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, Open and Close prices of stock are used for creating new variables which are used as inputs to the prediction models. The models are evaluated using standard strategic indicators and the market news with respect to market sentiment.

Bitcoin data for the last six months 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). Lastly, we want to determine the influence of market mentality and emotion on stock prices and the global market. Sentiment analysis using NLP 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 6 months of data from 9/27/2020 to 3/27/2021. Although for the final project, i.e. once all targeted models have been built, we would be using the dataset for the past 5-10 years.

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.

```
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-09-28	10771.641602	10949.123047	10716.676758	10721.327148	10721.327148	2.272037e+10
1	2020-09-29	10712.462891	10858.939453	10665.344727	10848.830078	10848.830078	2.045987e+10
2	2020-09-30	10845.411133	10856.528320	10689.670898	10787.618164	10787.618164	2.075962e+10
3	2020-10-01	10785.010742	10915.843750	10493.552734	10623.330078	10623.330078	2.717823e+10
4	2020-10-02	10624.390625	10662.813477	10440.311523	10585.164063	10585.164063	2.312784e+10
...	...	...	...	...	...	...	...
177	2021-03-24	54710.488281	57262.382813	52514.332031	52774.265625	52774.265625	7.056722e+10
178	2021-03-25	52726.746094	53392.386719	50856.570313	51704.160156	51704.160156	6.799981e+10
179	2021-03-26	51683.011719	55137.312500	51579.855469	55137.312500	55137.312500	5.665220e+10
180	2021-03-27	55137.566406	56568.214844	54242.910156	55973.511719	55973.511719	4.726654e+10
181	2021-03-28	55974.941406	56610.312500	55071.113281	55950.746094	55950.746094	4.768658e+10

182 rows × 7 columns

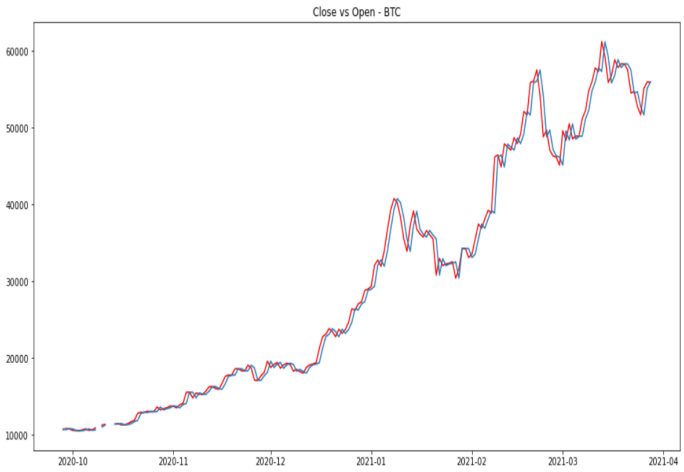
Statistics of the dataset that is used for training and testing are as follows: There is a total of 182 days of data available. For Training first 135 days of data from 9/27/2020 onwards is used. For Testing, data for remaining days after training data days is used.

Description of Data

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

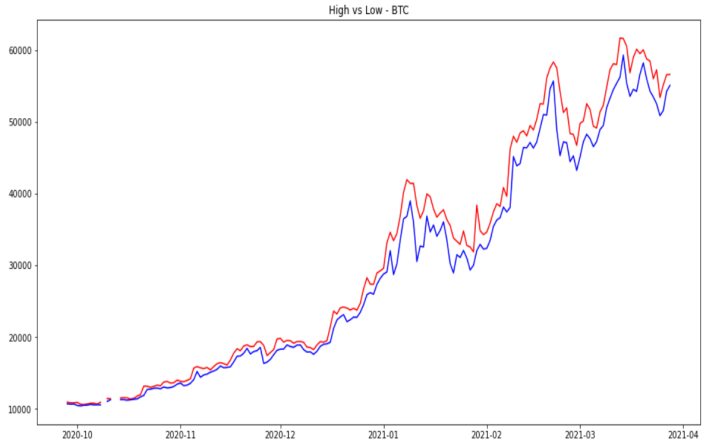
Open data gives us the opening value of the bitcoin on a certain day. Close data gives us the closing value of the bitcoin on a certain day. Comparing the opening and closing data gives us valuable information about the rise or fall in the value of the Bitcoin for a certain day. This data is visualized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

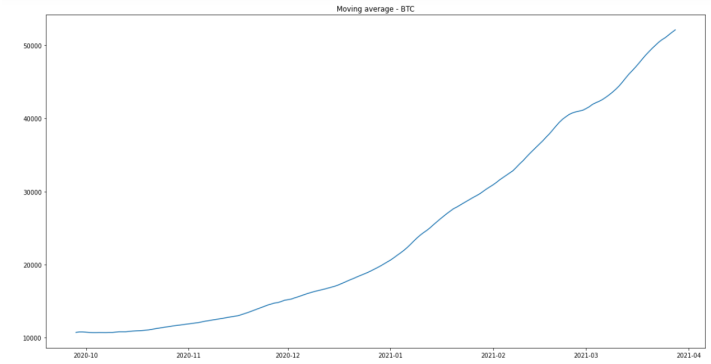
Methodology

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]

Moving Average:



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
```

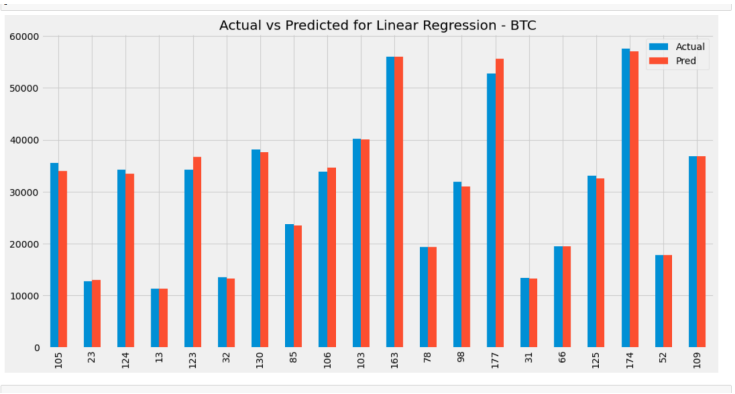
Mathematically linear regression is expressed as follows

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.

Actual observations vs Predictions of Linear Regression:



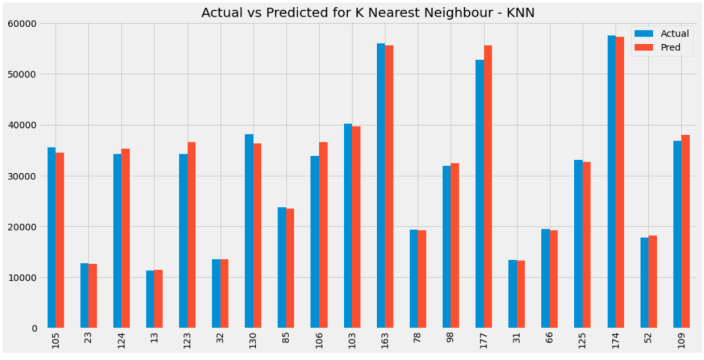
KNN

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.

In our case we have taken k = 3 to make three classifications of the data points.

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,

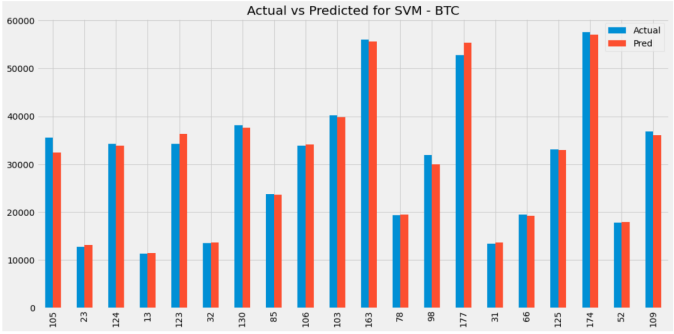
$\beta 0 + \beta 1X1 + ... + \beta nXn = \beta 0 + {}^{n}\sum_{i=1} \beta iXi = 0$

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.

We have used a **linear** kernel for 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 SVM model.

Actual observations vs Predicted values for SVM:



Accuracy Metric:

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]

Accuracy of Linear Regression : 99.45140056984066

Accuracy of KNN : 97.27282197928652

Accuracy of SVM : 99.51760568460489

Test Measures:

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

R2 measure for each model:

```
Linear R2:  0.9967615457168489
KNN R2:  0.9952195466292775
SVM R2:  0.9961194303943055
```

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

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

Mse measure for each model:

```
Linear Model Root mean square error 852.5939308346567
KNN Model Root mean square error 1035.8762948456213
SVM Model Root mean square error SVM 933.2996230456216
```

Pending Work

For final submission we plan to integrate: NN like RNN and CNN, LTSM, Random forest and produce the statistical metric. Secondly, we plan to add the influence of other technical indicators like MACD and RSI apart from MA. Lastly, we plan on adding sentimental analysis and benchmark all algorithms we use and deduce the best one with highest accuracy.

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