Building The Car of The Future

Rajashekar Korutla

Applied Machine Intelligence, Roux Institute at Northeastern University

ALY 6020: Predictive Analytics

CONTENTS

Problem Statement/Business Requirements

A car manufacturer known for making large automobiles is struggling with sales and has asked for your help in designing an energy efficient car. Using data gathered, determine which attributes may contribute to higher gas mileage so that they can design a more fuel-efficient automobile.

Overview Of Data

From the car dataset, it can clearly be seen that there are 398 rows and 8 columns in the dataset.

So, in order to get the list of columns, I used the df.columns line of code and stored the list of columns in the variable named col_list. Here is the line of code to get the columns from the dataset:

```
col_list = list(df.columns)
```

After getting the list of columns, I have used the following command to get the datatypes of the columns that are present in the dataset.

```
df.info()
```

Data Pre-Processing Including Data Exploration Analysis

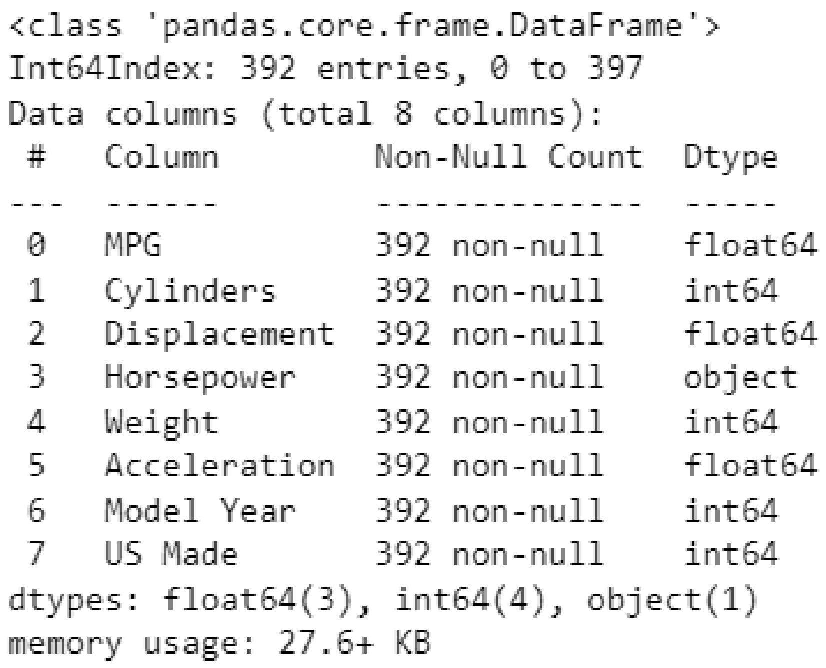
Then it happened to realize that there are certain question marks in the column of horsepower, which are to be removed from the dataset for smooth processing of data.

For which, I first took all the rows that contain the question mark, and got rid of all the rows from the dataset, I used the lines of code:

```
df.loc[df['Horsepower'] == '?']
df = df.drop(labels = [32,126,330,336,354,374] , axis = 0 )
```

This line of code would remove all the question marks of the column Horsepower.

This is the information left when the process is done,that is obtained by the line of code, df.info() :


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 392 entries, 0 to 397
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   MPG           392 non-null    float64
 1   Cylinders     392 non-null    int64  
 2   Displacement  392 non-null    float64
 3   Horsepower    392 non-null    object 
 4   Weight        392 non-null    int64  
 5   Acceleration  392 non-null    float64
 6   Model Year    392 non-null    int64  
 7   US Made       392 non-null    int64  
dtypes: float64(3), int64(4), object(1)
memory usage: 27.6+ KB
```

Then, to convert the datatype of the horsepower,from object to integer,this line of code is being used:

```
df['Horsepower'] = pd.to_numeric(df['Horsepower'])
```

Details Of Modeling Strategy

I did the normalization of data, to make the data normalized i.e. between 0 to 1. I did it with the following line of code,where ‘i’ being the i th column among the columns:

```
df_min_max_scaled[i] = (df_min_max_scaled[i] - df_min_max_scaled[i].min()) / (df_min_max_scaled[i].max() - df_min_max_scaled[i].min())
```

The target variable ‘MPG’ has been separated as ‘y’. All the other features have been put together into 'x’, to form the feature matrix. The data has been split for training data and test data, as 70% for training and the remaining 30% for test data.

This is implemented by the following lines of code:

```
x = df.drop(labels = [‘MPG’],axis = 1)

y = df['MPG']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3 , shuffle = 42)
```

Implementing Linear Regression

After the data has been split into training data and test data, I have imported the logistic regression algorithm from the scikit learn library, with the following line of code:

```
from sklearn.linear_model import LinearRegression
```

then the training data has been fit in the model, and predicted the test data with the following lines of code:

```
reg = LinearRegression().fit(X_train, y_train)
y_pred = reg.predict(X_test)
y_pred
```

Estimation Of Model's Performance

The metric that is being used to measure the model's performance is that , mean_squared_error and root mean squared error. Mean Squared Error is the squared difference between actual and predicted values.

As the name suggests, root mean squared error is the squared root of the mean squared error.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y_i}\right)^2$$

While the root mean squared error can be calculated from the mean squared error.

Insights And Conclusions

The linear regression model is performing good on the dataset , as the error that we got is quite less ,the output is 0.01 for mean squared error and 0.1 for the root mean squared error, which is quite desired.