Bonus project

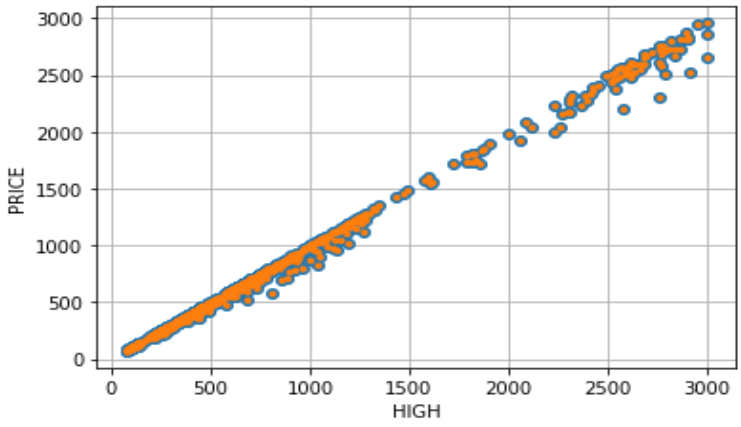
Bitcoin price prediction:
Dataset used :-
https://drive.google.com/file/d/19Kr-gxptHL5RCVxPomrSOWStvcoiGYuv/view?usp=sharing

Column with the name “Close” is used as the target column for prediction of bitcoin price.

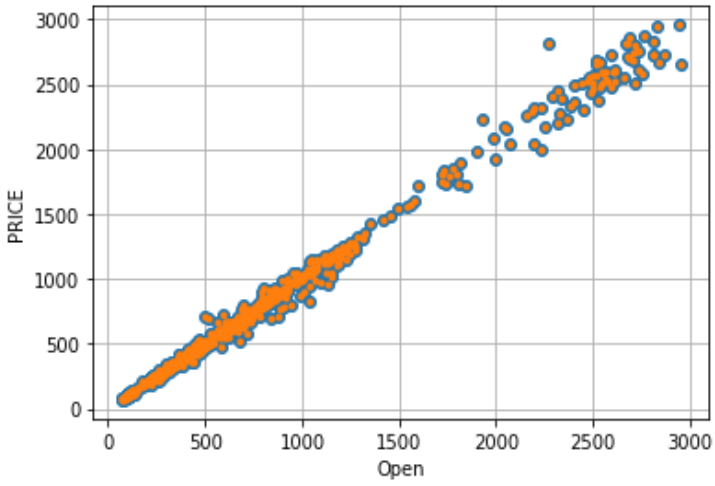
Preprocessing :

Converting string data type columns to integer type.
Assigning 0 to empty places.
With the help of ***to_datetime*** made three different columns with names ***year, month, day*** respectively.

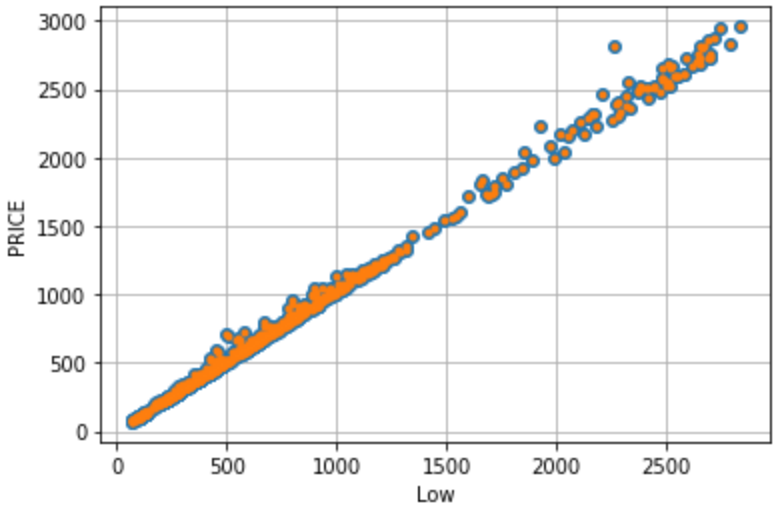
1. **Visualization:**



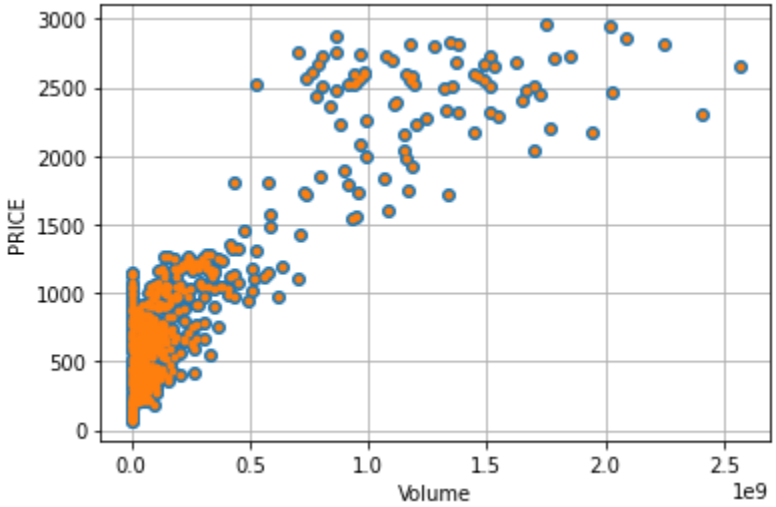
a.



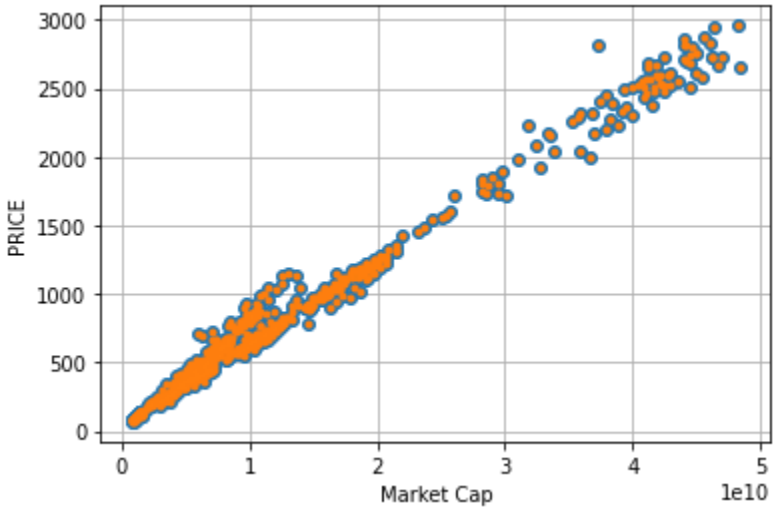
b.


PRICE
Low
0
500
1000
1500
2000
2500
3000

c.

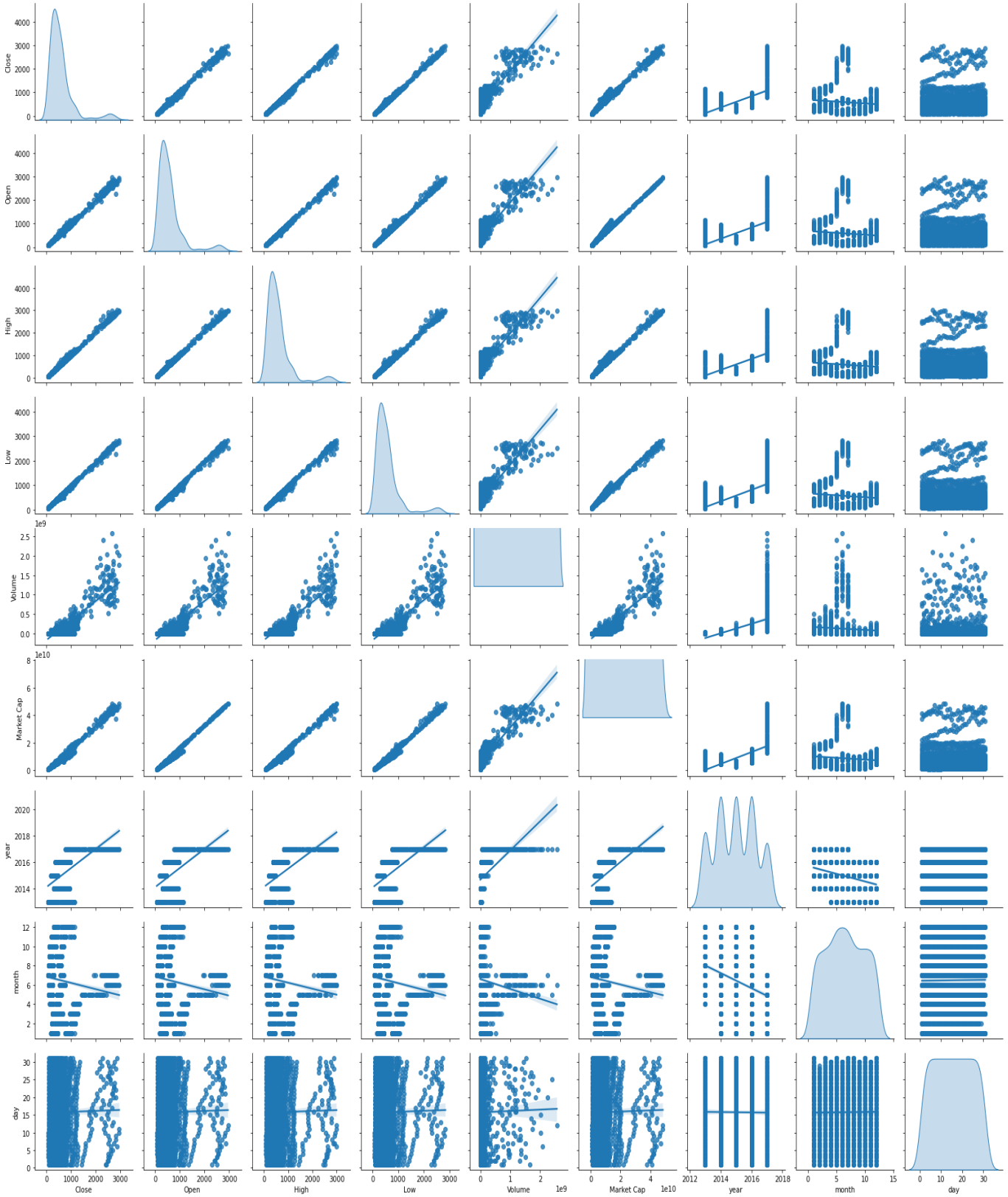

PRICE
Volume
1e9
0.0
0.5
1.0
1.5
2.0
2.5

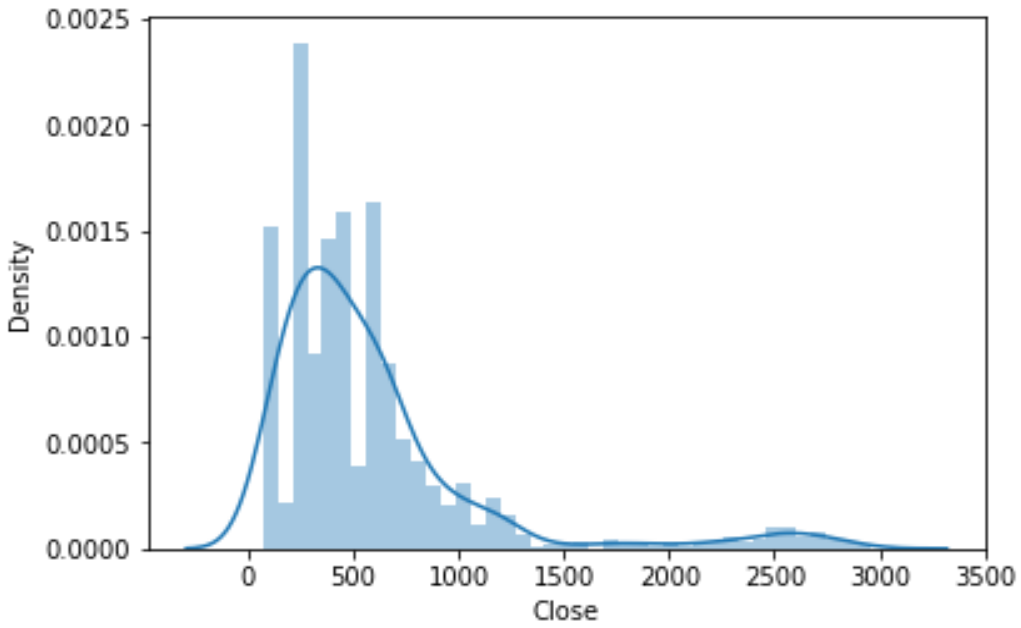
d.


PRICE
Market Cap
1e10
0
1
2
3
4
5

e.

f. This is the complete analysis of data with target variable:





Models:

- **Linear regression:**
  - LinearRegression 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.
  - `from sklearn.linear_model import LinearRegression`
  - Score of the model is given as R-square of self.predict(X) wrt Y.
  - Score of the model is 1.0.
- **Decision tree regressor:**
  - The decision tree is used to fit a sine curve with additional noisy observation. As a result, it learns local linear regressions approximating the sine curve. The decision trees learn too fine details of the training data and learn from the noise, i.e. they overfit.
  - `from sklearn.tree import DecisionTreeRegressor`
  - Score in this case is obtained using cross validation score from model selection.
  - Score of the model is in range (0.9972, 0.9998)
- **Random forest regressor:**
  - A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

- The sub-sample size is controlled with the **max_samples** parameter if **bootstrap=True** (default), otherwise the whole dataset is used to build each tree.
- `from sklearn.ensemble import RandomForestRegressor`
- Training the model on train data set then predicting the bitcoin price on the test data set.
- Score of this model is produced by **out of bag score.**
- Score got is 0.9989.

❖ Bagging regressor:

- A Bagging regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.
- `from sklearn.ensemble import BaggingRegressor`
- Score of this model is produced by **out of bag score.**
- Score got is 0.9713..

Sudip kumar
B20CS072