sum_{i=1} \beta i Xi = y$

We have implemented model using sklearn.svm.SVR package.

Hyperparameters:
kernel='linear'

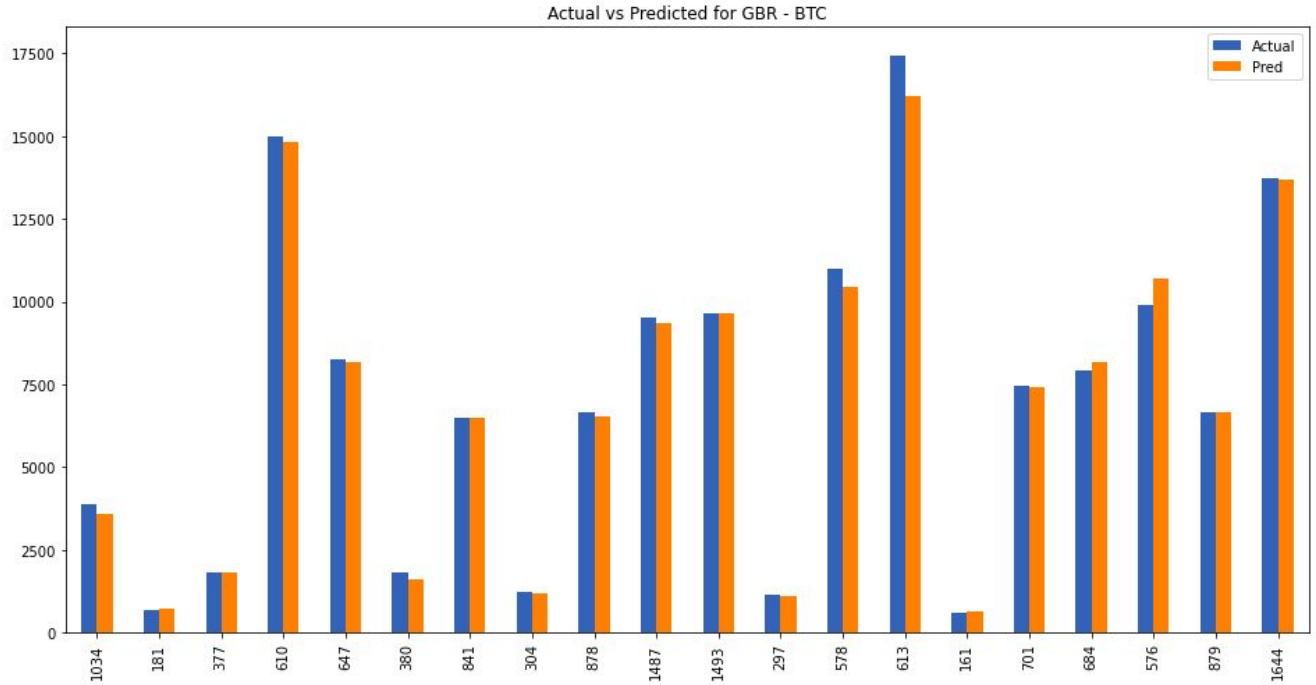
Gradient Boost Regression (GBR)

```
cross validation score mean  99.73510696464585
```

```
plot_svm_df=pd.DataFrame({'Actual':y_test,'Pred':y_GBR_predict})
plot_svm_df.head(20).plot(kind='bar',figsize=(16,8))

plt.title('Actual vs Predicted for GBR - BTC')

plt.show()
```



Gradient Boost Regression (GBR):

Boosting is a method of converting weak learners into strong learners.In boosting, each new tree is a fit on a modified version of the original data set.

Gradient Boosting trains many models in a gradual, additive and sequential manner.

Gradient boosting performs the identifying of the shortcomings of weak learners by using gradients in the loss function (y=ax+b+e , e needs a special mention as it is the error term).

For our model, we have used sklearn.ensemble.GradientBoostingRegressor as base library.

Hyperparameters:
n_estimators=100
learning_rate=0.1
max_depth=2
random_state=2

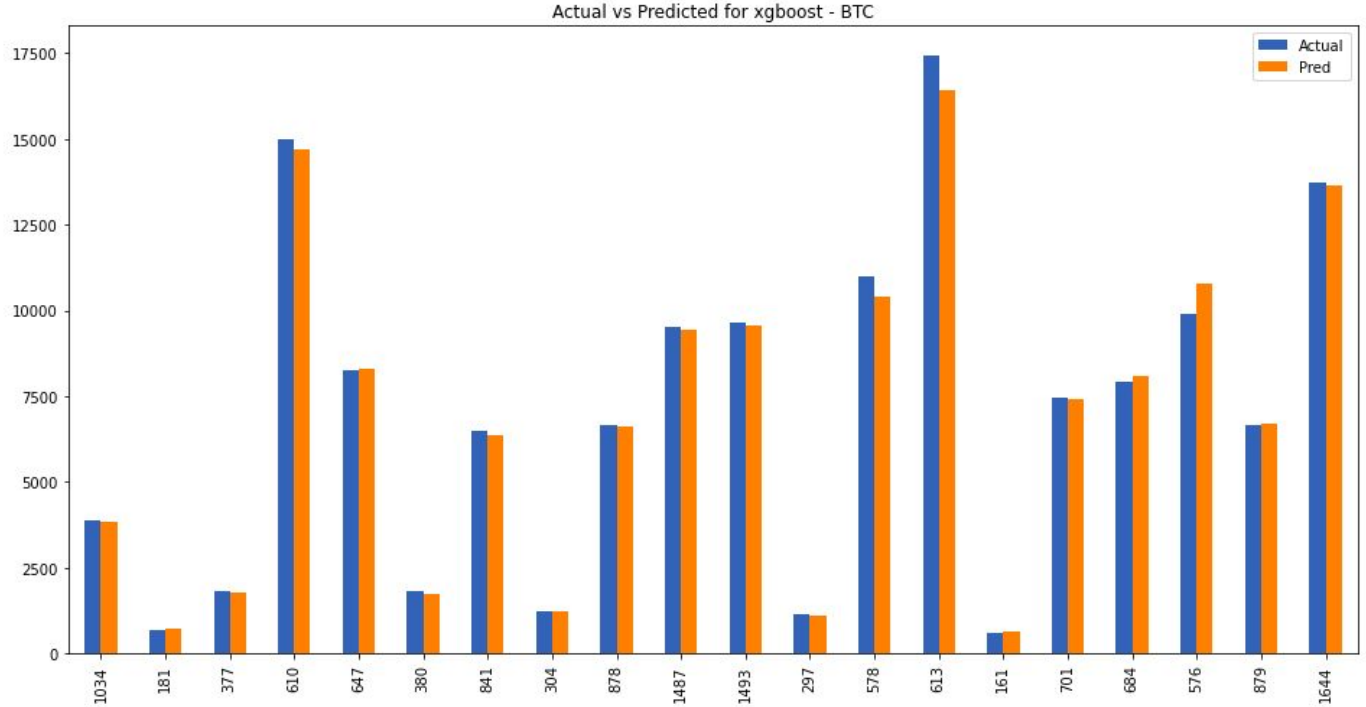
XGBoost (XGB)

```
cross validation score mean:  99.66938371730882
```

```
plot_svm_df=pd.DataFrame({'Actual':y_test,'Pred':y_xgb_predict})
plot_svm_df.head(20).plot(kind='bar',figsize=(16,8))

plt.title('Actual vs Predicted for xgboost - BTC')

plt.show()
```



XGBoost:

Extreme Gradient Boosting, or XGBoost for short, is an efficient open-source implementation of the gradient boosting algorithm

Two main reasons to use XGBoost are execution speed and model performance.

The objective function contains loss function and a regularization term.

The loss functions we used in XGBoost for regression problems is reg:linear.

Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods.

LTSM:

The Long Short-Term Memory network or LSTM network is a type of recurrent neural network designed to use sequential data such as time-series and mainly used in deep learning.

Long Short-Term Memory (LSTM) is a specialized RNN to mitigate the gradient vanishing problem. LSTMs can learn long-term dependencies using a mechanism called gates. These gates can learn what information in the sequence is important to keep or throw away.

LSTMs where they also handle noise, distributed representations, and continuous values. They also solve vanishing gradient problems and incapability of RNN to handle long term dependencies.

We used Sequential model with two LTSM layers and one dense layer. Loss and optimizers used were mean_squared_error and adadelta respectively with 50 epochs and batch size=1 as data was not too huge.

Achieved final loss: 0.0102

LTSM

```
Epoch 45/50
195/195 - 2s - loss: 3.7991e-06
Epoch 46/50
195/195 - 2s - loss: 3.7920e-06
Epoch 47/50
195/195 - 2s - loss: 3.7955e-06
Epoch 48/50
195/195 - 2s - loss: 3.7942e-06
Epoch 49/50
195/195 - 2s - loss: 3.7949e-06
Epoch 50/50
195/195 - 2s - loss: 3.7970e-06
```

Long short-term memory output.

Comparison of Error

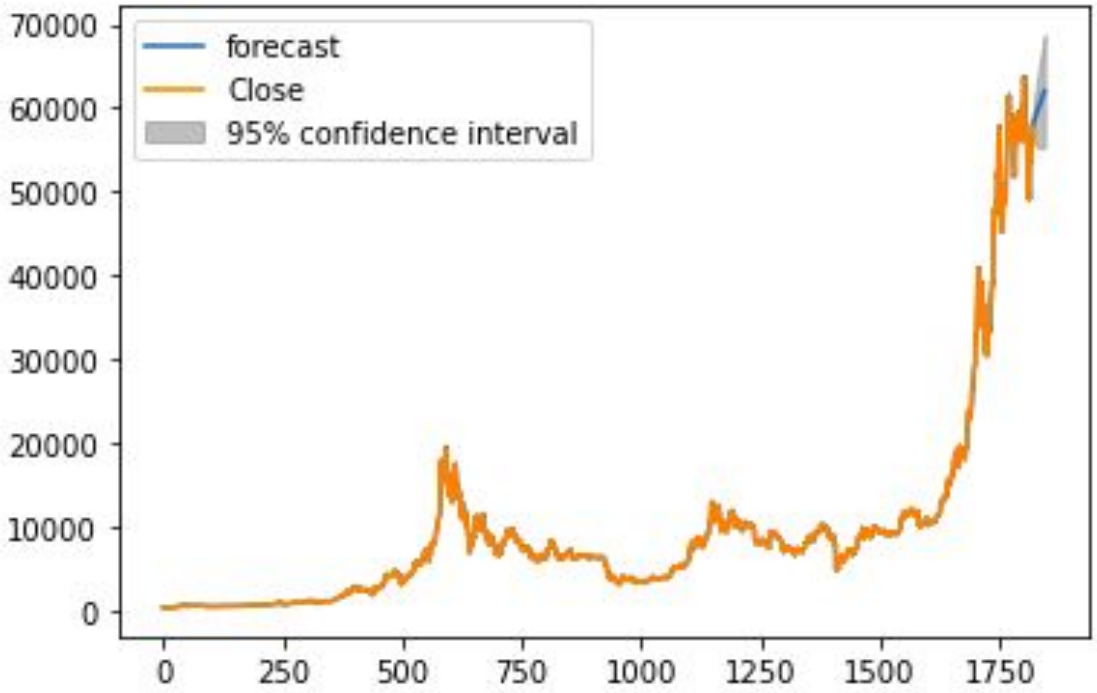
Model	RMSE (L2)	R2
Linear Regression	205.551	0.9996
K Nearest Neighbour	278.598	0.9993
SVM	218.535	0.9995
GBR	290.232	0.9992
XGB	306.705	0.9991
LSTM	1968.756	0.9901

Comparison of Accuracy

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

ARIMA

```
model_fit.plot_predict(start=2, end=len(df_arima)+25)
plt.show()
```



Arima:

Time-series forecasting models are the models that are capable to predict future values based on previously observed values. Time-series forecasting is widely used for non-stationary data

AutoRegressive Integrated Moving Average (ARIMA) model. ARIMA models are capable of capturing a suite of different standard temporal structures in time-series data.

- AR: < Auto Regressive > means that the model uses the dependent relationship between an observation and some predefined number of lagged observations (also known as “time lag” or “lag”).
- I:< Integrated > means that the model employs differencing of raw observations (e.g. it subtracts an observation from an observation at the previous time step) in order to make the time-series stationary.MA:
- *MA: < Moving Average >* means that the model exploits the relationship between the residual error and the observations.

Arima requires:

- p is the number of lag observations.
- d is the degree of differencing.
- q is the size/width of the moving average window.

For our model: The optimal model is: ARIMA(0, 2, 1).

Sentimental Analysis:

```
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

Google pynews API.

NLP: TextBlob library

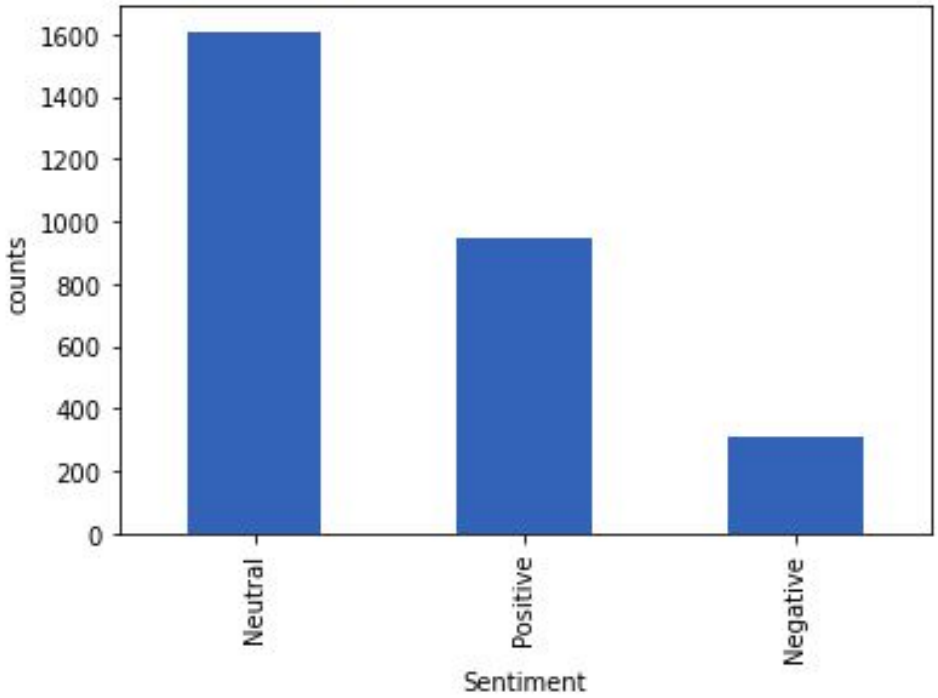
TextBlob is a NLP based library. We are using it to calculate the polarity or sentiment of our data to understand market emotion as most of the market buy or sell on the basis of market sentiment.

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

Textblob:

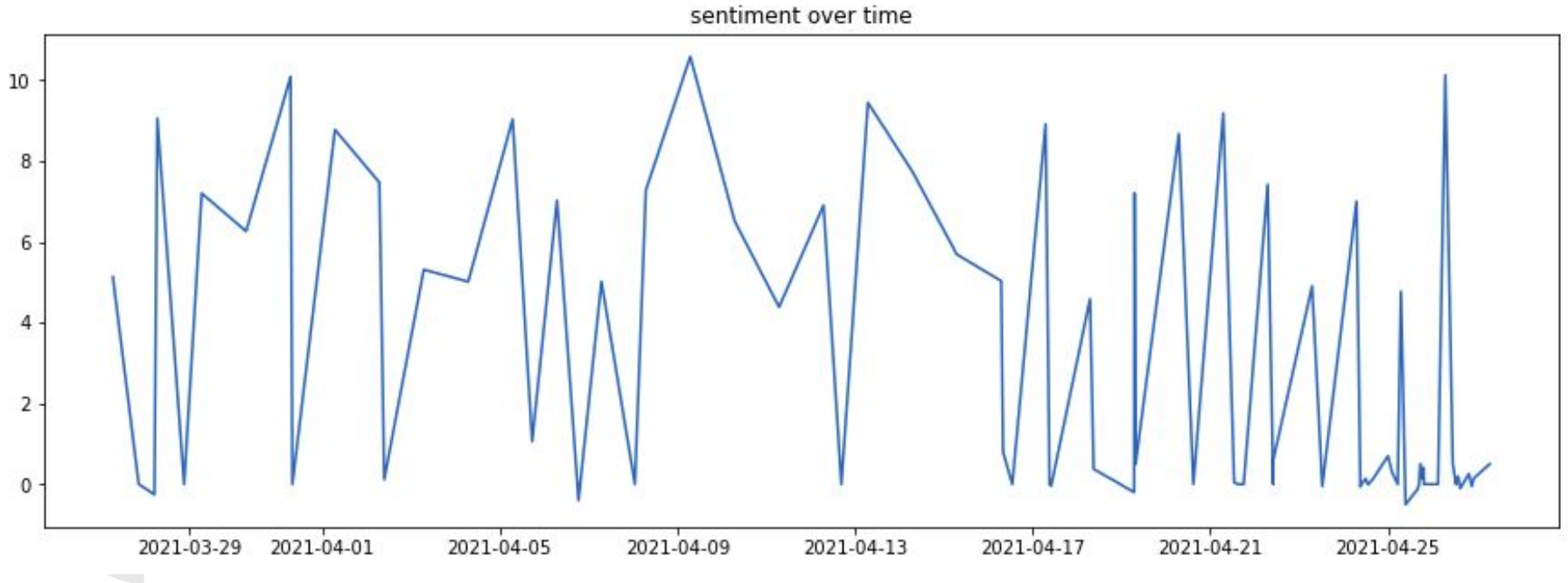
- Splits the sentence into works and use tokens and lemmas.
- Tags function return the list of tuples:(work, POS tag)
- These lemmas are passed to function textblob.en.sentiments.NaiveBayesAnalyzer
- This naive bayes algorithm is trained on movie reviews and is used to classify text into positive, negative and neutral.

```
df_sentiment['sentiment'].value_counts().plot(kind='bar')
plt.xlabel('Sentiment')
plt.ylabel('counts')
plt.show()
```



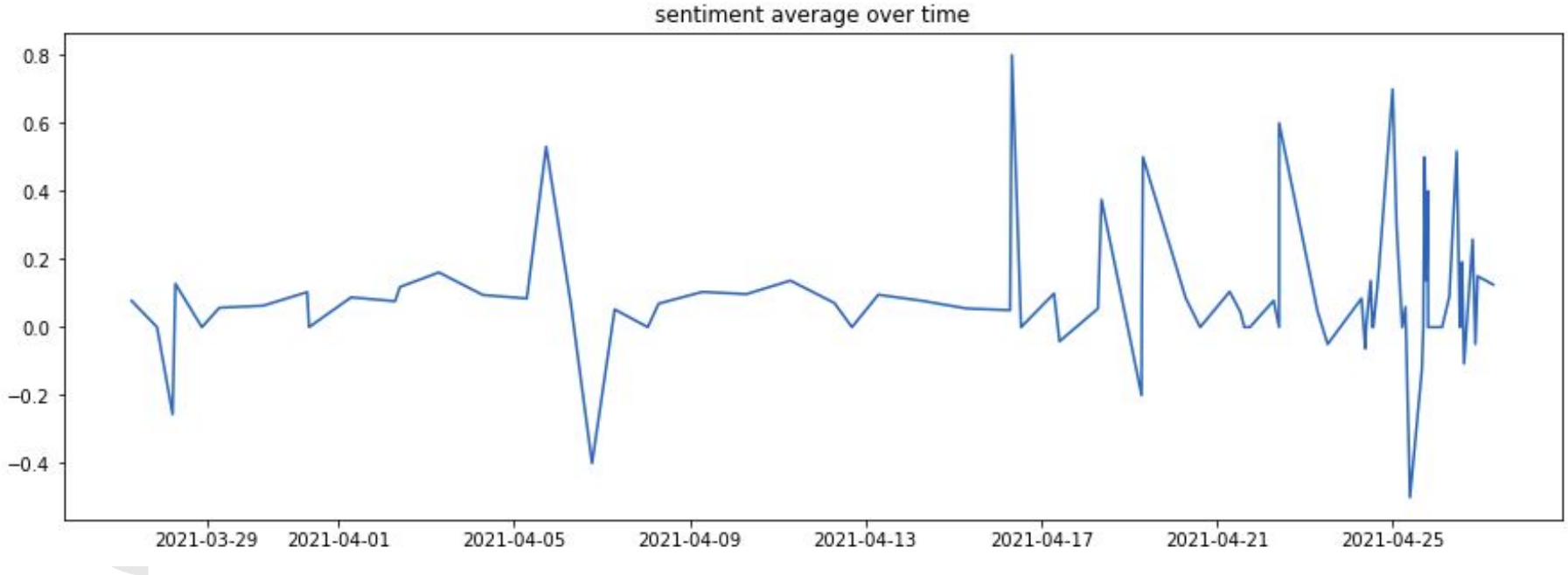
```
plt.figure(figsize=(15,5))                                    #sentiment over a month
plt.title('sentiment over time')
polarity=df_sentiment.groupby(['published']).sum()['polarity']
plt.plot(polarity.index,polarity)
```

```
[<matplotlib.lines.Line2D at 0x1a30bddffa0>]
```



```
plt.figure(figsize=(15,5))                                    #average sentiment over a month
plt.title('sentiment average over time')
polarity_count=df_sentiment.groupby(['published']).count()['polarity']
avg=polarity/polarity_count
plt.plot(avg.index,avg)
```

```
[<matplotlib.lines.Line2D at 0x1a30d47de50>]
```



Technical Indicators:

MACD:

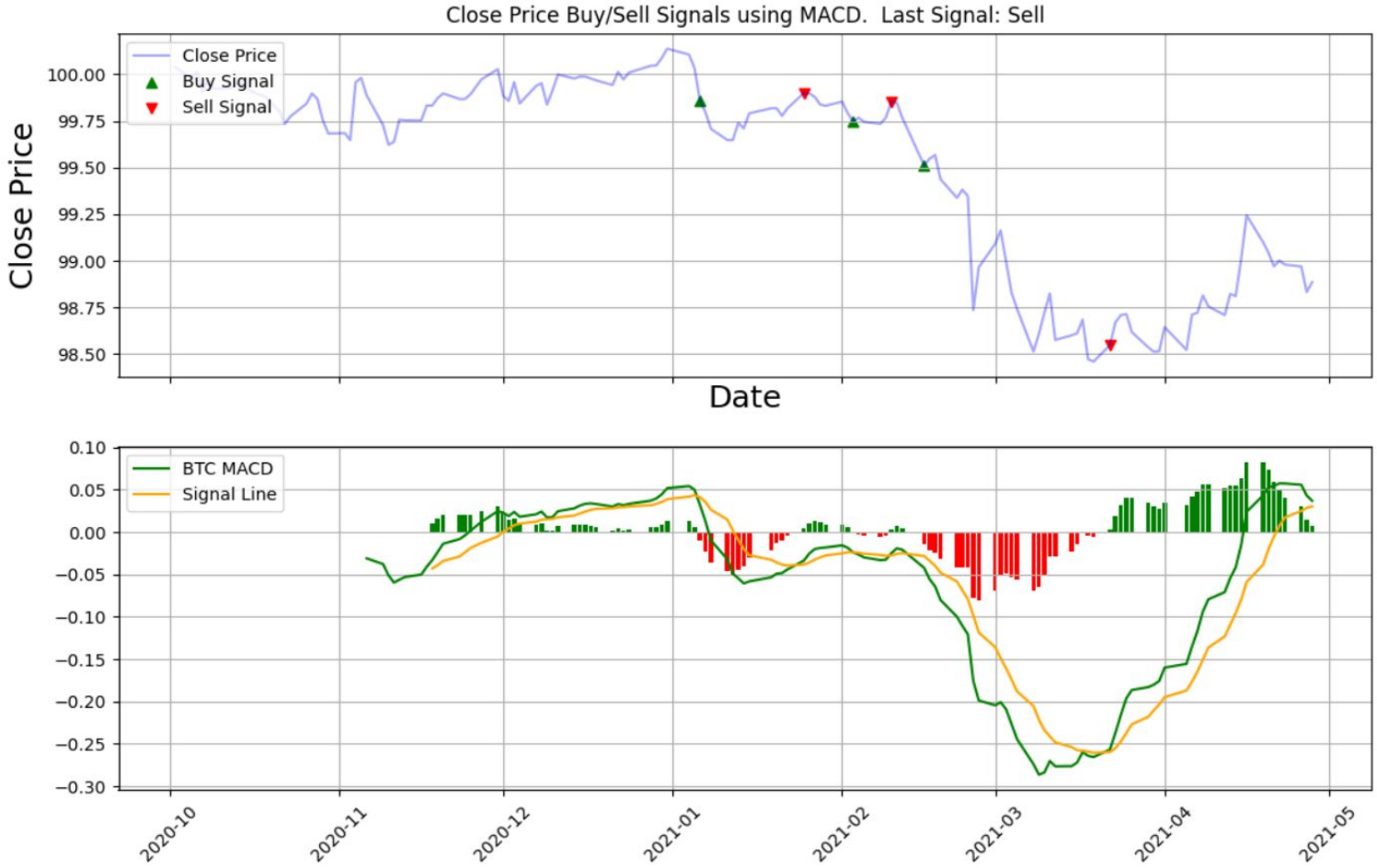
1. The MACD indicator is one of the most popular technical oscillator indicators

2. 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.

3. 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

Top: BTC Stock Price. Bottom: MACD



RSI:

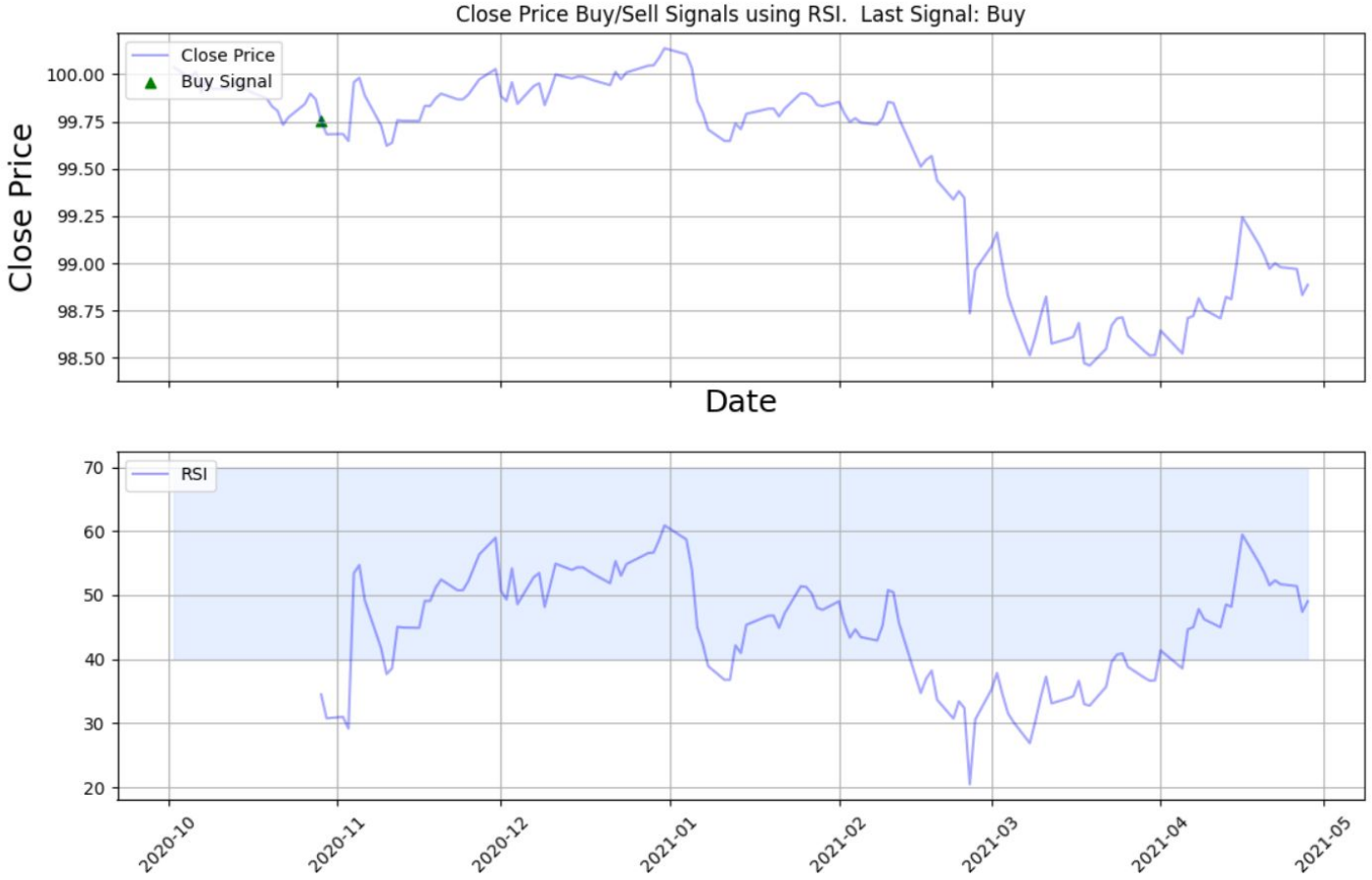
1. RSI stands for Relative Strength Index. It’s a widely used technical indicator and this is mainly due to its simplicity.

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

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

4. The price of an overbought stock usually decreases in price.

Top: BTC Stock Price. Bottom: RSI

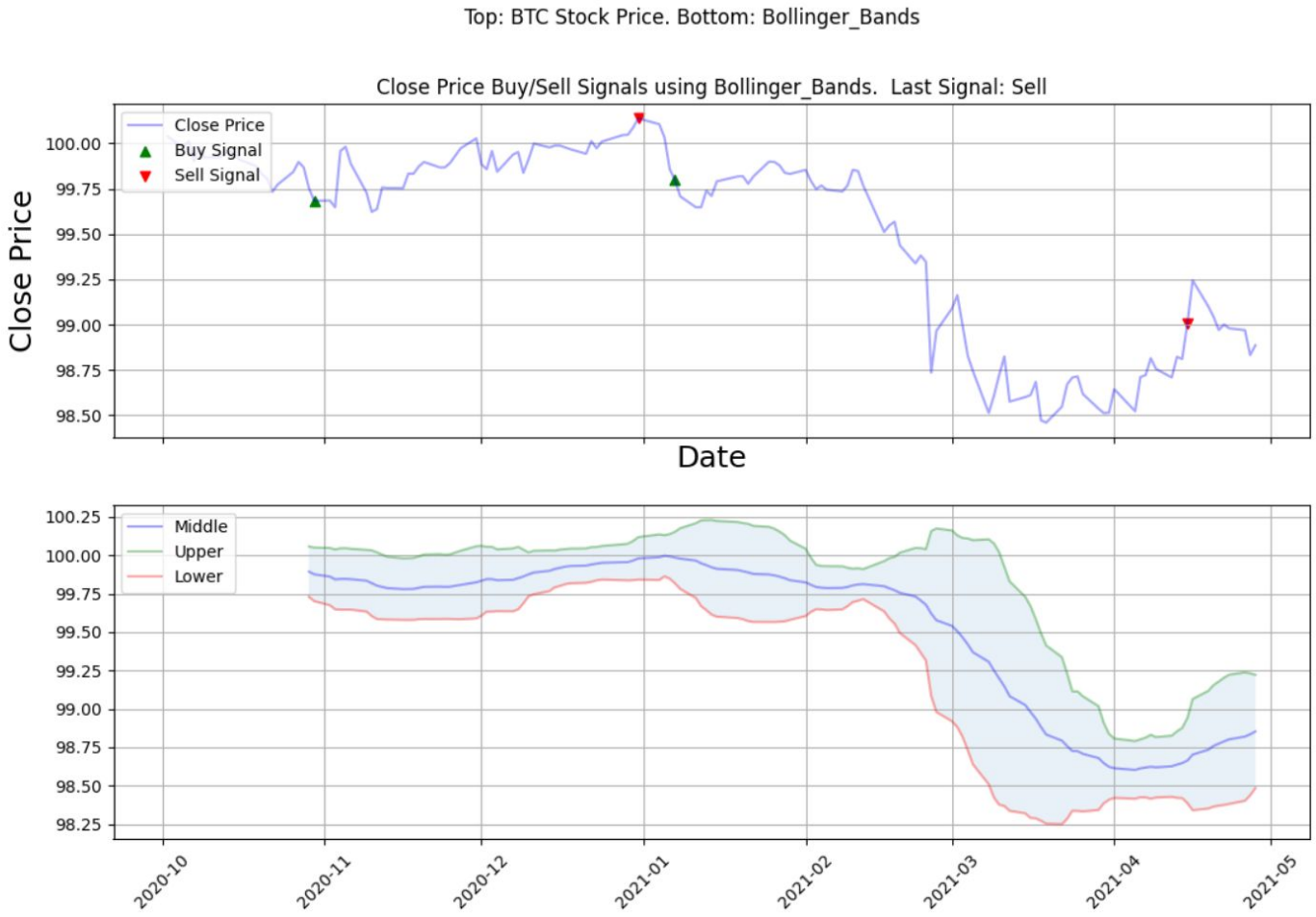


Bollinger Bands indicator:

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

2. **Sell**: As soon as the market price touches the upper Bollinger band
3. **Buy**: As soon as the market price touches the lower Bollinger band

4. 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.


Top: BTC Stock Price. Bottom: Bollinger_Bands
Close Price Buy/Sell Signals using Bollinger_Bands. Last Signal: Sell
Close Price
Buy Signal
Sell Signal
Close Price
100.00
99.75
99.50
99.25
99.00
98.75
98.50
Date
Middle
Upper
Lower
100.25
100.00
99.75
99.50
99.25
99.00
98.75
98.50
98.25
2020-10
2020-11
2020-12
2021-01
2021-02
2021-03
2021-04
2021-05

References:

[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

Thank You