Bitcoin Price Prediction

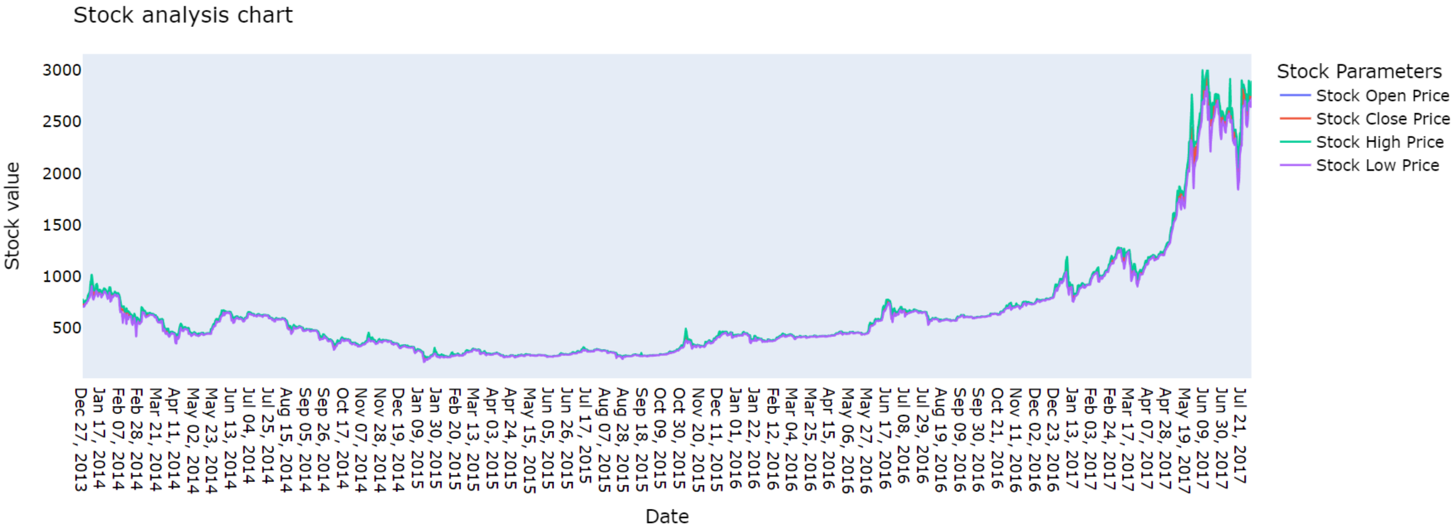
Kshitij Jaiswal B20CS028

***Abstract* – This paper reports my experience on building a regressor for predicting the bitcoin "Market Capital". On various dates from 28th April 2013 to 31st July 2017, it contained the bitcoin market opening price, closing price, the highest value of the bitcoin in a day, the lowest value of the bitcoin in a day and volume. Along with these, it also contained a column which had the Market Capital value of bitcoin for each day. Bitcoin price prediction is an integral part of stock price prediction in daily lives and is of immense importance to stock market**

I. Introduction

Bitcoin Price Prediction is an essential part of a Stock trader's life. For this dataset, we had to predict the Market Capital of Bitcoin based on Date and other features. This dataset helped me learn some very new kinds of regressors like Polynomial and LSTM Regressor models as it demands time series prediction i.e. just on the basis of previous few values of an attribute, we had to predict its value in the near future. I have predicted the value of "Market Cap" but just by changing the pred variable, this whole analysis can be done on any other variable.

The following graph shows some of the features of the Bitcoin Price Prediction Dataset -



Datasets

GAN : The file heart.csv is used as the dataset.

The dataset contains 1556 rows where each row represents the bitcoin market results for each day from 28-04-2013 to 31-07-2017. There were 242 entries with volume value equal to '-'. They were dropped and the dataset contained the latest market value at the top and hence the dataset was also reversed so as to get the latest market value at the bottom and the oldest at the top.

Furthermore, after dropping 242 rows (as mentioned above), there were only 1314 rows left, further split into training and testing datasets (70:30, shuffle=False).

II. Methodology

Overview

There are various regression algorithms present out of which I implemented the following -

- *Decision Tree Regressor*
- *Linear Regression*
- *XGboost Regressor*
- *Polynomial Regressor*
- *LSTM Regressor*

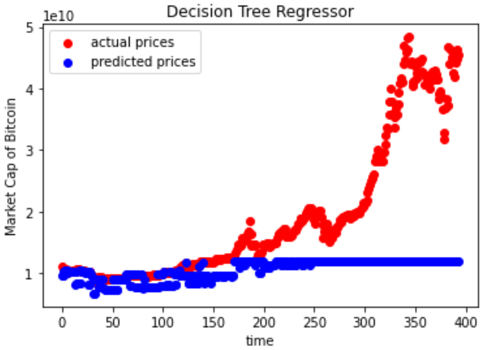
I used Sequential Feature Selector with forward = True and floating = False with k_features=2.

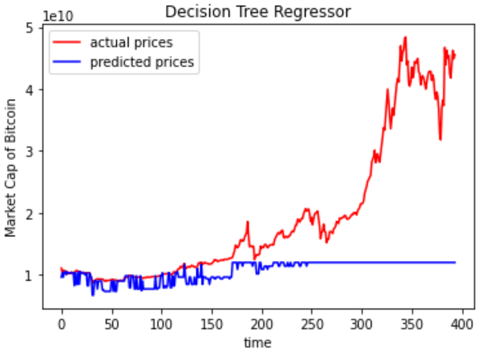
Exploring the dataset and pre-processing

On counting the number of NULL values in the train dataset , it was found that there are 242 rows where volume contained '-' values. The column type for 'volume' and 'Market Cap' were object and so the datatype of these columns was changed to 'integer64'. The column 'Date' was dropped as index serves the same purpose.

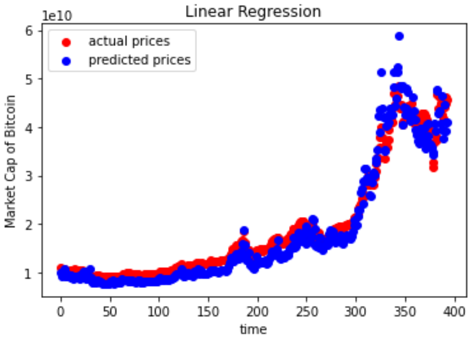
Implementation of regression algorithms

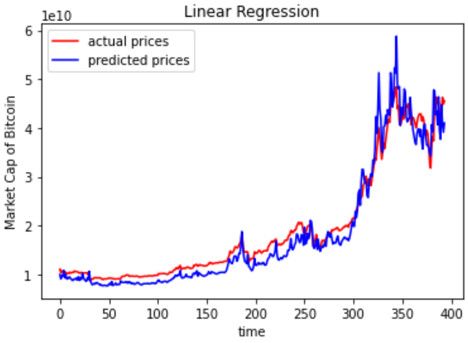
- *Decision Tree Regressor :* Decision Tree is primarily used for classification problems, however regressor using decision trees is built using multiple trees accounting for various features. It performed poorly on the bitcoin dataset.



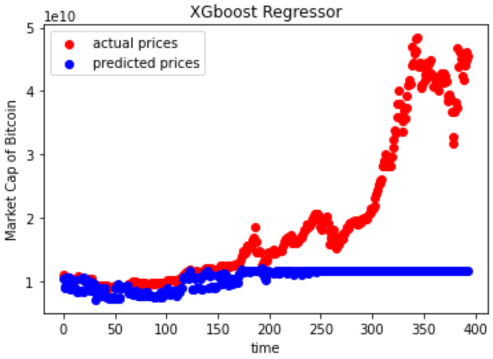


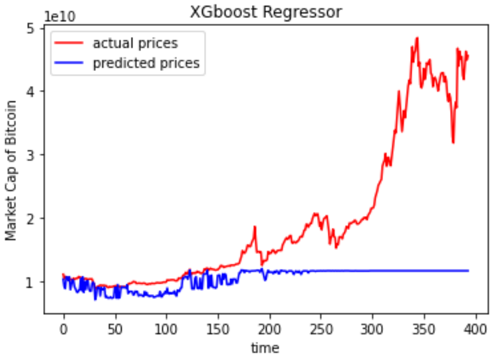
- *Linear Regression :* Linear Regression tries to fit a curve as close as possible by finding linear relationships in the data. It performed really nicely on the dataset.



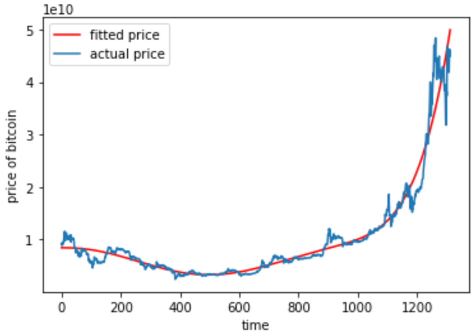


- *XGboost Regressor :* Adaboost Regressor is based on ensemble learning in which weights are reassigned to every sample data point which has a relatively large r2 score. It also performed poorly on the dataset.

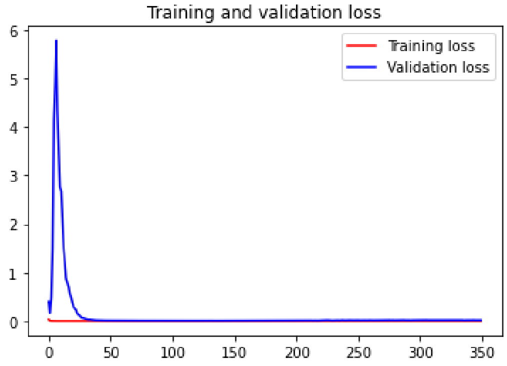


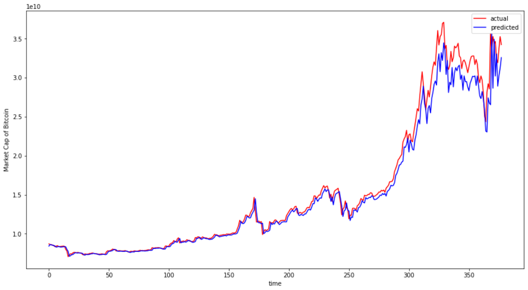


●*Polynomial Regressor :* Polynomial Regressor fits a polynomial curve on the given data with a degree specified by the user. I fit the entire dataset using polynomial regressor for better results. It performed well in the crude prediction.



- *LSTM Regressor :* Long Short Term Memory (LSTM) is a RNN (recurrent neural network) which helps stores feature values for future classification or regression. It was used for making the advanced prediction and was the best regressor for this problem.

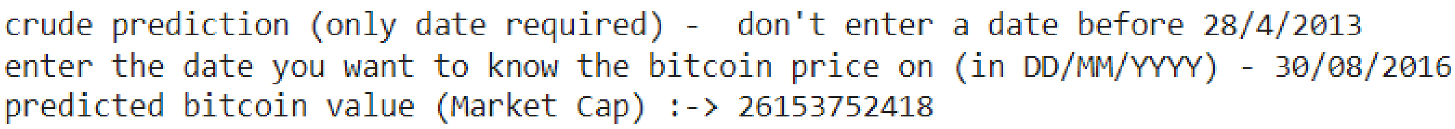




Feature Selection Technique : SFS

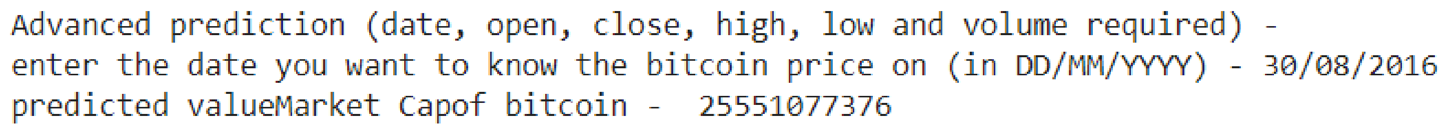
III.Regression on Input data Points

Crude Prediction - It was done using Polynomial regression which used only the date for prediction (hence crude)



```
crude prediction (only date required) -  don't enter a date before 28/4/2013
enter the date you want to know the bitcoin price on (in DD/MM/YYYY) - 30/08/2016
predicted bitcoin value (Market Cap) :-> 26153752418
```

Advanced Prediction - It was done using LSTM which used all the features for predicting the Market Cap of bitcoin.



```
Advanced prediction (date, open, close, high, low and volume required) -
enter the date you want to know the bitcoin price on (in DD/MM/YYYY) - 30/08/2016
predicted valueMarket Capof bitcoin -  25551077376
```

IV. Evaluation of models

The models implemented were evaluated using techniques like - Classification report : precision , recall , f1 score and support , Confusion matrix , ROC plots , accuracy score and cross validation scores.

Table 1.1 and 1.2 contains the results obtained from the above techniques.

Table 1.1

	Decision Tree	*Linear Regression*	*XGBoost*	*Polynomial Regression*	*LSTM*
R2 score	*-0.39*	*0.95*	*-0.43*	*0.95*	*0.97*
RMSE (1e9)	*13.55*	*2.46*	*13.76*	*1.93*	*1.37*
MAE (1e8)	*85.24*	*20.17*	*87.32*	*10.88*	*7.63*

Result and Analysis

The above table shows the drastic difference in performance among the regressors for the same regression problem. It is clear that the tree based regression algorithms perform poorly when it comes to graph fitting and sequential data analysis. Linear regression fits the data to some extent with a good r2 score but polynomial regression does better as expected because linear regression fits a curve considering only the linear dependencies in the data whereas polynomial regression takes into account polynomial curve fitting. Using hit and trial, I chose the value of degree = 6 for polynomial regression. The curve, as expected, was smooth like a polynomial function and Polynomial Regression was done completely on the basis of curve fitting i.e. no other feature was taken into account. Long Short Term Memory (LSTM) outperformed all the other regressors. It used a time_step = 15 i.e. after each prediction made, 15 days of the data was fitted in the model. LSTM is an extension of the Recurrent Neural Network (RNN) and helps store feature values for some time period. It is a key component of NLP (Natural Language Processing).

References

1. https://www.kaggle.com/code/meetnagadia/bitcoin-price-prediction-using-lstm/notebook
2. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
3. https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/
4. https://towardsdatascience.com/getting-started-with-xgboost-in-scikit-learn-f69f5f470a97
5. https://scikit-learn.org/stable/modules/tree.html