PROJECT # 01 : CAR PRICE PREDICTION

LINEAR REGRESSION VS RANDOM FOREST REGRESSION MODEL

Submitted by

- **Rangarajan Srinivasan**
  IIT Madras

HENRY HARVIN® EDUCATION

1. Overview

- Types of Car models and its available features was provided - Limited set of data

2. Problem

- Predict car "price" – with given set of data (features) by using different regression method – Linear Regression & Random Forest.

3. Objective

- Identify Independent variables (features) which are closely associated with car price and its "Error – Predict vs Actual" .

4.Methodolgy

Problem Definition

- Car "Price" prediction

Data Understanding

- Number of variables (features) available in data file

Exploratory Data Analysis

- Understanding Categorical and quantitative variables

4.Methodolgy

Data Preparation

- Data cleaning and conversion of additional data variables.

Model Building & Evaluation

- Data splitting technique , model using regression technique and evaluate model in terms of its accuracy.

Recommendation

- Features highly dependent on car “price” predictions.

4.Data Overview

Features

Data Understanding

- 26 set of variables and 205 data points of different car models were provided.
- 25 variables treated as - Independent
- 26th Variable – "Price" treated as Dependent.

	Features - Independent Variables
1	car_ID
2	symbolling
3	Car Name
4	fuel type
5	aspiration
6	door number
7	carboy
8	drive wheel
9	engine location
10	wheelbase
11	car length
12	car width
13	Car height

	Features - Independent Variables
14	Curb weight
15	engine type
16	cylinder number
17	engine size
18	fuel system
19	bore ratio
20	stroke
21	compression ratio
22	horsepower
23	peakrpm
24	citympg
25	highway

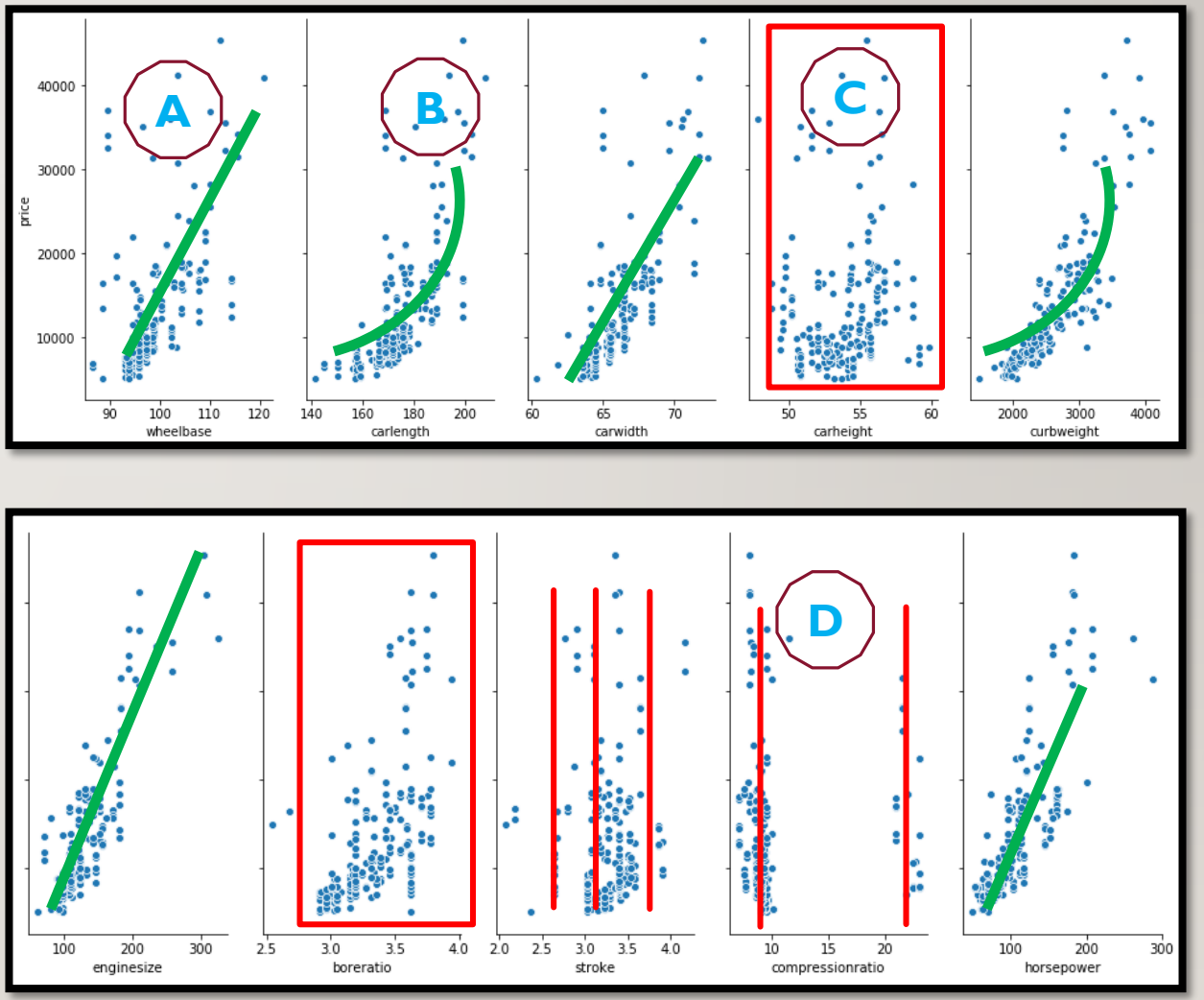
4.Data Overview

❖ **Price Influencing on different set of variables**

❖Data relation understanding -

- Dependent variable Vs. Independent variables

A ➔ Linear
B ➔ Exponential
C ➔ Scattered
D ➔ Segregated



5.Data Manipulation

✓ Categorical Data conversion and manipulation – "String" to "Integer"

Categorical Data into Integer data
26. Fuel type (Gasoline & Diesel)
27. Aspiration (Standard & Turbo)
28. Number of doors (2- "Coupe", 4 – Normal)
29. Car Body (Sedan, Hatchback, Wagon, Hardtop & Convertible)
30. Wheel Drive (Front wheel, Rear wheel, Four Wheel)
31. Engine Location (Front side, Rear Side)
32. Engine Type (DOHC, OHCV, OHC, L –ROTOR, OHCF, DOHCV)
33. Number of Cylinders (2, 3,4,5,6,8 & 12)
34. Fuel System (MPFI , 2BBL, MFI, 1BBL, SPFI, 4BBL, IDI, SPDI)

5.Data Manipulation

✓ Generating more variables using existing variables based on formulation

➔ Adding more data features will enhance in better accuracy

Adding more data Features

```
35. cars_data['area_car'] = cars_data['carlength'] * cars_data['carwidth']
36. cars_data['volume_car'] = cars_data['carlength'] * cars_data['carwidth']*cars_data['carheight']
37. cars_data['weight_to_volume ratio'] = cars_data['curbweight'] / cars_data['volume_car']
38. cars_data['enginesize_to_powerratio'] = cars_data['horsepower'] / cars_data['enginesize']
39. cars_data['avg_milage'] = (cars_data['citympg']+cars_data['highwaympg'])/2
```

✓ Removing features from data set (“Superfluous data”)

- Car Model
- Car_ID

5.Data Manipulation

Final Data Set

1	Car_ID
2	fueltype
3	aspiration
4	doornumber
5	enginelocation
6	wheelbase
7	carlength
8	carwidth
9	carheight
10	curbweight
11	cylindernumber
12	enginesize
13	boreratio
14	stroke
15	compressionratio
16	horsepower
17	peakrpm
18	citympg
19	highwaympg
20	CarModel
21	area_car
22	volume_car
23	weight_to_volume ratio
24	enginesize_to_powerratio
25	avg_milage
26	symboling_-1
27	symboling_0
28	symboling_1
29	symboling_2
30	symboling_3
31	carbody_HARDTOP
32	carbody_HATCHBACK
33	carbody_SEDAN
34	carbody_WAGON
35	drivewheel_FWD
36	drivewheel_RWD
37	enginetype_DOHCV
38	enginetype_L
39	enginetype_OHC
40	enginetype_OHCF
41	enginetype_OHCV
42	enginetype_ROTOR
43	fuelsystem_2BBL
44	fuelsystem_4BBL
45	fuelsystem_IDI
46	fuelsystem_MFI
47	fuelsystem_MPFI
48	fuelsystem_SPDI
49	fuelsystem_SPFI
50	CarCompany_AUDI
51	CarCompany_BMW
52	CarCompany_BUICK
53	CarCompany_CHEVROLET
54	CarCompany_DODGE
55	CarCompany_HONDA
56	CarCompany_ISUZU
57	CarCompany_JAGUAR
58	CarCompany_MAZDA
59	CarCompany_MERCURY
60	CarCompany_MITSUBISHI
61	CarCompany_NISSAN
62	CarCompany_PEUGEOT
63	CarCompany_PLYMOUTH
64	CarCompany_PORSCHE
65	CarCompany_RENAULT
66	CarCompany_SAAB
67	CarCompany_SUBARU
68	CarCompany_TOYOTA
69	CarCompany_VOLKSWAGEN
70	CarCompany_VOLVO
71	CarCompany_VOLVO

HENRY HARVIN® EDUCATION

5.Preparing Training & Test Data Set

- **Preparing Independent variables**

```
features = cars_data.columns.tolist()
```

- **Preparing dependent variables**

```
features.remove('price')
```

- **Putting feature variable to X**

```
X = cars_data[features]
```

- **Putting response variable to y**

```
y = cars_data['price']
```

- **Split into train and test from sklearn.model_selection import train_test_split**

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7,test_size = 0.3, random_state=100)
```

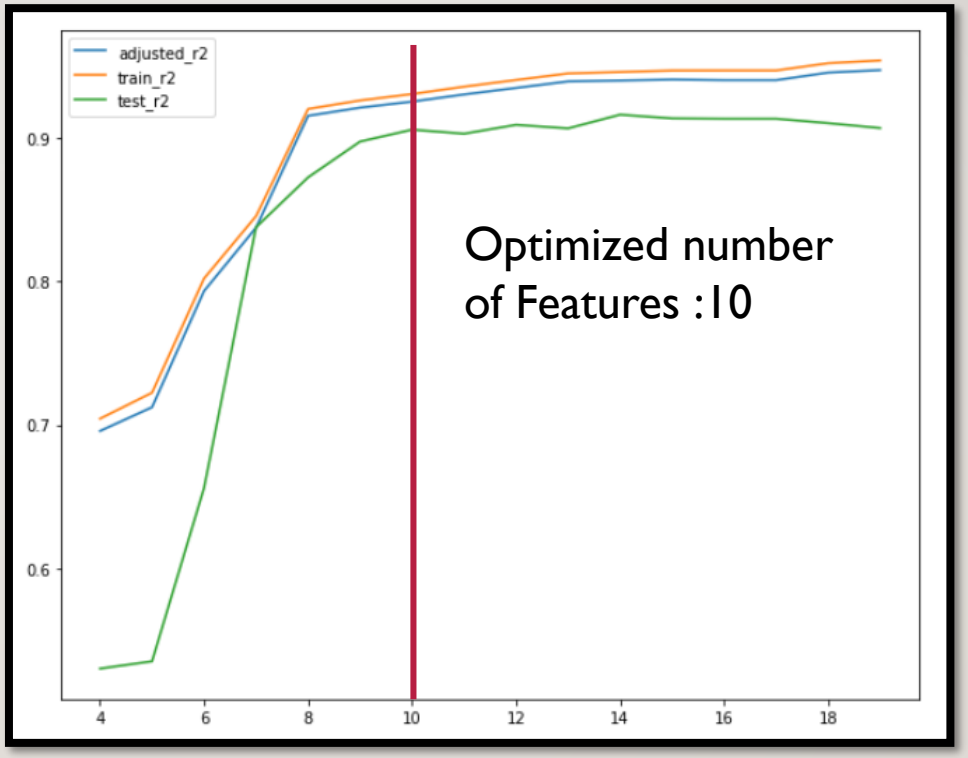
6.Model Building – Linear Regression

Linear Regression - Ordinary Least Squares & ranking based on Recursive Feature Elimination.

Dep. Variable:	price	R-squared:	0.930
Model:	OLS	Adj. R-squared:	0.925
Method:	Least Squares	F-statistic:	176.4
Date:	Fri, 11 Oct 2019	Prob (F-statistic):	2.89e-71
Time:	11:37:08	Log-Likelihood:	-1293.6
No. Observations:	143	AIC:	2609.
Df Residuals:	132	BIC:	2642.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.319e+04	180.641	73.039	0.000	1.28e+04	1.36e+04
enginelocation	-1853.3711	337.510	-5.491	0.000	-2521.000	-1185.742
carlength	-3.671e+04	5605.222	-6.548	0.000	-4.78e+04	-2.56e+04
carwidth	-1.521e+04	2801.370	-5.430	0.000	-2.08e+04	-9670.647
carheight	8206.6102	1993.989	4.116	0.000	4262.302	1.22e+04
curbweight	3996.0829	462.346	8.643	0.000	3081.517	4910.649
area_car	7.32e+04	1.05e+04	6.966	0.000	5.24e+04	9.4e+04
volume_car	-2.659e+04	5631.189	-4.721	0.000	-3.77e+04	-1.54e+04
CarCompany_BMW	2147.0989	179.803	11.941	0.000	1791.431	2502.766
CarCompany_BUICK	881.3579	236.023	3.734	0.000	414.480	1348.236
CarCompany_PORSCHE	820.7856	282.008	2.911	0.004	262.947	1378.625

Omnibus:	21.589	Durbin-Watson:	1.903
Prob(Omnibus):	0.000	Jarque-Bera (JB):	33.775
Skew:	0.764	Prob(JB):	4.63e-08
Kurtosis:	4.826	Cond. No.	162.



- ✓ **Mean Absolute Error (OLS): 1863.34 price.**
- ✓ **Accuracy OLS: 86.37 %.**
- ✓ r2_score : 0.9056

6.Model Building – Random Forest Regression

```
# Using Skikit-learn to split data into training and testing sets
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size = 0.25, random_state = 42)
```

- Training Features Shape: (153, 68);Training Labels Shape: (153,)
- Testing Features Shape: (52, 68); Testing Labels Shape: (52,)

```
# Limit depth of tree to 3 levels
rf_small = RandomForestRegressor(n_estimators=10, max_depth = 3)
rf_small.fit(train_features, train_labels)
```

```
# Get numerical feature importance
importances = list(rf.feature_importances_)
```

✓ **Mean Absolute Error (RFE): 1278.39 price.**

✓ **Accuracy RFE: 90.24 %.**

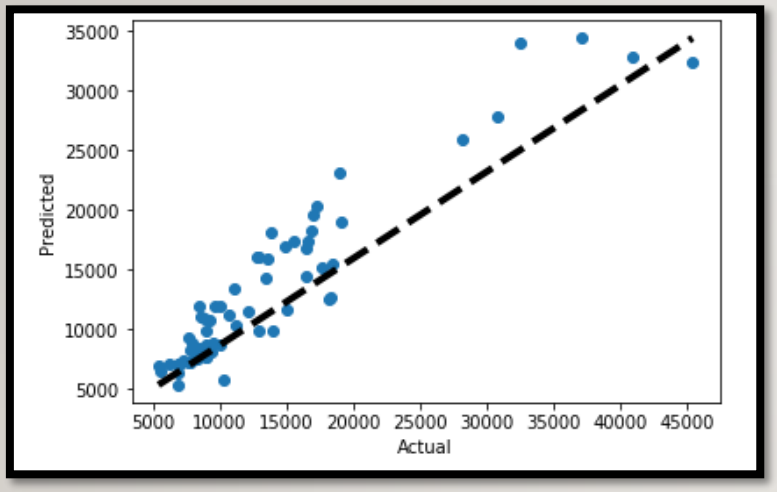
Feature : Importance

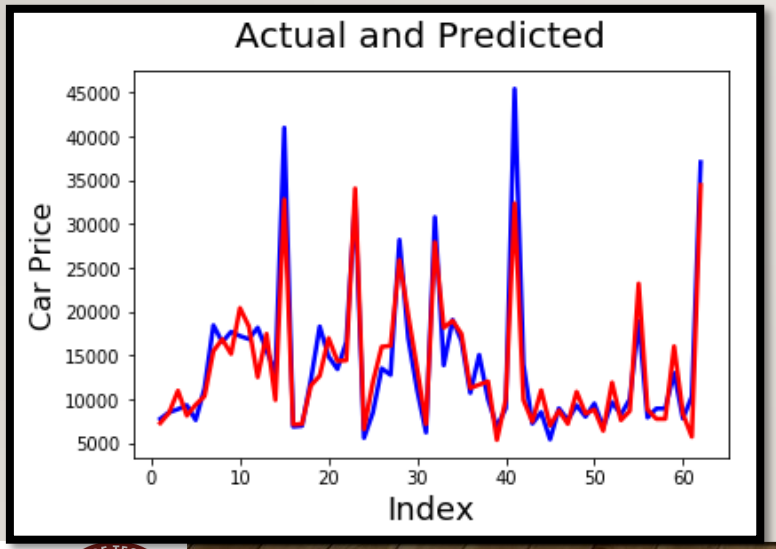
```
Variable: enginesize                Importance: 0.61
Variable: curbweight                Importance: 0.17
Variable: highwaympg                Importance: 0.07
Variable: avg_milage                Importance: 0.05
Variable: horsepower                Importance: 0.03
Variable: carlength                 Importance: 0.01
Variable: carwidth                  Importance: 0.01
Variable: area_car                  Importance: 0.01
Variable: enginesize_to_powerratio Importance: 0.01
Variable: CarCompany_BMW            Importance: 0.01
```

HENRY HARVIN® EDUCATION

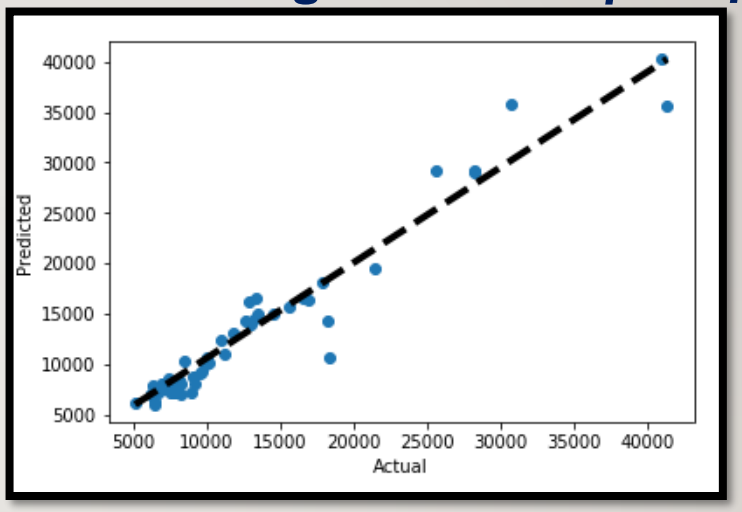
6.Model Building – Results Comparison

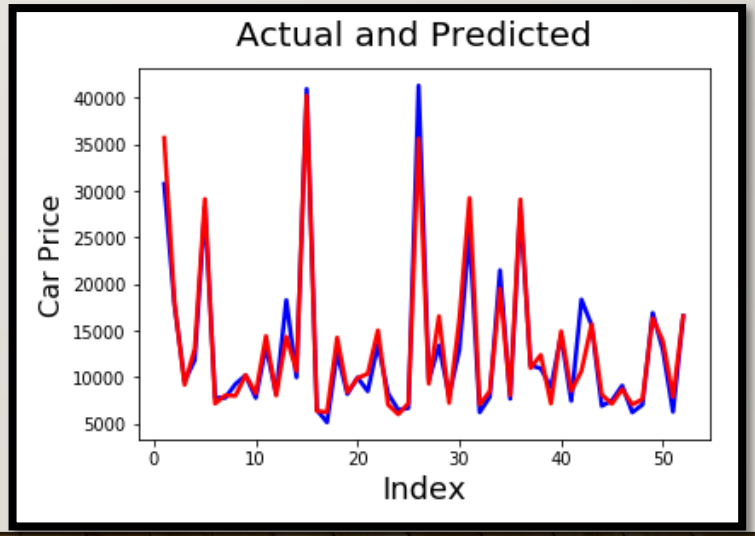
Linear Regression – OLS & RFE





Random Forest Regression – Depth of tree :03

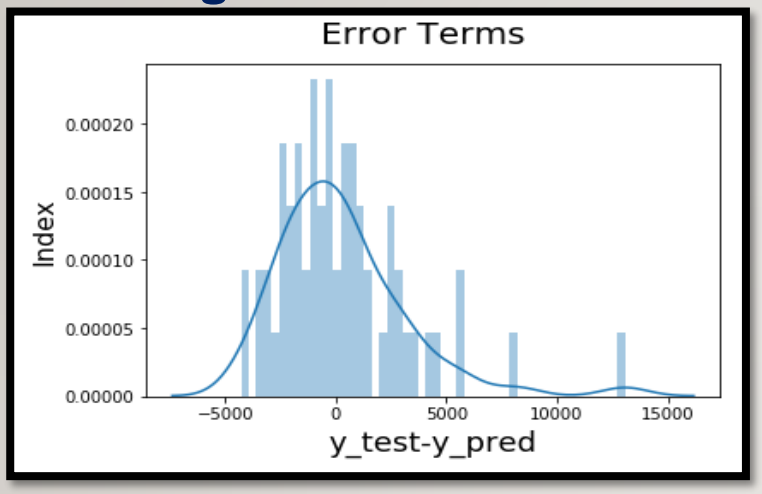




HENRY HARVIN® EDUCATION

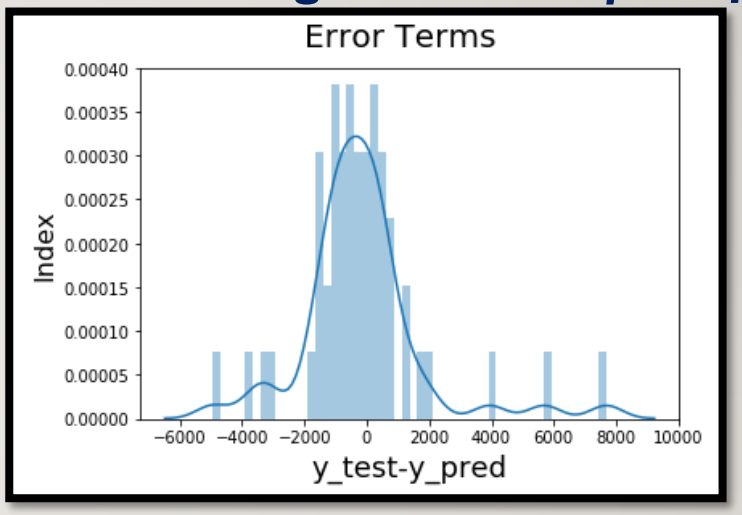
6.Model Building – Results Comparison

Linear Regression – OLS & RFE



Number of Features	Linear Regression with OLS & RFE	P>|t|
	const	
1	enginelocation	0
2	carlength	0
3	carwidth	0
4	carheight	0
5	curbweight	0
6	area_car	0
7	volume_car	0
8	CarCompany_BMW	0
9	CarCompany_BUICK	0
10	CarCompany_PORSCHE	0

Random Forest Regression – Depth of tree :03



Number of Features	Random Forest Regression & Tree depth level :03	Importance
1	enginesize	0.61
2	curbweight	0.17
3	highwaympg	0.07
4	avg_milage	0.05
5	horsepower	0.03
6	carlength	0.01
7	carwidth	0.01
8	area_car	0.01
9	enginesize_to_powerratio	0.01
10	CarCompany_BMW	0.01

7.Recommendation

- Prediction of "CAR PRICE " based on given set of data was carried out using Linear Regression (OLS& RFE) and compared with Random Forest Regression Model with number of tree level :03.
- Random Forest Regression Model gives better prediction in terms of accuracy than Linear Regression Model.

Linear Regression (OLS & RFE)	Random Forest – Tree Level :03
Mean Absolute Error OLS: 1863.34 price	Mean Absolute Error RFE: 1278.39 price.
Accuracy OLS: 86.37 %.	Accuracy RFE: 90.24 %.

- Prediction of Car price depends on following features as tabulated below.

S.No	Features
1	enginesize
2	curbweight
3	highwaympg
4	avg_milage
5	horsepower
6	carlength
7	carwidth
8	area_car
9	enginesize_to_powerratio
10	CarCompany_BMW