FLIP ROBO

USED CAR PRICE PREDICTION

Submitted by:

Shubham Nalawade

ACKNOWLEDGMENT

With the covid 19 impact, we have seen lot of changes in the Automobile market. Now some cars are in demand hence making them costlier and some are not in demand hence it is cheaper. One of our clients who works with small traders, sells used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price evaluation machine learning models. So, they are looking for new machine learning models from new data.

Based on the business requirements of the Client, I have scraped the Data from the well-known e-commerce websites such as cars 24. Based on the Data collected, we will be predicting the prices of used cars. We will be building various Machine Learning models. In the end, we will see how all the machine learning models performs. And based on which we will sort the best machine learning model and hyperparameter tune the same to get the improved performance.

INTRODUCTION

Business Problem Framing.

Impact of COVID-19 on Indian automotive sector

The Indian automotive sector was already struggling in FY20. before the Covid-19 crisis. It saw an overall degrowth of nearly 18 per cent. This situation was worsened by the onset of the Covid-19 pandemic and the ongoing lockdowns across India and the rest of the world. These two years (FY20 and FY21) are challenging times for the Indian automotive sector on account of slow economic growth, negative consumer sentiment, BS-VI transition, changes to the axle load norms, liquidity crunch, low-capacity utilisation and potential bankruptcies.

The return of daily life and manufacturing activity to near normalcy in China and South Korea, along with extended lockdown in India, gives hope for a U-shaped economic recovery. Our analysis indicates that the Indian automotive sector will start to see recovery in the third quarter of FY21. We expect the industry demand to be down 15-25 per cent in FY21. With such degrowth, OEMs, dealers and suppliers with strong cash reserves and better access to capital will be better positioned to sail through.

Auto sector has been under pressure due to a mix of demand and supply factors. However, there are also some positive outcomes, which we shall look at.

- With India's GDP growth rate for FY21 being downgraded from 5% to 0% and later to (-5%), the auto sector will take a hit. Auto demand is highly sensitive to job creation and income levels and both have been impacted. CII has estimated the revenue impact at $2 billion on a monthly basis across the auto industry in India.

- Supply chain could be the worst affected. Even as China recovers, supply chain disruptions are likely to last for some more time. The problems on the Indo-China border at Ladakh are not helping matters. Domestic suppliers are chipping in but they will face an inventory surplus as demand remains tepid.
- The Unlock 1.0 will coincide with the implementation of the BS-VI norms and that would mean heavier discounts to dealers and also to customers. Even as auto companies are managing costs, the impact of discounts on profitability is going to be fairly steep.
- The real pain could be on the dealer end with most of them struggling with excess inventory and lack of funding options in the post COVID-19 scenario. The BS-VI price increases are also likely to hit auto demand.

There are two positive developments emanating from COVID-19. The China supply chain shock is forcing major investments in the “Make in India” initiative. The COVID-19 crisis has exposed chinks in the automobile business model and it could catalyse a big move towards electric vehicles (EVs). That could be the big positive for auto sector.

Conceptual Background of the Domain Problem

Understanding the above business problem, there are certain factors that will influence the automotive industries in the future. Some of them include digital technologies, changing customer preferences, electrical vehicles, intelligent ability, and technical advancements. Technologies such as artificial intelligence, machine learning, cloud computing, and internet of things will also play an important role in developing new business models. Apart from that, they enable customers to ensure a better mobility experience. In other words, technologies may impact automotive industry units significantly that will change the markets. The introduction of electrical cars and hybrid vehicles may transform the automobile industries in coming years.

Review of Literature.

As per the requirement of our client, I have scrubbed data from different used cars selling merchants websites, and so based on the data collected I have tried analysing based on what factors the used car price is decided? What is the relationship between cost of the used cars and other factors like Fuel type, Brand and Model, year the car is purchased and No. Of owners before selling? And so based on all the above consideration I have developed a model that will predict the price of the used cars.

Motivation for the Problem Undertaken

I have taken this problem based on the requirement of the client and also, with a curiosity to know how the used cars markets are at the time of pandemic.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

On importing the data that is collected I have done some initial play around to understand the data and to cleanse the data.

Data Cleansing.

On data cleansing I have detecting duplicate records in the data collected and the null values in the data.

```
In [10]: df.duplicated().sum()
Out[10]: 2260
```

As like we can see I had 2260 duplicate records out of 6451 records, I decided to drop all the duplicate records because it will not help us in creating a perfect model.

```
In [12]: df = df.drop_duplicates() #Deleting duplicates

In [13]: df.isnull().sum()

Out[13]: LOCATION          0
         YEAR              0
         BRAND             0
         MODEL             0
         VARIANT         169
         KM_DRIVEN         0
         FUEL              0
         NO_OF_OWNERS      0
         PRICE             0
         dtype: int64
```

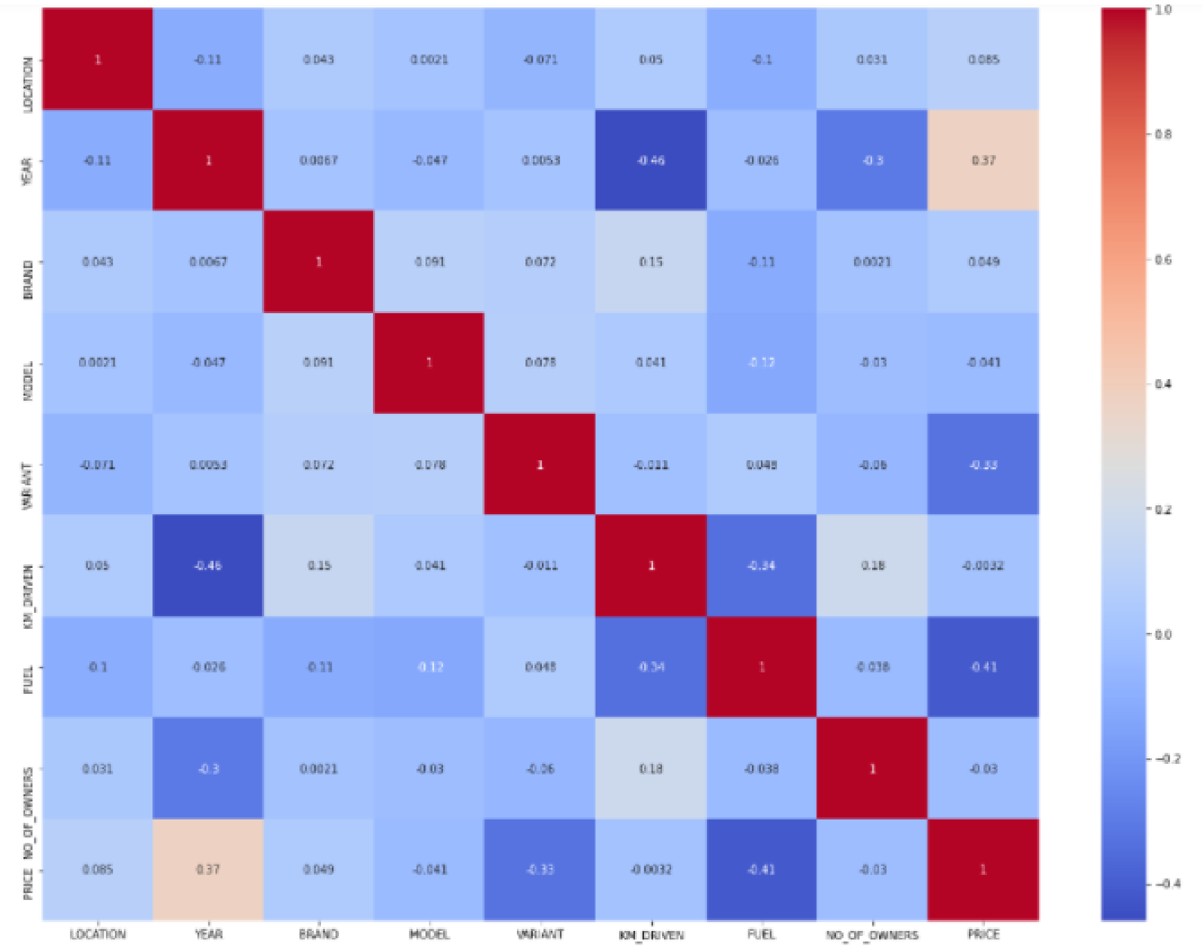
VARIANT has 169 null values. It will be replaced with NA.

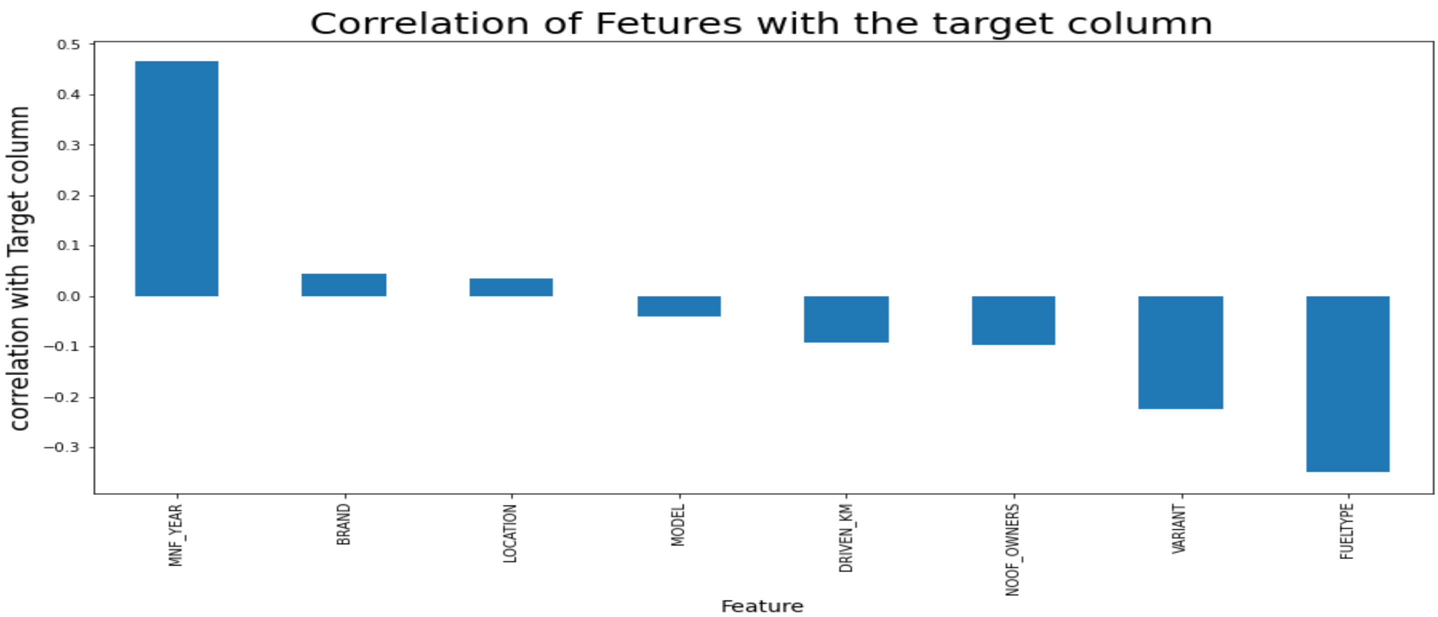
```
In [15]: df['VARIANT'] = df['VARIANT'].fillna('NA')
```

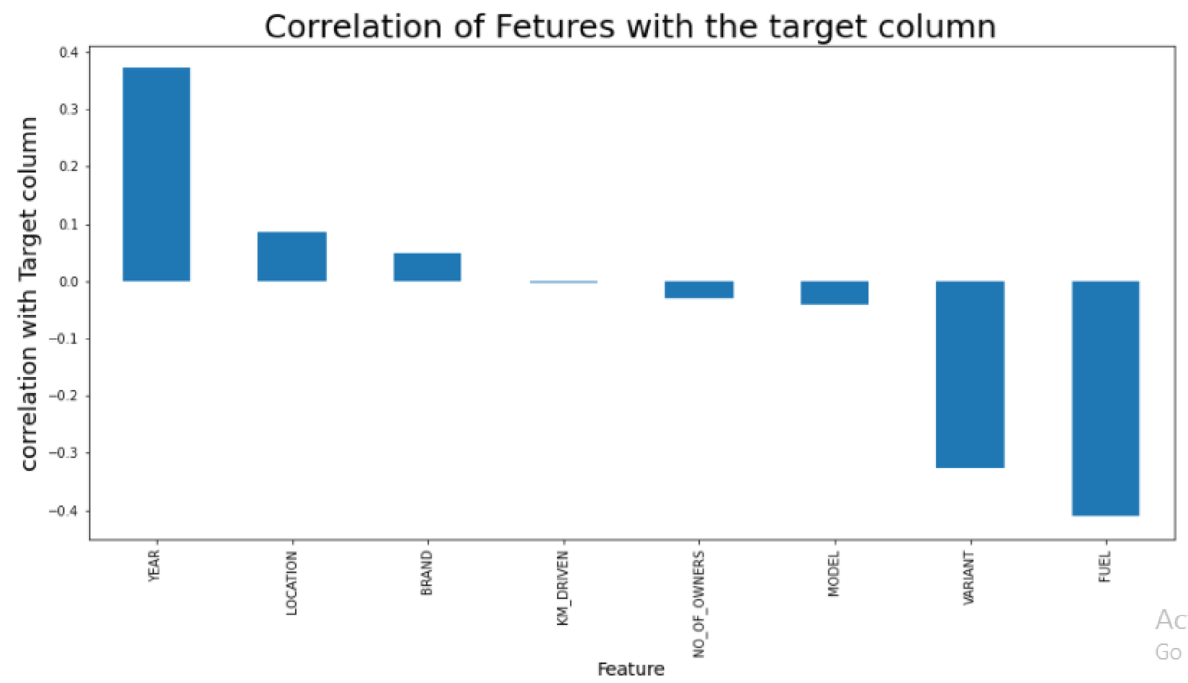
After dropping the duplicate records, I checked for null values in the data.

We have 169 null values in Variant column, I have used those null values in the model building but I have changed null values as NA, this will also help the client to predict the values on the used cars without the Variant values.

Data correlations:







Key Observations:

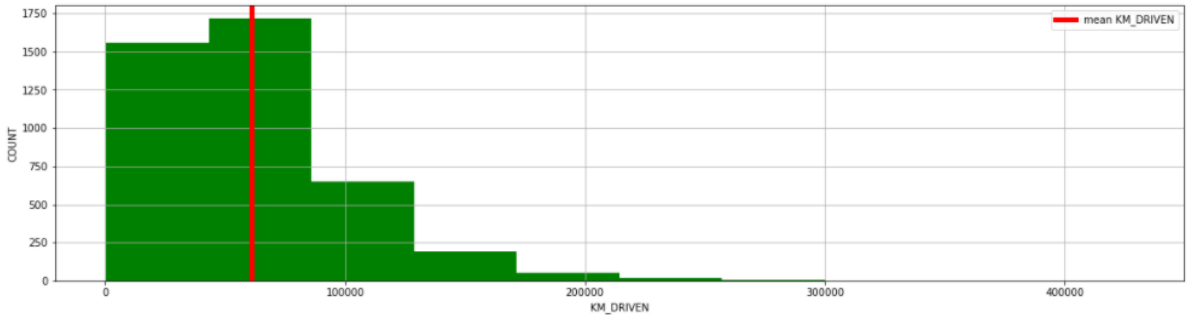
- From above we can clearly see that YEAR is positively correlated to PRICE and FUEL and VARIANT is negative correlated to PRICE.

Univariate Analysis.


```
MATHEMATICAL SUMMARY OF PRICE :
 count     4.191000e+03
mean       6.171740e+05
std        4.005465e+05
min        7.200000e+04
25%        3.498990e+05
50%        4.993990e+05
75%        7.768990e+05
max        4.725000e+06
Name: PRICE, dtype: float64
```

Key observations:

- Mean of the price is Rs: 6,17,174, the price is distributed between Rs: 72,000 to Rs: 47,25,000.
- Most of the Car price is leser than the Mean i.e Rs: 6,17,174.




```
MATHEMATICAL SUMMARY OF PRICE :
 count       4191.000000
mean       61036.793605
std        40382.806319
min          411.000000
25%        31995.000000
50%        53590.000000
75%        81983.000000
max       428123.000000
Name: KM_DRIVEN, dtype: float64
```

Key observations:

- Mean of the KM_DRIVEN is 61036.79 kms and the maximum KMS driven is 428123 kms.
- Most of the Cars comes to selling around below 61036 kilometers driven.

Data Sources and their formats

The Data is scraped from cars 24. These data are scraped and stored in a CSV format. Data contains following columns.

1. 'LOCATION' – It will tell which location the car is sold.
2. 'YEAR' – At what year the car is manufactured
3. 'BRAND' – Brand is manufacturer or which company made
4. 'MODEL' – It is basically the model of the car.
5. 'VARIANT' – Gear shift variant is (Automatic, Manual, Semi-Automatic)
6. 'KM_DRIVEN' – No of Kms driven before selling
7. 'FUELTYPE' – Petrol, diesel, CNG, LPG, Electric
8. 'NO_OF_OWNERS' – 1end, 2end or 3end car
9. 'PRICE' – our target variable that tells what is the price of the used car.

Data Preprocessing Done

On pre-processing the data, I have tried in finding out the skewness of the data and the outliers, have changed the data into numbers with label encoders.

```
In [16]: DF = df.copy()
```

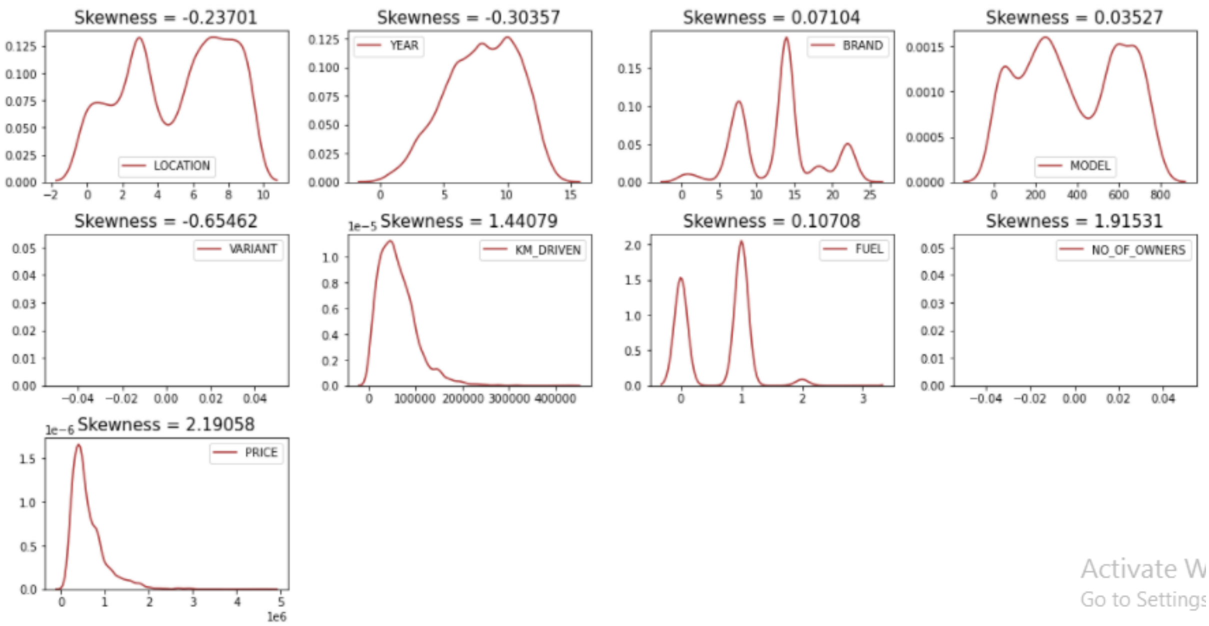
```
In [17]: from sklearn.preprocessing import LabelEncoder
         LE= LabelEncoder()
         catagorical_data = ['LOCATION' , 'YEAR', 'BRAND', 'MODEL', 'VARIANT', 'FUEL' , 'NO_OF_OWNERS' ]

         for i in catagorical_data:
             DF[i]=DF[i].astype('str')
             DF[i]=LE.fit_transform(DF[i])
```

```
In [18]: DF['PRICE'] = DF['PRICE'].str.replace(r'\D', '').astype(int)
         DF['KM_DRIVEN'] = DF['KM_DRIVEN'].str.replace(r'\D', '').astype(int)
```

After changing all the data with label encoders, I have tried in identifying skewness and outliers as follows.

We actually can see there are more skewness in the data let also see about the outliers in the data.




```
In [39]: from scipy.stats import zscore

         z= np.abs(zscore(DF))
         threshold= 3
         df_new = DF[(z < 3).all(axis=1)]
```

```
In [72]: print(f"Orginal Data {DF.shape}\nAfter Removing outliers {df_new.shape}\nThe percentage of data loss {((4191-3928)/4191)*100}%")

         Orginal Data (4191, 9)
         After Removing outliers (3928, 9)
         The percentage of data loss 6.275351944643283%
```

Loss of 6.27% Data is managabable. For better results, Let's go for outliers removed data.

We have 6.27% outliers and we also have skewness in the data. So I removed the outliers.

And final step in pre-processing I have split the data and converted it into an array with Pre-processing Standard Scaler.

```
In [41]: x_1=df_new.drop(["PRICE"], axis = 1)
         y_1=df_new.PRICE
```

```
In [42]: x_1
```

Out[42]:

	LOCATION	YEAR	BRAND	MODEL	VARIANT	KM_DRIVEN	FUEL	NO_OF_OWNERS
0	7	12	22	373	1	31235	0	0
1	7	8	13	617	0	65766	0	0
2	7	8	14	289	1	65754	0	0
3	7	12	18	609	2	48151	1	0
4	7	11	14	674	1	28944	0	0
...	...	...	...	...	...	...	...	...
6412	0	12	14	255	1	80770	1	0
6414	0	9	22	378	0	176897	0	1
6415	0	8	14	521	1	163159	0	0
6416	0	10	13	397	1	39802	0	0
6417	0	11	21	436	0	63621	1	0

3928 rows × 8 columns

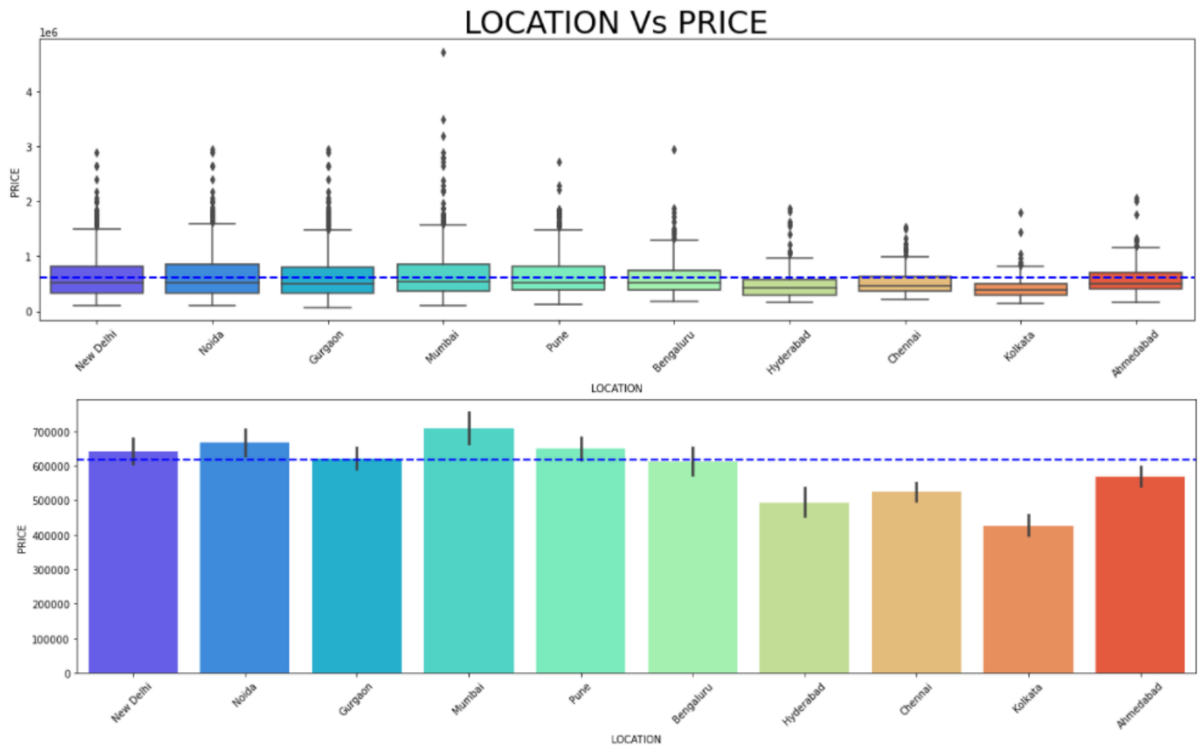
```
In [43]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         x_1 = sc.fit_transform(x_1)
```

```
In [44]: x_1
```

```
Out[44]: array([[ 0.672856  ,  1.39366677,  1.8058948 , ..., -0.7562947 ,
                 -1.16881693, -0.50611862],
                [ 0.672856  , -0.04457903,  0.06029031, ...,  0.2196599 ,
                 -1.16881693, -0.50611862],
                [ 0.672856  , -0.04457903,  0.25424637, ...,  0.21932074,
                 -1.16881693, -0.50611862],
                ...,
                [-1.73956394, -0.04457903,  0.25424637, ...,  2.97229222,
                 -1.16881693, -0.50611862],
                [-1.73956394,  0.67454387,  0.06029031, ..., -0.51416436,
                 -1.16881693, -0.50611862],
                [-1.73956394,  1.03410532,  1.61193875, ...,  0.15903546,
                  0.71355818, -0.50611862]])
```

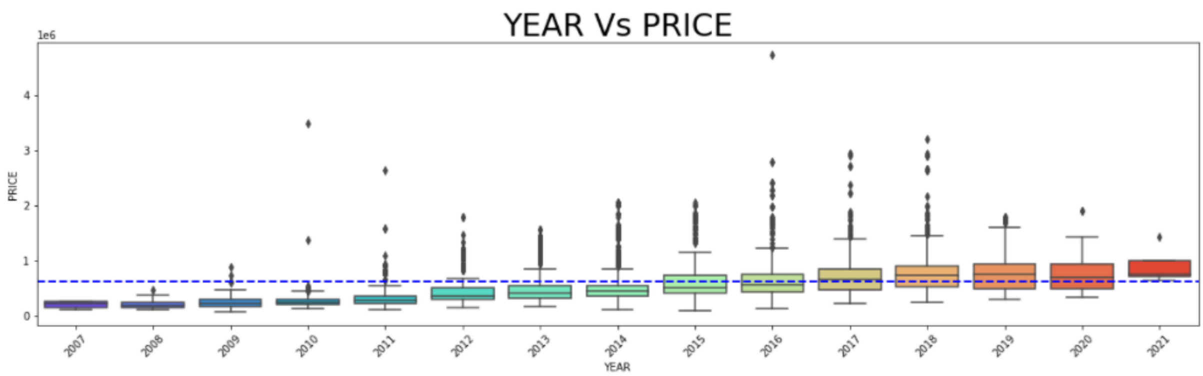
Data Inputs- Logic- Output Relationships

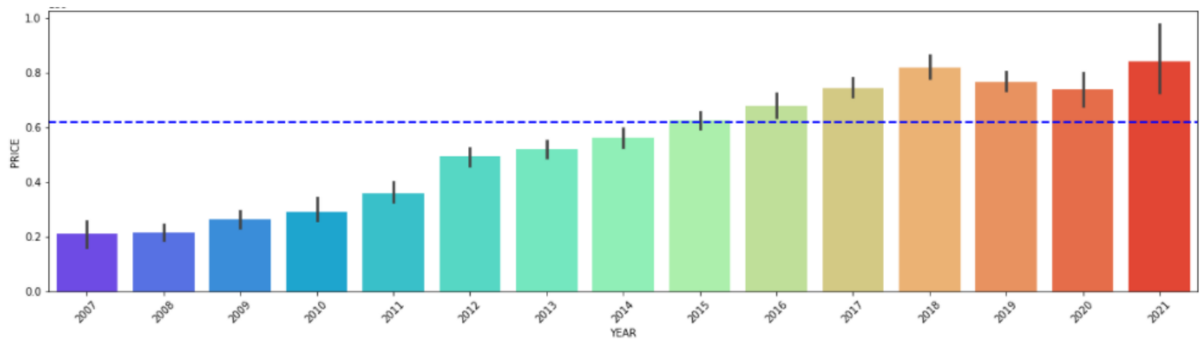
I have done some visualizations to understand the input output logic of the data collected.



Key observations:

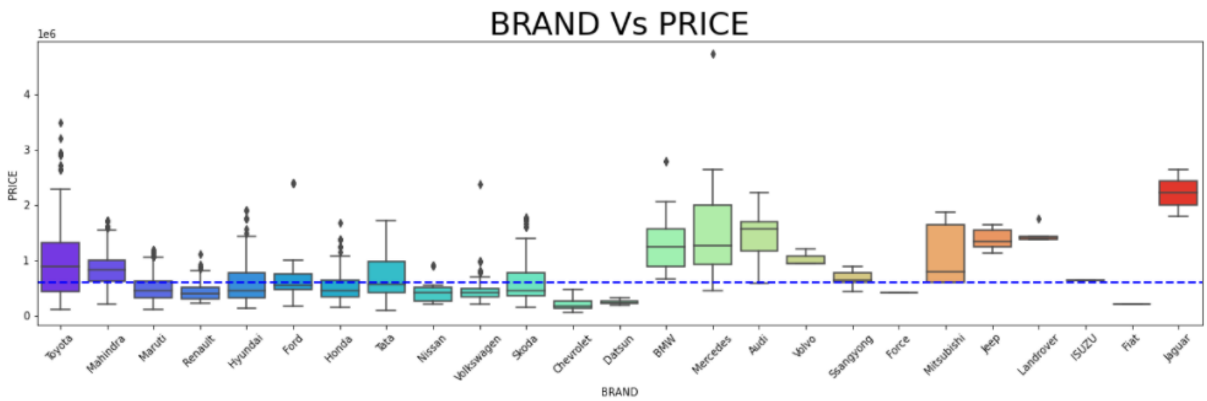
- New Delhi, Noida, Gurgaon, Mumbai,Bengaluru have the costliest cars.
- Hyderabad, Chennai, Kolkata have cars comparatively cheaper.

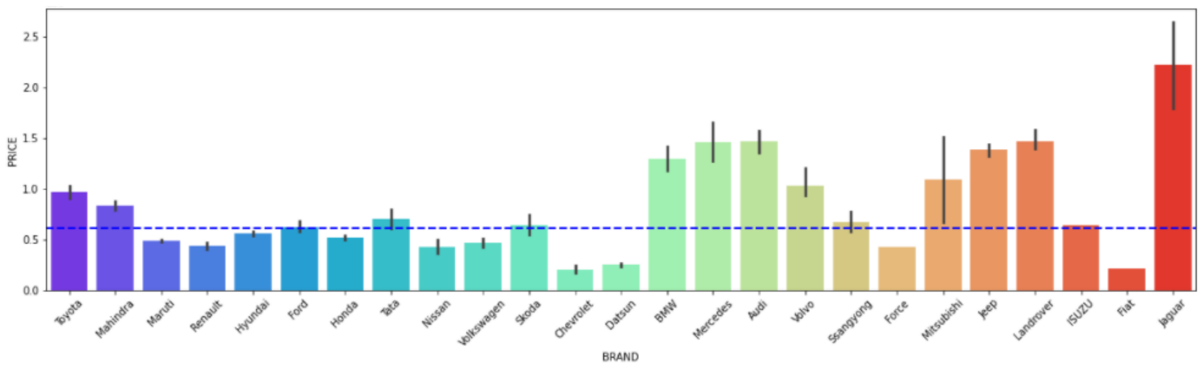




Key observations:

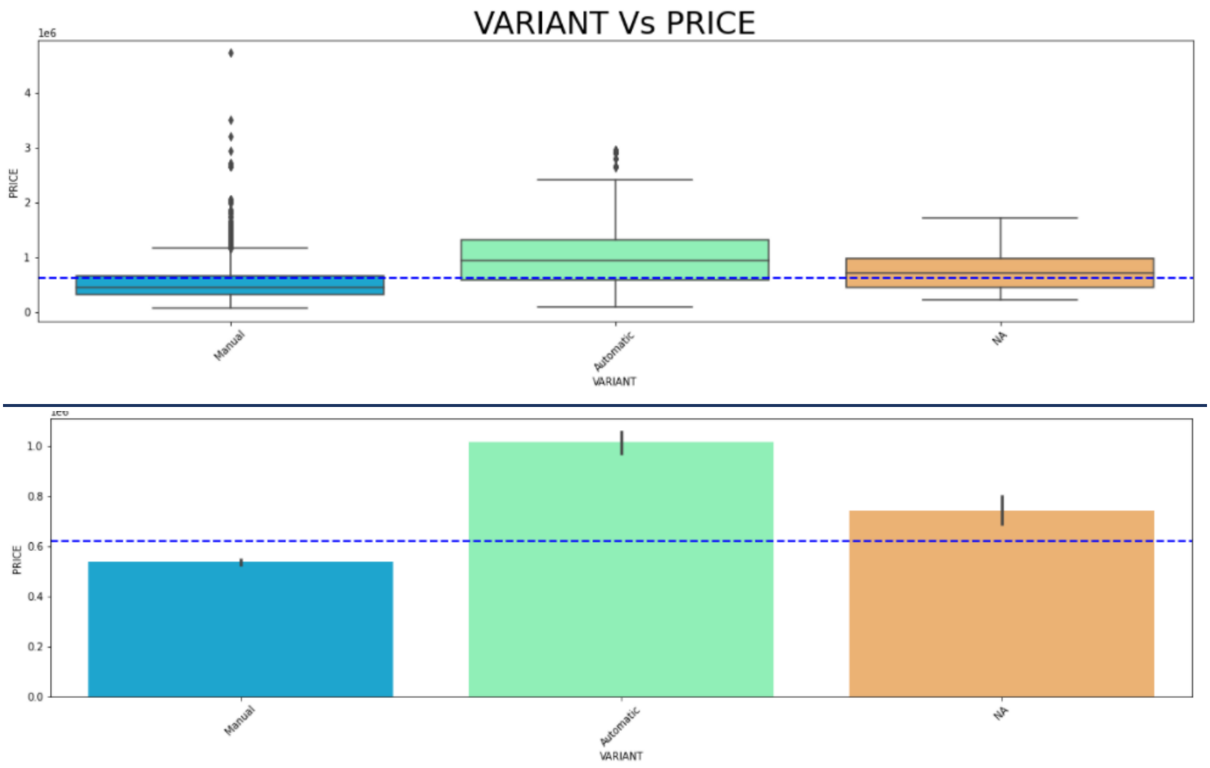
- From 2016 to 2021 cars are being sold higher in PRICE and also above average PRICE.





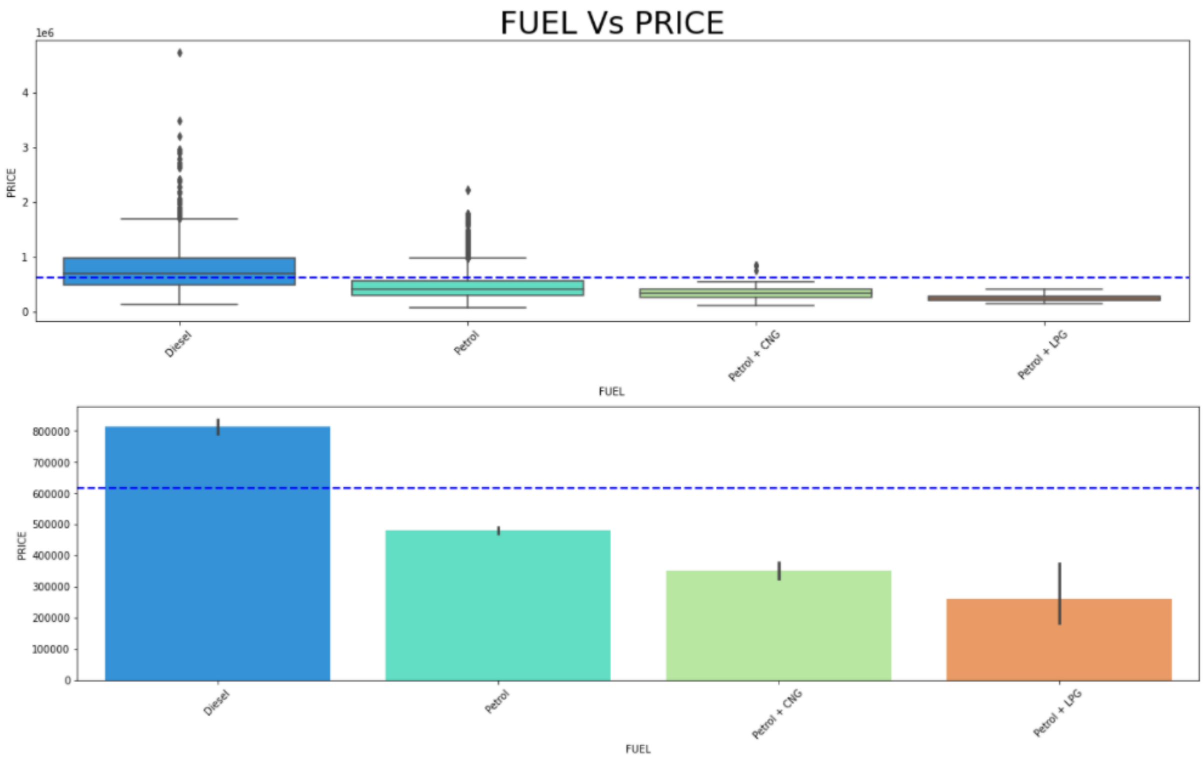
Key observations:

- Luxury Cars are obviously highly priced.
- Chevrolet, Datsun and Fiat Cars are much below than the mean price.



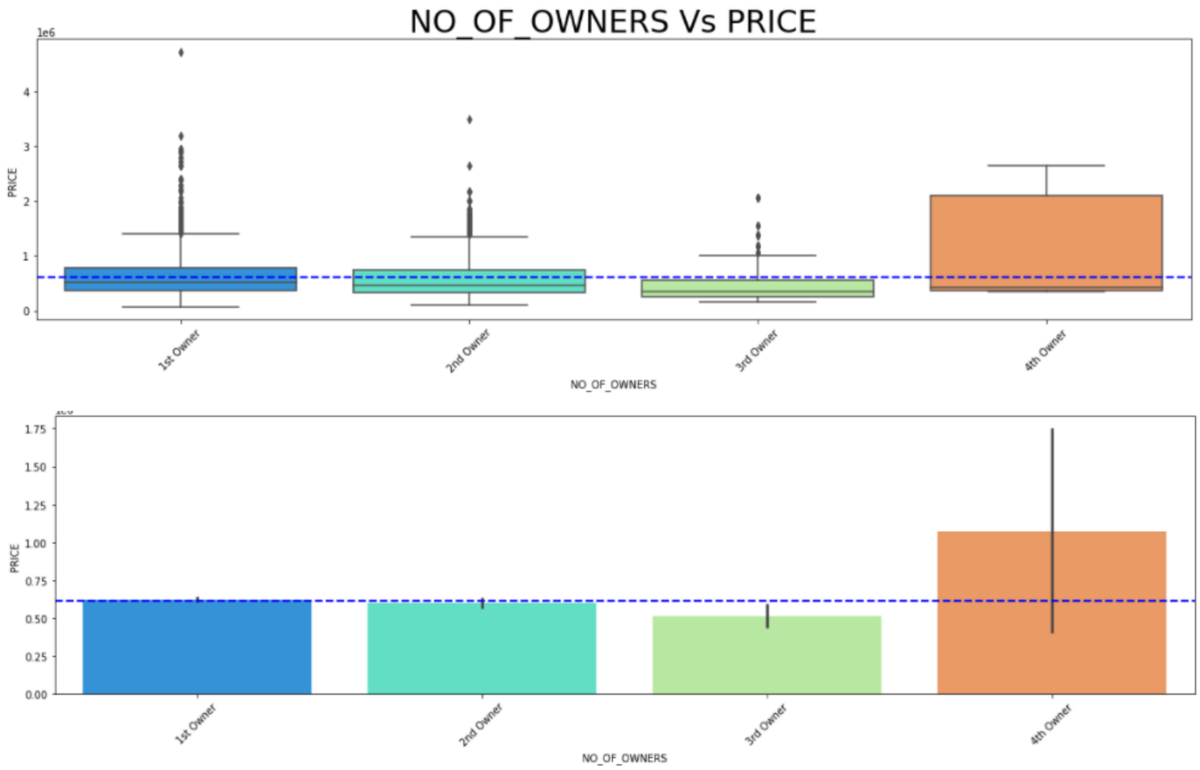
Key observations:

- Automatic Variants are costlier.



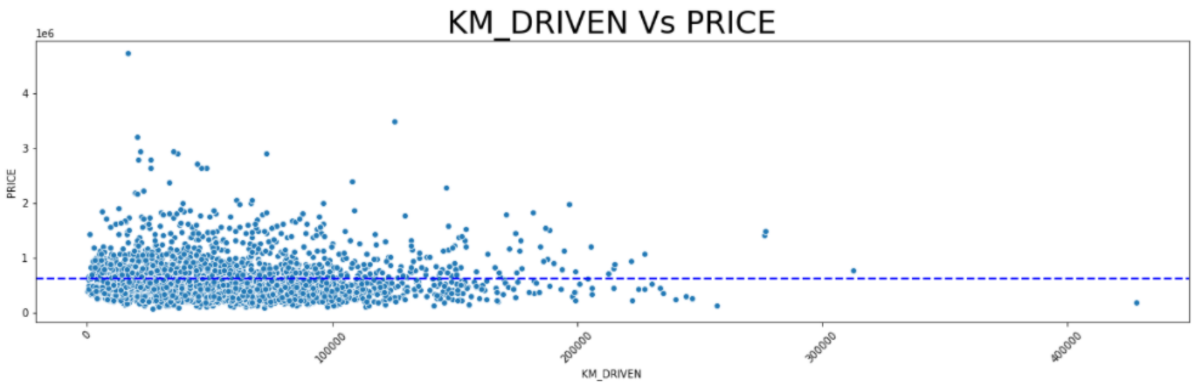
Key observations:

- Diesel Cars are expensive.
- Petrol Cars are costlier than CNG and LPG fuel types.



Key observation:

- 1st owner cars are costliest followed by second and third.



Key Observations:

- The lesser kms driven cars are costilier.

Hardware and Software Requirements and Tools Used

1. Python 3.8.
2. NumPy.
3. Pandas.
4. Matplotlib.
5. Seaborn.
6. Data science.
7. Machine Learning
8. SciPy
9. Sklearn.
10. Anaconda Environment
11. Jupyter Notebook.

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods).

Considering the business requirement of the client, I have collected the precise data to predict the used car price but there where multiple data of the car that are available. Data like colour of the car, sun roof attached, music system brand, electronics in the car, tyre brands, seat colours and much more. But after analysing all these data I have selected the data that have more correlation with the price of the car. Data like manufacturing year, number of owners used before, mode, fuel variant, gear shift variant, Brand of the car. I experimented and visualized how these variables contributed more towards the deciding factor of the car price. Based on such visualization I have built the model.

Testing of Identified Approaches (Algorithms)

After the pre-processing of the data that is collected, I have split the data as x_1 and y_1 and I have imported the required libraries to train my model.

```
In [41]: x_1=df_new.drop(["PRICE"], axis = 1)
         y_1=df_new.PRICE
```

```
In [43]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         x_1 = sc.fit_transform(x_1)
```

```
In [45]: from sklearn.model_selection import train_test_split, GridSearchCV
         from sklearn.linear_model import LinearRegression
         from sklearn.model_selection import cross_val_score, cross_val_predict, cross_validate
         from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_error

         accu = 0
         for i in range(0,1000):
             x_train_1, x_test_1, y_train_1, y_test_1 = train_test_split(x_1,y_1,test_size = .30, random_state = i)
             mod = LinearRegression()
             mod.fit(x_train_1,y_train_1)
             y_pred_1 = mod.predict(x_test_1)
             tempacc = r2_score(y_test_1,y_pred_1)
             if tempacc> accu:
                 accu= tempacc
                 best_rstate=i

         print(f"Best Accuracy {accu*100} found on randomstate {best_rstate}")

         Best Accuracy 47.312716496928076 found on randomstate 220

In [46]: x_train, x_test, y_train, y_test = train_test_split(x_1,y_1,test_size = .25, random_state = best_rstate)

In [47]: from sklearn.linear_model import LinearRegression, Lasso, Ridge, ElasticNet
         from sklearn.svm import SVR
         from sklearn.neighbors import KNeighborsRegressor
         from sklearn.tree import DecisionTreeRegressor
         from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor
```

On selecting the best random state parameter, I have used nine regression algorithms to train my model. Based on the best model score and best CV score I have selected Random Forest Regressor as the final model.

Run and evaluate selected models

I have used nine different regression algorithms to shortlist the best model.

Cross validation mean score and the model score.

```
In [48]: models = [LinearRegression(), Lasso(), Ridge(alpha=1, random_state=42), ElasticNet(), SVR(), KNeighborsRegressor(), DecisionTreeF

         model_names = ["LinearRegression", "Lasso", "Ridge", "ElasticNet", "SVR", "KNeighborsRegressor", "DecisionTreeRegressor", "AdaBoo
```

```
In [49]: score= []
         mean_abs_e=[]
         mean_sqr_e=[]
         root_mean_e=[]
         r2=[]

         for m in models:
             m.fit(x_train,y_train)
             print("Score of", m, "is:", m.score(x_train,y_train))
             score.append(m.score(x_train,y_train))
             predm=m.predict(x_test)
             print("\nERROR:")
             print("MEAN ABSOLUTE ERROR: ",mean_absolute_error(y_test,predm))
             mean_abs_e.append(mean_absolute_error(y_test,predm))
             print("MEAN SQUARED ERROR: ", mean_squared_error(y_test,predm))
             mean_sqr_e.append(mean_squared_error(y_test,predm))
             print("ROOT MEAN SQUARED ERROR :",np.sqrt(mean_squared_error(y_test,predm)))
             root_mean_e.append(np.sqrt(mean_squared_error(y_test,predm)))
             print("R2 SCORE: ", r2_score(y_test,predm))
             r2.append(r2_score(y_test,predm))
             print("****************************************************************************************************************")
             print('\n\n')
```


Out[51]:

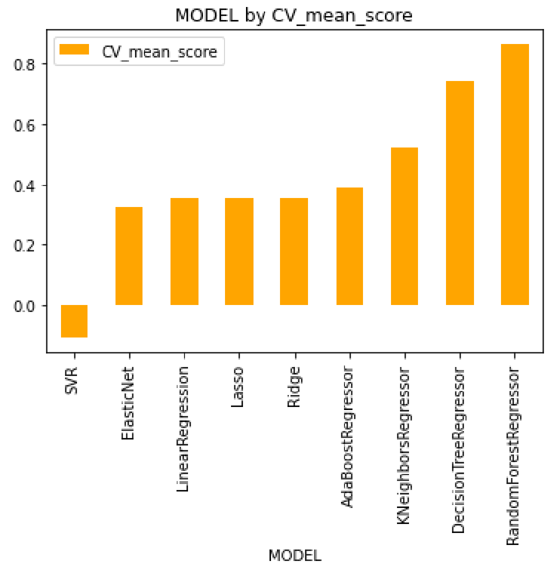
	MODEL	SCORE	CV_mean_score	CV_STD	MBE	MSE	RMSE	R2
8	RandomForestRegressor	0.986186	0.865242	0.093318	55337.211157	8.082784e+09	89904.302130	0.922804
6	DecisionTreeRegressor	1.000000	0.742404	0.178304	59330.901222	1.364676e+10	116819.353396	0.869664
5	KNeighborsRegressor	0.753990	0.523069	0.081861	141374.874338	4.305722e+10	207502.343079	0.588775
7	AdaBoostRegressor	0.464049	0.388622	0.164240	193888.710170	5.434633e+10	233122.989858	0.480957
2	Ridge	0.421513	0.354826	0.079857	174388.272171	5.558295e+10	235760.355592	0.469146
1	Lasso	0.421513	0.354795	0.079890	174396.671545	5.558184e+10	235758.004557	0.469157
0	LinearRegression	0.421513	0.354795	0.079888	174396.867820	5.558177e+10	235757.855664	0.469157
3	ElasticNet	0.373074	0.324409	0.066822	180851.637306	6.200755e+10	249013.148867	0.407787
4	SVR	-0.078356	-0.106140	0.134593	234775.069353	1.124140e+11	335281.974797	-0.073628

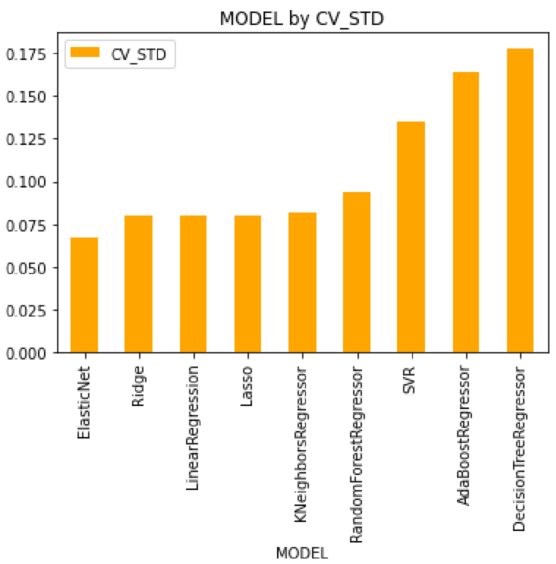
As live saw above Random forest Regression model stands at the top with the r2 score of 92.28 with the CV score of 86.52 further I am going to hyperparameter tune the model to reduce over fitting and to increase the performance of the model.

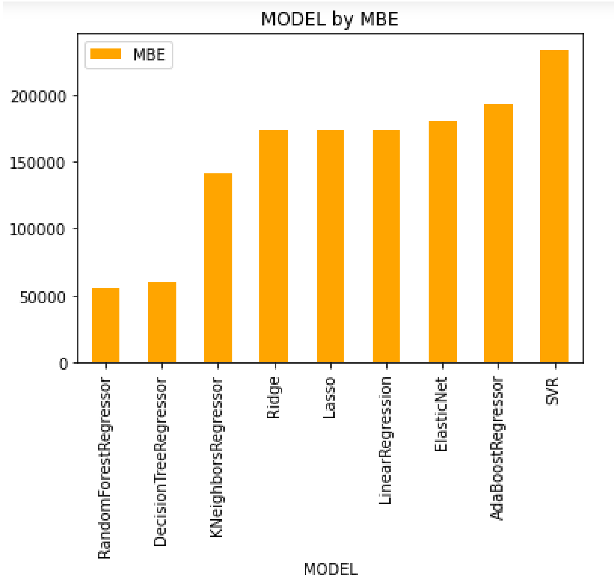
Visualizations

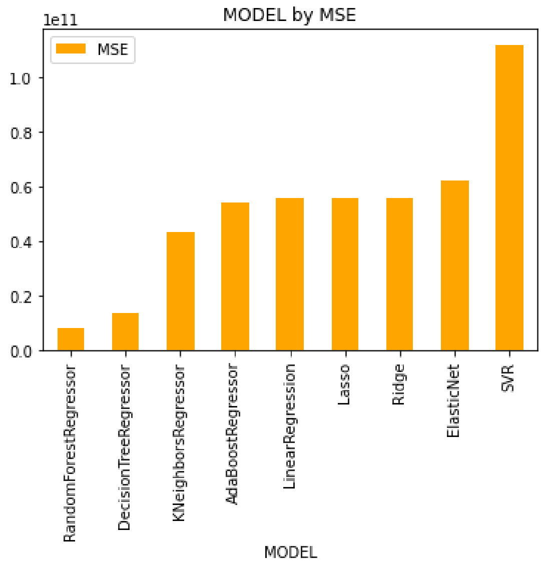
I have already explained some visualizations related to the input output logic.

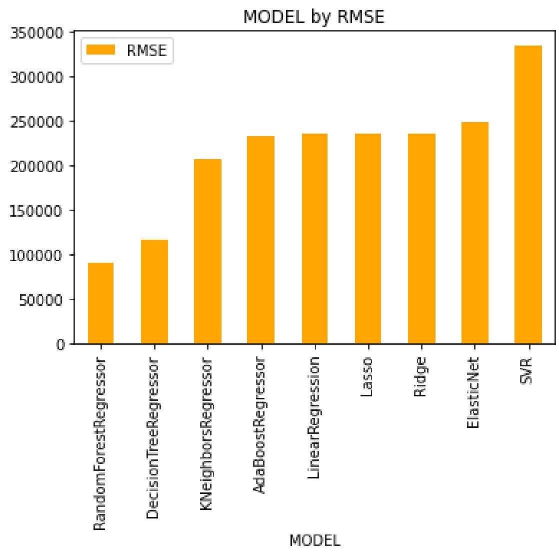
Now Model Visualization is explained.

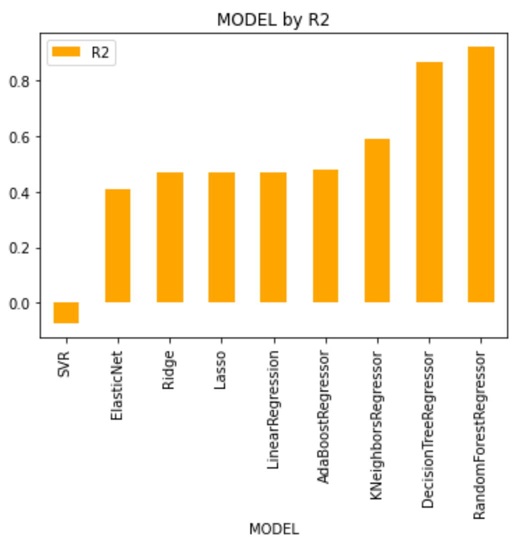












From above observation, we can come to a conclusion that Random Forest is the best model with r2 Score of 92.28 let's try in Hyper tuning the same for improved performance and also to reduce the over fitting the Data.

Interpretation of the Results

From the visualization above we can clearly understand that the used car price factors are decided by the factors such as brand, location, model, year made, number of owners used the car before, fuel type of the car.

From that we can clearly say that the used car price depending on the Brand that is the manufacturer and model it varies. The manufacturer like Land Rover, Benz, BMW cars are costliest used car in the market comparatively to other cars, the low kilometres driven and also if the manufacturing year is lesser on these brands those card sells in much higher rates or closest to the buying new car rates. The Diesel variant and Automatic shift variants are also costliest user car variants in the used car market

CONCLUSION

Key Findings and Conclusions of the Study

The manufacturer like Land Rover, Benz, BMW cars are costliest used car in the market comparatively to other cars, the low kilometres driven and also if the manufacturing year is lesser on these brands those card sells in much higher rates or closest to the buying new car rates. The Diesel variant and Automatic shift variants are also costliest user car variants in the used car market.

Learning Outcomes of the Study in respect of Data Science

The above research will help our client to study about the latest used car market and with the help of the model built he can easily predict the price ranges of the cars, and also will helps him to understand based on what factors the Car Price is decided.

Limitations of this work and Scope for Future Work

The limitation of the study is that in the volatile changing market we have taken the data, to be more precise we have taken the data at the time of pandemic, so when the pandemic ends the market correction might happen slowly. So based on that again the deciding factors of the used car prize might change and we have shortlisted and taken these data from the important cities across India, if the seller is from the different city our model might fail to predict the accurate prize of that used car.