Car Price Prediction Project

Submitted by:

Vikash Kumar Singh

ACKNOWLEDGMENT

In this project, I took references from our DataTrained's video of Shankargouda Tegginmani sir and some websites which are- https://www.youtube.com https://www.kaggle.com , https://www.github.com , https://stackoverflow.com and https://scikit-learn.org in completion of this project. The data source is provided by Flip Robo company.

INTRODUCTION

- **Problem statement**

  Used car price prediction problem has a certain value because different studies show that the market of used cars is destined to a continuous growth in the short term. In fact, leasing cars is now a common practice through which it is possible to get hold of a car by paying a fixed sum for an agreed number of months rather than buying it in its entirety. Once leasing period is over, it is possible to buy the car by paying the residual value, i.e., at the expected resale price. It is therefore in the interest of vendors to be able to predict this value with a certain degree of accuracy, since if this value is initially underestimated, the instalment will be higher for the customer which will most likely opt for another dealership. It is therefore clear that the price prediction of used cars has a high commercial value, especially in developed countries where the economy of *leasing* has a certain volume.

  This problem, however, is not easy to solve as the car's value depends on many factor including year of registration, manufacturer, model, mileage, horsepower, origin and several other specific information such as type of fuel and braking system, condition of bodywork and interiors, interior materials (plastics of leather), safer index, type of change (manual, assisted, automatic, semi-automatic), number of doors, number of previous owners, if it was previously owned by a private individual or by a company and the prestige of the manufacturer.

  Unfortunately, only a small part of this information is available

and therefore it is very important to relate the results obtained in terms of accuracy to the features available for the analysis. Moreover, not all the previously listed features have the same importance, some are more so than others and therefore is essential to identify the most important ones, on which to perform the analysis.
Since some attributes of the dataset aren't relevant to our analysis, they have been discarded; so, as mentioned above, this fact must be taken into account when conclusions on the accuracy are drawn.

- ## Conceptual Background of the Domain Problem
    1. In preparation of any Machine Learning Model, we have to have conceptual knowledge of cleansing, scaling, outlier and many more method to perform the EDA analysis
    2. The knowledge of Data Visualization is also required to observe the skewness, outlier, correlation and multicollinearity of the given dataset and you are also required to have knowledge to reduce the following observation.
    3. Use standard techniques (PCA, Scaling, encoding, cross-validation, grid search, ensemble techniques) wherever applicable.

- ## Review of Literature
    The literature study gives an overview of this project and the EDA methods, the feature engineering methods that have been used in this study. As well as about evaluation metrics to measure the performance of the algorithms.

- **Motivation for the Problem Undertaken**

  Presently, I am doing an internship with FlipRobo company and they have given us this project to build a Machine Learning Model to predict whether the customer is a defaulter or not i.e., he is paying his loaned amount on time or not. This project gives me an opportunity to recall the teaching of Datatrained's mentor and to build multiple classification model with better accuracy. This will help me in upgrading my skills and knowledge.

Analytical Problem Framing

- **Mathematical/ Analytical Modeling of the Problem**

  In the given dataset, target is in continuous form so, the model will be regression model (supervised learning) in which, we will design a predictive model to analyses the relation between features and target variables (sale price of used cars).

Data Sources and their formats

We have scraped the data from the website: car24 with different location in India and saved the required data as CSV file.

Link: - https://www.cars24.com/

- ## Data Pre-processing Done

In pre-processing, we have done the following steps:

1.) Imported some important libraries in our jupyter notebook.
2.) Imported the given dataset
3.) Check the details of each feature from the given dataset.
4.) Cleanse the dataset i.e., check for null values, duplicate value, shape of dataset etc.
5.) Drop the unnecessary features.
6.) Checked the skewness
7.) Transform the dataset with square root transformation
8.) Outlier detection
9.) Checked for correlation
10.) Visualization
11.) Separation of features and label
12.) Scaling of the data
13.) Train test split.

- ## Data Inputs- Logic- Output Relationships

Many things are taken as considered before buying any used cars. It mainly depends on manufacturing year of car and kilometres driven as of now, and how many people were using it. As we know time can reduce the value of any materialistic things, this is also applied here. Older car gets very less value. These same is also applying in this model. We will thoroughly analyse each feature to get logic behind the label and features i.e., input- output relationships.

- ## State the set of assumptions (if any) related to the problem under consideration

My only presumption was that the car price will decrease with the age(manufacturing year).

- ## Hardware and Software Requirements and Tools Used

- **Hardware tools:**

1. Windows laptop
2. i5 processor
3. 4GB ram
4. 250 GB SSD card

- **Software tools:**

1. windows 10
2. Anaconda Navigator
3. Jupyter Notebook
4. Python

- **Libraries and packages:**

1. Pandas

2. NumPy
3. SciPy
4. Seaborn
5. Mat plot
6. Sklearn

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

  From the domain knowledge and my own understanding, I have followed these steps:

  1. Importing the given test and train dataset
  2. Checking null values in the dataset
  3. Imputation of given dataset
  4. Transformation of given dataset
  5. Standardization of given dataset
  6. Splitting the train and test dataset

- **Testing of Identified Approaches (Algorithms)**

  1. Linear Regression
  2. Random Forest Regressor
  3. Decision Tree Regressor
  4. Support Vector Regressor
  5. KNeighbors Regressor
  6. XGBRegressor

7. Lasso
8. Ridge
9. AdaBoost Regressor
10. Gradient Boosting Regressor

- **Run and evaluate selected models**
    1. **Linear regression**

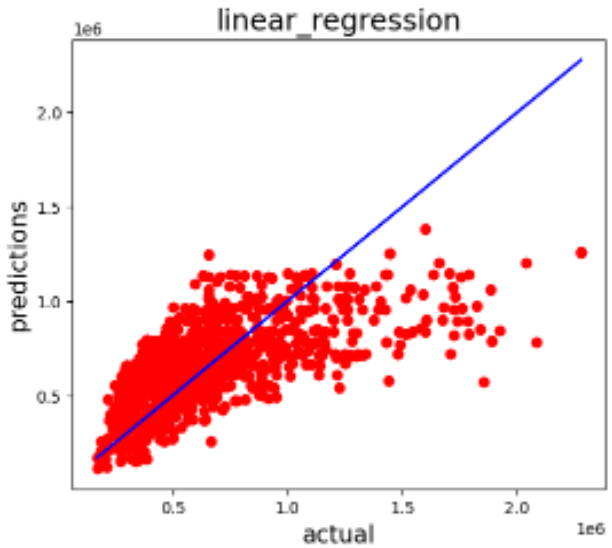
1.LinearRegression

```
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm

regressor=LinearRegression()
regressor.fit(X_train,y_train)
y_pred = regressor.predict(X_test)
regression_results(y_test,y_pred)
model_accuracy(regressor)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('linear_regression', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.427
R2:  0.4266
Adjusted_r2:  0.4235
MAE:  173336.3776
MSE:  60181021981.5205
RMSE:  245318.2056
Accuracy: 45.20 %
Standard Deviation: 2.39 %
```



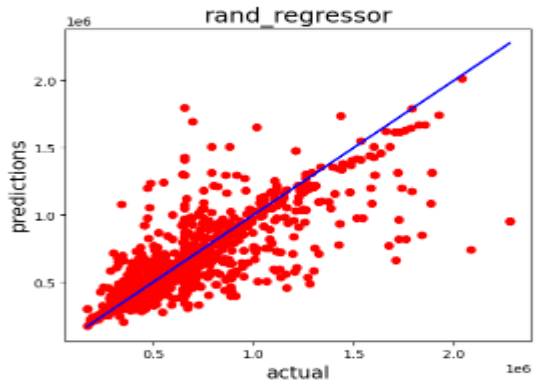
2. Random forest regressor

2. RandomForestRegressor

```
rand_regressor = RandomForestRegressor()
rand_regressor.fit(X_train, y_train)
y_pred_rf = rand_regressor.predict(X_test)
regression_results(y_test,y_pred_rf)
model_accuracy(rand_regressor)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_rf, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('rand_regressor', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.6214
R2:  0.6198
Adjusted_r2:  0.6177
MAE:  111104.9625
MSE:  39910458090.9337
RMSE:  199776.0198
Accuracy: 64.64 %
Standard Deviation: 2.93 %
```



3. Decision tree regressor

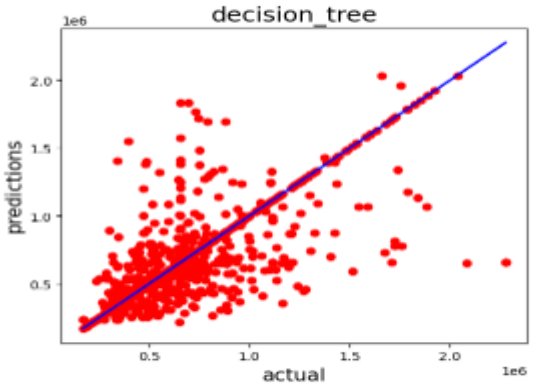
3. DecisionTreeRegressor

```
from sklearn.tree import DecisionTreeRegressor

decision_tree=DecisionTreeRegressor(criterion='mse',splitter='random',random_state=10)
decision_tree.fit(X_train, y_train)
y_pred_dt = decision_tree.predict(X_test)
regression_results(y_test,y_pred_dt)
model_accuracy(decision_tree)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_dt, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('decision_tree', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.5529
R2:  0.551
Adjusted_r2:  0.5486
MAE:  92824.7741
MSE:  47122909135.2476
RMSE:  217078.1176
Accuracy: 55.96 %
Standard Deviation: 4.03 %
```



4. SVR

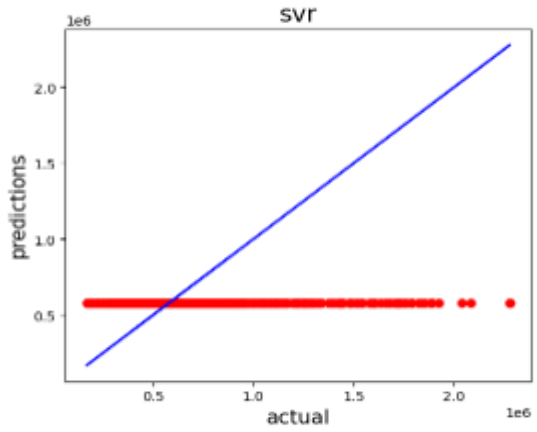
```
from sklearn.svm import SVR

svr=SVR()
svr.fit(X_train, y_train)
y_pred_svr = svr.predict(X_test)

regression_results(y_test,y_pred_svr)
model_accuracy(svr)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_svr, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('svr', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.0009
R2:  -0.0504
Adjusted_r2:  -0.0661
MAE:  227483.2255
MSE:  111296212787.9657
RMSE:  333610.8703
Accuracy: -5.26 %
Standard Deviation: 1.28 %
```



5. KNeighbors regressor

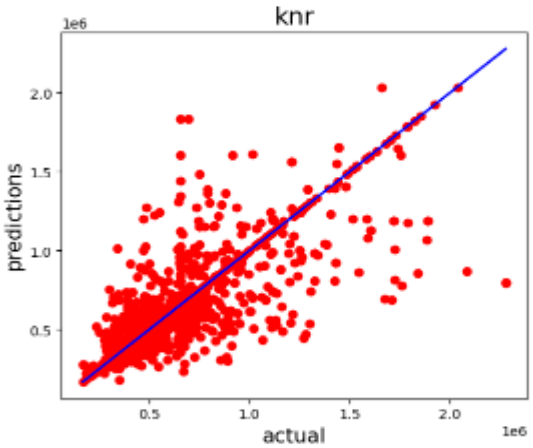
```
from sklearn.neighbors import KNeighborsRegressor

knr=KNeighborsRegressor(n_neighbors=2)
knr.fit(X_train, y_train)
y_pred_knr = knr.predict(X_test)

regression_results(y_test,y_pred_knr)
model_accuracy(knr)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_knr, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('knr', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.5632
R2:  0.5617
Adjusted_r2:  0.5593
MAE:  114598.922
MSE:  46008562155.8453
RMSE:  214496.0656
Accuracy: 56.71 %
Standard Deviation: 3.88 %
```



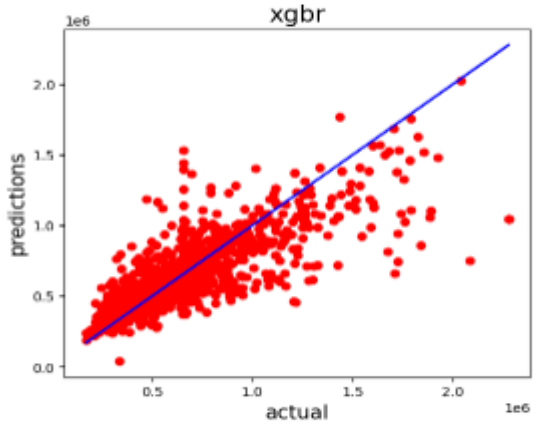
6. **XGBRegresso**

```
from xgboost import XGBRegressor
xgbr=XGBRegressor(random_state=10)
xgbr.fit(X_train, y_train)
y_pred_xgbr = xgbr.predict(X_test)

regression_results(y_test,y_pred_xgbr)
model_accuracy(xgbr)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_xgbr, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('xgbr', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.6167
R2:  0.6151
Adjusted_r2:  0.613
MAE:  129581.4118
MSE:  48403849848.7953
RMSE:  201005.0091
Accuracy: 63.98 %
Standard Deviation: 2.04 %
```



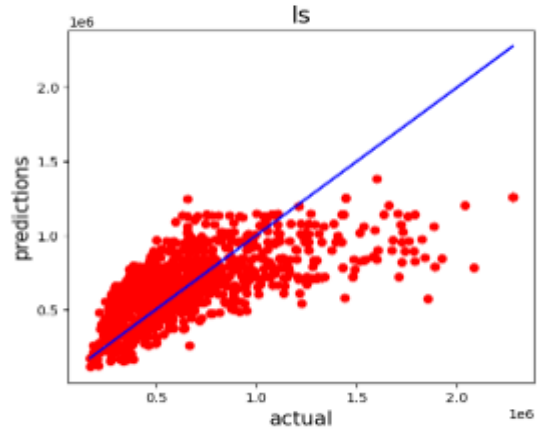
7. **Lasso**

```
ls = Lasso()
ls=Lasso(random_state=10)
ls.fit(X_train, y_train)
y_pred_ls = ls.predict(X_test)

regression_results(y_test,y_pred_ls)
model_accuracy(ls)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_ls, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('ls', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.427
R2:  0.4266
Adjusted_r2:  0.4235
MAE:  173336.2309
MSE:  60181055215.8438
RMSE:  245318.2733
Accuracy: 45.18 %
Standard Deviation: 2.43 %
```



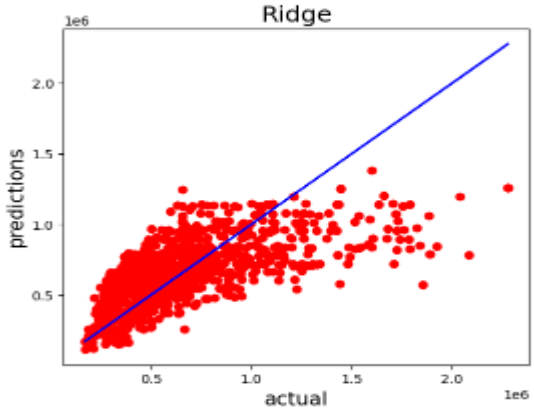
8. **Ridge**

```
rd = Ridge()
rd=Ridge(random_state=10)
rd.fit(X_train, y_train)
y_pred_rd = rd.predict(X_test)

regression_results(y_test,y_pred_rd)
model_accuracy(rd)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_rd, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('Ridge', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.427
R2:  0.4266
Adjusted_r2:  0.4235
MAE:  173327.9116
MSE:  60180354944.9183
RMSE:  245316.846
Accuracy: 45.19 %
Standard Deviation: 2.43 %
```



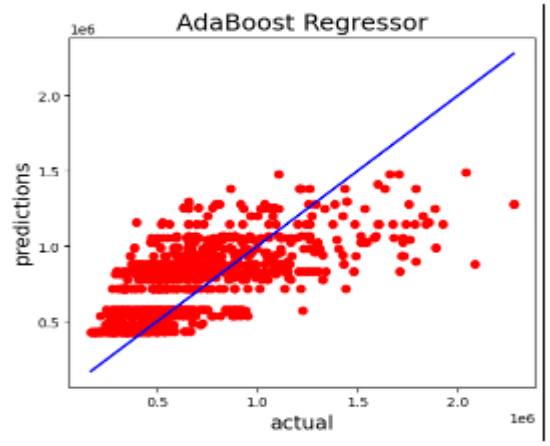
9. **AdaBoosting regressor**

```
abr = AdaBoostRegressor()
abr=AdaBoostRegressor(random_state=10)
abr.fit(X_train, y_train)
y_pred_abr = abr.predict(X_test)

regression_results(y_test,y_pred_abr)
model_accuracy(abr)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_abr, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('AdaBoost Regressor', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.4142
R2:  0.3164
Adjusted_r2:  0.3127
MAE:  211656.4315
MSE:  71754581182.1588
RMSE:  267870.456
Accuracy: 36.39 %
Standard Deviation: 5.89 %
```



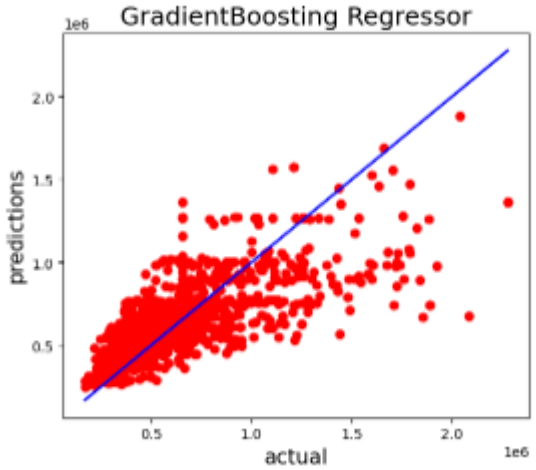
10. **GradientBoosting classifier**

```
gbr = GradientBoostingRegressor()
gbr=GradientBoostingRegressor(random_state=10)
gbr.fit(X_train, y_train)
y_pred_gbr = gbr.predict(X_test)

regression_results(y_test,y_pred_gbr)
model_accuracy(gbr)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_gbr, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('GradientBoosting Regressor', fontsize = 18)
plt.show()
```

```
Explained_variance:  0.501
R2:  0.4997
Adjusted_r2:  0.497
MAE:  157503.3217
MSE:  52507396916.3644
RMSE:  229144.9256
Accuracy: 53.61 %
Standard Deviation: 3.23 %
```



We have built different models of regression, in which we are evaluating them with cross validation score, Explained variance,r2 score ,adjusted r2 score, MSE, RMSE .Among all, Random forest regressor model has highest accuracy.

For further improvement in the accuracy, we have done hyperparameter tuning for this this model and it improved the accuracy.

After hyperparameter tuning.

```
param_grid = {
            "n_estimators"      : [10,20,30],
            "max_features"      : ["auto", "sqrt", "log2"],
            "min_samples_split" : [2,4,8],
            "bootstrap": [True, False],
            }
```

```
grid_search=GridSearchCV(estimator=rand_regressor,cv=10,param_grid=param_grid,verbose=True,n_jobs=-1)
```

After tuning its accuracy has increased by very little margin.

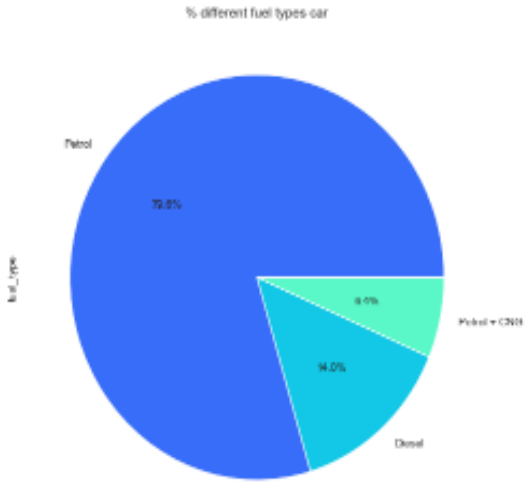
```
rand_regressor= RandomForestRegressor(bootstrap= True,
                                      max_features='auto',
                                      min_samples_split= 2,
                                      n_estimators=30)
rand_regressor=RandomForestRegressor(random_state=10)
rand_regressor.fit(X_train, y_train)
y_pred_rand_regressor =rand_regressor.predict(X_test)

regression_results(y_test,y_pred_rand_regressor)
model_accuracy(rand_regressor)

plt.figure(figsize = (6,5))
plt.scatter(x = y_test, y=y_pred_rand_regressor, color = 'r')
plt.plot(y_test,y_test, color = 'b')
plt.xlabel('actual', fontsize = 15)
plt.ylabel('predictions', fontsize = 15)
plt.title('RandomForestRegressor', fontsize = 18)
plt.show()
```

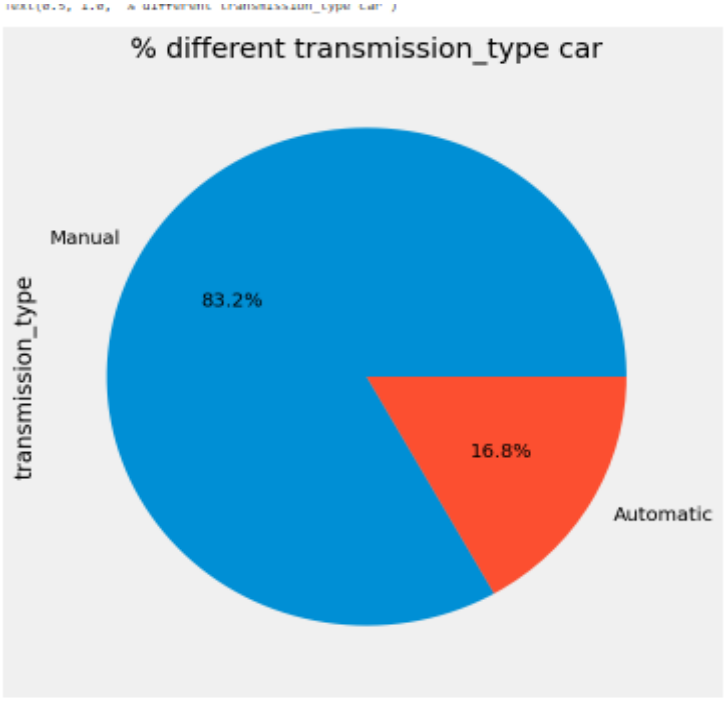
```
Explained_variance:  0.6271
R2:  0.6257
Adjusted_r2:  0.6236
MAE:  110290.2387
MSE:  39290863657.0183
RMSE:  198219.2313
Accuracy: 64.98 %
Standard Deviation: 3.02 %
```

- **Key Metrics for success in solving problem under consideration**
  a. Cross validation score
  b. Explained variance
  b. R2 score
  c. Adjusted r2 score
  d. MSE
  e. RMSE score
  f. Accuracy
- **Visualizations**
  1. Pie plot of ‘fuel_type’ column.



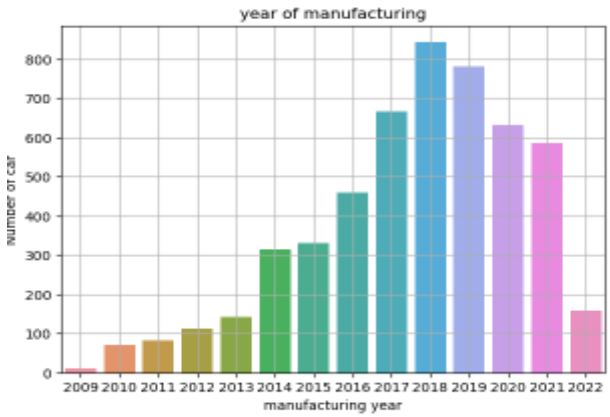
In this plot, we can clearly see that the car variants of patrol is maximum and followed by diesel and least from the variant of hybrid i.e., Patrol +CNG.

2. Pie plot of ‘tranmission_type’ column



In this feature , Manual model car is maximum .

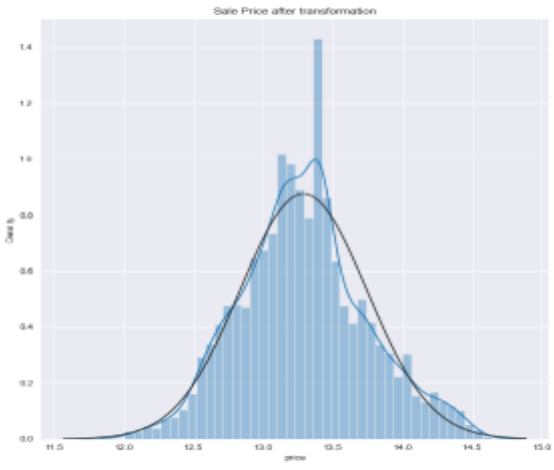
3. barplot for 'year' column



As per this dataset, the most number of cars were manufactured in year 2018 followed by 2019, 2017, and 2020. Least were from year 2009.

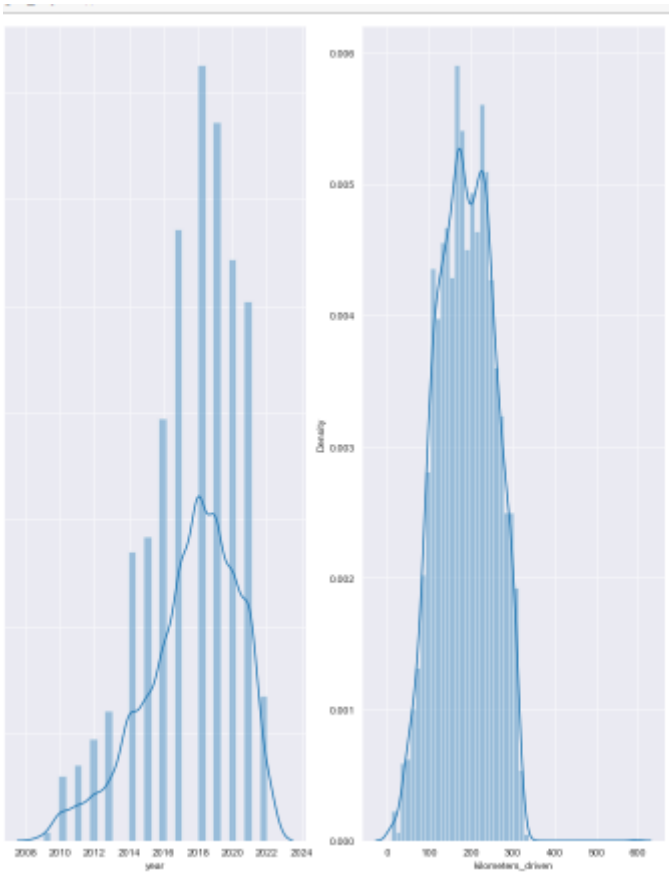
4. Distplot of target data

This is before the transformation.



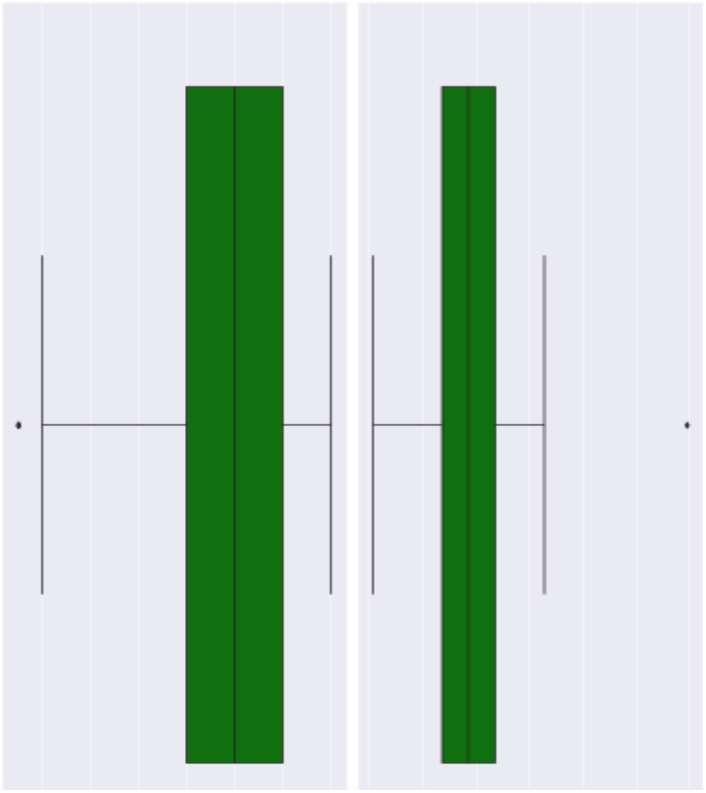
This is after transformation of target.

5. distribution of numerical features



6. Box plot to detect outlier

```
plt.tight_layout()
```



- ## Interpretation of the Results

  After Visualization, we have observed the skewness, outlier and multicollinearity of the data, and have tried to reduce it by square root transformation technique, log transformation technique and interquartile range and respectively.

  After doing 10 different modelling, we analyzed that the random forest regressor model is best among all.

CONCLUSION

- ## Key Findings and Conclusions of the Study

The increased prices of new cars and the financial incapability of customer to buy them, used cars sales are on global increase. Therefore, there is an urgent need for a used car price prediction model, which effectively determines the worthiness of car using variety of features. This model will help in predicting sale prices of used cars. In this model, we will first do the cleaning and exploring of the input data. We have performed Linear regression, Random Forest regressor, Decision Tree regressor, SVR, KNeighbours regressor, Lasso, Ridge, Adaboost regressor, XGB regressor and Gradient boosting regressor and hyperparameter tuning as we understand that parameterization can drive the significant result in the performance.

- ## Learning Outcomes of the Study in respect of Data Science

  i. From different visualization, we have learned about the skewness, outlier and multicollinearity of the dataset and it helped in making better model.

ii. Data cleansing- we learned, how to fill or drop the null values data for the dataset by performing multiple statistical operation.

iii. Machine learning algorithm- we have performed 10 multiple regression model to predict the sale price of house, out of all Gradient boosting regressor have performed best among the others. We have also the Hyper Para meter tuning to the improve of model accuracy.

- ## Limitations of this work and Scope for Future Work

  Limitations of this project were that we haven't cover all regressor algorithms in our Data Science course, instead of it, they have focused on the basic algorithm. In Hyper Para Meter tuning, parameter for different model is very difficult to choose.