TOPIC: USED CAR PRICE PREDICTION

Group No: 12
Participant List:
Chinmay Bhat
Ram Warutkar
Sakshi Jain

PROBLEM STATEMENT:

Buying or selling a second hand car from either an individual or any dealer, there is always a room for negotiation, therefore knowing the fair market price is very important. The price of a used car depends on various factors like Make and Model, Kilometers Driven, how old vehicle is, Seating Capacity and many other factors. We have tried to build an Algorithm which can predict the selling price of the used car by applying various machine learning techniques. For we have scrapped the data from droom.in

OVERALL SUMMARY:

DATA UNDERSTANDING

A snapshot of data:

Name_of_Vehical	On_Road_Price	Rating	Transparency_Score	Seller_Score	Health_Score	Pricing_Score	Make	Model	...	Seating_Capacity	F
Hyundai i10 Sportz 1.2 AT 2012	646545	7.0	6.4	9.5	3.8	8.9	hyundai	i10	...	5.0	
Hyundai i10 Magna 1.2 2014	485837	6.6	6.4	9.5	3.8	7.0	hyundai	i10	...	5.0	
Hyundai i10 Sportz 1.2 AT 2012	646545	6.3	6.1	6.7	4.0	9.1	hyundai	i10	...	5.0	
Hyundai i10 Magna 2011	533375	6.9	6.9	9.9	4.0	6.8	hyundai	i10	...	5.0	
Hyundai i10 Magna 2011	533375	6.9	6.9	9.9	4.0	6.8	hyundai	i10	...	5.0	

; × 28 columns

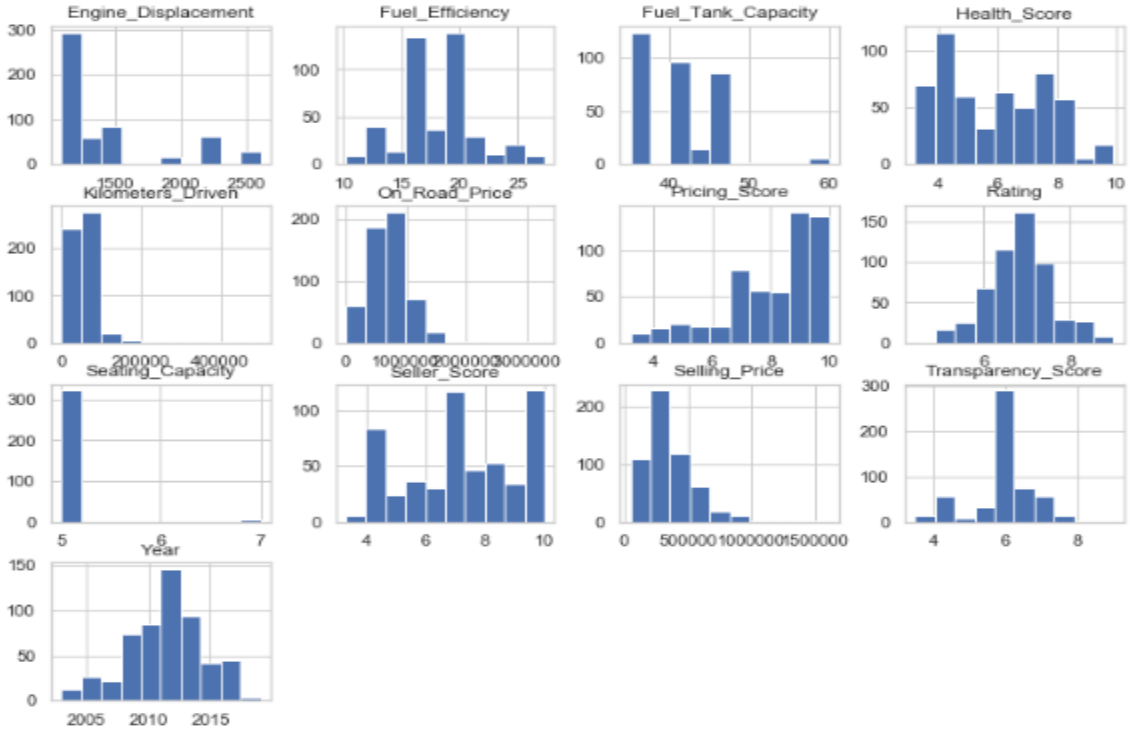
The fields of the dataset are as follows:

Name of Vehicle, Selling Price, On Road Price, Rating, Transparency Score, Seller Score, Health Score, Pricing Score, Make, Model, Trim, Year, Exterior Color, Interior Color, Fuel Type, Transmission Type, Body Type, Interior Furnishing, Seating Capacity, Fuel Tank Capacity, Location, Registration State, Kilometers Driven, Seller Type, Number of Owners, Engine Displacement, Fuel Efficiency.

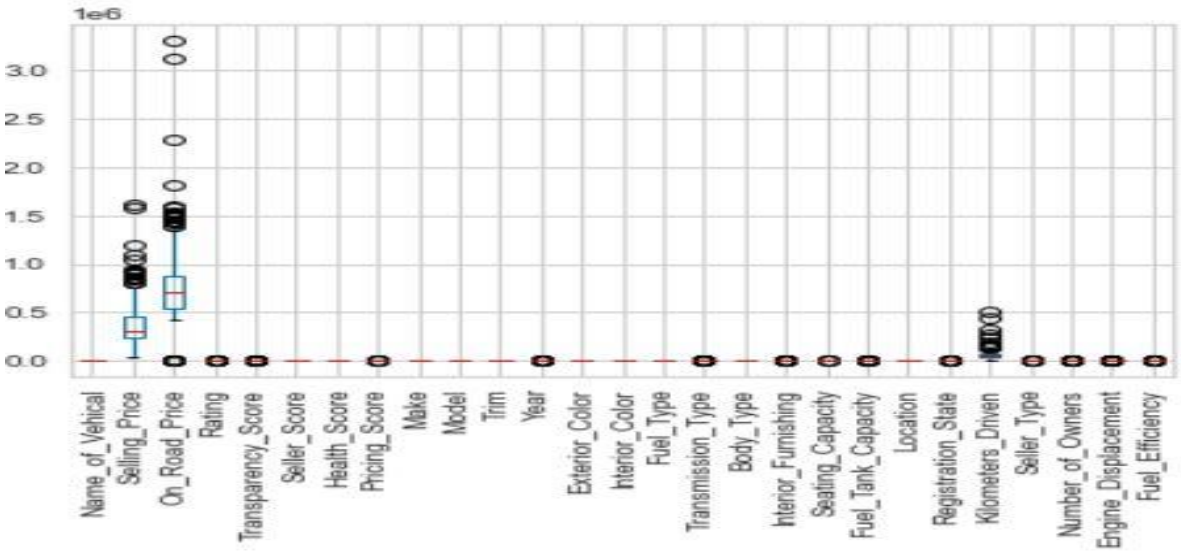
There are **597 rows and 27 columns**, in which 14 filed is in the form of object and 9 float and 4 int. There are **351 Null Values**. So, the data is a mixture of **both categorical and numerical fields**. Initial observation suggests the range of numerical fields greatly varies, such as On Road Price and Health Score has huge difference in magnitude. In categorical data type the unique values ranges 4 to 49 except for Name of Vehicle it is 331.The fields Rating, Transparency Score, Seller Score, Health Score, Pricing Score are the review scores given by the Droom.in after

taking into account all important trust factors such as auto inspection, warranty, verified seller, attractiveness of pricing for buyer, and level of disclosures by the sellers etc.

EXPLORATORY DATA ANALYSIS

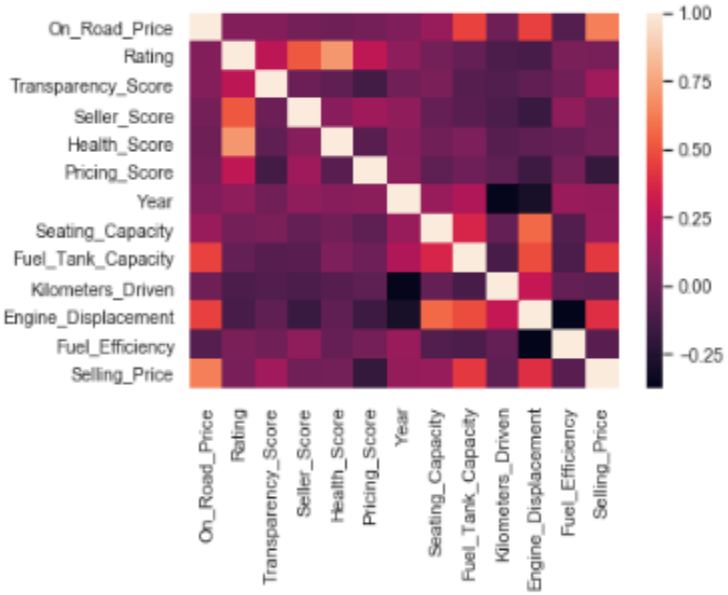


The Histogram suggests that the dependent variable i.e. selling price is positively skewed and other factors like Fuel Efficiency, Transparency score and Year are distributed symmetrically.



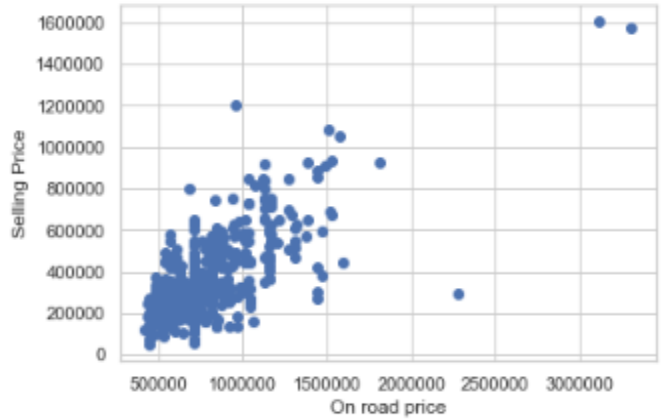
The Box Plot tells that the on-road price and selling price have many outliers and also the range of all the features highly differ like for selling price it is in lakhs and for rating it is upto 6.

The Correlation matrix:

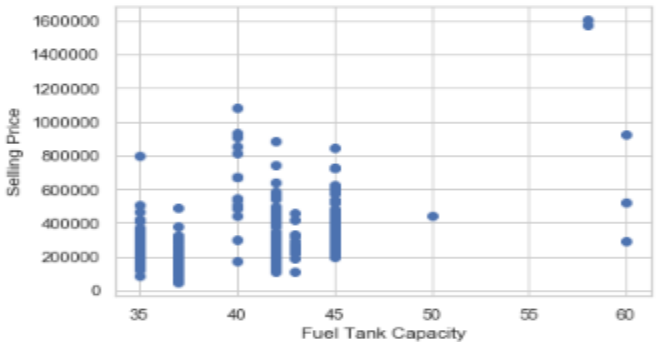


The heat-map suggest that the feature on road price has the highest correlation and then Fuel tank capacity and Engine Displacement are important factors in predicting the selling price.

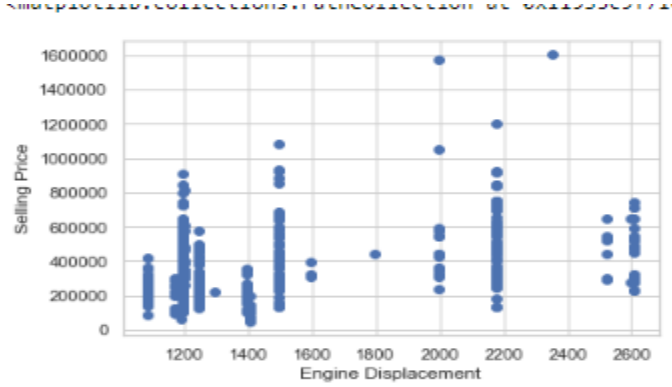
On the basis of heatmap plotting the features which have highest correlation with the response variable to see how it affects the selling price of the vehicle



The plot of selling price against on road price shows the positive correlation and shows as the on-road price increases the selling price also increases.



The fuel tank capacity as such not show any positive correlation because for higher value of fuel tank capacity the selling price is clustered in the same range as it is for lower value of fuel tank capacity.



For higher values of Engine Displacement, the selling price is clustered in the same as it is clustered for lower value of Engine Displacement.

VARIANCE INFLATION FACTOR

	VIF Factor	features
0	22.662306	On_Road_Price
1	662.381627	Rating
2	101.576398	Transparency_Score
3	36.542658	Seller_Score
4	61.791701	Health_Score
5	50.259027	Pricing_Score
6	6.936224	Make
7	3710.573985	Year
8	2054.911625	Seating_Capacity
9	183.311208	Fuel_Tank_Capacity
10	3.407388	Kilometers_Driven
11	224.364644	Engine_Displacement
12	52.974772	Fuel_Efficiency
13	5.012817	Petrol
14	1.037067	Petrol + CNG
15	31.648849	Automatic
16	414.259298	Manual
17	327.039362	Hatchback
18	2.600399	MUV
19	152.686514	SUV
20	28.771236	Sedan
21	1.204940	Leather
22	1.306401	Leatherette
23	1.040072	Other
24	341.787776	First Owner
25	5.230322	Fourth Owner
26	101.117255	Second Owner
27	14.491587	Third Owner
28	57.160646	Dealer

The above table shows that some features are highly corelated, these features can cause standard error of coefficient to grow and therefore resulting in wider confidence interval for coefficient.

The Dataset also has missing values and categorical features which need to be filled and encoded respectively in order to fit a model.

- The missing value is filled with mean when the feature was categorical and with the mode or by method ffill of pandas when the feature was categorical.
- The categorical features were encoded with OneHotEncoding.
- The features like Name of vehicle, model were dropped because we want to predict the price on the basis of the condition of car and of which model it belongs. Also we have dropped the features like Location And State because these columns consists of single value which is not going to bring any change in the predictive power of the model
- After this the dataset was divided into training and testing part, the model was fitted on training part and was tested on testing dataset. The testing dataset contains 20 per cent of whole data.

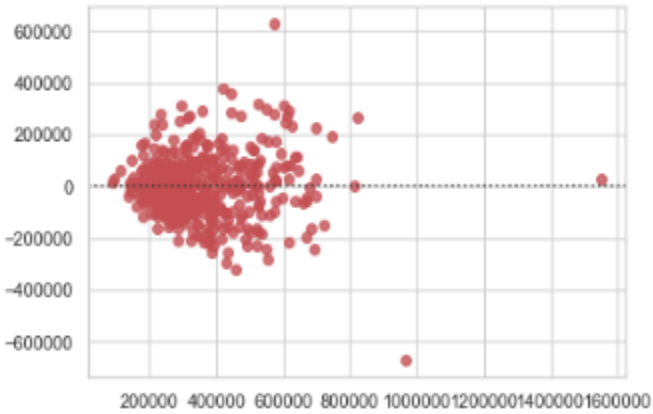
FITTING MULTIPLE LINEAR REGRESSION

```
1 from sklearn.linear_model import LinearRegression
2 model = LinearRegression()
3 model.fit(X_train,y_train)
```

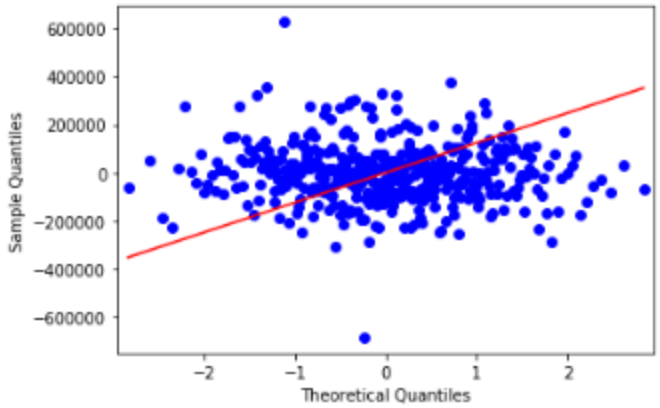
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=True)
```

- The R-square for the multiple linear model is 59 per cent.
- Training Root Mean Square: 123425
- Testing Root Mean Square: 125658
- The training RMSE is quite high because underestimating or overestimating the price of a vehicle by 125000 is huge and can cost either to buying party or to the selling party.
- The difference between the training RMSE and Testing RMSE is not much. So, clearly the model is not overfitting on testing dataset

The residual plot:



The residual plot is random, It does not show any pattern thus we can conclude that residuals are homoscedastic and follows no particular pattern.



The residuals are not normally distributed that means the amount of error in the model is not consistent across full range of data.

After removing the features whose p-value was more then 0.05: The result is:

OLS Regression Results

Dep. Variable:	Selling_Price	R-squared:	0.571
Model:	OLS	Adj. R-squared:	0.562
Method:	Least Squares	F-statistic:	63.37
Date:	Tue, 21 Apr 2020	Prob (F-statistic):	3.05e-73
Time:	11:15:18	Log-Likelihood:	-5779.7
No. Observations:	439	AIC:	1.158e+04
Df Residuals:	429	BIC:	1.162e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-2.203e+07	4.32e+06	-5.097	0.000	-3.05e+07	-1.35e+07
On_Road_Price	0.4477	0.031	14.551	0.000	0.387	0.508
Transparency_Score	2.272e+04	7497.259	3.030	0.003	7982.369	3.75e+04
Pricing_Score	-2.367e+04	4156.712	-5.694	0.000	-3.18e+04	-1.55e+04
Make	4179.9148	2909.218	1.437	0.152	-1538.180	9898.010
Year	1.105e+04	2153.830	5.131	0.000	6818.343	1.53e+04
Petrol	3.787e+04	1.53e+04	2.481	0.013	7866.141	6.79e+04
Hatchback	-9.345e+04	2.27e+04	-4.108	0.000	-1.38e+05	-4.87e+04
Sedan	-1.189e+05	3.18e+04	-3.742	0.000	-1.81e+05	-5.64e+04
Dealer	-1.025e+05	4.62e+04	-2.219	0.027	-1.93e+05	-1.17e+04

- The R-square for this model is 57 per cent.
- Training Root Mean Square: 123350
- Testing Root Mean Square: 126329
- Removing the unimportant features also does not bring any change in the model . The difference bet ween the training RMSE and Testing RMSE remains the same.

After this we removed the outliers from the dataset, and build the model on the same. It has significant impact on the training and testing RMSE as both decreased to around 1 lakh. But for this model also the residual plot is not normally distributed. We tried building the model on scaled data as well but for this also t he residual plot is not normally distributed. So we can not accept these models as they violate the important assumption of linear regression. So we can conclude that the model cannot be build through linear regression The model can build through other algorithm which does not assume anything about the data.

DECISION TREE

```
from sklearn.tree import DecisionTreeRegressor
model_tree=DecisionTreeRegressor(max_depth=3,min_samples_split=4, min_samples_leaf=3)
model_tree.fit(X_train,y_train)
```

The decision tree model has slight improvement over linear model.

- The R-square for decision tree model is 64 per cent.
- Training Root Mean Square: 129461
- Testing Root Mean Square: 167871
- The difference between training RMSE and testing RMSE is quite huge, and it suggest that the model has overfitted the testing dataset.
- The optimized tuning parameters were found by GridSearchCV.

RANDOM FOREST

```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor(max_depth=6, min_samples_leaf=3, min_samples_split= 2, n_estimators= 100)
rf.fit(X_train,y_train)
```

- The optimized parameter were found by GridSearch CV.
- The random forest model is performing good on the training dataset
- The R-square for the model is 79 per cent.
- The training RMSE is: 88353
- The Testing RMSE is: 152659
- But the random forest model is performing poorly on the test data set.
- So, we can say that the random forest algorithm is overfitting the dataset

BOOSTING

```
from sklearn.ensemble import AdaBoostRegressor
ad_boost=AdaBoostRegressor(n_estimators=300)
ad_boost.fit(X_train,y_train)
```

- As compare to other models except random forest, boosting is performing slightly good on the training dataset but performing poor on the testing dataset.
- The R-square for the boosting algorithm is 67 per cent
- The Training RMSE is 113401
- The testing RMSE is 159083

SUPPORT VECTOR REGRESSION

```
svr_1=SVR(kernel='rbf',gamma=0.1)
svr_1.fit(X_train_s,y_train_s)
```

```
svr=SVR(kernel='linear',C=100)
svr.fit(X_train_s,y_train_s)
```

We tried support vectors on the scaled data, so that execution becomes faster.

- For the kernel linear the R-square is 56 per cent.
- The training RMSE is 0.08
- The testing RMSE is also 0.08.
- So we can say that the support vector isn't overfitting the testing dataset.

For the kernel rbf, the results are same the R-Square is 56 per cent and the training and testing RMSE is same as for kernel linear that is 0.08.

K NEIGHBORS REGRESSION

```
model_k=neighbors.KNeighborsRegressor()
model_K = neighbors.KNeighborsRegressor(n_neighbors = 10)
model_k.fit(X_train,y_train)
```

The model for K neighbors is overfit model because there is a huge difference between training and Testing RMSE.

- The R-Square for the model is 61 per cent.
- Training RMSE is: 119713
- Testing RMSE is: 160414

COMPARISION OF ALL THE MODELS

Algorithm	R- Square	Training RMSE	Testing RMSE
Multiple Linear Regression	59 %	123425	125658
Regression after removing the features	57 %	123350	126329
Decision Tree	54 %	129416	167871
Random Forest	79 %	88353	152659
Boosting	65 %	113401	159081
Support Vector Machine (kernel linear)	56 %	0.08	0.08
Support Vector Machine- Kernel rbf	56%	0.08 (On scaled data)	0.08 (On scaled data)
K Neighbor Regression	61 %	119713	160414

CONCLUSION

- Ideally there is no model which can predict the selling price accurately. The random forest model is working quite good on the training dataset but not on testing dataset.
- We cannot opt for linear regression because the residuals are not normally distributed which suggest that the error is not constant across the full range of the data. And when we think about the predictions that is the amount of predictive ability they have (that is as calculated in their beta weights) is not the same across the full range of the dependent variable. Thus the

predictors technically mean different things at different level of the dependent variable, which is not good for interpretation.

- ❖ All the models are overfitting the data except for support vector regression which has same RMSE for training as well as for testing dataset.
- ❖ For the SVR the data was scaled in range of 0 to 1, in according to that the RMSE of 0.08 is high. If this error is acceptable to the seller party or the buyer party. Then they can opt for this model at a cost of underestimated or overestimated selling price by 80,000.

LEARNING OUTCOME

- ❖ Through this project we were able to understand that every time the data will not perform good on the algorithm linear regression. This is may be because the linear regression has many assumptions which needs to fulfilled in order to perform better on the dataset.
- ❖ The Supports vectors performs fast on scaled data.
- ❖ Any of the model didn't perform good or the which perform good on training set but fail to perform on the testing set- For this the reason could be the unbalanced data.
- ❖ Due to this project we are now familiar with data scrapping.
