PROJECT REPORT

on

Car Price Prediction project

By Sreekari I

INTRODUCTION

With the covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. With the change in market due to covid 19 impact, there are changes in the car price valuation.

DATA SOURCE

The data is collected from the cardekho website. The details include the various features of the cars that are helpful to predict the price of the car.

OBJECTIVE

We are going to analyse the given data and check the factors that affect the price of the car and make a car price valuation model to predict the car price.

I have used jupyter notebook for data analysis.

DATA ANALYSIS

The given dataset has 7000 rows and 14 columns. The names of the columns are:

'Unnamed: 0'
'Name',
'Price(in Rs.)',
'Year',
'Fuel',
'KMs Driven',
'Engine Displacement',
'No Of Owners',
'RTO',
'Transmission',
'Insurance Type',
'Mileage',
'Max Power',
'Torque'

Now, let us analyse the clean and pre-process the data.

I have removed the 'Unnamed: 0' column since it is not useful in creating the model.

-→ Removing unnecessary details in the column values.

In this step, I have removed unwanted characters and values from the columns by using replace method of pandas dataframe and substituted with suitable values from all the columns.

-→ Data Analysis

By checking the df.info(), I see that there are 6 columns of float type and 7 columns of object datatype. Also we can see that there are some missing values.

Handling null values:

There are null values in Price, Kms driven, Engine displacement, RTO, Insurance type, Mileage, Max power and Torque columns.

I have dropped the null values in Insurance type since there are only 6 null values.

I have filled null values in Price, Kms driven, Engine displacement, Mileage, Max power and Torque columns with mean.

For RTO column, I have replaced null values with 'NA'.

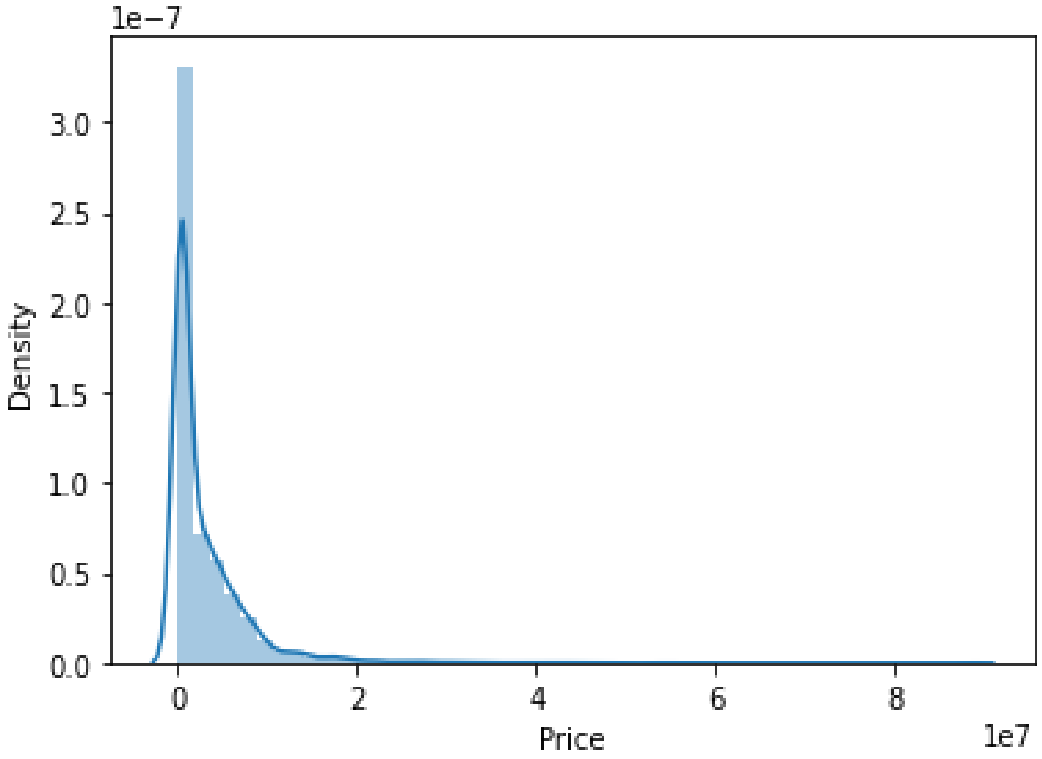
Now, we have removed all the null values. The shape of the data after removing null values in 6994 rows and 13 columns.

Let us analyse these columns now.

Analysing target variable

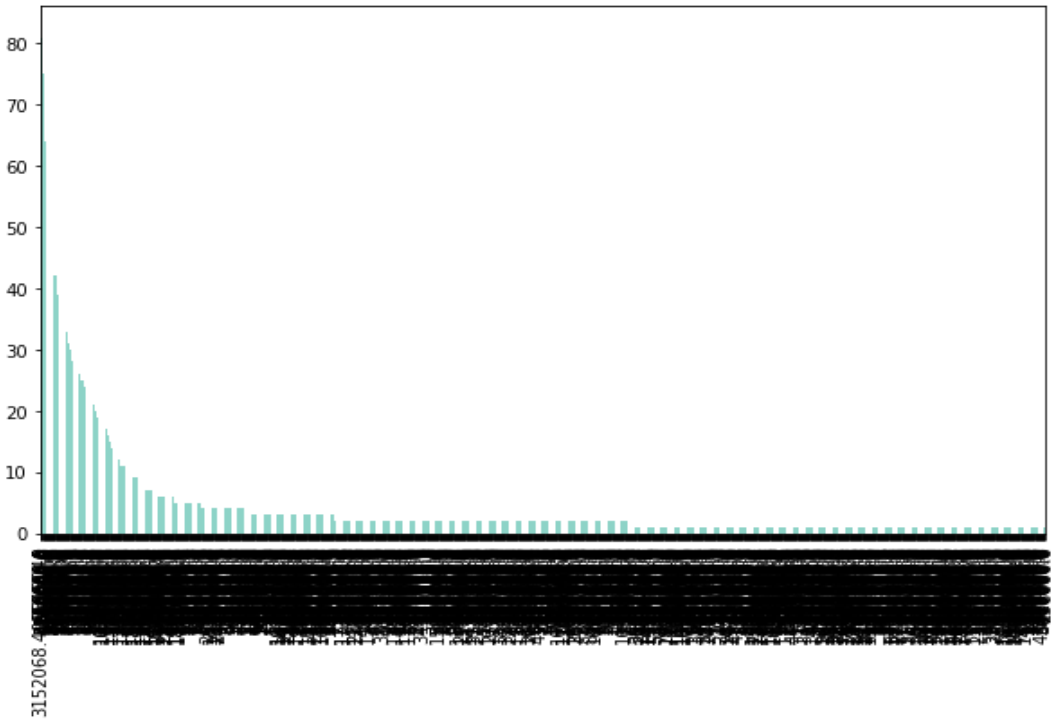
df.Price.describe()

```
count       6994.00000
mean     3152068.49552
std      5272558.90385
min        10000.00000
25%       430000.00000
50%       884000.00000
75%      4250000.00000
max     88150000.00000
Name: Price, dtype: float64
```

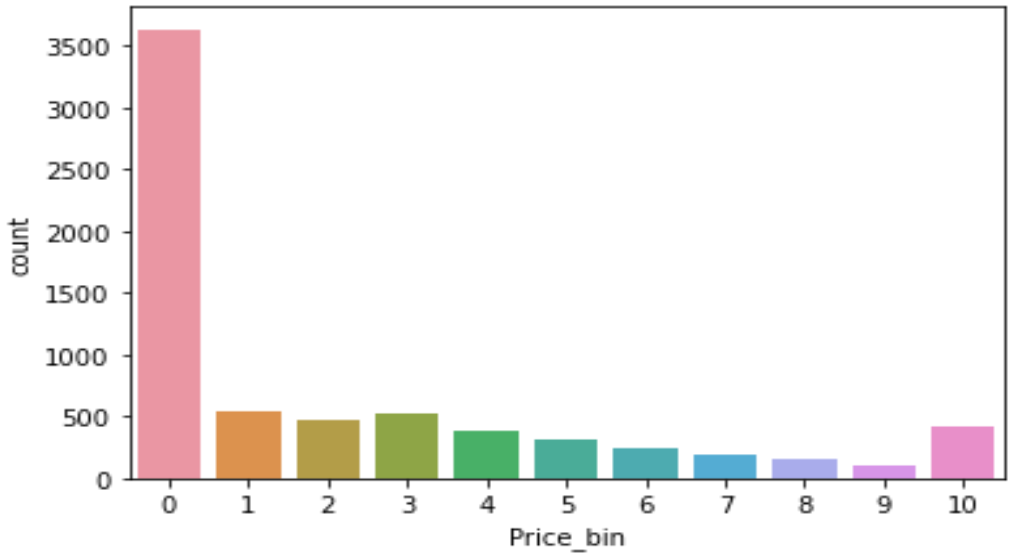


The above image is the distplot showing the Price column distribution.

The below image is the barplot showing the number of cars available in various price range.

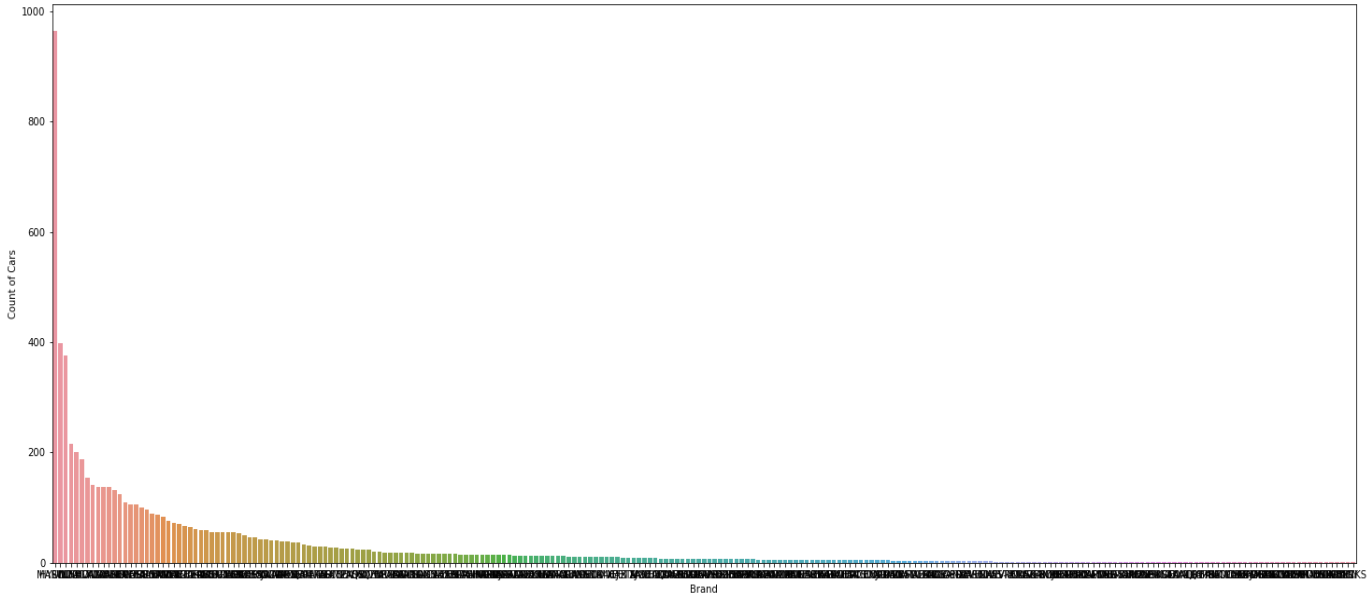


I have created bins(0-10) for Price less than 1000000 as bin-0, 1000000-2000000 as bin-1 and so on till 9000000-10000000 as bin-9 and price above 10000000 as bin-10.



It is observed that Car Name consists of two parts 'Car company' + ' ' + 'Car Model'. So I have split out car company to a new column.

Now, let's see companies and their no of models of each type. I see that there are 242 different models.



Let us encode the data to create the data model.

```
df.replace({"1st Owner": 1, "1st ": 1, 'First ':1, 'First Owner': 1,
            "Second Owner": 2, "2nd ": 2, 'Second ':2, '2nd Owner':2,
            "Third Owner": 3, '3rd Owner':3, '3rd ':3, 'Third ':3,
           'Fourth & Above Owner':4, 'Fourth & Above ': 4, '4th ': 4, 'Test Drive Car':0, '-':0}, inplace = True)
```

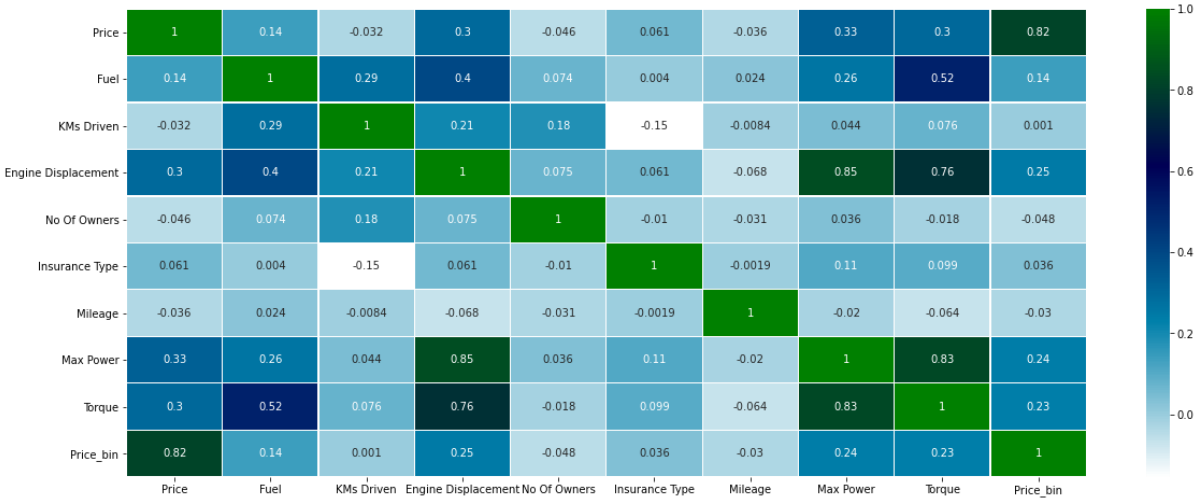
```
df.replace({"Petrol": 1, "Diesel": 2, 'CNG':3, 'LPG': 4,
            "Electric": 5}, inplace = True)
```

```
df.replace({"Available": 1, 'Insurance': 1, "Manual": 2, 'Comprehensive':4,
'Corporate': 5,
        "Third Party insurance": 3, 'Third Party':3, 'Automatic':6, 'Individual': 7,
        'No Insurance':0, 'Not Available':0,'No insurance' : 0}, inplace = True)
```

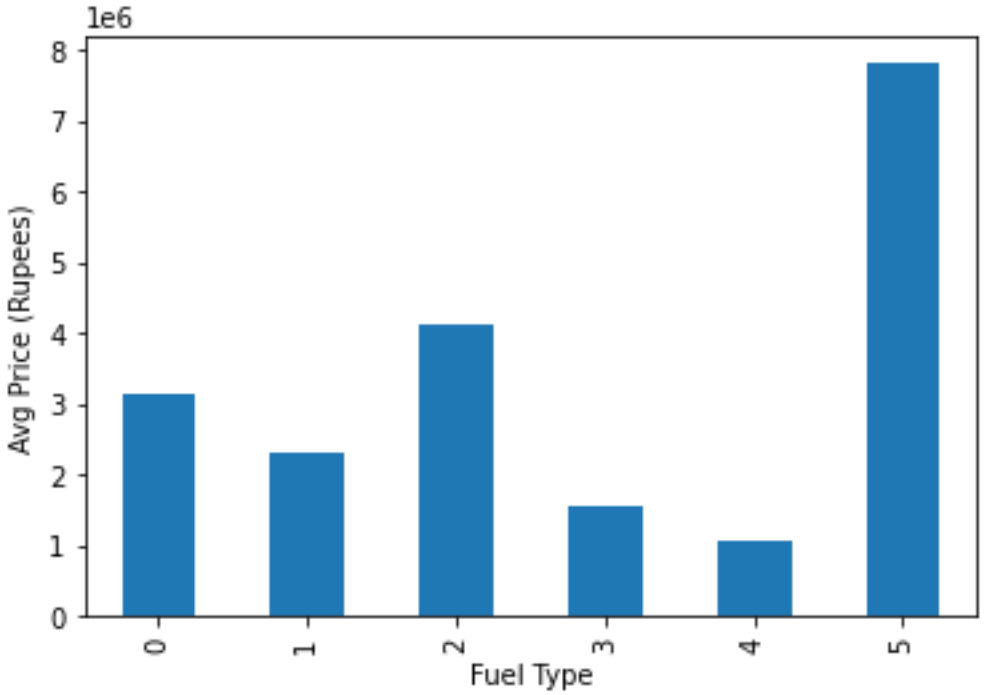
```
# As RTO is Nominal Categorical data we will perform OneHotEncoding
rto = df[["RTO"]]
rto = pd.get_dummies(rto, drop_first= True)
```

```
# As Transmission is Nominal Categorical data we will perform
OneHotEncoding
transmission = df[["Transmission"]]
transmission = pd.get_dummies(transmission, drop_first= True)
```

The below image is the heat map. We can see the correlation between the columns of the dataset with this heat map.



Let's see how price varies with Fuel Type.



Now that we have done encoding for all the object data type columns except for Name column.

Let us encode Name column so that we can proceed with data modelling.

The following is the code I have used to encode the Name column.

```
from sklearn import preprocessing
labelencoder=preprocessing.LabelEncoder()
for column in df.columns:
    df['Name'] = labelencoder.fit_transform(df["Name"])
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df = df.apply(lambda col: le.fit_transform(col.astype(str)), axis=0,
result_type='expand')
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
```

```
for column_name in df.columns:
    if df[column_name].dtype == object:
        df[column_name] = le.fit_transform(df[column_name])
    else:
        pass
```

MODEL CREATION

Firstly, dividing the data into X and y sets for the model building.

```
X=df.drop(['Price'], axis=1)
y=df['Price']
```

Next, splitting the data into training and test data test.

```
We specify this so that the train and test data set always have the same rows, respectively
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

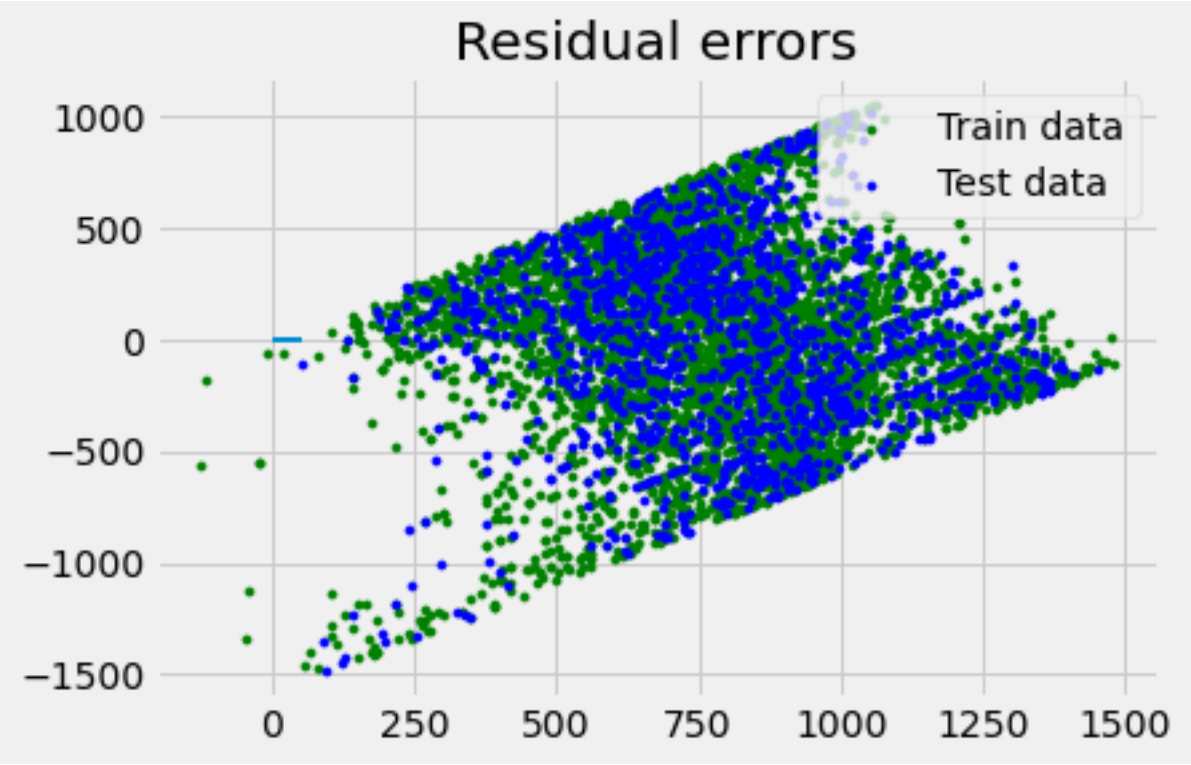
I have used Linear Regression model and Random Forest Models for analysis.

LINEAR REGRESSION MODEL:

Accuracy is 75.05.

Regression Coefficients are [3.57174617e+01 2.97252817e+01 -8.95962500e-03 -1.10160589e-01 -2.37282034e+01 -2.33252153e-02 1.06051891e+00 -2.74244380e+00
5.02138530e-01 9.39486536e-02 -1.90581270e+00 4.69114244e+01 -1.54695821e+00]

Variance score: 0.27136831275125595
Regression intercept : 160.20766244703452



RANDOM FOREST MODEL:

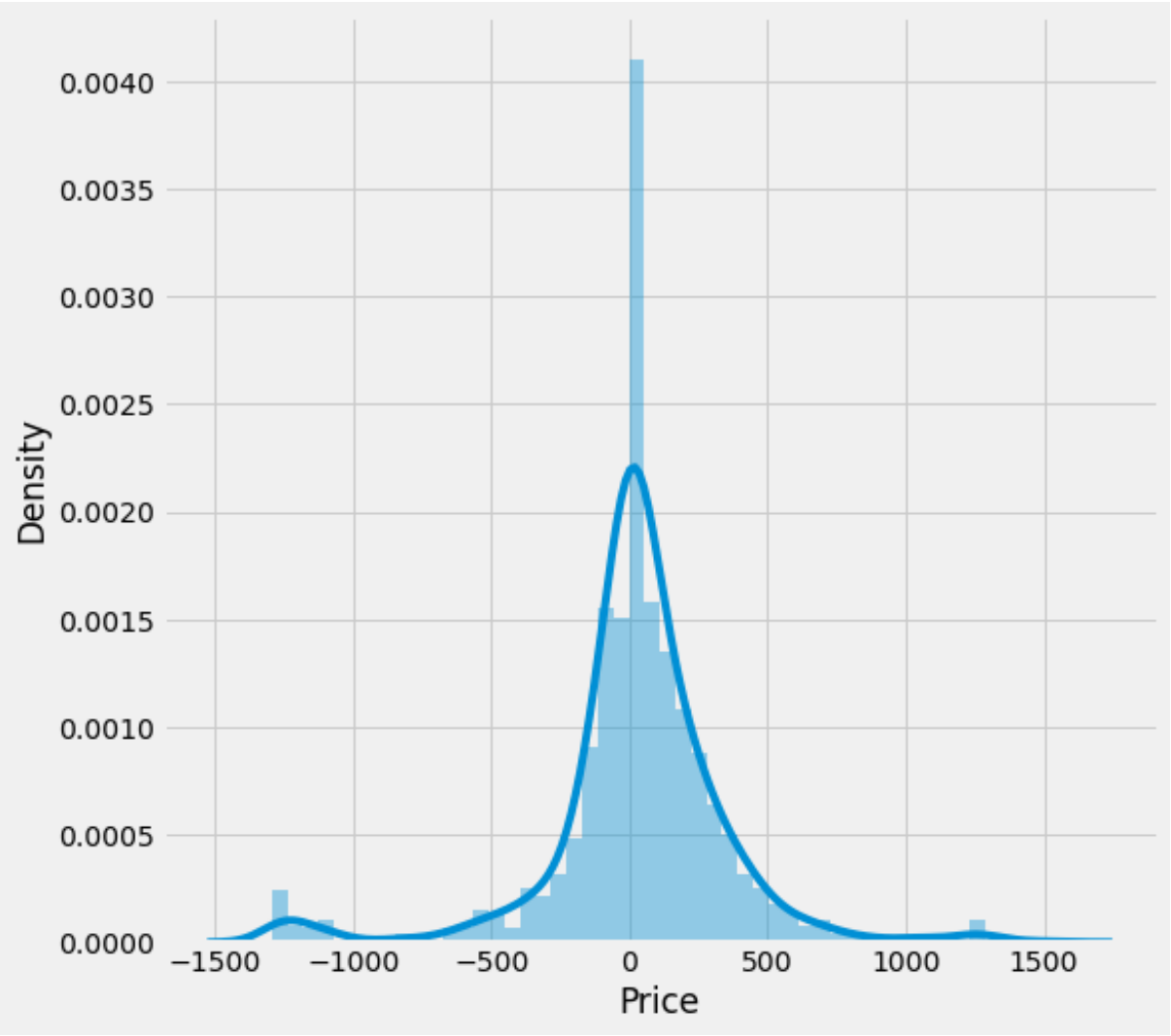
RandomForestClassifier(max_depth=5, n_jobs=-1, oob_score=True, random_state=42)

OOB Score: 0.08954423592493298

Hyper Parameter Tuning:

I have used GridSearchCV for hyper parameter tuning. I have applied hyper parameter tuning for random forest model.

Grid search best score: 0.2865039508089265



MAE: 204.64403145103645
MSE: 110955.76626161544
RMSE: 333.1002345565302

CONCLUSION

From the above analysis we can see that various features of the cars affect the price of the cars. If it has more features, the price of the car will be more when compared to the car with less number of features. So the car price can be predicted based on these factors.