Assignment 1: Linear Regression

Kavya kongara 11441102

Linear regression: It is a supervised machine learning algorithm. It takes input values of one variable and predict output values of another variable.

Data set: monset.csv is the name of the dataset that I used for my analysis. It contains 6 attributes and 437 rows

First model: (Size Vs price)

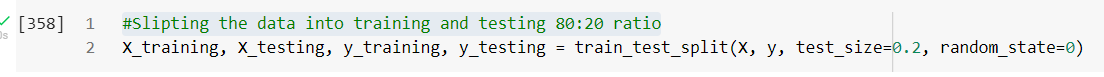
* I created first model, by combining the 2 attributes i.e., width and height and created a new attribute called SIZE.
* Data preparation:

X is independent variable which stores the values of Size

y is dependent variable which stores the values of price

* Splitting the data:

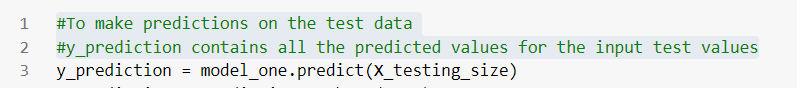
I am splitting the data for training and testing in 80:20 ratio by using sklearn library

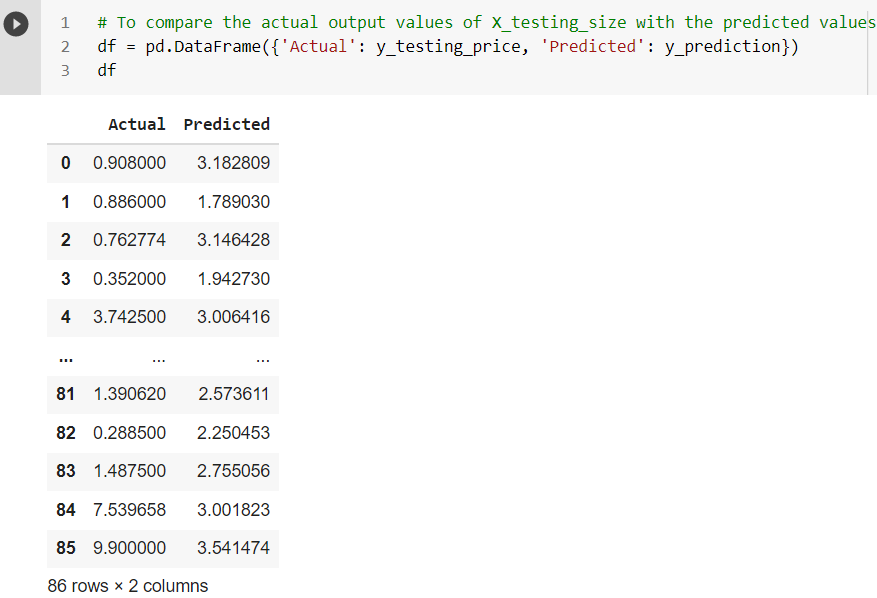


* Training the model:

Now, applying the linear regression model for training data by using scikit-learn.

For every increase in the size the prize increases by 0.02

* Predictions:
* For predictions, I am using the test data to check how accurate are the price values. 
* To compare the actual results for X\_testing\_size with the predicted values, pushed the values inside a dataframe

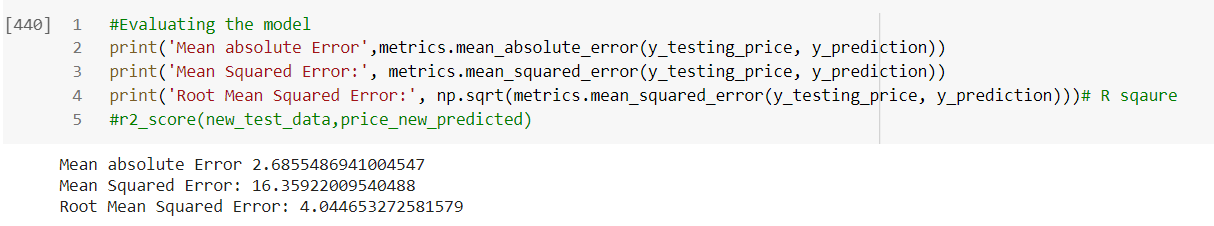


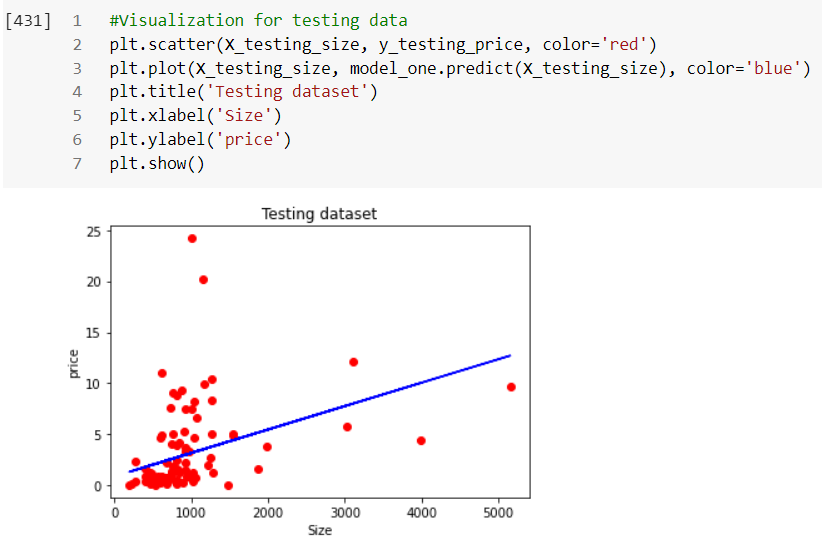
Here, the predicted values are bit high when compared to the actual ones.

* Evaluating the model:

For model evaluation I am using Scikit-learn library

* Mean Absolute error
* Mean squared error
* Root mean squared error



* Visualizations:
* Training data:
* Testing data: 

Conclusion: The values are very close to the linear line, which minimizes the error to be a best fit model (based on the accuracy)

Second model: (height Vs price)

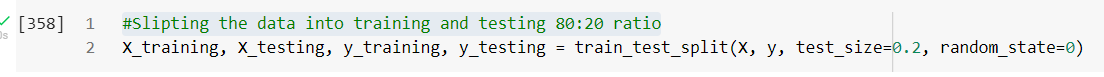
* I created second model, by taking 2 attributes i.e., height and price.
* Data preparation:

X is independent variable which stores the values of height

y is dependent variable which stores the values of price

* Splitting the data:

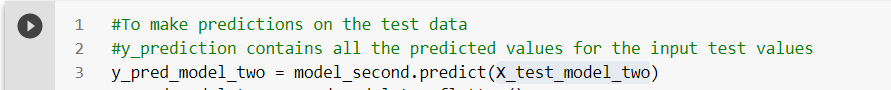
I am splitting the data for training and testing in 80:20 ratio by using sklearn library

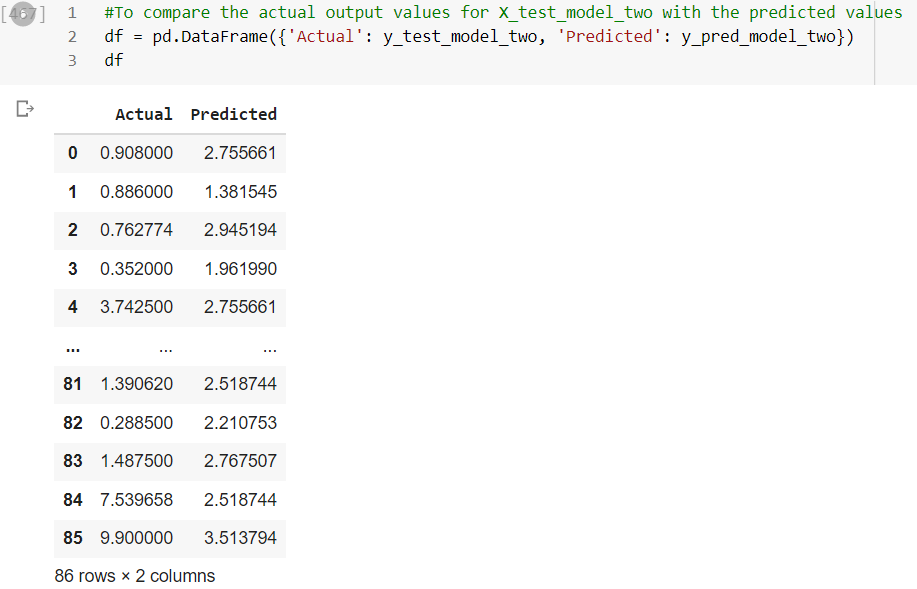


* Training the model:

Now, applying the linear regression model for training data by using scikit-learn.

For every increase in the height the prize increases by 0.02

* Predictions:
* For predictions, I am using the test data to check how accurate are the price values. 
* To compare the actual results for X\_test\_model\_two with the predicted values, pushed the values inside a dataframe

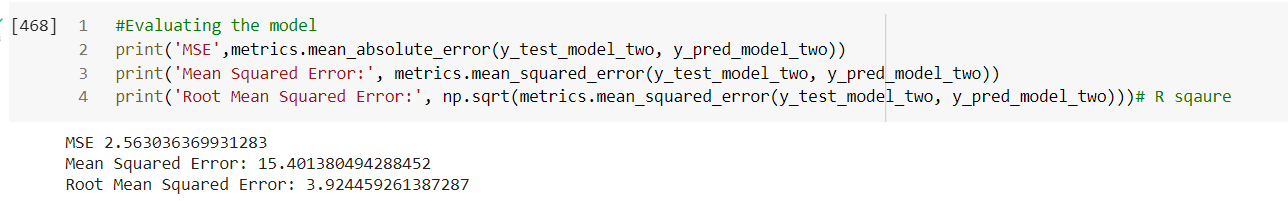


Here, the predicted values are bit high when compared to the actual ones.

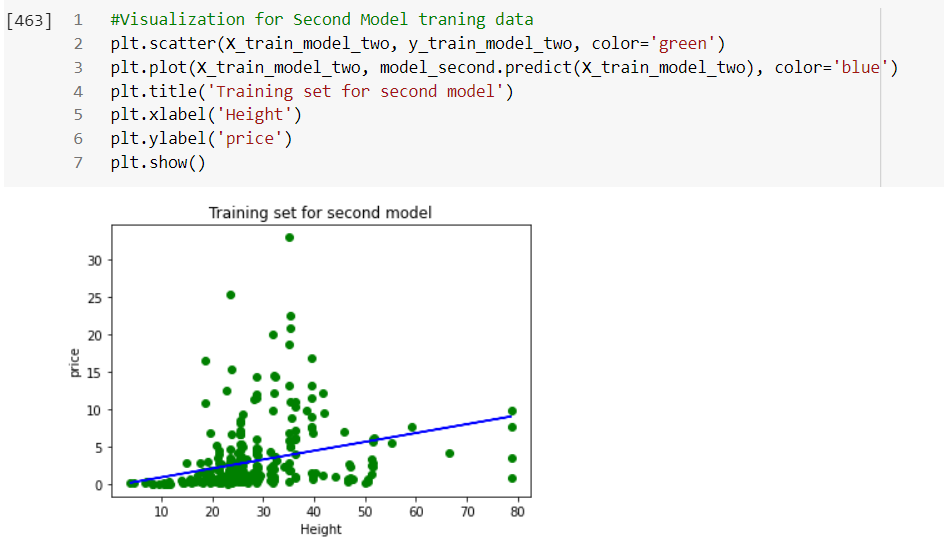
* Evaluating the model:

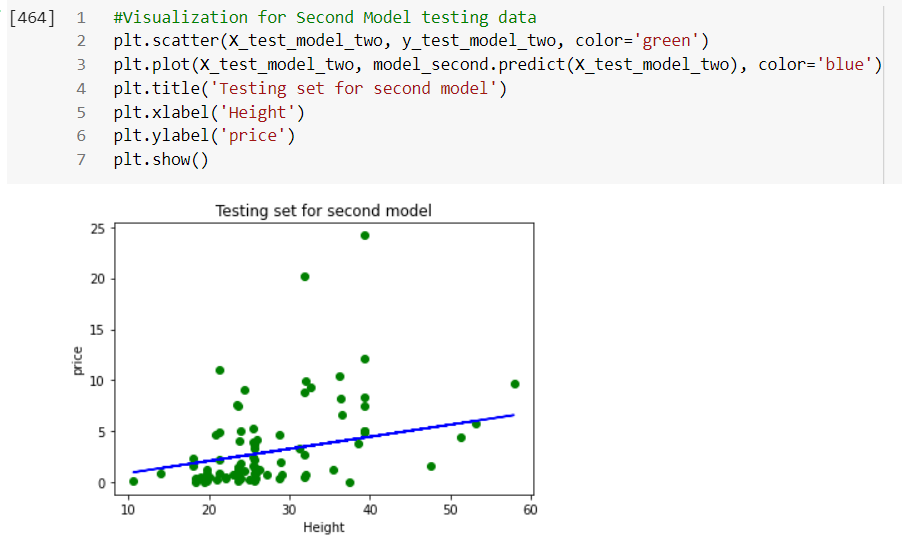
For model evaluation I am using Scikit-learn library

* Mean Absolute error
* Mean squared error
* Root mean squared error



* Visualizations:
* Training data:



* Testing data: 

Conclusion: From the training and testing graphs the points lie on the linear line and near to the line which reduces the error.

Multivariant Linear regression: (['HEIGHT', 'WIDTH', 'SIGNED','PICTURE', 'SIZE', 'HOUSE '] VS price)

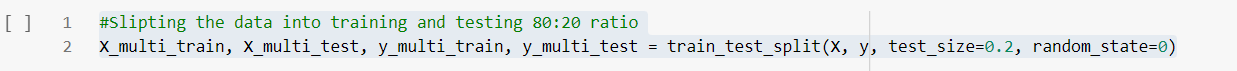
* I created Multivariant model, by taking 7 attributes
* Data preparation:

X is independent variable which stores the values of 'HEIGHT', 'WIDTH', 'SIGNED','PICTURE', 'SIZE', 'HOUSE

y is dependent variable which stores the values of price

* Splitting the data:

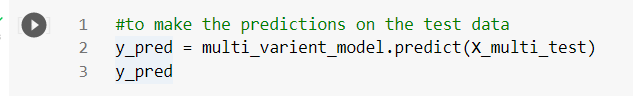
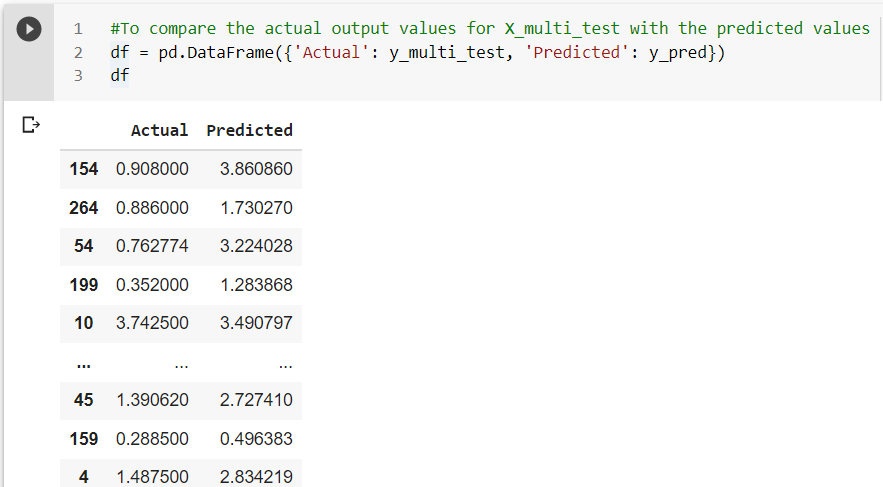
I am splitting the data for training and testing in 80:20 ratio by using sklearn library



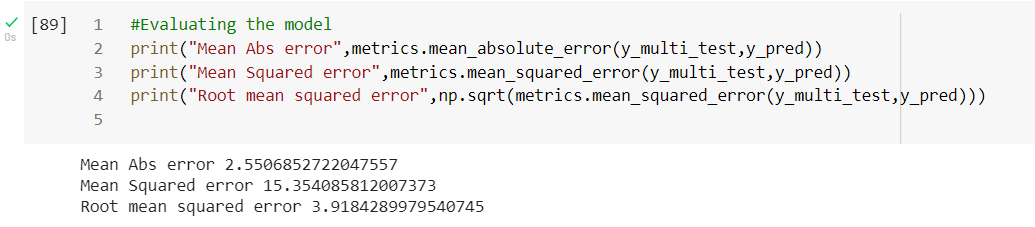
* Training the model:

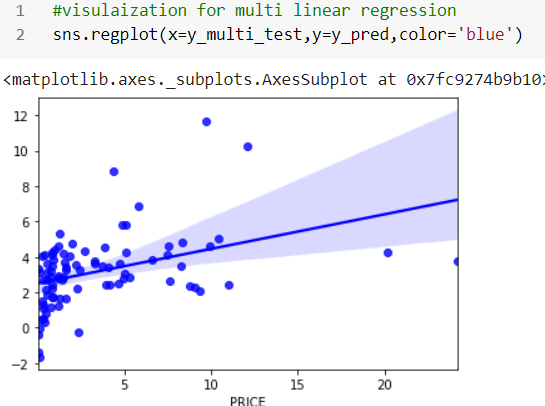
Now, applying the linear regression model for training data by using scikit-learn.

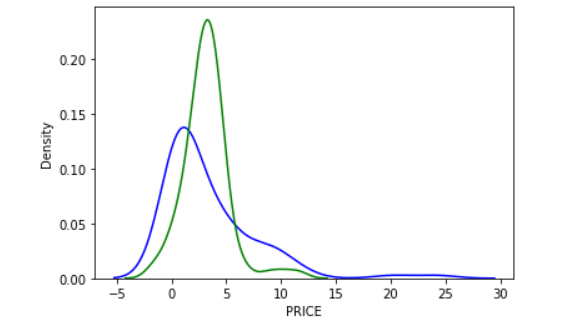
For every increase in the height the prize increases by 0.02

* Predictions:
* For predictions, I am using the test data to check how accurate are the price values. 
* To compare the actual results for X\_multi\_test with the predicted values, pushed the values inside a dataframe
* Evaluating the model:

For model evaluation I am using Scikit-learn library

* Mean Absolute error
* Mean squared error
* Root mean squared error 
* Visualizations:
* For multivariant regression visualization I am using seaborn library





Blue line gives the testing values and the green line on the graph shows the prediction values. So, the predicted values are high for the some give test values.

Finally, my model is good fit in terms of predicting the values which are close to the regression line and minimizes the error.