Bitcoin Prediction

Dhruv (B20EE016)

Abstract

This project reports my experience with building a Bitcoin Price predictor. The dataset of bitcoin from 2013 to 2017 was used. We use various Regression algorithms and compare their results in this report.

Introduction

Bitcoin is a decentralized digital currency, that can be sent from user to user on the peer-to-peer bitcoin network without the need for intermediaries. It was created in 200 As a most valuable cryptocurrency, and with its high volatility, it offers a new opportunity for future price prediction

This project aims to put in use the concepts of regression and various other machine learning and data processing techniques taught in the course or PRML to predict the future values like open and close price, high and low price, volume, etc of Bitcoin

DATA DESCRIPTION AND PREPROCESSING

Dataset included 1556 rows and 7 columns. Different features related to Bitcoin like opening and closing price on each day, High and Low prices on each day Volume and Market Cap, etc. Data range from 2013 to 2017

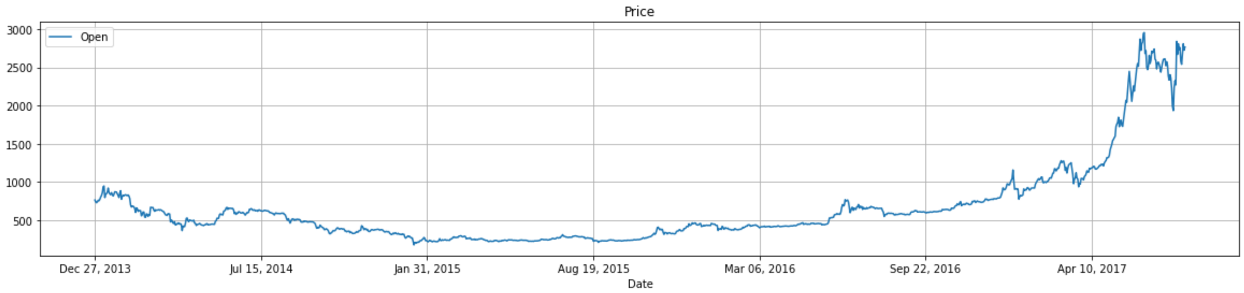
PREPROCESSING

The volume column had some missing values, there were converted to nan values and dropped from the dataset making dataset rows getting reduced from 1556 to 1313. Data present in the volume and market cap column were of object type hence to for further analysis they were converted to integer datatype. Input data columns were scaled using a standard scaler

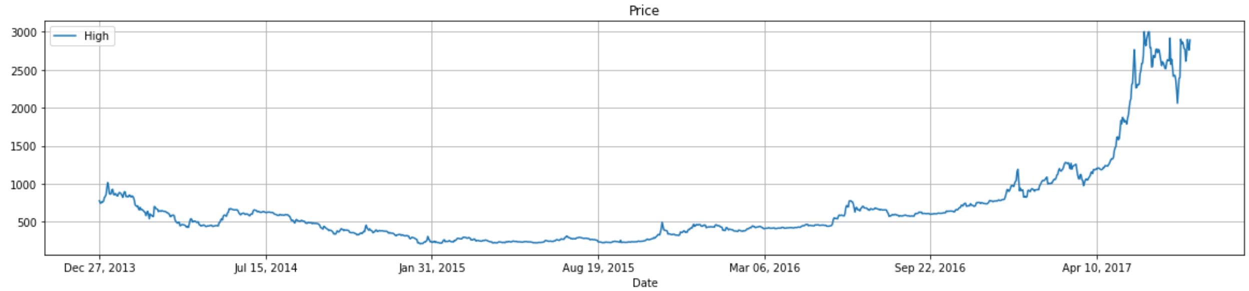
DATA VISUALIZATION

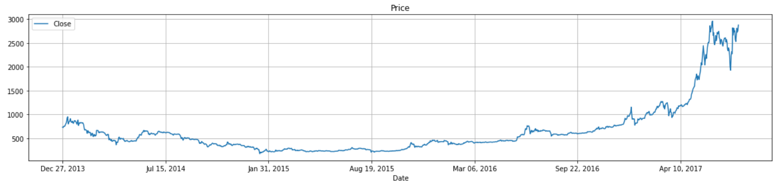
Fluctuation of various features with date:

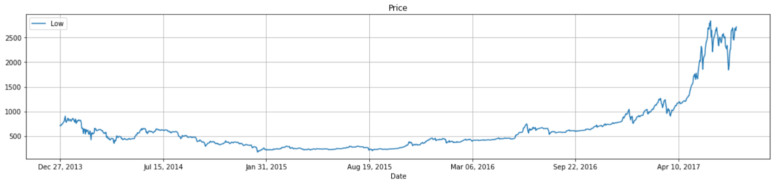
- Opening price vs date. it can be seen opening price has risen exponentially from 2016



- High price vs date. similarly, the high price of bitcoin started rising exponentially between 2016-2017

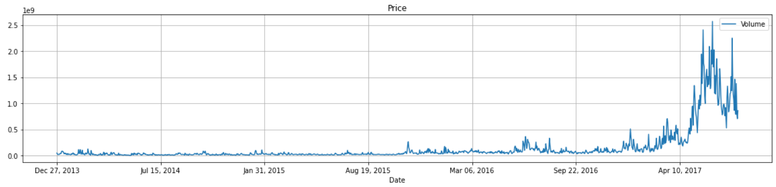


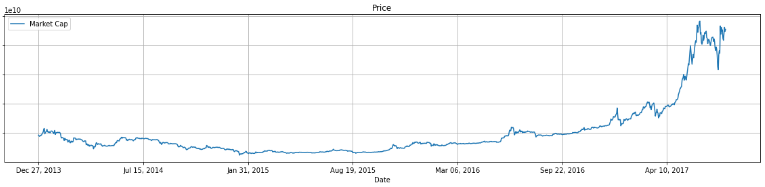




Close Price vs Date

Low Price vs Date





Volume vs Date

Market Cap vs Date

All these graphs clearly explain that the popularity of Bitcoin has risen exponentially between 2016-2017

MACHINE LEARNING MODELS

The price that we have to predict after n days is a regression problem so various regression models were used

Regression is generally used in finance and investing. By definition regression attempts to determine the relationship between one dependent variable (Y) and other variables (known as independent variables).

Various Regression models:

1) SVR

   SVR(Support Vector Regressor) is a supervised learning technique it aims at reducing the error by determining the hyperplane and minimizing the range between the predicted and the observed values

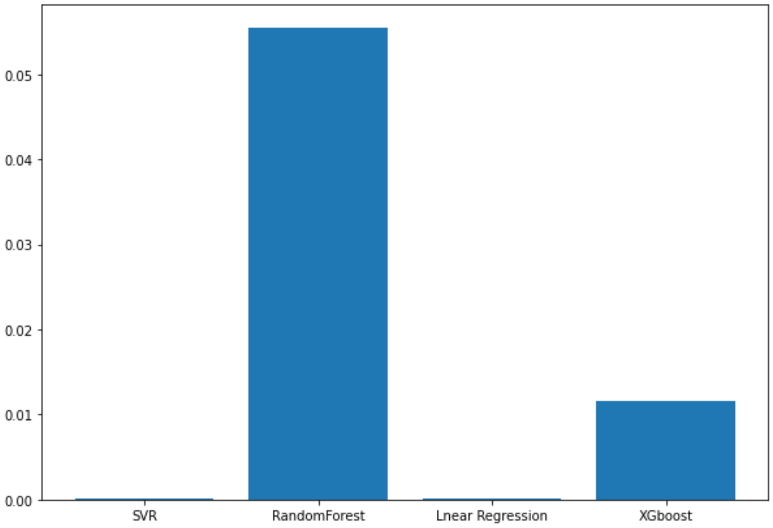
2) Random Forest Regressor
Random Forest has multiple decision trees as base learning models. It is an ensemble technique that makes use of multiple decision trees and a technique commonly known as bagging. The basic idea is to use combine multiple decision trees in determining the final output rather than relying on individual decision trees.

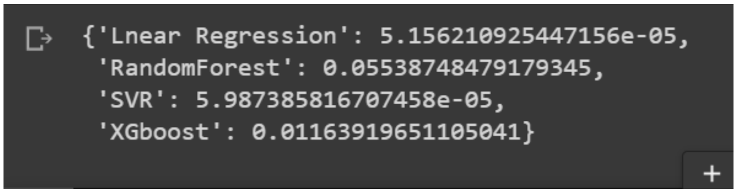
3) Linear Regression
It is a machine learning algorithm based on supervised learning. It performs the task to predict a dependent variable value (y) based on a given independent variable (x). This regression technique finds out a linear relationship between x and y.

4) XGboost Regressor
XGBoost is an efficient implementation of gradient boosting and can be used for regression problems

All these above-mentioned models were trained and tested on each column (open,close, high, etc) of the dataset, and mse were reported

- Feature - Open

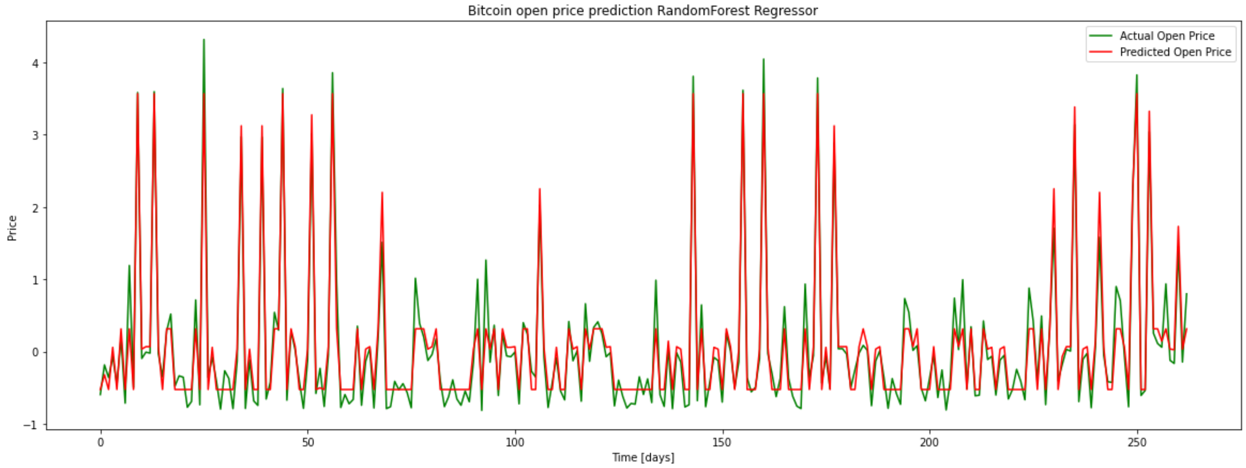


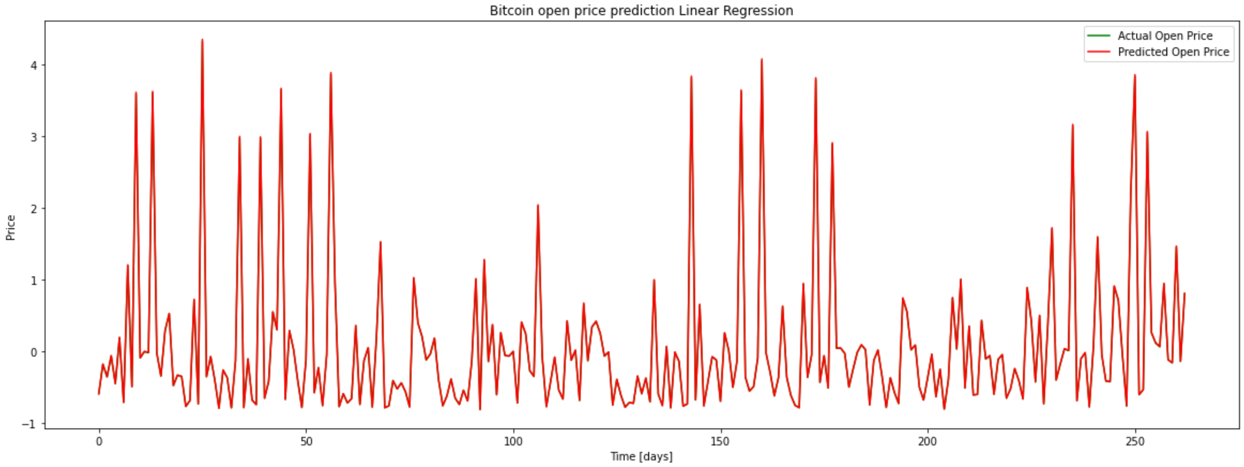


```
{'Lnear Regression': 5.156210925447156e-05,
 'RandomForest': 0.05538748479179345,
 'SVR': 5.987385816707458e-05,
 'XGboost': 0.01163919651105041}
```

The plot of MSE of Different Models for Open Feature

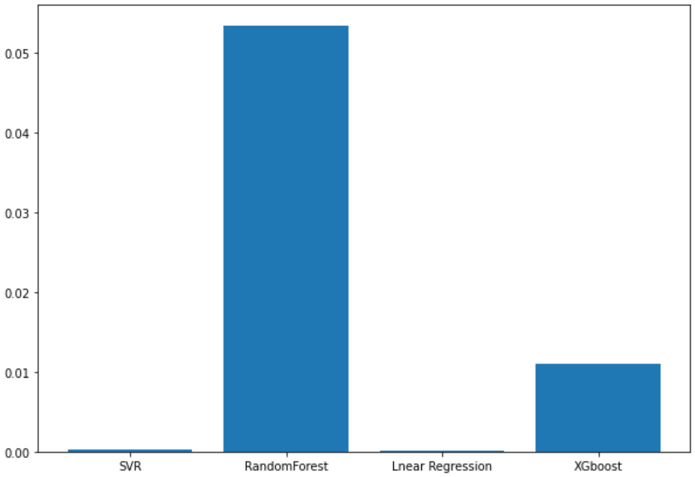
The best Model is Linear Regression and the worst model is Random Forest

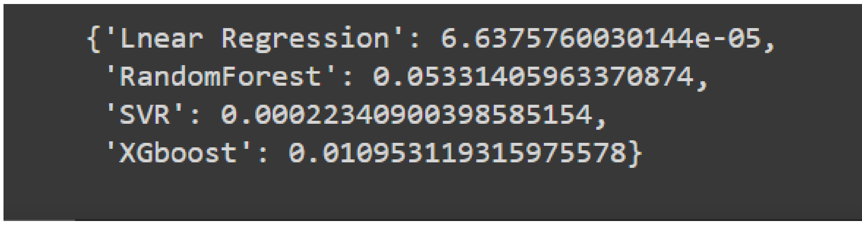




As we can see the predicted and actual values of the linear regression model has almost overlap while the random forest model has less overlap

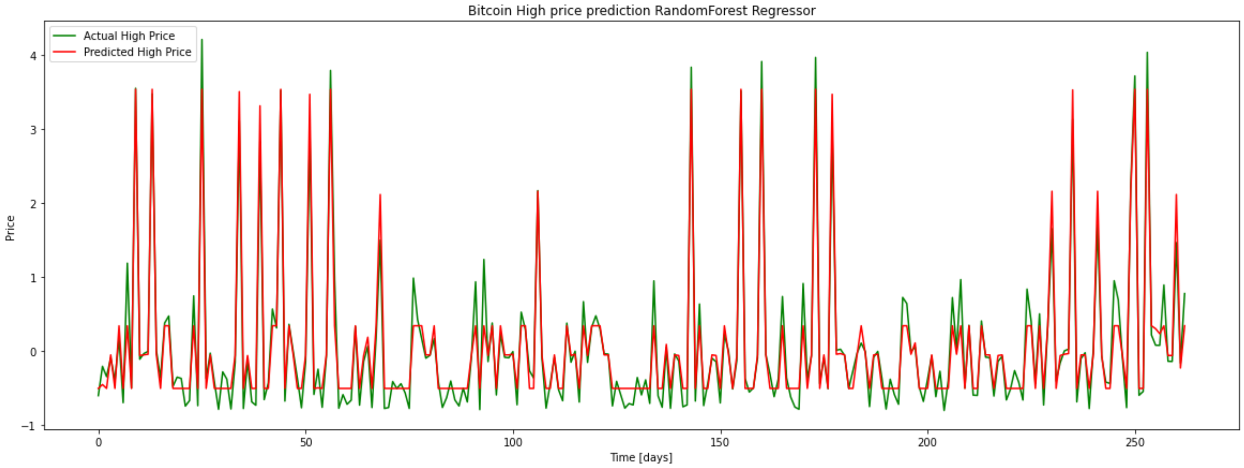
- Feature -High

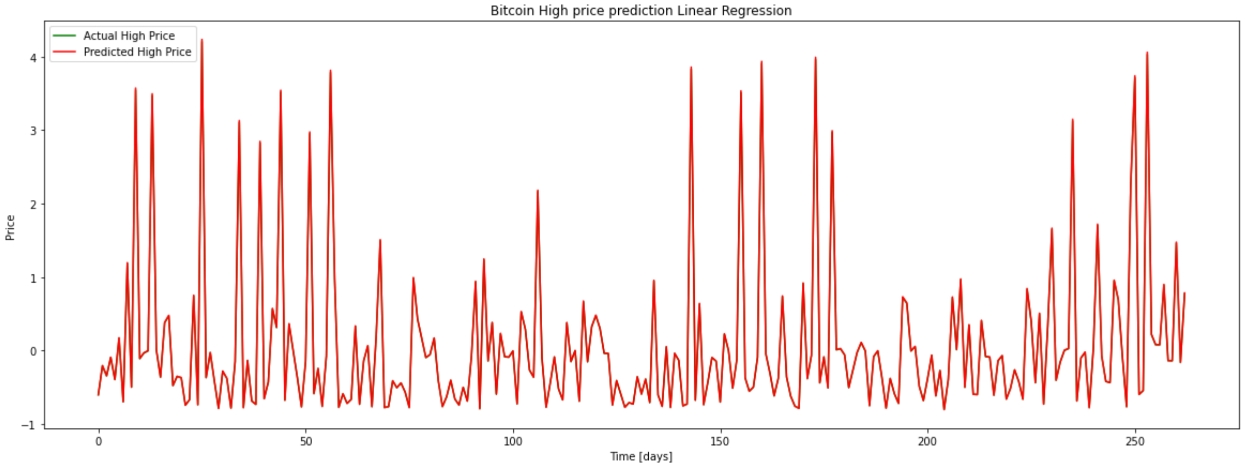




```
{'Lnear Regression': 6.6375760030144e-05,
 'RandomForest': 0.05331405963370874,
 'SVR': 0.00022340900398585154,
 'XGboost': 0.010953119315975578}
```

The plot of MSE of Different Models for High Feature

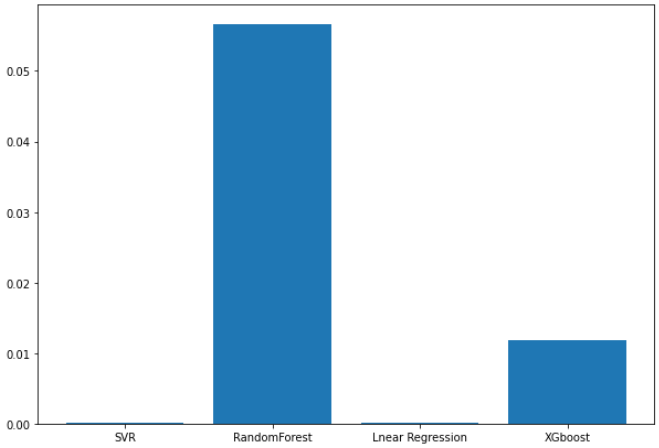


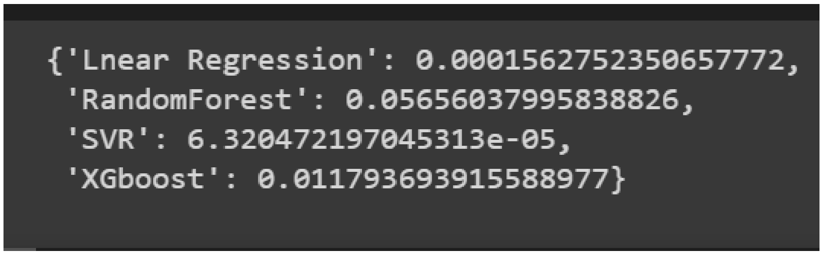


As we can see the predicted and actual values of the linear regression model has almost overlap while the random forest model has less overlap

Similar results can be seen for this feature also Logistic regression has the best overlap

- Feature -Close

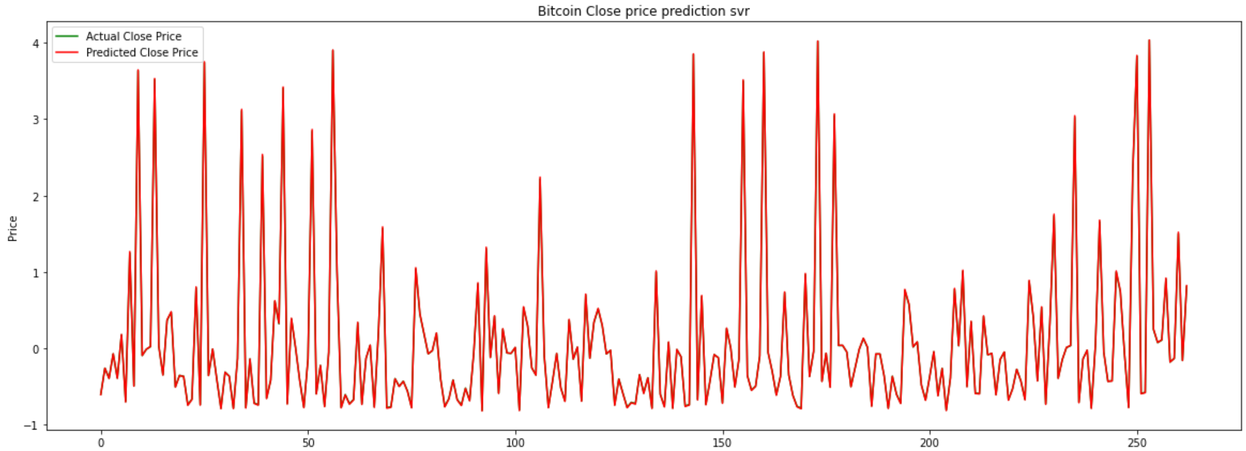


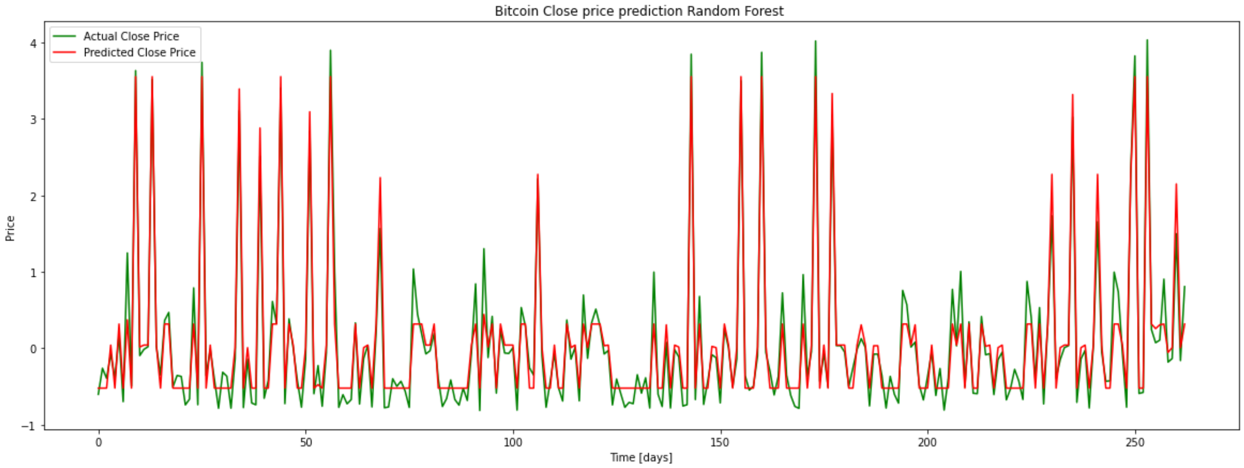


```
{'Lnear Regression': 0.00015627523506577772,
 'RandomForest': 0.05656037995838826,
 'SVR': 6.320472197045313e-05,
 'XGboost': 0.011793693915588977}
```

The plot of MSE of Different Models for Close Feature

The best model is Svr for this feature

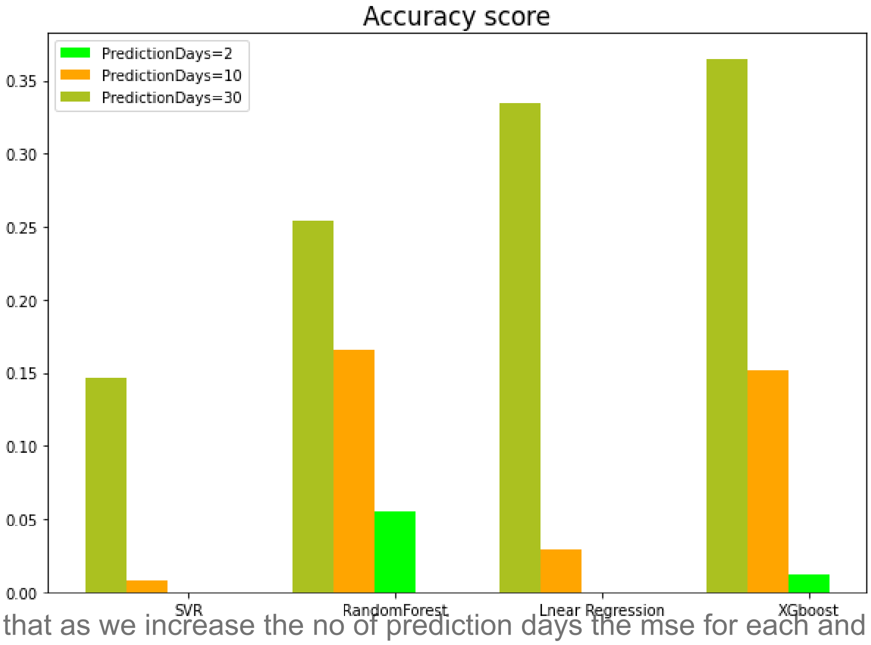




A similar analysis for the rest of the features has been done in the Colab file

Comparing Models on the basics of No of Prediction Days

The above-mentioned models were trained on different values of prediction days (2, 10 30). The graph obtained was :



It can be seen that as we increase the no of prediction days the mse for each and every model is increasing. This is happening because with more no of prediction days more data is lost as we are using the shift in the data to predict the future values hence more error while predicting

CONCLUSION

Svr and Linear Regression have been giving the best results.

REFERENCES

https://machinelearningmastery.com/xgboost-for-regression/
https://scikit-learn.org/stable/