FLIP ROBO

Car Price Prediction

Submitted by:

Sumit Santra

ACKNOWLEDGMENT

I Would to like to express my Special thanks of gratitude to my SME **Shubham Yadav,** Who gave me thr golden opportunity to do this wonderful project of **Price Prediction Project**

Who also helped me in completing my project. I came to know about so many new things , I am really thankful to them.

INTRODUCTION

- Business Problem Framing
  - ➔ 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. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models. So, they are looking for new machine learning models from new data. We have to make carprice valuation model. This project contains two phase

- Conceptual Background of the Domain Problem
  - ➔ Here we need to understand the Car Industry to understand the Problem Easily.
  - ➔ We need to know the companies which Manufactures the car and which company Manufacture the high-end cars as we need to predict the price we should know the car Company Closely.
  - ➔ As for Used cars the the price is dependent on how many owners it has, Which Transmission it uses, What knd of fuel it uses and How many kilometer it ran this things decide the price of the car.
  - ➔ We Should Know this basic things to understand the Project.
- Review of Literature
  - ➔ First I started Researching about the cars how cars work which company manufactures high end cars and what are the fuels the car uses to run.
  - ➔ After this I researched about how the price of used car is determined the main point was number of owners, Kilometers Run and what kind of transmission the car uses Manual Transmission has less value as compared to Automatic Tranmission. Then Fuel also decides the price of the car.

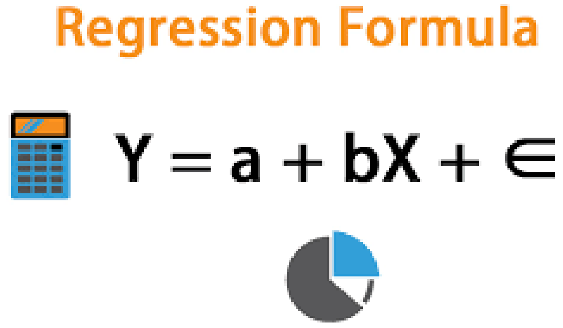
- Motivation for the Problem Undertaken

➔ My motivation to undertake this Project is my mentor Shubham Yadav Sir & to get the knowledge about the Web Scraping and Machine Learning So that it will be easy for me to undertake this kind of project easily as I know the Concepts behind Machine Learning and how to pre process the data and predict the dependent variable.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

➔ As this is a regression model I used regression technique to solve this problem i.e Predicting the Price of Used car.



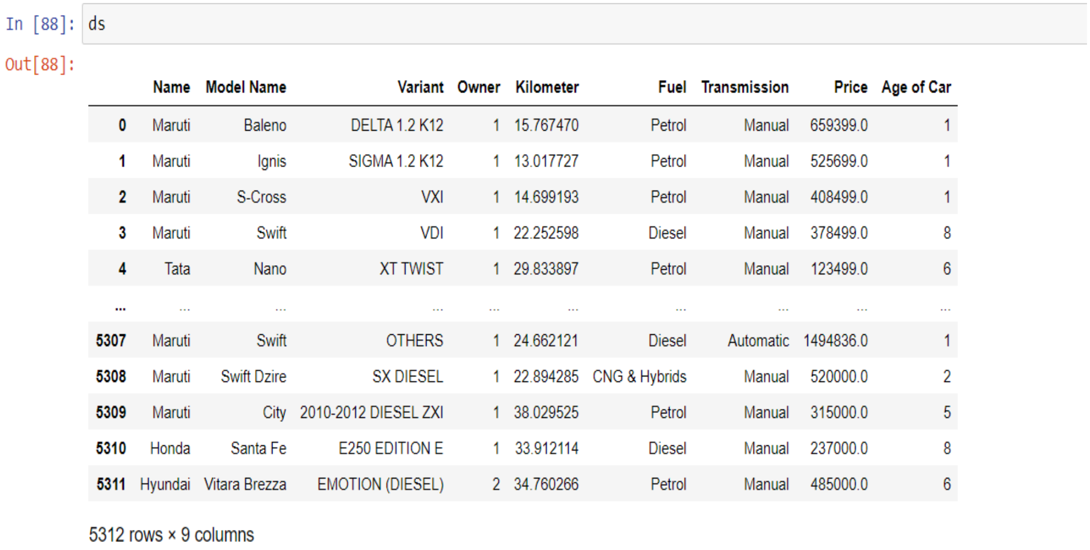
➔ This is the Main Formulae that we used while doing the regression problem

- Data Sources and their formats

➔ Data Sources are different used cars website from which I scraped all the important data. They where in html format I converted them into list and later converted them into dictionary and later to dataframe format.

➔ The necessary things in this data are the company Name, Model Name, Variant, Kilometer run, No.of Owner, Transmission and the Price of the car.

➔ After Scraping the Cars from 2 Websites olx and Cars24 I created a dataset of 5312 rows and 9 columns



```
In [88]: ds
Out[88]:
```

	Name	Model Name	Variant	Owner	Kilometer	Fuel	Transmission	Price	Age of Car
0	Maruti	Baleno	DELTA 1.2 K12	1	15.767470	Petrol	Manual	659399.0	1
1	Maruti	Ignis	SIGMA 1.2 K12	1	13.017727	Petrol	Manual	525699.0	1
2	Maruti	S-Cross	VXI	1	14.699193	Petrol	Manual	408499.0	1
3	Maruti	Swift	VDI	1	22.252598	Diesel	Manual	378499.0	8
4	Tata	Nano	XT TWIST	1	29.833897	Petrol	Manual	123499.0	6
...	...	...	...	...	...	...	...	...	...
5307	Maruti	Swift	OTHERS	1	24.662121	Diesel	Automatic	1494836.0	1
5308	Maruti	Swift Dzire	SX DIESEL	1	22.894285	CNG & Hybrids	Manual	520000.0	2
5309	Maruti	City	2010-2012 DIESEL ZXI	1	38.029525	Petrol	Manual	315000.0	5
5310	Honda	Santa Fe	E250 EDITION E	1	33.912114	Diesel	Manual	237000.0	8
5311	Hyundai	Vitara Brezza	EMOTION (DIESEL)	2	34.760266	Petrol	Manual	485000.0	6

5312 rows × 9 columns

➔

➔ I cleaned the data using Regular Expression by removing Unwanted things and replace them with important data

- Data Inputs- Logic- Output Relationships

➔ The relation between the Independent variable and the dependent variable is that from all the columns we have to predict the price of the used cars

➔ By using machine learning model we have to teach the Model to predict the price of the used car by using all the independent columns such as Company Name, Model Name, Variant, No.of Owner, Age of the Car, Kilometer Run, Fuel Used and the transmission.

- Hardware and Software Requirements and Tools Used
- **<u>Hardware Used:</u>**

➔ CPU – AMD RYZEN 5800HX

➔ RAM – 16GB

➔ STORAGE – 1 TB SSD

- **Software Used:**
  - ➔ OS – Windows 10
  - ➔ IDE – JUPYTER NOTEBOOK

- **Python Libraries Used:**
  - ➔ Sklearn – Machine Learning Library
  - ➔ Pandas – Panel and Data
  - ➔ Numpy – Numeric Python
  - ➔ Seaborn – Visualization
  - ➔ Matplotlib.pyplot – Visualization
  - ➔ Joblib – Saving the model
  - ➔ XGBOOST – Boosting Algorithm

Model/s Development and Evaluation

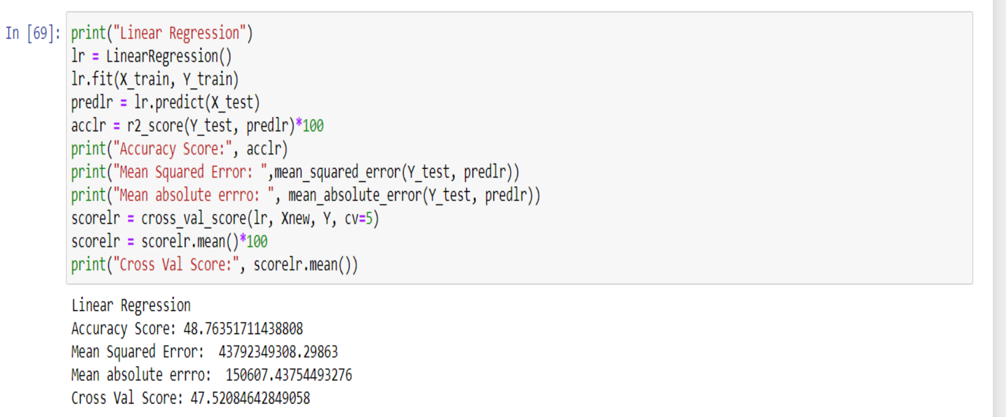
- Identification of possible problem-solving approaches (methods)
  - ➔ First I loaded the data in a Temp variable ds then I searched for Null values in the dataset I found out that there are 237 missing values in the transmission column so I filled the Nan rows with the mode of the column that is Manual.
  - ➔ After Removing all the Nan values and replacing them with the mode of the column I found another problem with the dataset In the Name column that is Maruti and Maruti Suzuki are the Same Company but they are categorised differently so by using replace method I replace all the Maruti Suzuki to Maruti now they are categorised as same there were also other Companies so I fixed them using this method
  - ➔ Then I came to model Column in that many Model of Same name are uppercase and Lower case so I changed all the model name to uppercase and reduced the Categories in the Dataset
  - ➔ After this my model was good to perform Machine Learning task and I performed Label Encoder to Convert Categories to Number as our model can only understand numbers.

- Testing of Identified Approaches (Algorithms)
- **Algorithms Used:**

➔ **Linear Regression**
➔ **Lasso Regression**
➔ **Ridge Regression**
➔ **Decision Tree Regression**
➔ **Random Forest Regression**
➔ **AdaBoost Regression**
➔ **SVR**
➔ **Kneighbors Regression**
➔ **XGBRegressor**
➔ **XGBRFRegressor**

- Run and Evaluate selected models

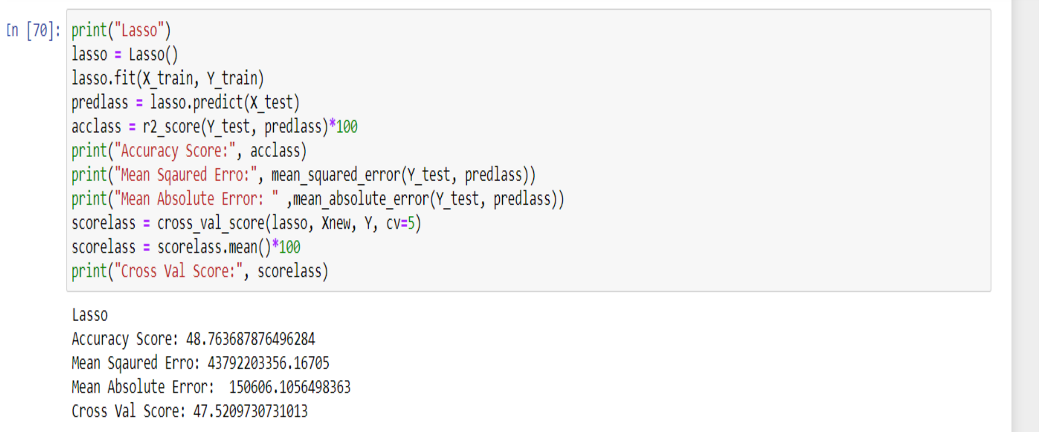
➔ Linear Regression



```
In [69]: print("Linear Regression")
         lr = LinearRegression()
         lr.fit(X_train, Y_train)
         predlr = lr.predict(X_test)
         acclr = r2_score(Y_test, predlr)*100
         print("Accuracy Score:", acclr)
         print("Mean Squared Error: ",mean_squared_error(Y_test, predlr))
         print("Mean absolute errro: ", mean_absolute_error(Y_test, predlr))
         scorelr = cross_val_score(lr, Xnew, Y, cv=5)
         scorelr = scorelr.mean()*100
         print("Cross Val Score:", scorelr.mean())
```

```
Linear Regression
Accuracy Score: 48.76351711438808
Mean Squared Error:  43792349308.29863
Mean absolute errro:  150607.43754493276
Cross Val Score: 47.52084642849058
```

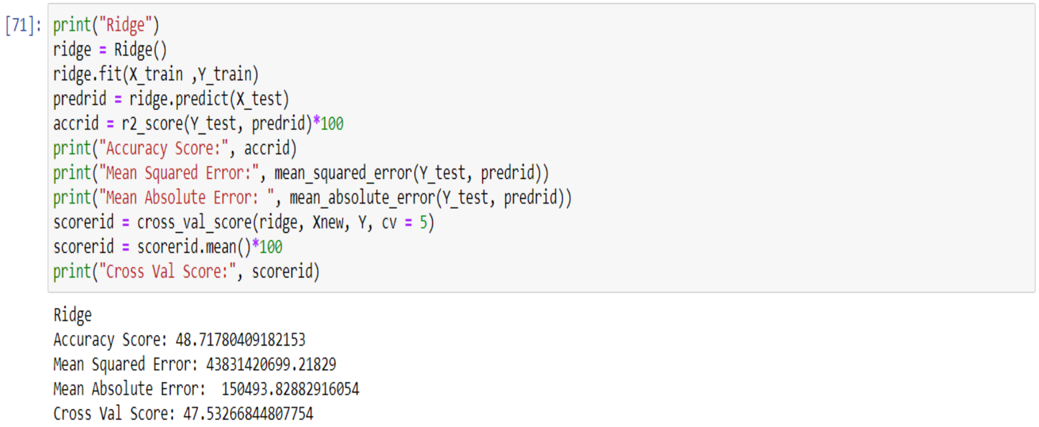
➔ Lasso Regression



```
In [70]: print("Lasso")
         lasso = Lasso()
         lasso.fit(X_train, Y_train)
         predlass = lasso.predict(X_test)
         acclass = r2_score(Y_test, predlass)*100
         print("Accuracy Score:", acclass)
         print("Mean Sqaured Erro:", mean_squared_error(Y_test, predlass))
         print("Mean Absolute Error: " ,mean_absolute_error(Y_test, predlass))
         scorelass = cross_val_score(lasso, Xnew, Y, cv=5)
         scorelass = scorelass.mean()*100
         print("Cross Val Score:", scorelass)
```

```
Lasso
Accuracy Score: 48.763687876496284
Mean Sqaured Erro: 43792203356.16705
Mean Absolute Error:  150606.1056498363
Cross Val Score: 47.5209730731013
```

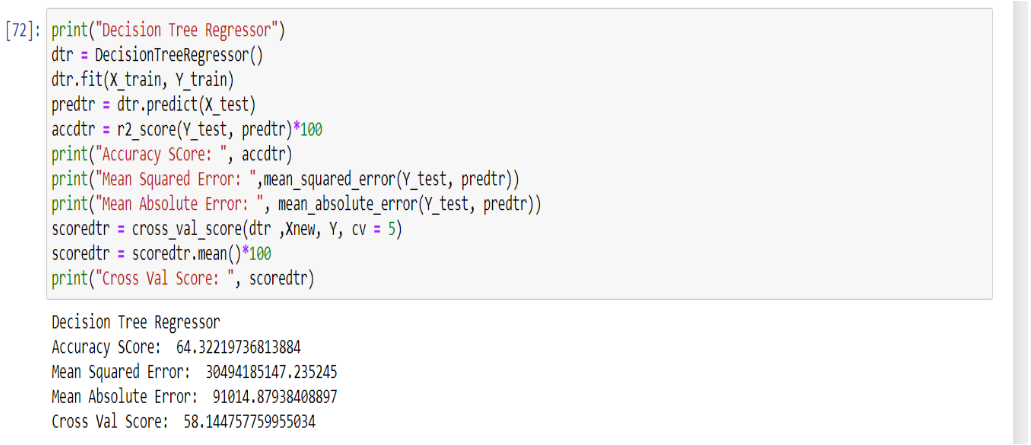
➔ Ridge Regression



```
[71]: print("Ridge")
      ridge = Ridge()
      ridge.fit(X_train ,Y_train)
      predrid = ridge.predict(X_test)
      accrid = r2_score(Y_test, predrid)*100
      print("Accuracy Score:", accrid)
      print("Mean Squared Error:", mean_squared_error(Y_test, predrid))
      print("Mean Absolute Error: ", mean_absolute_error(Y_test, predrid))
      scorerid = cross_val_score(ridge, Xnew, Y, cv = 5)
      scorerid = scorerid.mean()*100
      print("Cross Val Score:", scorerid)

      Ridge
      Accuracy Score: 48.71780409182153
      Mean Squared Error: 43831420699.21829
      Mean Absolute Error:  150493.82882916054
      Cross Val Score: 47.53266844807754
```

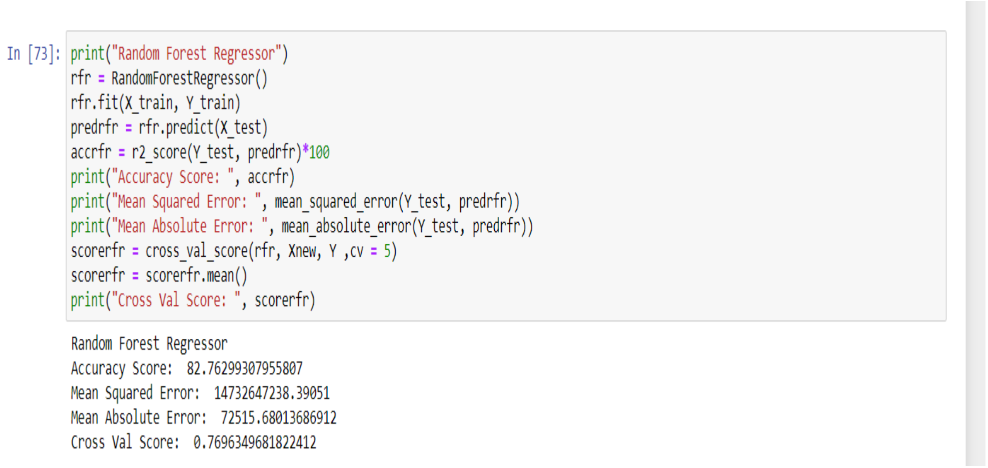
➔ Decision Tree Regression



```
[72]: print("Decision Tree Regressor")
      dtr = DecisionTreeRegressor()
      dtr.fit(X_train, Y_train)
      predtr = dtr.predict(X_test)
      accdtr = r2_score(Y_test, predtr)*100
      print("Accuracy SCore: ", accdtr)
      print("Mean Squared Error: ",mean_squared_error(Y_test, predtr))
      print("Mean Absolute Error: ", mean_absolute_error(Y_test, predtr))
      scoredtr = cross_val_score(dtr ,Xnew, Y, cv = 5)
      scoredtr = scoredtr.mean()*100
      print("Cross Val Score: ", scoredtr)

      Decision Tree Regressor
      Accuracy SCore:  64.32219736813884
      Mean Squared Error:  30494185147.235245
      Mean Absolute Error:  91014.87938408897
      Cross Val Score:  58.144757759955034
```

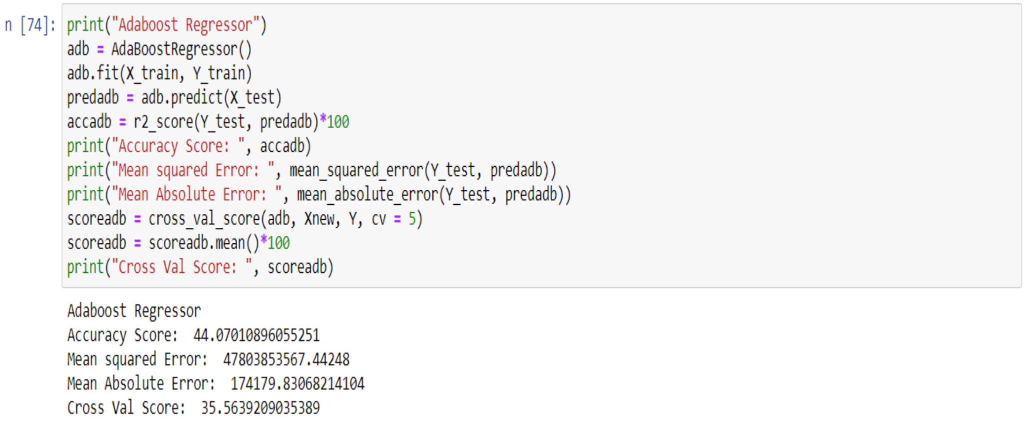
➔ Random Forest Regression



```
In [73]: print("Random Forest Regressor")
         rfr = RandomForestRegressor()
         rfr.fit(X_train, Y_train)
         predrfr = rfr.predict(X_test)
         accrfr = r2_score(Y_test, predrfr)*100
         print("Accuracy Score: ", accrfr)
         print("Mean Squared Error: ", mean_squared_error(Y_test, predrfr))
         print("Mean Absolute Error: ", mean_absolute_error(Y_test, predrfr))
         scorerfr = cross_val_score(rfr, Xnew, Y ,cv = 5)
         scorerfr = scorerfr.mean()
         print("Cross Val Score: ", scorerfr)

         Random Forest Regressor
         Accuracy Score:  82.76299307955807
         Mean Squared Error:  14732647238.39051
         Mean Absolute Error:  72515.68013686912
         Cross Val Score:  0.7696349681822412
```

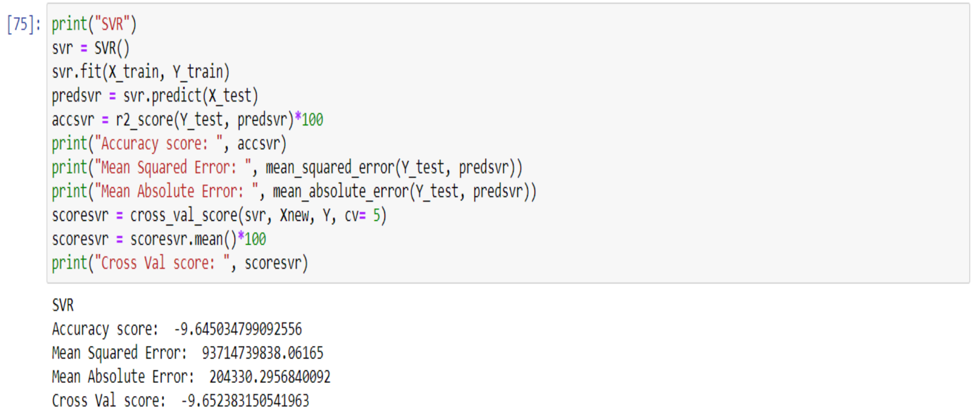
➔ AdaBoost Regression



```
n [74]: print("Adaboost Regressor")
        adb = AdaBoostRegressor()
        adb.fit(X_train, Y_train)
        predadb = adb.predict(X_test)
        accadb = r2_score(Y_test, predadb)*100
        print("Accuracy Score: ", accadb)
        print("Mean squared Error: ", mean_squared_error(Y_test, predadb))
        print("Mean Absolute Error: ", mean_absolute_error(Y_test, predadb))
        scoreadb = cross_val_score(adb, Xnew, Y, cv = 5)
        scoreadb = scoreadb.mean()*100
        print("Cross Val Score: ", scoreadb)

        Adaboost Regressor
        Accuracy Score:  44.07010896055251
        Mean squared Error:  47803853567.44248
        Mean Absolute Error:  174179.83068214104
        Cross Val Score:  35.5639209035389
```

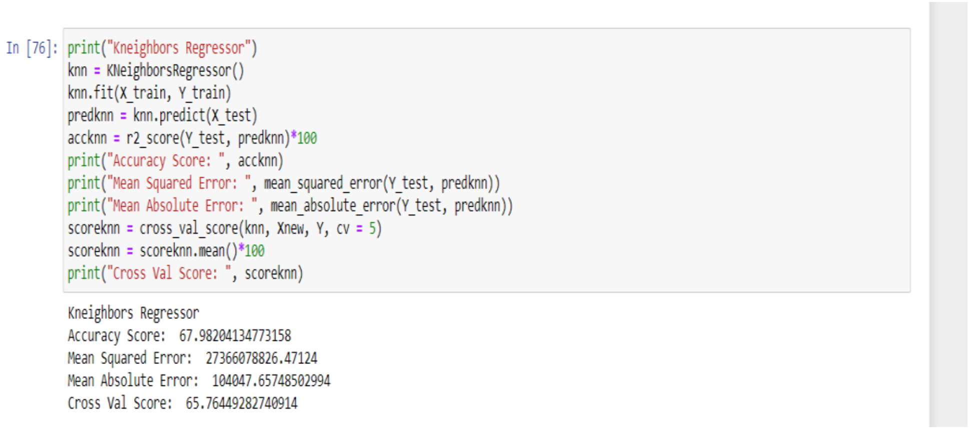
➔ SVR



```
[75]: print("SVR")
      svr = SVR()
      svr.fit(X_train, Y_train)
      predsvr = svr.predict(X_test)
      accsvr = r2_score(Y_test, predsvr)*100
      print("Accuracy score: ", accsvr)
      print("Mean Squared Error: ", mean_squared_error(Y_test, predsvr))
      print("Mean Absolute Error: ", mean_absolute_error(Y_test, predsvr))
      scoresvr = cross_val_score(svr, Xnew, Y, cv= 5)
      scoresvr = scoresvr.mean()*100
      print("Cross Val score: ", scoresvr)

      SVR
      Accuracy score:  -9.645034799092556
      Mean Squared Error:  93714739838.06165
      Mean Absolute Error:  204330.2956840092
      Cross Val score:  -9.652383150541963
```

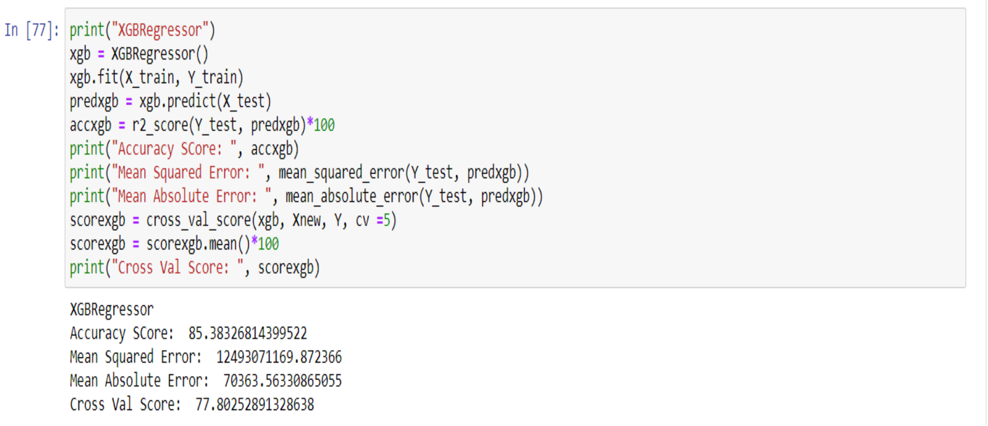
➔ Kneighbors Regression



```
In [76]: print("Kneighbors Regressor")
         knn = KNeighborsRegressor()
         knn.fit(X_train, Y_train)
         predknn = knn.predict(X_test)
         accknn = r2_score(Y_test, predknn)*100
         print("Accuracy Score: ", accknn)
         print("Mean Squared Error: ", mean_squared_error(Y_test, predknn))
         print("Mean Absolute Error: ", mean_absolute_error(Y_test, predknn))
         scoreknn = cross_val_score(knn, Xnew, Y, cv = 5)
         scoreknn = scoreknn.mean()*100
         print("Cross Val Score: ", scoreknn)

         Kneighbors Regressor
         Accuracy Score:  67.98204134773158
         Mean Squared Error:  27366078826.47124
         Mean Absolute Error:  104047.65748502994
         Cross Val Score:  65.76449282740914
```

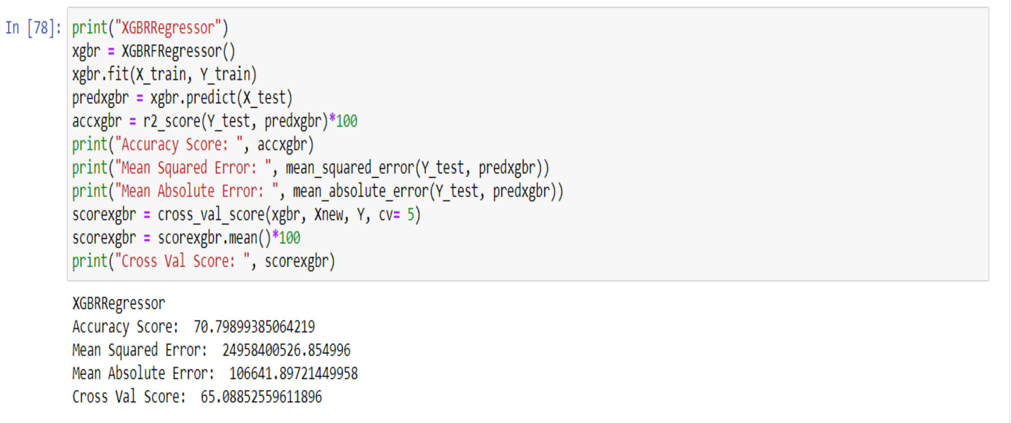
➔ XGBRegressor



```
In [77]: print("XGBRegressor")
         xgb = XGBRegressor()
         xgb.fit(X_train, Y_train)
         predxgb = xgb.predict(X_test)
         accxgb = r2_score(Y_test, predxgb)*100
         print("Accuracy SCore: ", accxgb)
         print("Mean Squared Error: ", mean_squared_error(Y_test, predxgb))
         print("Mean Absolute Error: ", mean_absolute_error(Y_test, predxgb))
         scorexgb = cross_val_score(xgb, Xnew, Y, cv =5)
         scorexgb = scorexgb.mean()*100
         print("Cross Val Score: ", scorexgb)
```

```
XGBRegressor
Accuracy SCore:  85.38326814399522
Mean Squared Error:  12493071169.872366
Mean Absolute Error:  70363.56330865055
Cross Val Score:  77.80252891328638
```

➔ XGBRFRegressor



```
In [78]: print("XGBRRegressor")
         xgbr = XGBRFRegressor()
         xgbr.fit(X_train, Y_train)
         predxgbr = xgbr.predict(X_test)
         accxgbr = r2_score(Y_test, predxgbr)*100
         print("Accuracy Score: ", accxgbr)
         print("Mean Squared Error: ", mean_squared_error(Y_test, predxgbr))
         print("Mean Absolute Error: ", mean_absolute_error(Y_test, predxgbr))
         scorexgbr = cross_val_score(xgbr, Xnew, Y, cv= 5)
         scorexgbr = scorexgbr.mean()*100
         print("Cross Val Score: ", scorexgbr)
```

```
XGBRRegressor
Accuracy Score:  70.79899385064219
Mean Squared Error:  24958400526.854996
Mean Absolute Error:  106641.89721449958
Cross Val Score:  65.08852559611896
```

- Key Metrics for success in solving problem under consideration

  ➔ Three metrics Used:

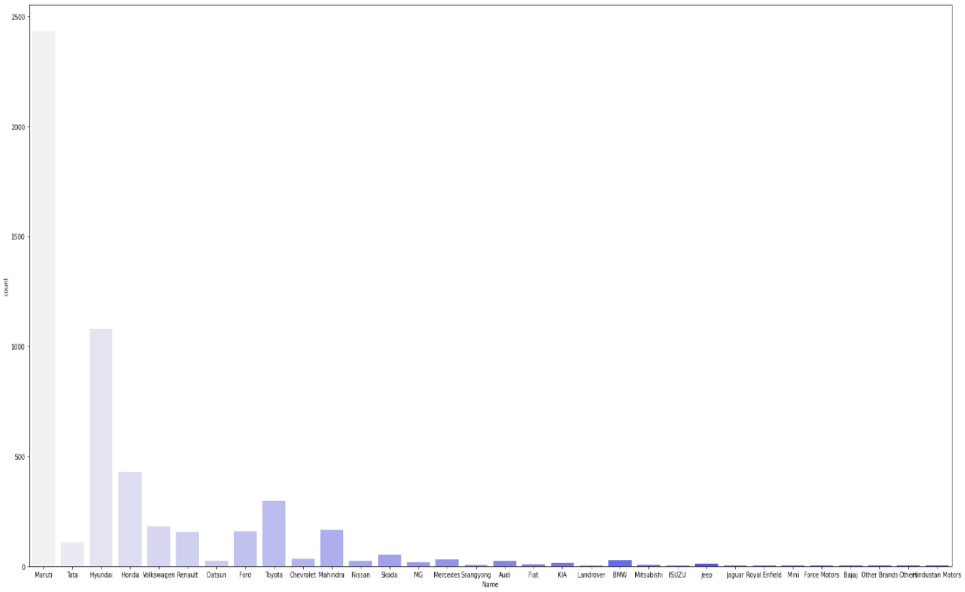
  1. R2 Score
  2. Mean Squared Error
  3. Mean Absolute Error

1. R2 Score:- R-squared (R^2) is a statistical measure that **represents the proportion of the variance for a dependent variable that's explained by an independent variable** or variables in a regression model.

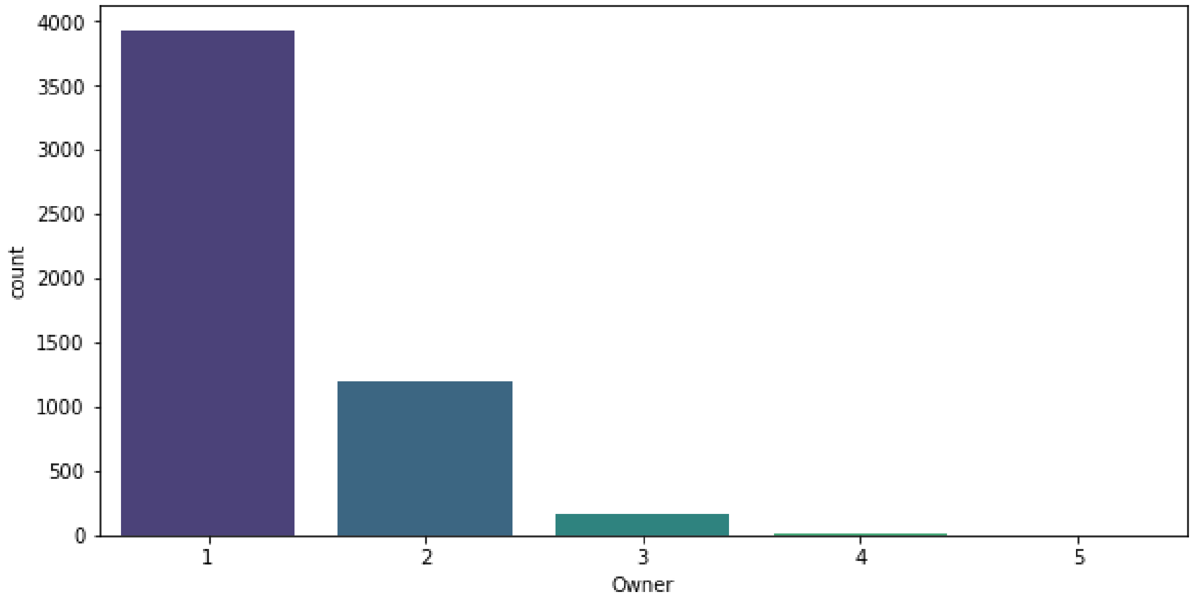
2. Mean Square error:- The mean squared error (MSE) tells **you how close a regression line is to a set of points**. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them.

3. Mean Absolute Error:- , mean absolute error (MAE) is **a measure of errors between paired observations expressing the same phenomenon**. ... This is known as a scale-dependent accuracy measure and therefore cannot be used to make comparisons between series using different scales.

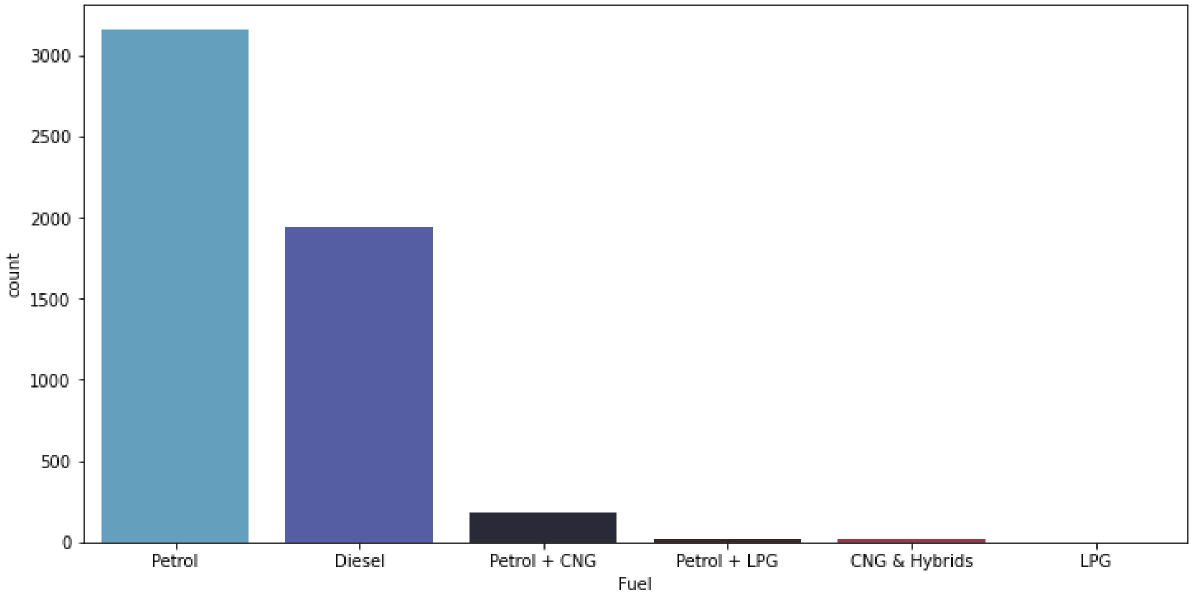
- Visualizations



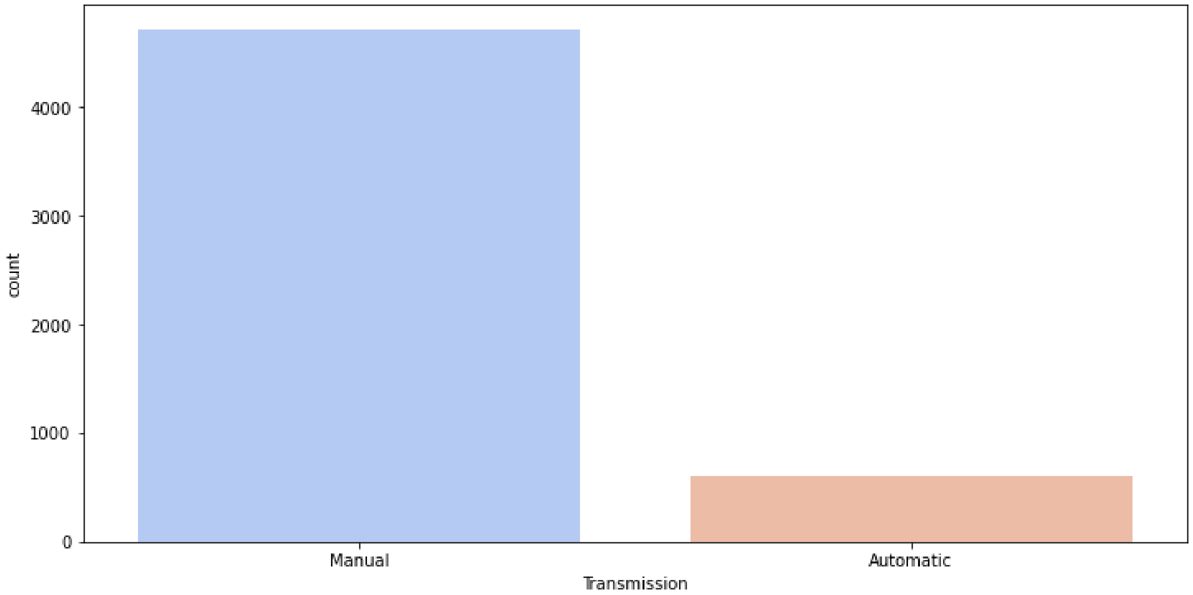
from above graph we can see that most data is from Maruti Company followed by Hyundai



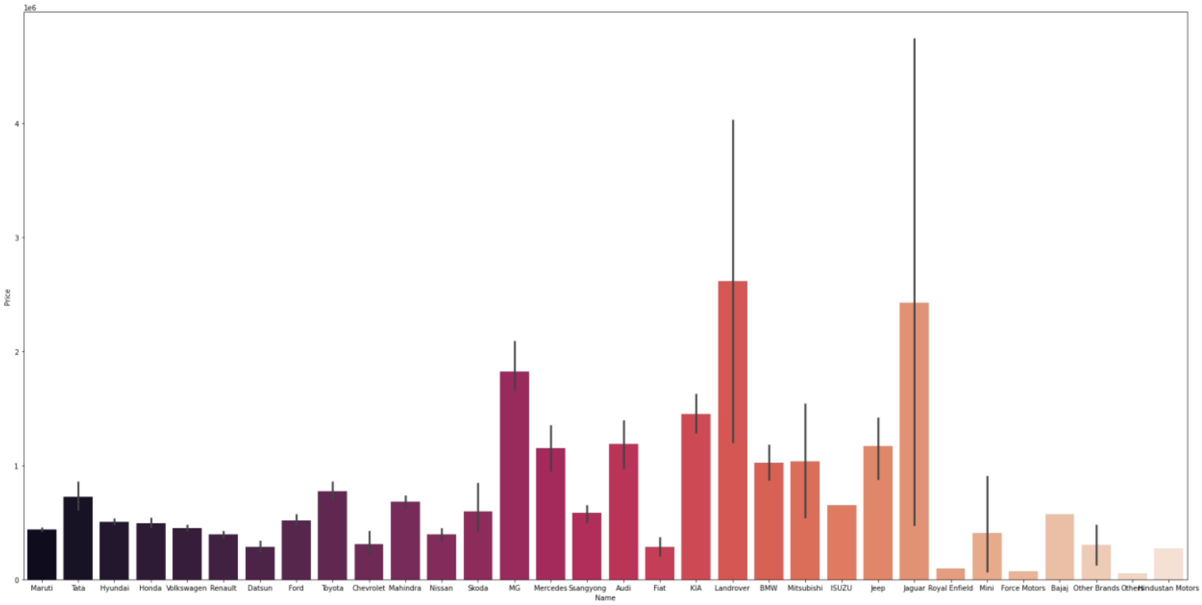
from above we can see that in our dataset the car which are present has been Sold by 1st Owner followed by 2nd Owner



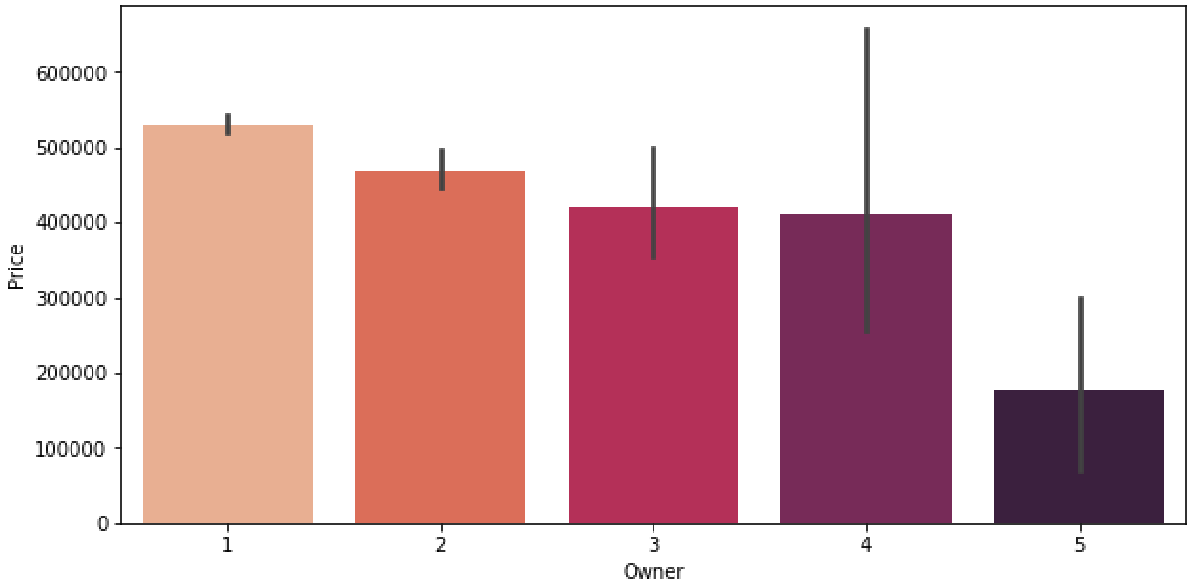
from above we can see that most of the Cars in our dataset has Fuel type as Petrol followed By Diesel and We also has some Hybrid Cars in our dataset



from above we can see that most of the Cars in our dataset has Manual Transmission

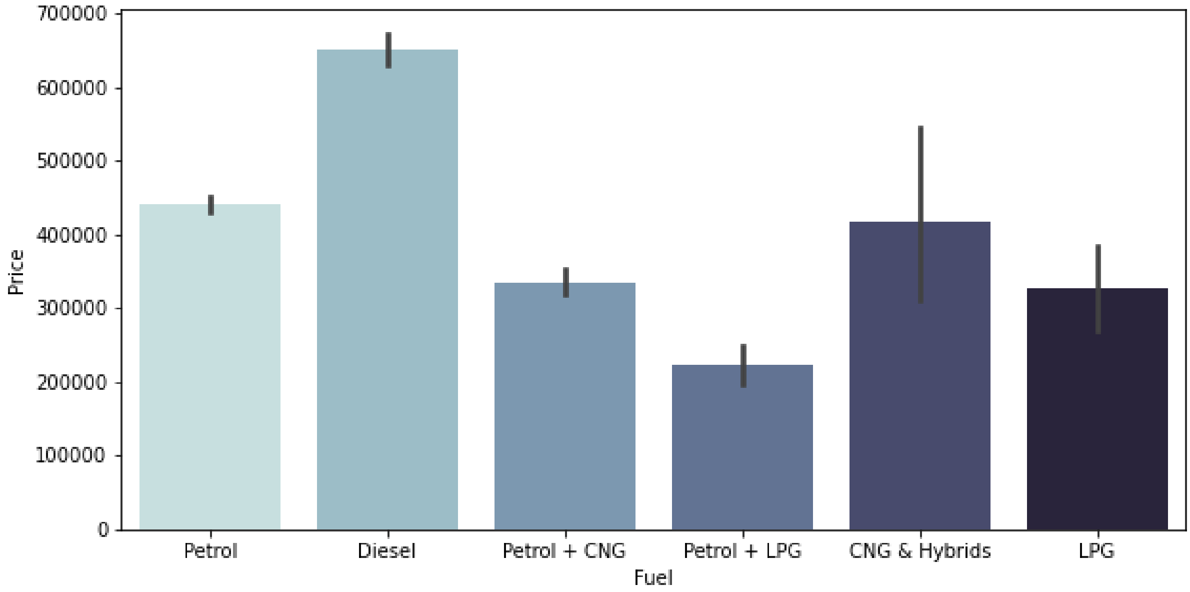


from above we can see that Toyota, Jaguar, Skoda, Honda has better selling Price in the market than other Car Companies

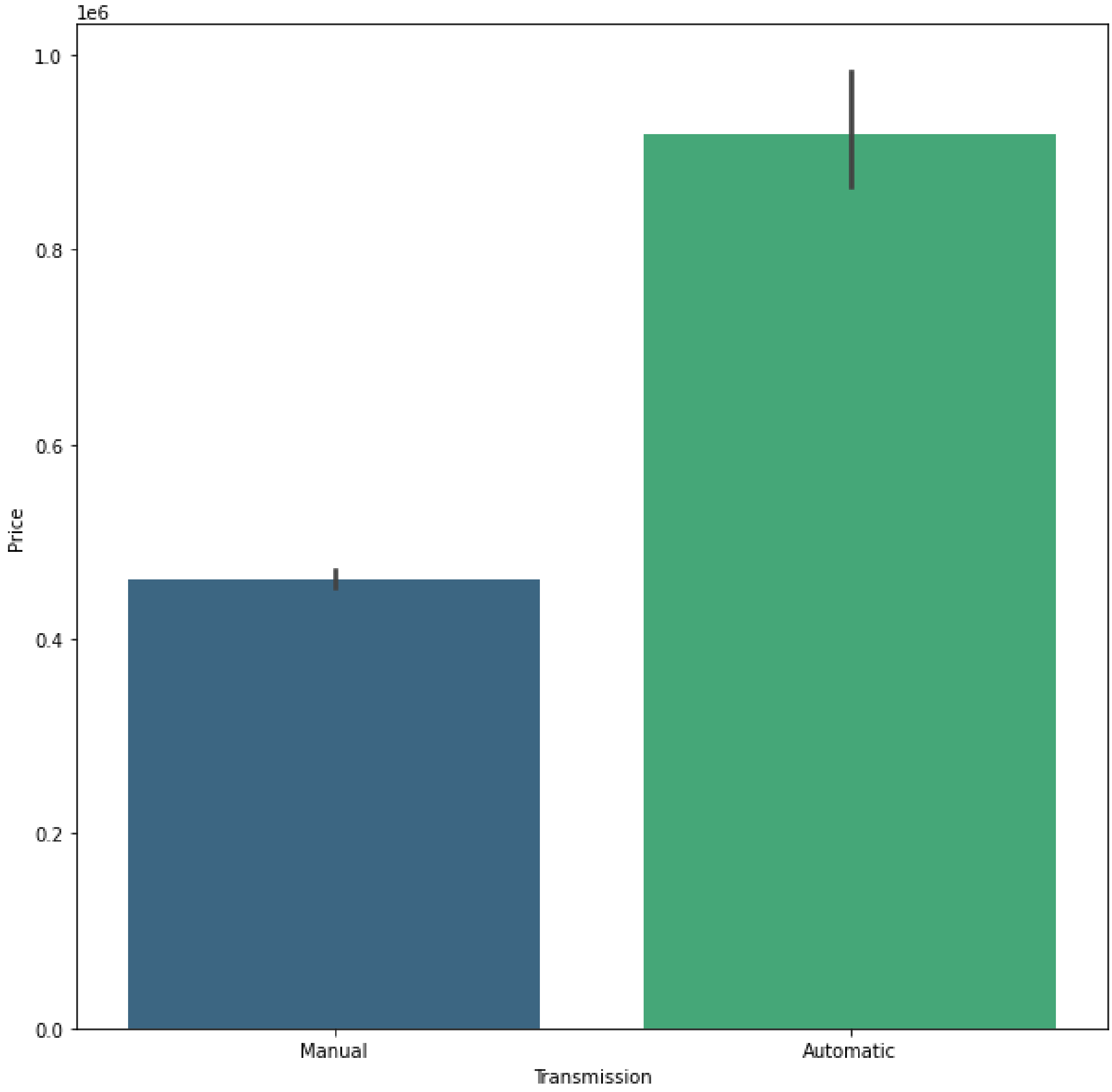


from above we can see that selling price of 2 owner is Highest followed by 1 owner

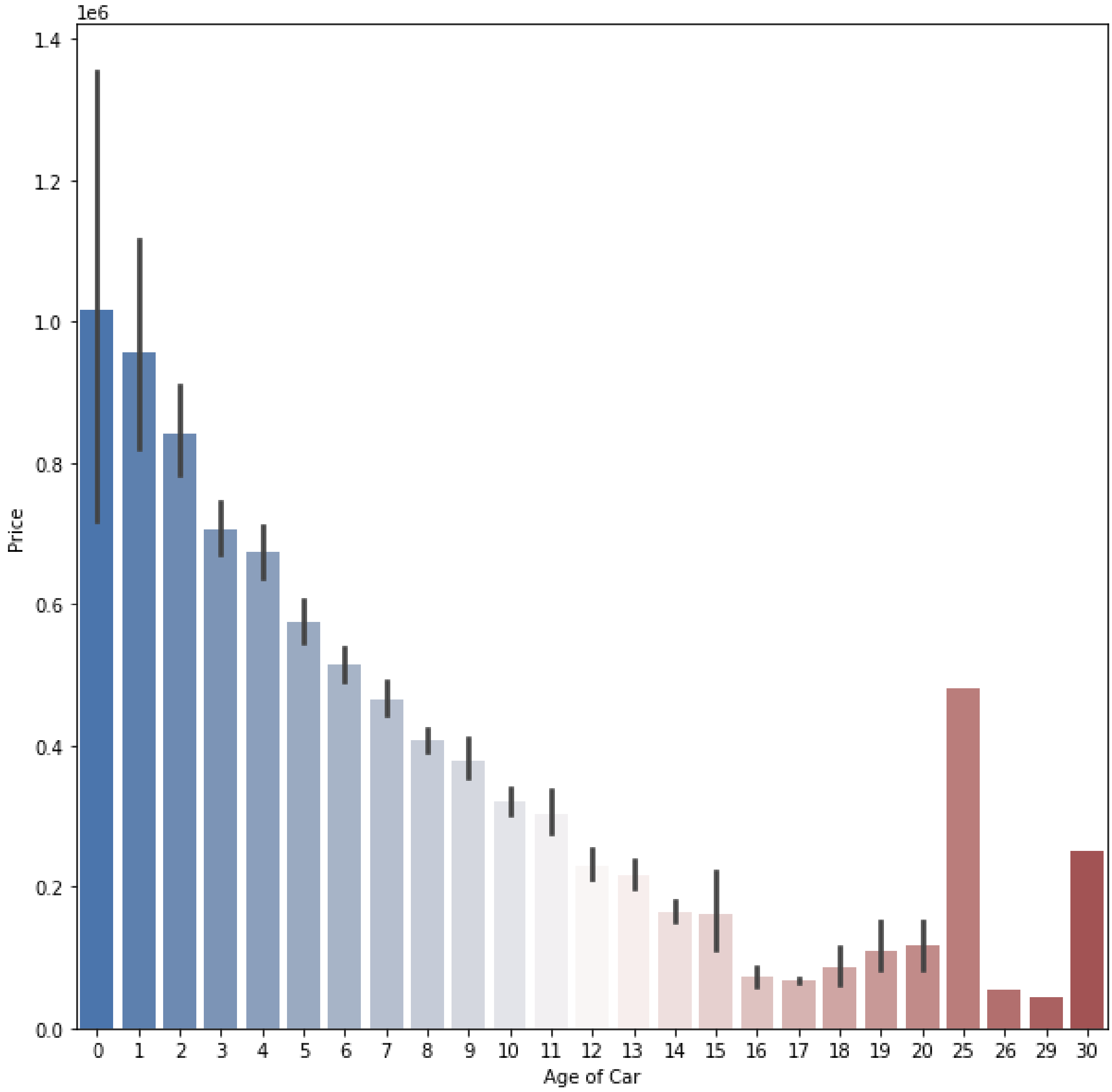
we can also see if the owner exceeds 2 then the selling price goes down drastically



from above we can see Diesel engine has higher selling price as compared to other type of Engines



from above we can see that Automatic Transmission has Higher price than Manual Transmission i.e Technology has Higher price



from above we can see that as the age of the car increases the price of the car decreases its natural but we have cares which is 25 but the selling value of the car is high we will see if it is a ourlier or not and then we will fix it

- Interpretation of the Results

1) From Car Company Graph we Understand that Car Companies Like Skoda, Mercedes, Jaguar, Toyota, Honda has Higher Selling Price as they are big companies more Customers are attracted to them so the selling price of this Company Cars are the Highest, These Car Companaies also has good Customer Service so most of the People tends to buy this Cars.

2) From Owner vs Price we can understand that as the number of owner increases the price of the Car decrease

3) From Fuel vs Price we can see that the price is higher for Diesel car than Petrol Because most of the People in India wants less runnning cost and using diesel as a fuel the running cost Decreases as Price of Diesel is cheaper than Petrol so Car which uses Diesel as Fuel has Higher Selling Price.

4) From Transmission vs Price we can see that the price of Automatic transmission is Higher than the Manual as in Automatic Transmission user don't need to Switch gear which is less tiring to the driver and it is easy to ride in city, traffic roads than the Manual gear cars

5) From Age vs Price we see that newer the car higher is the selling price as the age of the car increases the selling price of the car decreases.

CONCLUSION

- Key Findings and Conclusions of the Study
    - ➔ From this project I came to know that the Machine Learning algorithms performs well when the data is normalized before normalizing I got 60% accuracy after normalizing I got 86% accuracy.

- Learning Outcomes of the Study in respect of Data Science
    - ➔ From this project I have faced many problems that are I scraped the data from two different websites i.e olx and Cars24 so I faced some challenges while scraping the data both the website has different different sections so I need to code twice for that
    - ➔ As I normalized the data the accuracy of the model increased and after that XGBRegressor has the best accuracy and Mean Squared Error so I choose this model as the best model for this problem.

- Limitations of this work and Scope for Future Work
  - ➔ There are some limitations from this project as we have scraped only 5000 rows from the Website so we may have not got all the Car Manufacturing Companies so it may mis calculate the price of the car which are not in the dataset
  - ➔ To improve the model we can add more data of the cars which are not in the dataset.