NAME OF THE PROJECT

CAR PRICE PREDICTION PROJECT

Submitted by: **AKSHAY RUNTHALA**

ACKNOWLEDGMENT

I took the help of various sources (mentioned below) to complete my project:-

- **SCIKIT-LEARN DOCUMENTATION(https://scikit-learn.org/stable/)**
- **NOTES PREPARED BY MYSELF AND RESOURCES PROVIDED BY MY TRAINING INSTITUTE(DATA TRAINED)**
- **PANDAS DOCUMENTATION(https://pandas.pydata.org/docs/)**
- **MATPLOTLIB DOCUMENTATION(https://matplotlib.org/)**
- **SEABORN DOCUMENTATION(https://seaborn.pydata.org/)**

INTRODUCTION

- **Business Problem Framing**
  Describe the business problem and how this problem can be related to the real world.

MY ANSWER-
The business problem I came across while preparing this project was that if any business/individual chooses to do business in used car selling,then it is very important to find out the value of the cars as accurately as possible.The used cars change from one owner to another and the owners consider price to be the main factor while purchasing the cars.**If the value of car is fixed at very high rate/very low rate then it will affect the future dealings of that car between the owners and car seller business because fixing of price depends on the historical cost of the used car**.

A car price estimator is a human being and though he is supposed to make an unbiased estimate for the value of the used car or property,**it takes time for him to decide the best price for the used car given the availability of data.A car estimator's estimate varies from person to person.**
I can say the accuracy of estimate and time taken to estimate the price/value of a property are the main limitations faced in today's world nowadays.The values predicted by the car estimate in the end are an estimate and actual price of the used car varies from estimate.Many businesses/individuals rely on the estimates/decision of the car estimate/agents to buy/or sell a particular property.
Buying/selling a used car or doing car selling businesses is a huge financial matter.If the human's estimate regarding a used car is even slightly wrong or varies,then it may cause huge financial loss to the businesses dealing in car selling . The individual choosing to

purchase a used car may suffer huge financial losses if he purchased a used car at a higher cost than the reasonable price in the industry. **There is risk of financial loss in both selling and buying a used car if the human's estimate is incorrect.**
There is a need for a pattern to estimate the used cars' values as accurately as possible. Human beings' abilities are limited due to time factor as well as accuracy in estimation of values.
Machine learning tasks can easily overcome these limitations of human beings because they are exponentially faster than human estimators to predict the value of a used car based on the given data (used car features).Most importantly,the machine learning models can predict the values of used cars more accurately than humans.

Conceptual Background of the Domain Problem

Describe the domain related concepts that you think will be useful for better understanding of the project.

MY ANSWER-

- Historical financial records of the used car
- Quality of materials used in repairs/service of the used car.
- Data available regarding cost of materials used in the used cars' repair (based on the country or region).

Review of Literature

This is a comprehensive summary of the research done on the topic. The review should enumerate, describe, summarize, evaluate and clarify the research done.

MY ANSWER:--

I searched the websites for understanding the meaning of the terms and concepts used in the car selling industry.I needed to understand these concepts so that I can perform my machine learning tasks better.

● Motivation for the Problem Undertaken

Describe your objective behind making this project, this domain and what is the motivation behind.

MY ANSWER:--

My objective for making this project is to-

- Make a machine learning model that can most accurately predict the **PRICES** of every used car regarding which data is available to the company.
- To make this model as efficient as possible to predict the future used cars based on data available to the company.
- Find out the **important variables** affecting the price of a single used car.
- Develop my skills for data science further by undertaking this project. This is a great project which directly relates to the real world scenario. I was motivated by the various ideas that came when I first looked into this project.
- Test and upgrade my various python coding skills while working on my project.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

I calculated the skewness of the whole dataset to know whether the data is normal or not. I also used skewness to impute missing values in numerical columns of the dataset. If my data is not normal,It is better to impute missing numerical values with the median value and if it is normal,then it is advisable to impute missing values with mean.

I imputed my categorical columns' missing values with the most frequent value.

SKEWNESS CHECK AND OBSERVATIONS:-

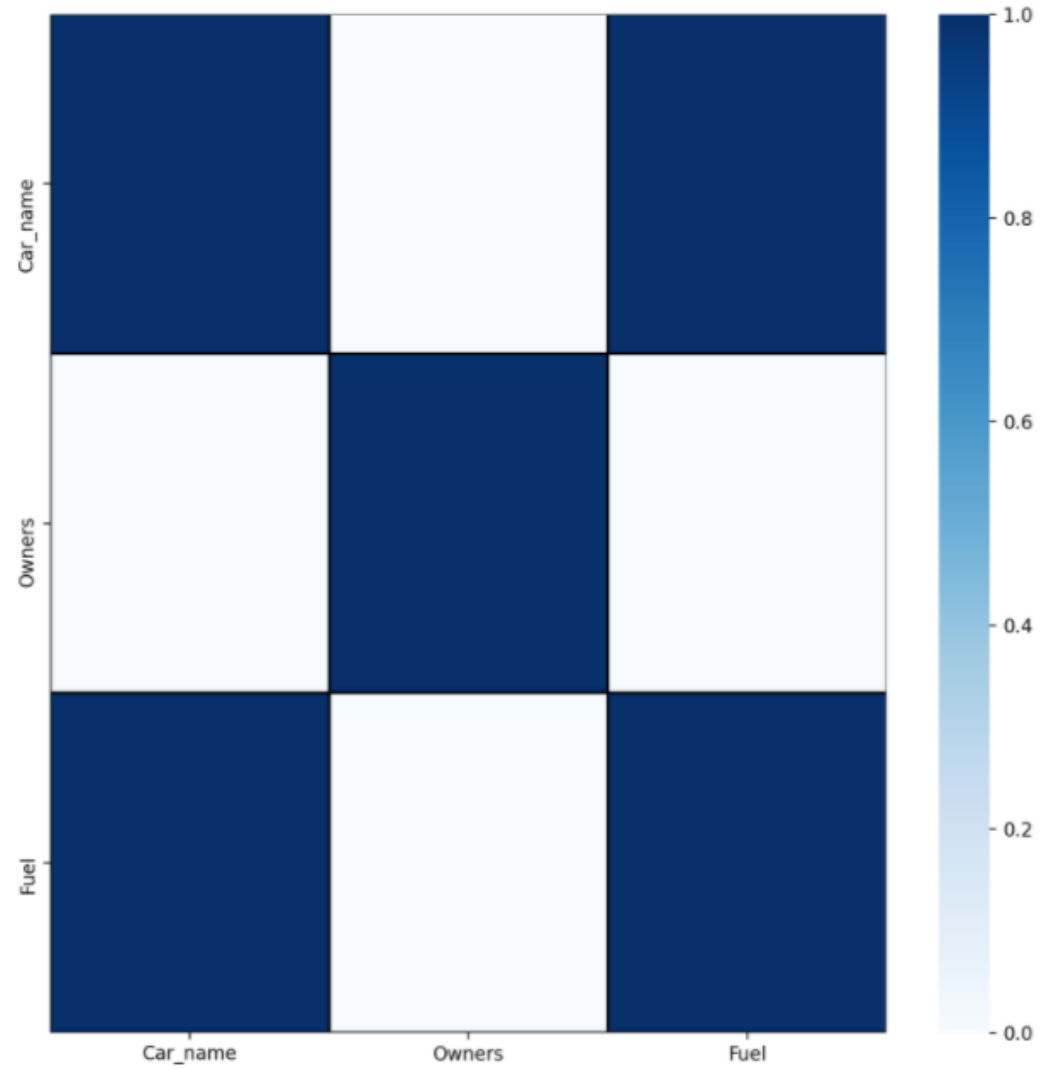
I Plotted Skewness For All Data Distributions Against Columns Of The Data. Plotted Green Horizontal Line For A Skewness Of Zero(Normal Distribution Has A Skewness Of 0). Plotted Red Lines Around The Green Line Denoting A Range Of (- 0.5,0.5).If Data Points Fall Within The Above Mentioned Range,Then They Are Approximately Normally Distributed . The rule for skewness seems to be: If the skewness is between -0.5 and 0.5, the data are fairly symmetrical. If the skewness is between -1 and -0.5 (negatively skewed) or between 0.5 and 1 (positively skewed), the data are moderately skewed. Skewness of the normal distribution is zero.Some of the dataset columns seem to have a skewness of more than 0.5 and even more than 1,so they all are highly skewed.They all have a right-skewed data or a positively-skewed data curve since the right tail is longer and mass of the distribution is concentrated on the left of the figure.**I did not consider skewness for the categorical columns as well as the target label - Price in my dataset.** Correlation coefficients are indicators of the strength of the linear relationship between two different variables, x and y. A linear correlation coefficient

that is greater than zero indicates a positive relationship. A value that is less than zero signifies a negative relationship. Finally, a value of zero indicates no relationship between the two variables x and y.

I calculated the correlation between my target label (PRICE) and other columns of the data to know which variables affect my Sale price the most.

I used chi square tests from scipy library to find relationships between the categorical variables of the data. Below is the heatmap for chi-square test results.

```
In [142]: plt.figure(figsize=(10,10),dpi=200)
          sns.heatmap(chi_square_matrix,linecolor='black',linewidths=1,cmap='Blues')
Out[142]: <AxesSubplot:>
```



As shown in the above heatmap figure,the **'blue'** colour shows the relationship status to be **'yes'or (1)**,whereas the **'white'** colour shows the relationship status to be **'no'or (0)** between the variables of the dataset.

- Data Sources and their formats

  What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data.

- ☐ I used web scraping tools for python to extract the data from the website --CarDekho: New Cars, Car Prices, Buy & Sell Used Cars in India
- ☐ I used the Selenium library to extract the data from the website.
- ☐ I found various important details regarding the car.They are mentioned below-
    - Owners
    - Engine
    - Kilometers driven by the car
    - Manufacturing year
    - Fuel used by used car

Data Preprocessing Done

> What were the steps followed for the cleaning of the data? What were the assumptions done and what were the next actions steps over that?

My Answer--

- There were many missing values in the dataset. I used the Simple imputer method from sklearn library to impute the missing values in the data.I calculated skewness for the data and based on skewness data,I imputed the missing values.
- I used the Label Encoding method from sklearn library to convert the ‘object’ data type columns in the data to float or integer.
- I used the Elliptic Envelope to detect the outliers and removed them(the removal was limited to 5% of the dataset).

- I used the **cube root** method from numpy library to adjust my skewness of the dataset.
- I used the Standard Scalar from sklearn.preprocessing to scale the data because scaling was necessary for better regression tasks.

Data Inputs- Logic- Output Relationships

Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

My Answer--

The relationship between data input and output is very important in machine learning model preparation. If the data we import is incorrect, then the output will also be incorrect. Every machine learning task is dependent on the data we import to prepare the machine learning model.

I used the correlation matrix to find the relationship between inputs and output. Correlation greater than 0.5 is considered a strong positive relationship.

There is strong positive relationship between below mentioned inputs/columns and the output (Price)--

- Engine

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

  My Answer--

  1. I chose a random state for all my regression models as well as a random state for splitting my data into training and test data to be **185.**
  2. I used an elliptic envelope method to detect the outliers in my dataset **because we can limit the amount of outliers to be removed from the dataset using this method.**

- Hardware and Software Requirements and Tools Used

  MY ANSWER-

  Libraries and packages used :-

  - Pandas-used for Exploratory data analysis

    I used-

    - ☐ .describe() method to understand the data
    - ☐ Pivot table method from pandas library to analyse my data better and derive meaningful insights for my exploratory data analysis.
    - ☐ .info() method to understand the data type of columns in the data.
  - Matplotlib- used for Exploratory data analysis
  - Seaborn- used for Exploratory data analysis

- ☐ Used bar plot,scatter plot to visualize my data for EXPLORATORY DATA ANALYSIS.
- ☐ Used **heatmap** to show **correlation/relationship** between the columns/variables of the data.
- ☐ Used boxplot to detect outliers in my data.

- Numpy- used for Exploratory data analysis
  - ☐ Used **log1p** method to reduce excess skewness of the data.
- Sklearn-used for machine learning
  - ☐ Imported various regression models like KNN,DECISION TREE,RANDOM FORESTS,LIGHT xgboost REGRESSOR,XGBOOST
  - ☐ Performed standard scaling using standard scaler from preprocessing.

- Scipy-used for statistics
  - ☐ For computing Chi-square tests.**CHI-SQUARE TEST IS USED TO DETERMINE THE RELATIONSHIP BETWEEN THE CATEGORICAL VARIABLES OF THE DATASET.**
- Light gbm regression –used for machine learning
- Xgboost regression -used for machine learning

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Describe the approaches you followed, both statistical and analytical, for solving this problem.

MY ANSWER-

- ☐ I calculated the skewness method using the pandas library and plotted the skewness data to determine the normality of the data because I found it more effective.
- ☐ I used the ***ELLIPTIC ENVELOPE*** method to detect my outliers upto a predefined limit(I chose around 5% of my data as outliers).

- Run and Evaluate selected models

Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

MY ANSWER-

Listing down all the algorithms used for the training and testing.

- ☐ RANDOM FOREST REGRESSOR
- ☐ ADABOOST-REGRESSOR
- ☐ XGBOOST REGRESSOR
- ☐ LIGHT GBM REGRESSOR
- ☐ ELASTIC NET REGRESSION
- ☐ K-NEAREST NEIGHBORS REGRESSOR
- ☐ DECISION TREE REGRESSOR
- ☐ SUPPORT VECTOR MACHINE REGRESSION

ElasticNet()

Root_mean_squared_error: 403130.92827303906

mean_absolute_error: 252202.49528002916

mean_squared_error: 162514545330.28217

r2: 0.518881767212354

cross validation scores below:-- ElasticNet()

root_mean_squared_error: -373620.1102201482

mean_absolute_error_cross_val_score: -243825.65894125952

mean_squared_error_cross_val_score: -141202504817.85315

DecisionTreeRegressor(random_state=185)

Root_mean_squared_error: 250894.4069773904

mean_absolute_error: 132826.0734720416

mean_squared_error: 62948003452.53641

r2: 0.813644790273725

cross validation scores below:-- DecisionTreeRegressor(random_state=185)

root_mean_squared_error: -263084.7586036847

mean_absolute_error_cross_val_score: -140380.13502439024

mean_squared_error_cross_val_score: -69542823059.67377

RandomForestRegressor(random_state=185)

Root_mean_squared_error: 221460.89845683082

mean_absolute_error: 118849.68259303983

mean_squared_error: 49044929545.30673

r2: 0.8548043205481882

cross validation scores below:--   RandomForestRegressor(random_state=185)

root_mean_squared_error: -218446.79162313495

mean_absolute_error_cross_val_score: -121204.17231962833

mean_squared_error_cross_val_score: -47942795184.647736

AdaBoostRegressor(random_state=185)

Root_mean_squared_error: 346526.593885571

mean_absolute_error: 262664.05942598113

mean_squared_error: 120080680269.93547

r2: 0.6445056375354203

cross validation scores below:--   AdaBoostRegressor(random_state=185)

root_mean_squared_error: -357646.4565794863

mean_absolute_error_cross_val_score: -273922.9871456755

mean_squared_error_cross_val_score: -128574763680.00943

GradientBoostingRegressor(random_state=185)

Root_mean_squared_error: 238112.78197642163

mean_absolute_error: 141677.76178988896

mean_squared_error: 56697696940.550896

r2: 0.8321485889172009

cross validation scores below:-- GradientBoostingRegressor(random_state=185)

root_mean_squared_error: -237159.5198236249

mean_absolute_error_cross_val_score: -142356.24905614095

mean_squared_error_cross_val_score: -56501802802.10709

SVR()

Root_mean_squared_error: 609193.4476028292

mean_absolute_error: 383113.25138604204

mean_squared_error: 371116656602.22107

r2: -0.09867698069515485

cross validation scores below:-- SVR()

root_mean_squared_error: -577865.7454527493

mean_absolute_error_cross_val_score: -372946.2388415694

mean_squared_error_cross_val_score: -341042421964.5959

LGBMRegressor(random_state=185)

Root_mean_squared_error: 229591.04540060167

mean_absolute_error: 132652.72612702145

mean_squared_error: 52712048128.141136

r2: 0.8439479531479019

cross validation scores below:-- LGBMRegressor(random_state=185)

root_mean_squared_error: -232209.10641140578

mean_absolute_error_cross_val_score: -135145.53933806362

mean_squared_error_cross_val_score: -54112996119.77082

KNeighborsRegressor()

Root_mean_squared_error: 300877.7695271833

mean_absolute_error: 176931.15838751628

mean_squared_error: 90527432195.65285

r2: 0.7319969230553529

cross validation scores below:-- KNeighborsRegressor()

root_mean_squared_error: -322739.16903865384

mean_absolute_error_cross_val_score: -197713.16979512194

mean_squared_error_cross_val_score: -104973989516.78403

XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,

colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,

importance_type='gain', interaction_constraints='',

learning_rate=0.300000012, max_delta_step=0, max_depth=6,

min_child_weight=1, missing=nan, monotone_constraints='()',

n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=185,

reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,

tree_method='exact', validate_parameters=1, verbosity=None)

Root_mean_squared_error: 210658.10715283063

mean_absolute_error: 118346.83395135729

mean_squared_error: 44376838109.21348

r2: 0.8686240306403521

cross validation scores below:-- XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,

colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,

importance_type='gain', interaction_constraints='',

learning_rate=0.300000012, max_delta_step=0, max_depth=6,

min_child_weight=1, missing=nan, monotone_constraints='()',

n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=185,

reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,

tree_method='exact', validate_parameters=1, verbosity=None)

root_mean_squared_error: -207191.23161848556

mean_absolute_error_cross_val_score: -118862.0075903201

mean_squared_error_cross_val_score: -43102950970.05803

I used various metrics to evaluate my model performance and based on them, I selected my models for hyper parameter tuning.

In [89]:
```
gr = GridSearchCV(rfr,param_grid=rfprr)
gl = GridSearchCV(lgr,param_grid=lgpr1)
gx = GridSearchCV(xgbr,param_grid=xgbrp)
```

I created grid search cv models for the best models chosen above.

In [90]:
```
print(gr)
gr.fit(sX_train,y_train)
y_pred=gr.predict(sX_test)
print('Mean_absolute_error: ',mean_absolute_error(y_test,y_pred))
print('Mean_squared_error: ',mean_squared_error(y_test,y_pred))
print('Root_mean_squared_error: ',np.sqrt(mean_squared_error(y_test,y_pred)))
print('R2_score: ',r2_score(y_test,y_pred))
print(gr.best_params_)
print('\n')
print(gr.best_estimator_)
```

```
GridSearchCV(estimator=RandomForestRegressor(random_state=185),
             param_grid={'max_depth': [None, 12, 10, 8],
                         'max_features': ['auto', 'log2', 'sqrt'],
                         'min_samples_leaf': [1, 2, 3],
                         'min_samples_split': [1, 2, 3],
                         'n_estimators': [100, 200, 300, 500, 700]})
Mean_absolute_error:  117983.26790271843
Mean_squared_error:  48908697184.83658
Root_mean_squared_error:  221153.10801532178
R2_score:  0.8552076313557513
{'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 500}


RandomForestRegressor(n_estimators=500, random_state=185)
```

In [91]:
```
print(gl)
gl.fit(sX_train,y_train)
y_pred=gl.predict(sX_test)
print('Mean_absolute_error: ',mean_absolute_error(y_test,y_pred))
print('Mean_squared_error: ',mean_squared_error(y_test,y_pred))
print('Root_mean_squared_error: ',np.sqrt(mean_squared_error(y_test,y_pred)))
print('R2_score: ',r2_score(y_test,y_pred))
print(gl.best_params_)
print('\n')
print(gl.best_estimator_)
```

```
GridSearchCV(estimator=LGBMRegressor(random_state=185),
             param_grid={'learning_rate': [0.1, 0.001, 0.2, 0.3, 0.5, 0.7, 0.9],
                         'max_depth': [6, 7, 8, 10, 12],
                         'n_estimators': [200, 300, 500, 700, 800],
                         'n_jobs': [1], 'reg_alpha': [0, 0.5, 1, 1.2],
                         'reg_lambda': [1.2, 0, 1, 0.5]})
Mean_absolute_error:  129209.8418583677
Mean_squared_error:  50280073464.59777
Root_mean_squared_error:  224232.18650451984
R2_score:  0.8511477231742116
{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 500, 'n_jobs': 1, 'reg_alpha': 1.2, 'reg_lambda': 0.5}


LGBMRegressor(max_depth=6, n_estimators=500, n_jobs=1, random_state=185,
              reg_alpha=1.2, reg_lambda=0.5)
```

In [92]:
```
print(gx)
gx.fit(sX_train,y_train)
y_pred=gx.predict(sX_test)
print('Mean_absolute_error: ',mean_absolute_error(y_test,y_pred))
print('Mean_squared_error: ',mean_squared_error(y_test,y_pred))
print('Root_mean_squared_error: ',np.sqrt(mean_squared_error(y_test,y_pred)))
print('R2_score: ',r2_score(y_test,y_pred))
print(gx.best_params_)
print('\n')
print(gx.best_estimator_)
```

```
GridSearchCV(estimator=XGBRegressor(base_score=0.5, booster='gbtree',
                                    colsample_bylevel=1, colsample_bynode=1,
                                    colsample_bytree=1, gamma=0, gpu_id=-1,
                                    importance_type='gain',
                                    interaction_constraints='',
                                    learning_rate=0.300000012, max_delta_step=0,
                                    max_depth=6, min_child_weight=1,
                                    missing=nan, monotone_constraints='()',
                                    n_estimators=100, n_jobs=8,
                                    num_parallel_tree=1, random_state=185,
                                    reg_alpha=0, reg_lambda=1,
                                    scale_pos_weight=1, subsample=1,
                                    tree_method='exact', validate_parameters=1,
                                    verbosity=None),
             param_grid={'learning_rate': [0.1, 0.001, 0.2, 0.3, 0.5, 1],
                         'max_depth': [6, 7, 8, 10, 12],
                         'n_estimators': [200, 300, 500, 700], 'n_jobs': [1],
                         'reg_alpha': [0, 0.5, 1, 1.2],
                         'reg_lambda': [1.2, 0, 1, 0.5]})
Mean_absolute_error:  113531.55275977324
Mean_squared_error:  41328577354.249565
Root_mean_squared_error:  203294.31215420063
R2_score:  0.8776482925888665
{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 700, 'n_jobs': 1, 'reg_alpha': 1, 'reg_lambda': 0.5}


XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.1, max_delta_step=0, max_depth=6,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=700, n_jobs=1, num_parallel_tree=1, random_state=185,
             reg_alpha=1, reg_lambda=0.5, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

I chose the **XGBOOST REGRESSOR, LIGHT GBM REGRESSOR and RANDOM FOREST REGRESSOR** as the models for hyperparameter tuning because they performed the best among all the models based on metrics and scores and also r2 score.

I created grid search cv models for the best models chosen above.

Xgboost regressor performed best among all grid search cv models.

I set randomized search cv for xgboost regressor to find best parameters **but the grid search cv found better parameters than randomized search cv**.

· Key Metrics for success in solving problem under consideration

What were the key metrics used along with justification for using it? You may also include statistical metrics used if any.

My answer-

The metrics used were--

- ☐ R2 score-- the closer it is to 1,the better the model performance is.
- ☐ Mean absolute error
- ☐ Mean squared error
- ☐ Root mean squared error(RMSE)

All the errors should be as minimal as possible.the lesser they are,the better the model performance is.

· Visualizations

Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.

If different platforms were used, mention that as well.

My answer-

1. Histogram

   I plotted it to know about the skewness and distribution of the numerical data columns.

In statistics, a positively skewed (or right-skewed) distribution is a type of distribution in which most values are clustered around the left tail of the distribution while the right tail of the distribution is longer.

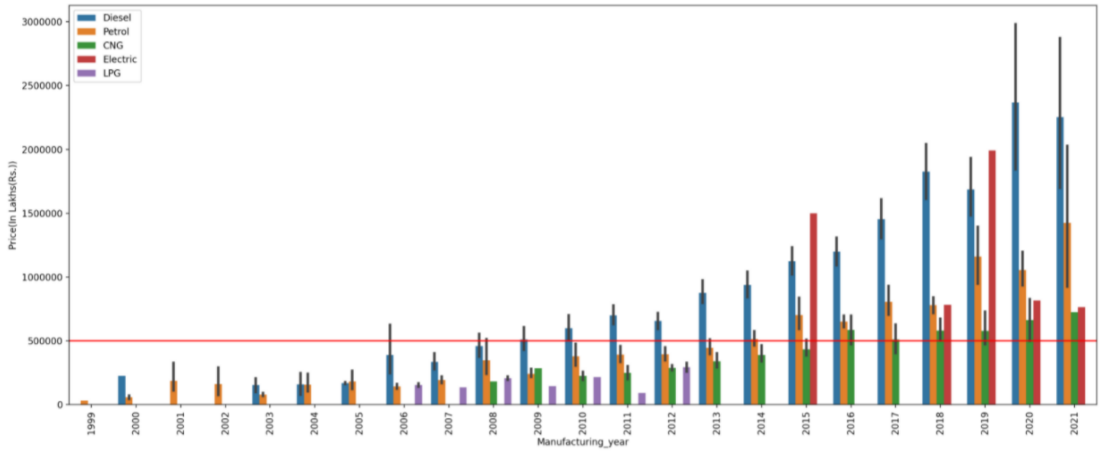
In statistics, a negatively skewed (also known as left-skewed) distribution is a type of distribution in which more values are concentrated on the right side (tail) of the distribution graph while the left tail of the distribution graph is longer.

While plotting histograms with kde for the numerical columns,I observed that-

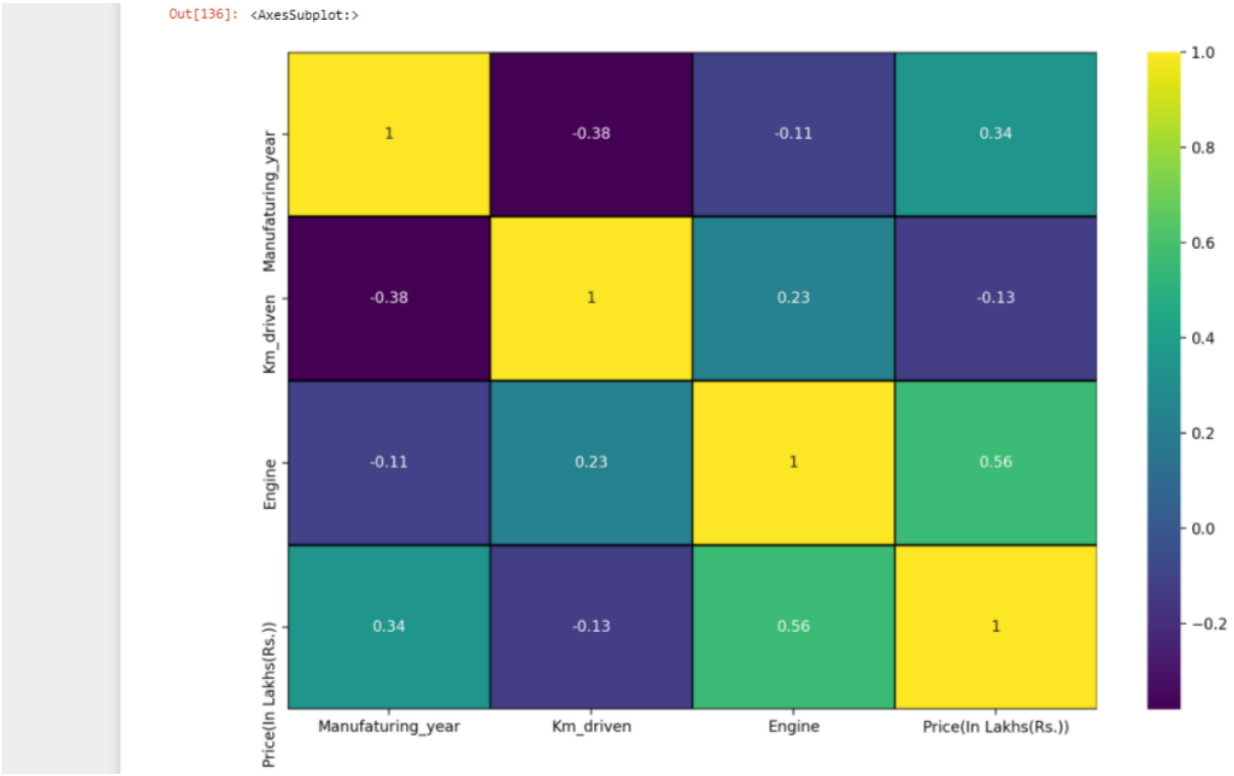
1. Manufacturing Year -> this column distribution is left-skewed.
2. Km_driven -> this column distribution is right-skewed.
3. Engine -> this column distribution is right-skewed.
4. Price(in Lakhs(Rs.)) -> the column distribution is right-skewed.

2. Bar plot

   To show comparison of different data based on categorical columns.

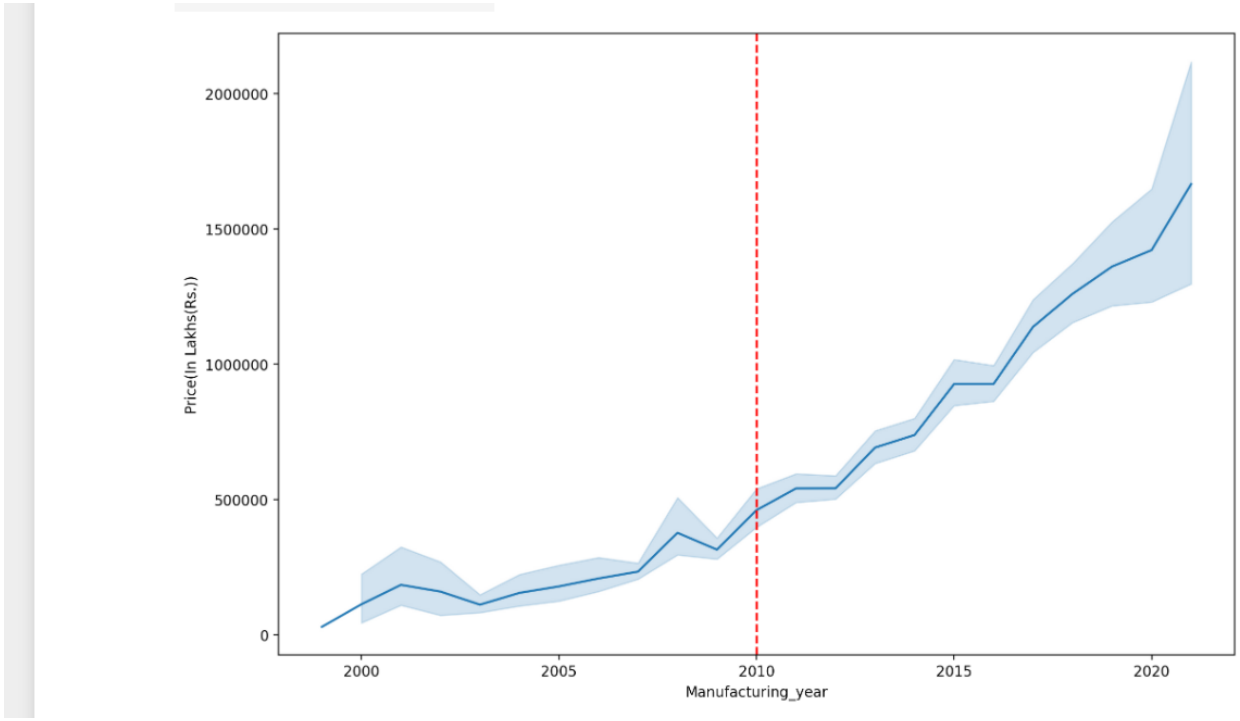


3. Heatmap
   - ☐ To show correlation between the variables and target label of the data.



- ☐ To show the chi square test results.

4. Line Plot -To show comparison of different data based on categorical columns year by year.



- Interpretation of the Results

Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

I found that while calculating skewness of the dataset,***many*** of the columns /variables were **not normally distributed**. When comparing all the models over r2 score,mean absolute error,mean squared error and root mean squared error---light gbm regression and xgboost regressor were consistent. After doing grid search cv, I found that xgboost regressor performed the best among all models.I used randomized search cv to find better scores for my xgboost regressor,but **grid search cv**

found better parameters for xgboost regressor. I saved that model with the help of the joblib library.

CONCLUSION

- Key Findings and Conclusions of the Study

  Describe the key findings, inferences, observations from the whole problem.

  I found that most of the data were highly skewed and there were some missing values in the dataset.

- Learning Outcomes of the Study in respect of Data Science

  List down your learnings obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you faced while working on this project and how you overcame that.

  My answer-

  - I learned to improve on my web scraping skills in python with selenium.
  - I learned how to organize my data for the machine learning process.
  - I applied one of my favourite methods for data analysis which I often see in Excel--**pivot tables.**

- I found out that skewness can be a great technique to know a lot about the data like--
    - ☐ whether it is normal or not,
    - ☐ how it affects outliers.
    - ☐ How it can be used to handle missing values.
- I found the seaborn heatmap plotting correlation to be better than scatterplots at finding out the relationships between the variables of the dataset.