PROJECT NAME:

CAR PRICE PREDICTION

SUBMITTED BY:

LALBIAK ZAUVA

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. 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 car price valuation model.

Car prices mostly depends on several factors; among which the number of owners could really reflect how people would have an opinion on the condition of the car. If several people already owned that particular car, then the value most likely drops. And also used-car-buyers prefer cars which has lesser number of kilometres driven even for the same type of car. This reflects whether the used-cars are still in good condition or not. Lesser number of kilometres driven would add value to the car and the price most likely increases.

In this project, data is collected from different cities including Ahmedabad, Bangalore and Mumbai. We have over 5511 rows with 9 features including the target variable (Price).

Brand	Engine (in cc)	Number of owner	Insurance	Manufacturing year	Driven kilometers	Fuel type	Mileage	Price (in lakhs)
Maruti	1248.0	First Owner	NaN	2015.0	57000.0	Diesel	26.21	6.25
Maruti	993.0	Second Owner	NaN	NaN	50000.0	Petrol	17.30	70000.00
Maruti	1197.0	First Owner	NaN	2019.0	3972.0	Petrol	21.01	8.00
Maruti	1197.0	First Owner	NaN	2019.0	6441.0	Petrol	19.56	8.40
Tata	1199.0	First Owner	NaN	2020.0	12524.0	Petrol	23.84	7.30

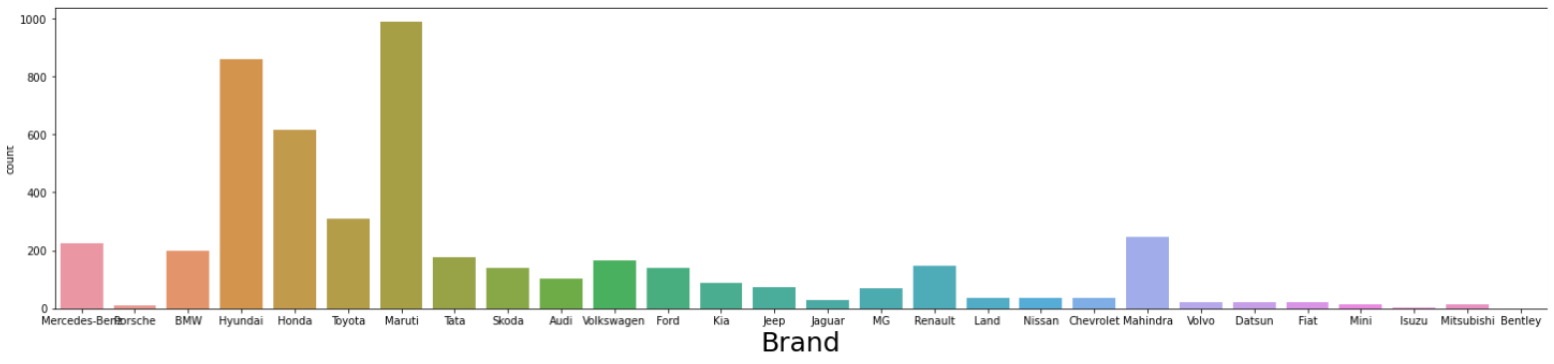
There are many null values. Some rows have null values for all the columns and they are dropped. But some rows have null and non-null values; so they are handled accordingly.

The dataframe contains 4 object type data while it contains 5 float64 type data. So, we have both categorical data and continuous data in our dataframe. Null values from categorical data are filled with mode of the column; while null values from the continuous data are filled with mean of the data.

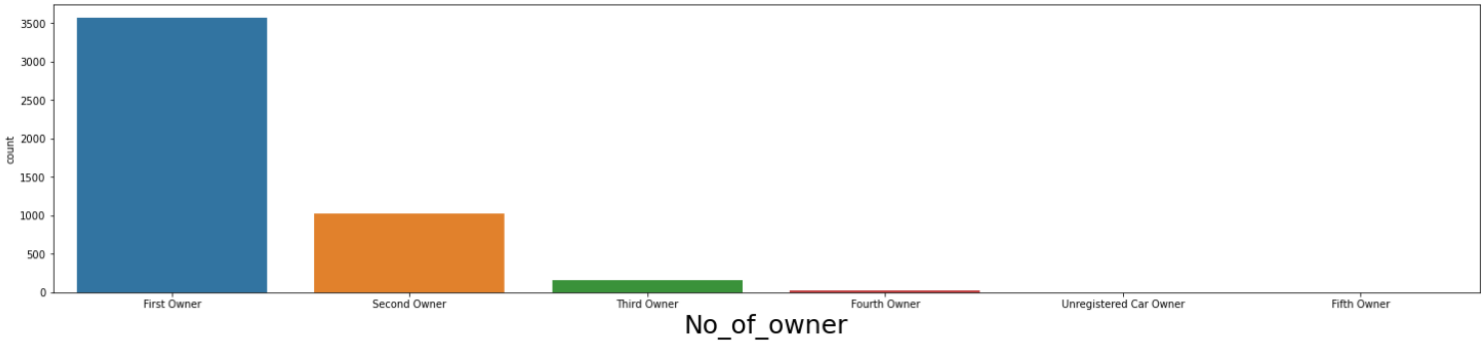
By describing the data using .describe() function, there are some unrealistic values which would surely make our data looks skewed. This skewness needs to be handled so that the machine learning algorithm performs its task at its best. We will see how this is done later; but first let us visualize the data to understand it better.

Exploratory Data Analysis:

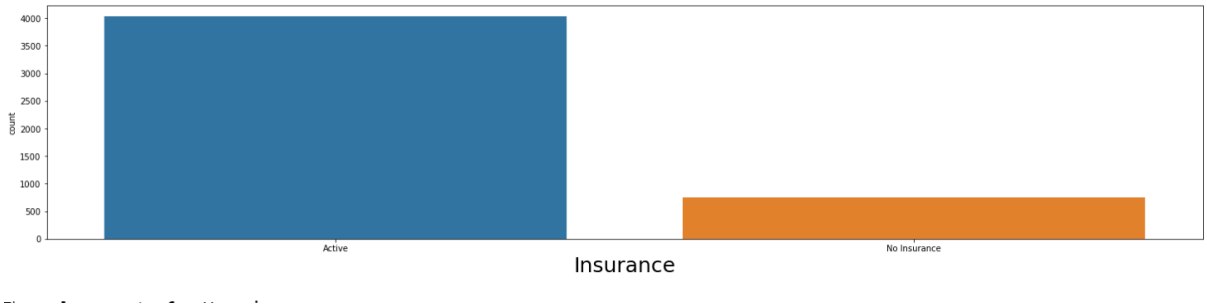
I have split the data into categorical and continuous data so that we can perform EDA separately.



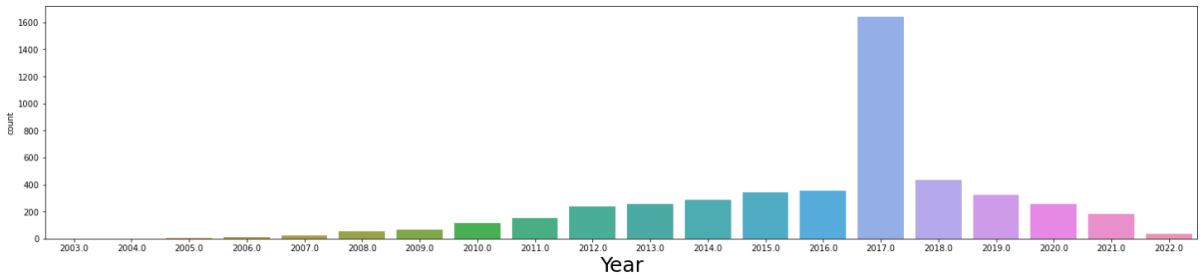
- As we can see, Maruti product is the best selling used-car from different cities followed by Hyundai and Honda. Bentley and Isuzu are not commonly seen in the used-car market.



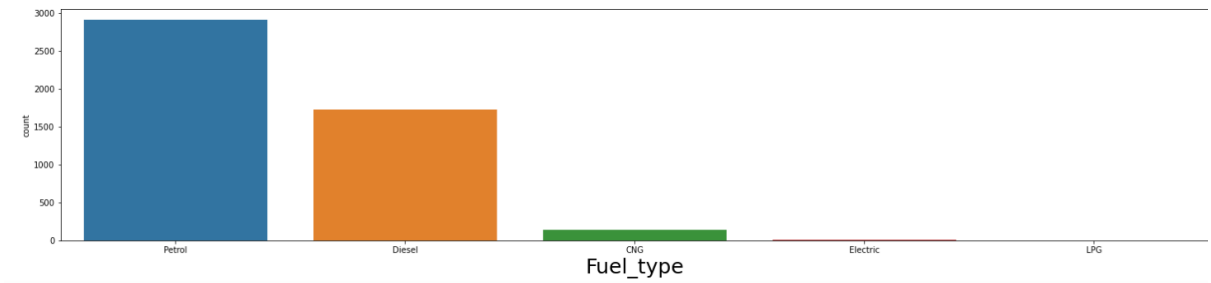
- Clearly, most people prefer to buy first-owner cars and hence this makes them more expensive than second owner cars.



- Active insurance adds up value to the car and would be more expensive than ‘no insurance’ cars.

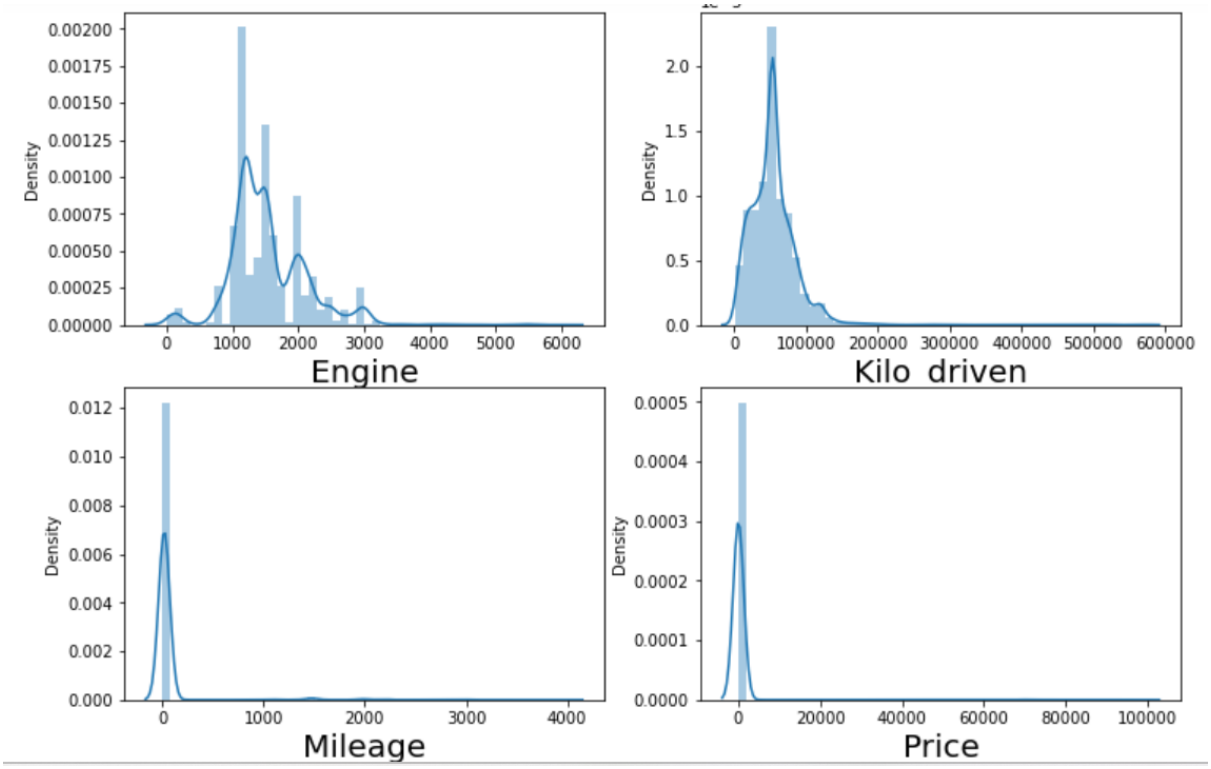


- 2017 model has the highest sales. This could imply that years before 2017 would be too old for most customers; and years after 2017 are rather new, but would be expensive, and so most customers would not prefer them and so 2017 model cars could be the right year for the car model.

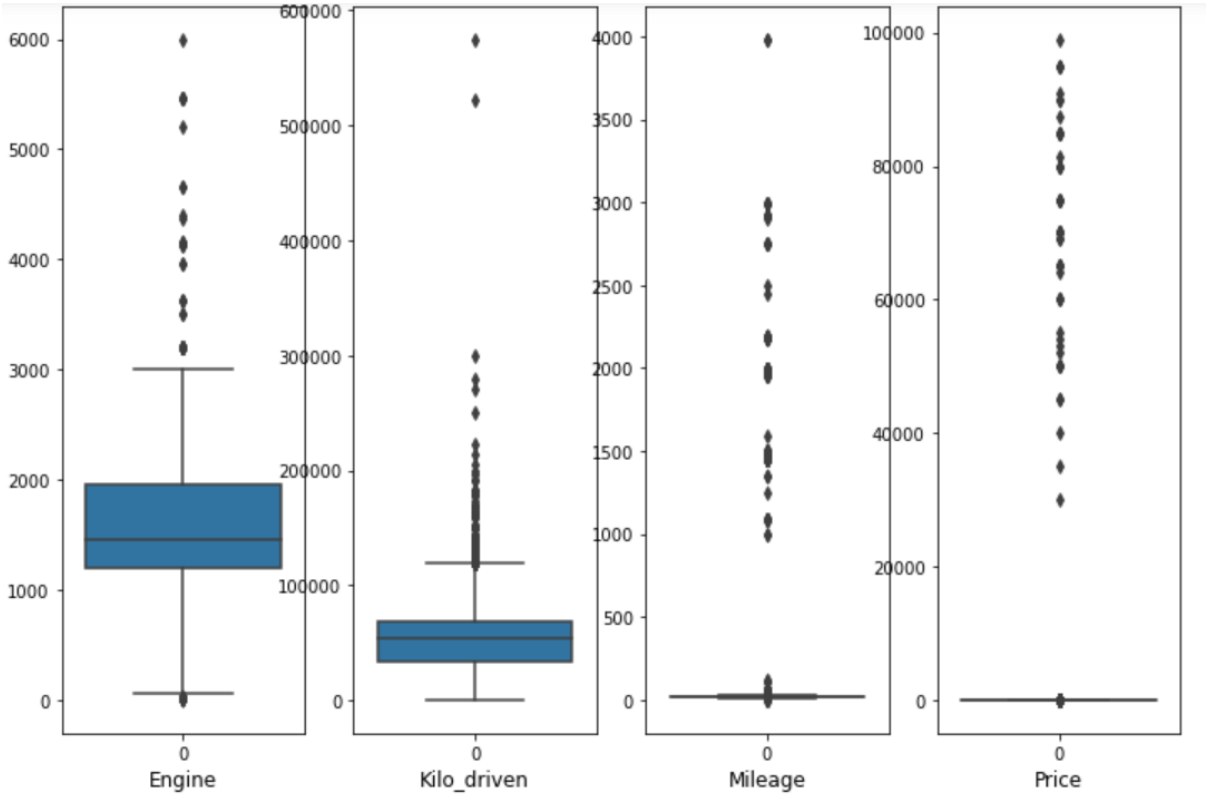


- Petrol using cars has the highest sales followed by that of diesel. Petrol using cars are the most common used-cars and people also prefer them and hence price will be higher than the other fuel consuming cars.

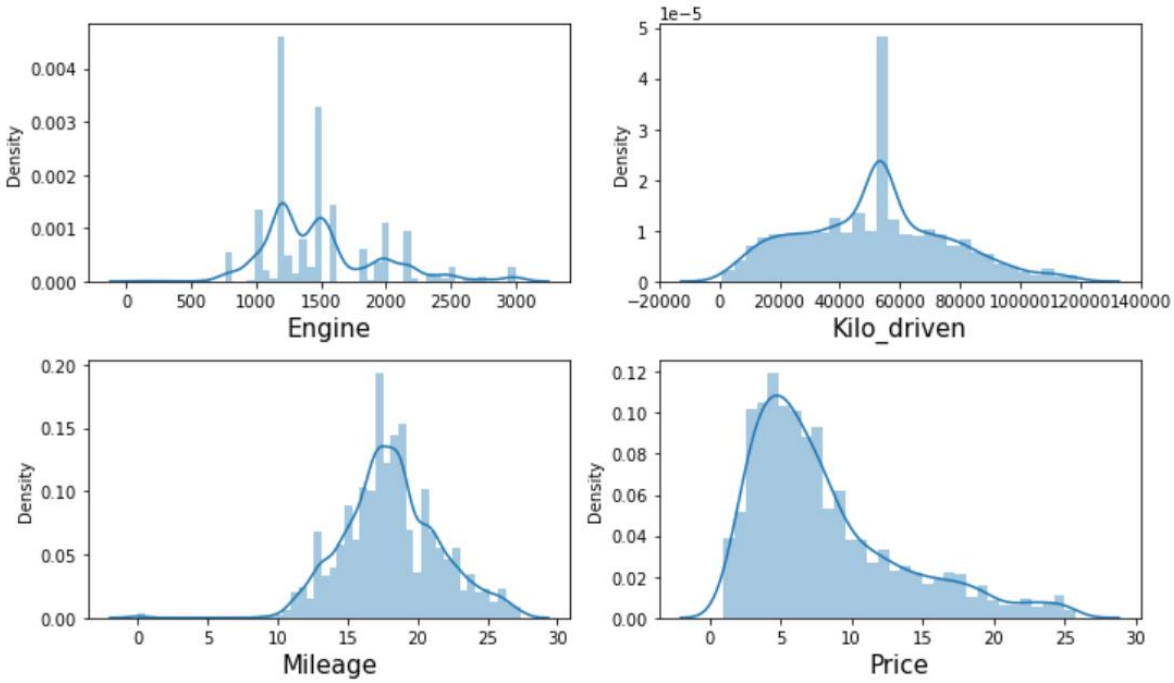
Now, plotting the distribution plot for the continuous data, we see that there is so much skewness in the data.



So, we need to handle this using box-plot and remove the outliers.



Using IQR (inter quartile range), we can identify the outliers and remove them.
After removing the outliers, the distribution plot looks like a normal distribution:



Model building:

Now that we have completed the data cleaning and pre-processing part, let's move on and build our model. But first let's import necessary libraries:

```
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
import sys
from sklearn import metrics
!{sys.executable} -m pip install xgboost
import xgboost as xgb
from sklearn.metrics import r2_score,mean_squared_error
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import StandardScaler
```

Since we are dealing with a regression problem, we have to calculate the R2 score.
Then we need to split the data into train data and test data.
After trying different algorithms, XGB Regressor algorithm gives the best result. So, we will choose this algorithm for our model and to make prediction.

```
y_test_pred = xgb.predict(x_test)
print(f"The accuracy score is {r2_score(y_test,y_test_pred)*100:.2f} %")
```

```
The accuracy score is 87.83 %
```

Cross Validation:

We need to perform cross validation of the performance of our model to see whether our model overfits or not. The cross-validation score here looks good and our model does not overfit.

```
cv_score = cross_val_score(xgb,x_scaled,y,cv = 14)
cv_mean = cv_score.mean()
cv_mean
```

```
0.8258767408560568
```

Regularization:

Also, the L1 form and L2 form has pretty much the same value which also tells us that our model does not overfit.

```
lasso_reg.score(x_test,y_test)
```

```
0.6261093815046923
```

```
ridge_model.score(x_test,y_test)
```

```
0.6261122046001555
```

Now, we can save the model for future predictions.

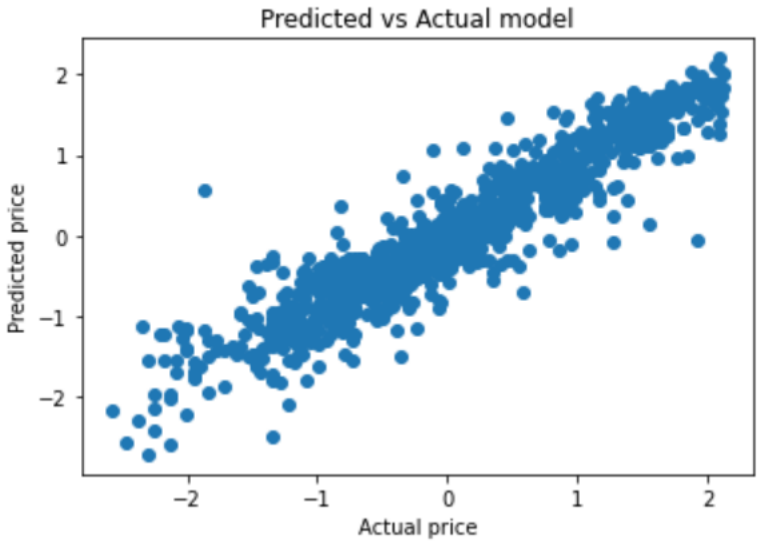
```
import pickle
filename = 'car_price_prediction.pickle'

pickle.dump(xgb,open(filename,'wb'))
```

Finally, let's see the scatter plot between the actual and predicted values.

```
plt.scatter(y_test,y_test_pred)
plt.xlabel('Actual price')
plt.ylabel('Predicted price')
plt.title('Predicted vs Actual model')
```

```
Text(0.5, 1.0, 'Predicted vs Actual model')
```



Conclusion:

The features that we are dealing with (8 features excluding the target) are all important factors and have strong correlation with the target variable. So, we are not dropping any column since they are all important. Most people do not prefer to buy very cheap cars since the quality and condition of the car would also be quite bad. They neither want to buy expensive cars even though they are still in good conditions. Most people rather want to buy cars of moderate price which still looks good.

Out of different brands, the most popular brand is Maruti. Since our target variable is continuous, we perform some checks on the accuracy score, but we find that our model works pretty well and does not overfit. Using XGB Regressor algorithm, we get an R2 score of 87.83 %.