Title - OLX car sellers are finding it difficult to put a price on the cars they are selling.So we predict Car selling price using Regression

Group number - 20

Participants list - 1. Bablu 2. Shakti Singh Rathore

Problem statement- Regression is an important machine learning model for these kinds of problems OLX Group is a global online marketplace operating in 45 countries. The OLX marketplace is a platform for buying and selling services .OLX car sellers are finding it difficult to put a price on the cars they are selling.So we predict Car selling price and we use features of the cars and their prices. For this kind of project of Price predict, we will apply the linear regression and Random forests and evaluate the result based on the training and testing a set of the data.

Overall summary of your solution-

1. Web Scraping-

   Scrap the data from OLX site using python library bs4 in BeautifulSoup and extract feature of Second hand car and around 800 rows and 16 columns and convert into the data frame.

2. Loading the Data and getting first intuitions
   A. Identifiying missing values
   B. Plausibility check of numerical attributes
   C. Checking values of categorical attributes
3. Visualization /Plot and Histogram of "Price"
   A. Rough Data Cleansing for plausible Visualization
   B. Checking the amount of data after rough cleansing
   C. Check Distribution and Data
4. Data Spliting

   Split data into traning and testing and after that using Linear Regression ,Random forests and more algorithm

5. Buidling Custom-Transformers and Preprocessing-Pipelines
   A. Calculating VIF and Noramlize data using Logrithm
   B. Data Cleansing (Filtering, Replacing NaN-Values, Frequency count and imput data)
   C. Custom-Transformers & Pipelines L1 and L2 Regularization
   D. Training and comparing Models

E. Feature Importance (Note: I can only show numerical features. Help is appreciated if you could help me out with categorical features.)

Detailed description and analysis of the solutions provided, including each technique used, supported by Tables / Graphs / Charts / Metrics

Description and Analysis of the solutions provided:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import LinearRegression
import statsmodels.formula.api as smf
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import seaborn as sns
from sklearn import model_selection
```

	Unnamed: 0	Variant	Make_Month	No_of_Owners	Color	Type_of_Car	Transmission	Insurance_Type	Condition	Registration_Place	...	Registr
0	0	CitySV	March	First	Grey	Sedans	Manual	Comprehensive	Used	MH	...	
1	1	VXCVTi-vtec	July	First	Brown	Sedans	Automatic	Comprehensive	Used	MH	...	
2	2	City1.5VManual	June	First	Brown	Sedans	Manual	Comprehensive	Used	MH	...	
3	3	SwiftVDi	June	First	Other	Hatchback	Manual	NoInsurance	Used	MH	...	
4	4	SwiftDzireVXi1.2BS-IV	March	First	Silver	Sedans	Manual	Comprehensive	Used	MH	...	
...	...	...	...	...	...	...	...	...	...	...	...	
803	803	NaN	November	NaN	Silver	NaN	NaN	NoInsurance	NaN	GJ	...	
804	804	SwiftLXi	May	First	Silver	Hatchback	Manual	ThirdParty	Used	WB	...	
805	805	WagonR1.0LXiCNG	August	First	Grey	Hatchback	Manual	NoInsurance	Used	MH	...	
806	806	AltoK10VXi	November	First	Silver	Sedans	Manual	ThirdParty	Used	MH	...	
807	807	ZenLXiBS-III	June	Second	Silver	Hatchback	Manual	ThirdParty	Used	KA	...	

i08 rows × 44 columns

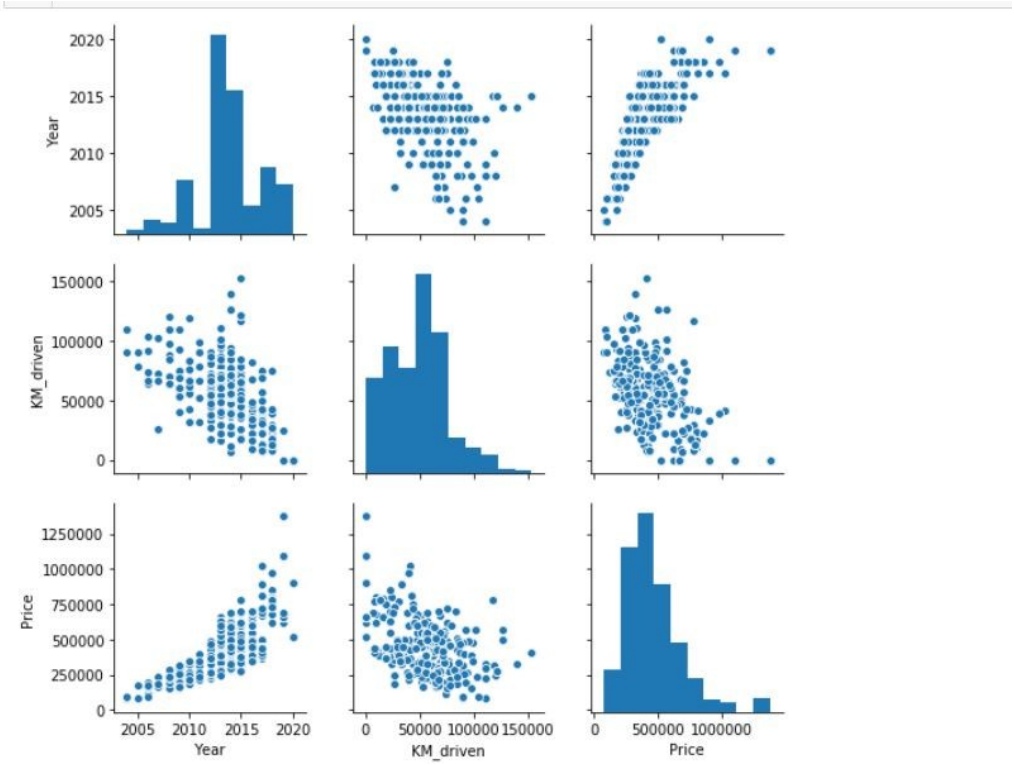
And then we read the csv file where data is stored by creating a dataframe as:

```
df=pd. read_csv('forestfires.csv')
df. head ()
```

Linearity: The relationship between the independent and dependent variables should be linear. The linearity assumption can best be tested with scatter plots.

import seaborn as sns

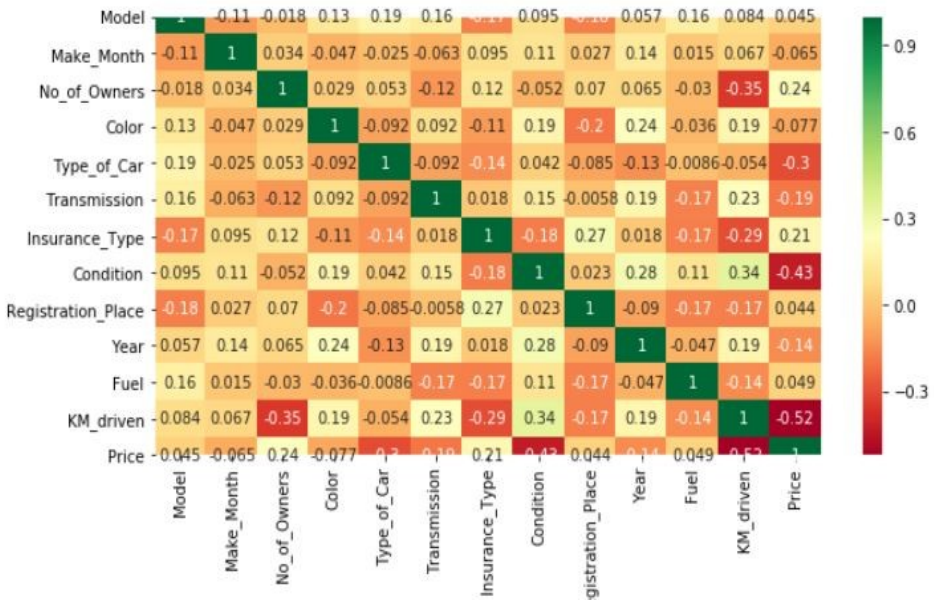
sns.pairplot(df)



)

Multicollinearity: Linear regression assumes that there is little or no multicollinearity in the data**.** It can be tested through VIF, Correlation matrix.

Heat Map:



Correlation Matrix:

	Model	Make_Month	No_of_Owners	Color	Type_of_Car	Transmission	Insurance_Type	Condition	Registration_Place	Year	
Model	1.000000	-0.107473	-0.017795	0.131404	0.187031	0.159551	-0.170226	0.094845	-0.176045	0.057243	0.
Make_Month	-0.107473	1.000000	0.034495	-0.046952	-0.025286	-0.063360	0.094561	0.109600	0.027268	0.140856	0.
No_of_Owners	-0.017795	0.034495	1.000000	0.028610	0.052684	-0.120552	0.122788	-0.051689	0.069733	0.065488	-0.
Color	0.131404	-0.046952	0.028610	1.000000	-0.092328	0.092372	-0.108337	0.191445	-0.196447	0.241763	-0.
Type_of_Car	0.187031	-0.025286	0.052684	-0.092328	1.000000	-0.092294	-0.142720	0.041608	-0.085349	-0.130715	-0.
Transmission	0.159551	-0.063360	-0.120552	0.092372	-0.092294	1.000000	0.017652	0.154278	-0.005781	0.194190	-0.
Insurance_Type	-0.170226	0.094561	0.122788	-0.108337	-0.142720	0.017652	1.000000	-0.177434	0.274687	0.017743	-0.
Condition	0.094845	0.109600	-0.051689	0.191445	0.041608	0.154278	-0.177434	1.000000	0.022891	0.279548	0
Registration_Place	-0.176045	0.027268	0.069733	-0.196447	-0.085349	-0.005781	0.274687	0.022891	1.000000	-0.090337	-0.
Year	0.057243	0.140856	0.065488	0.241763	-0.130715	0.194190	0.017743	0.279548	-0.090337	1.000000	-0.
Fuel	0.163855	0.015331	-0.030359	-0.036042	-0.008597	-0.169179	-0.172990	0.111569	-0.166335	-0.047073	1.
KM_driven	0.084150	0.066594	-0.348487	0.192295	-0.054129	0.233560	-0.294191	0.336426	-0.171979	0.191667	-0.
Price	0.044620	-0.064534	0.237494	-0.076728	-0.297439	-0.189390	0.208429	-0.426967	0.043609	-0.142946	0.

VIF:

```
17      print(names[i]   ,              ,rsq,     ,vif,     )
```

```
variable    Rsquare value      VIFvalue
Model           0.15    1.18
Make_Month            0.09    1.1
No_of_Owners            0.1    1.11
Color           0.1    1.11
Type_of_Car            0.08    1.09
Transmission            0.13    1.15
Insurance_Type            0.18    1.22
Condition           0.42    1.72
Registration_Place            0.18    1.22
Year           0.18    1.22
Fuel           0.16    1.19
KM_driven           0.46    1.85
```

Conclusion: With VIF > 5 there is an indication that multicollinearity may be present; with VIF > 10 there is certainly multicollinearity among the variables. We can see that there no variables that are greater than 10,if it is than centering the data the simplest way to address the problem is to remove independent variables with high VIF values

	Features	VIF
0	const	83.06
8	KM_driven	2.16
6	Year	2.08
7	Fuel	1.31
5	Condition	1.23
2	Type_of_Car	1.17
3	Transmission	1.15
1	Model	1.13
4	Insurance_Type	1.13

Feature Importances-

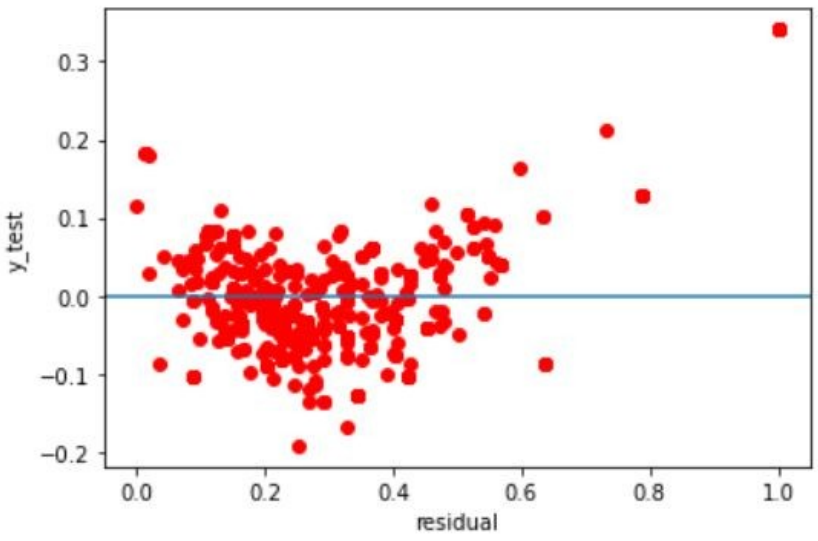
```
Feature ranking:
1. feature 11 (0.434944)
2. feature 4 (0.189643)
3. feature 9 (0.185361)
4. feature 0 (0.089601)
5. feature 10 (0.033305)
6. feature 3 (0.031574)
7. feature 1 (0.011403)
8. feature 5 (0.009501)
9. feature 2 (0.009366)
10. feature 6 (0.002554)
11. feature 8 (0.001727)
12. feature 7 (0.001019)
['Model' 'Make_Month' 'No_of_Owners' 'Color' 'Type_of_Car' 'Transmission'
 'Insurance_Type' 'Condition' 'Registration_Place' 'Year' 'Fuel'
 'KM_driven']
```

Models-

Linear Regression :-

- When used all the features which are encoded to fit the model and the R-squared value is 80.02%.
- There are show residual plot its show are random point.
- There are show Train RMSE ,Train MSE, and also show for test .
- As the difference between training and testing MSE and RMSE is less the model is not having overfitting

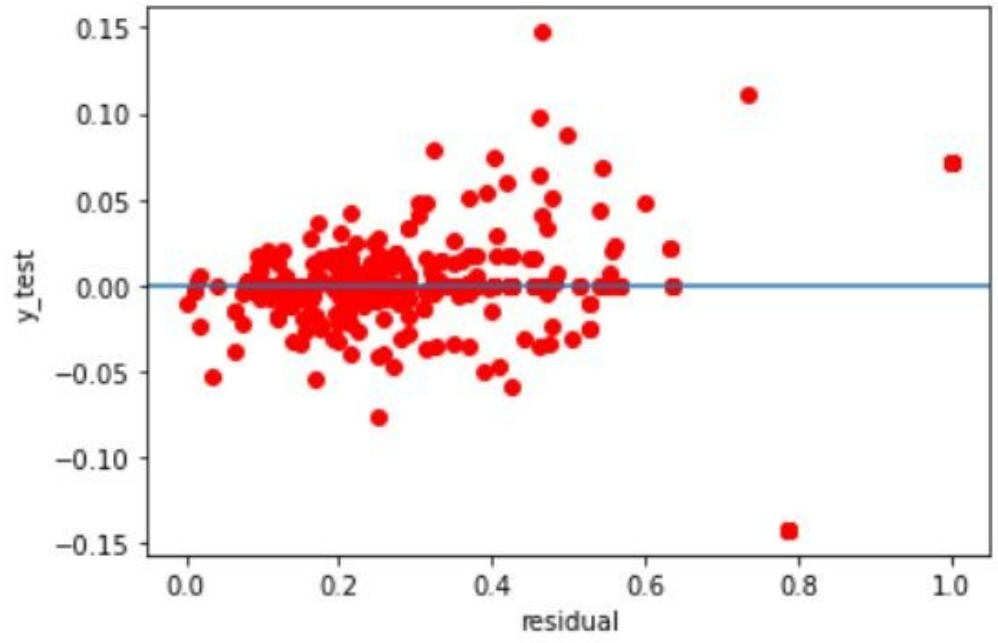
```
   Linear   Regression
Train MSE is 0.00596568068314689
Test MSE is 0.005411058157453982
Train RMSE is: 0.07723781899527517
Test RMSE is 0.0735598950342779
r2 score is 0.8002072860979079
   Linear   Regression
```



KNearest Neighbor :-

- When used all the features which are encoded to fit the model and the R-squared value is 96.11%.
- There are show residual plot its show are random point.
- There are show Train RMSE ,Train MSE, and also show for test .
- As the difference between training and testing MSE and RMSE is less the model is not having overfitting

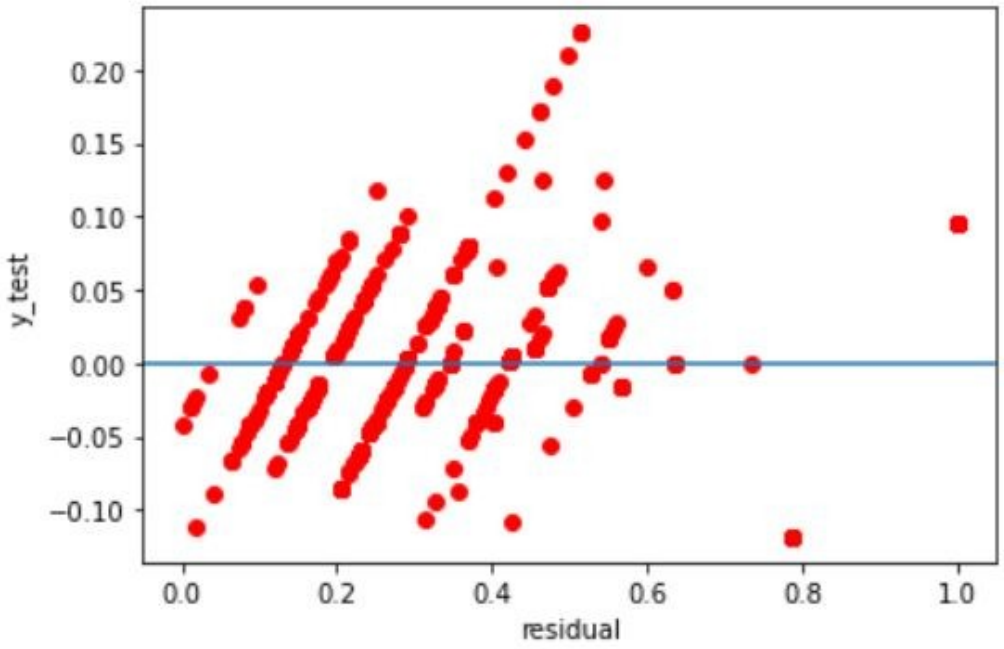
```
       KNearest Neighbor
Train MSE is 0.0007594520887611 39
Test MSE is 0.0010520565196397704
Train RMSE is: 0.027558158297700867
Test RMSE is 0.03243542075632395
r2 score is 0.9611548756045676
       KNearest Neighbor
```



Decision Tree :-

- When used all the features which are encoded to fit the model and the R-squared value is 88.69%.
- There are show residual plot its show are random point.
- There are show Train RMSE ,Train MSE, and also show for test .
- As the difference between training and testing MSE and RMSE is less the model is not having overfitting

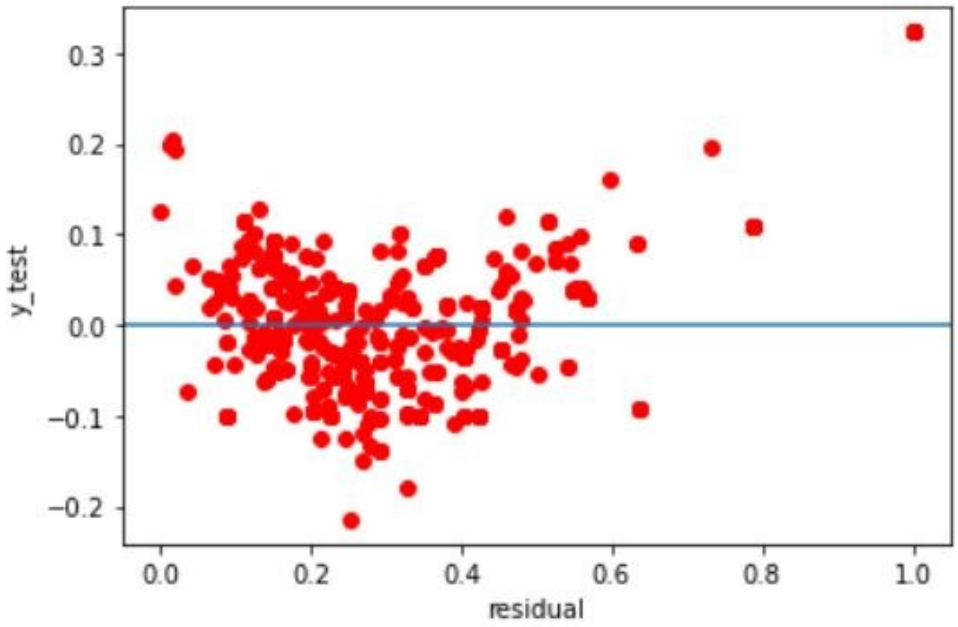
```
                Decision_Tree
Train MSE is 0.0028229480907654 67
Test MSE is 0.0030630343418223157
Train RMSE is: 0.0531314228189446
Test RMSE is 0.05534468666296988
r2 score is 0.886903462110277
                Decision_Tree
```



Support Vector Regression :-

- When used all the features which are encoded to fit the model and the R-squared value is 78.38%.
- There are show residual plot its show are random point.
- There are show Train RMSE ,Train MSE, and also show for test .
- As the difference between training and testing MSE and RMSE is less the model is not having overfit

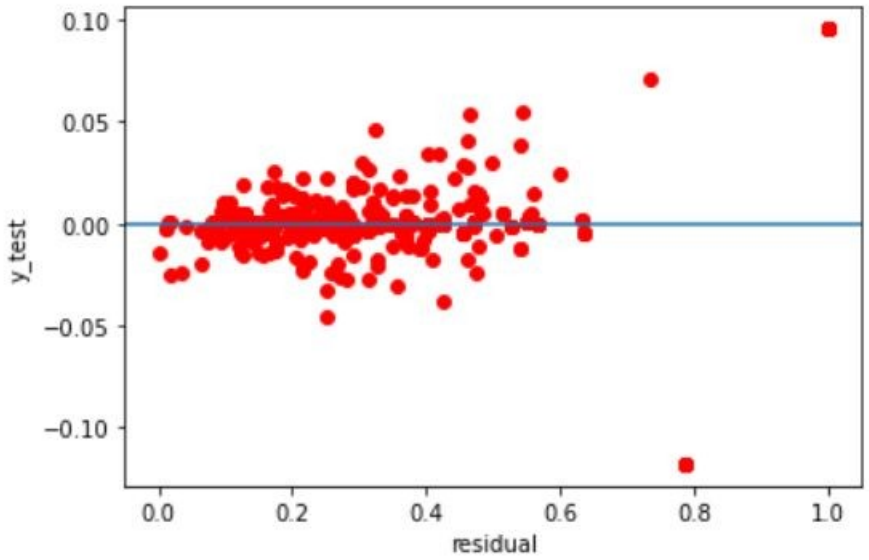
```
                Support Vector Regression
Train MSE is : 0.0062013065643427324
Test MSE is  : 0.0058613622836297214
Train RMSE is: 0.07874837499493391
Test RMSE is : 0.07655953424381395
r2 score is 0.7835806890752856
                Support Vector Regression
```



RandomForest Regressor :-

- When used all the features which are encoded to fit the model and the R-squared value96.70%.
- There are show residual plot its show are random point.
- There are show Train RMSE ,Train MSE, and also show for test .
- As the difference between training and testing MSE and RMSE is less the model is not having overfitting

```
       RandomForest Regressor
Train MSE is 0.0004924120766641 84
Test MSE is 0.0008913230966497077
Train RMSE is: 0.022190359994019564
Test RMSE is 0.02985503469516838
r2 score is 0.9670896421251823
       RandomForest Regressor
```



<u>Identification of the best model backed by the logic for selecting it</u>

Conclusions
I've tried to play with as much stuff as I could with this dataset in order to understand the very basic topics about:
data interpretation and selection Random forests and Ridge and lasso given good result

we are getting linear regression in accuracy 80 % and when we apply Regularization l1 and l2 we get more accuracy around 80 % and same as in random forests ,SVM,Decision tree gaving best accuracy 96% .

Lessons learnt from the project

Learn Various type of Machine learning algoritham and EDA process and Cleaning Dataset Learn about the of sklearn Library and Matplot lib libaray and learnt Web Scarping how to data scraping from any site .