CAR PRICE PREDICTION

REGRESSION MACHINE LEARNING MODEL

PREPARED BY :- RAJ KUMAR

ABSTRACT

Machine Learning is used across many ranges around the world. The automobile and consultancy business is no exception.

Cars of a particular make, model, year, and set of features start out with a price set by the manufacturer. As they age and are resold as used, they are subject to supply-and-demand pricing for their particular set of features, in addition to their unique history. The more this sets them apart from comparable cars, the harder they become to evaluate with traditional methods. Using Machine Learning algorithms to better utilize data on all the less common features of a car can more accurately assess the value of a vehicle.

This study compares the performance of Linear Regression, Ridge Regression, Lasso Regression,Support vector regression, Decision Tree Regression, KNeighborsRegressor, CatBoostRegressor and Random Forest Regression ML algorithms in predicting the price of used cars. The study has been conducted with a large public dataset of used cars. The results show that CatBoostRegressor and Random Forest Regression demonstrate the highest price prediction performance across all metrics used.

Table of content

INTRODUCTION

New cars of a particular make, model, and year all have the same retail price, excluding optional features. This price is set by the manufacturer. Used cars, however, are subject to supply-and-demand pricing. Further, used cars have additional attributes that factor into the price. These include the condition, mileage, and year of manufacturing etc., which sets cars that may have shared a retail price apart.

The used car market is generally divided into two categories, retail and wholesale. The retail price is the higher of the two prices and is what an individual should expect when buying a car at a dealership. The wholesale price is the lower price which dealers will pay. Whether the dealer has sourced the car from a trade-in, auction, or another dealer, this price is considerably lower to ensure that the dealer will make a profit on the vehicle. Prices for peer-to-peer car sales generally lie in-between the retail and wholesale price points. Because there is no “middle-man” in peer-to-peer transactions, there is only a single price point, rather than two. A difficulty in peer-to-peer transactions is for both parties to agree on a fair price. There are many tools which provide an approximation, but do not factor in the particularities of the car into the price. Car markets are to some extent local and therefore location also affects the price. There is therefore a need for a valuation method which can make use of more of the features particular to each car, and extract information from all other previous sales of cars with shared features.

METHODOLOGY

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts: training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data.

This system is implemented using the following modules.

1.)Collection of Dataset

2.)Data Preprocessing

3.)Model Building.

DATA COLLECTION

Initially, we collected a dataset for our car price prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training data is also split into training and validation data. The training dataset is used for prediction model learning while validation and testing data is used for evaluating the prediction model. For this project, 64% of the total data is used for training, 16% used for validation and 20% of data is used for testing.

The dataset used for this project is taken from Kaggle. The dataset consists of 11 attributes.

PRE- PROCESSING OF DATA

Data preprocessing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes.

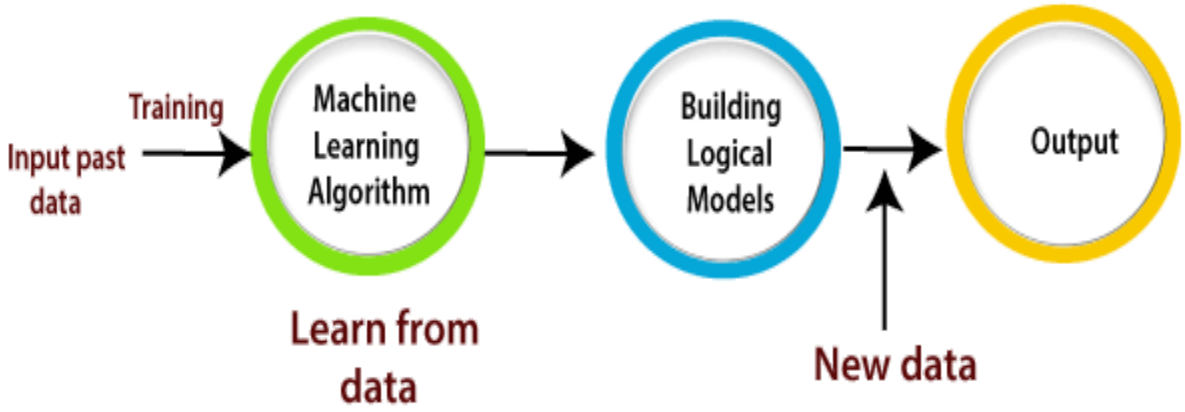
In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

Figure: Data Pre-processing

MODEL BUILDING

Various machine learning algorithms like linear regression,SVM, Decision Tree, Random Tree and catboost are used for Regression. Comparative analysis is performed among algorithms and the algorithm that gives the good outcome is used for Car price prediction.

Performance metrics are R2 score and Root Mean squared error to analyze models performance.

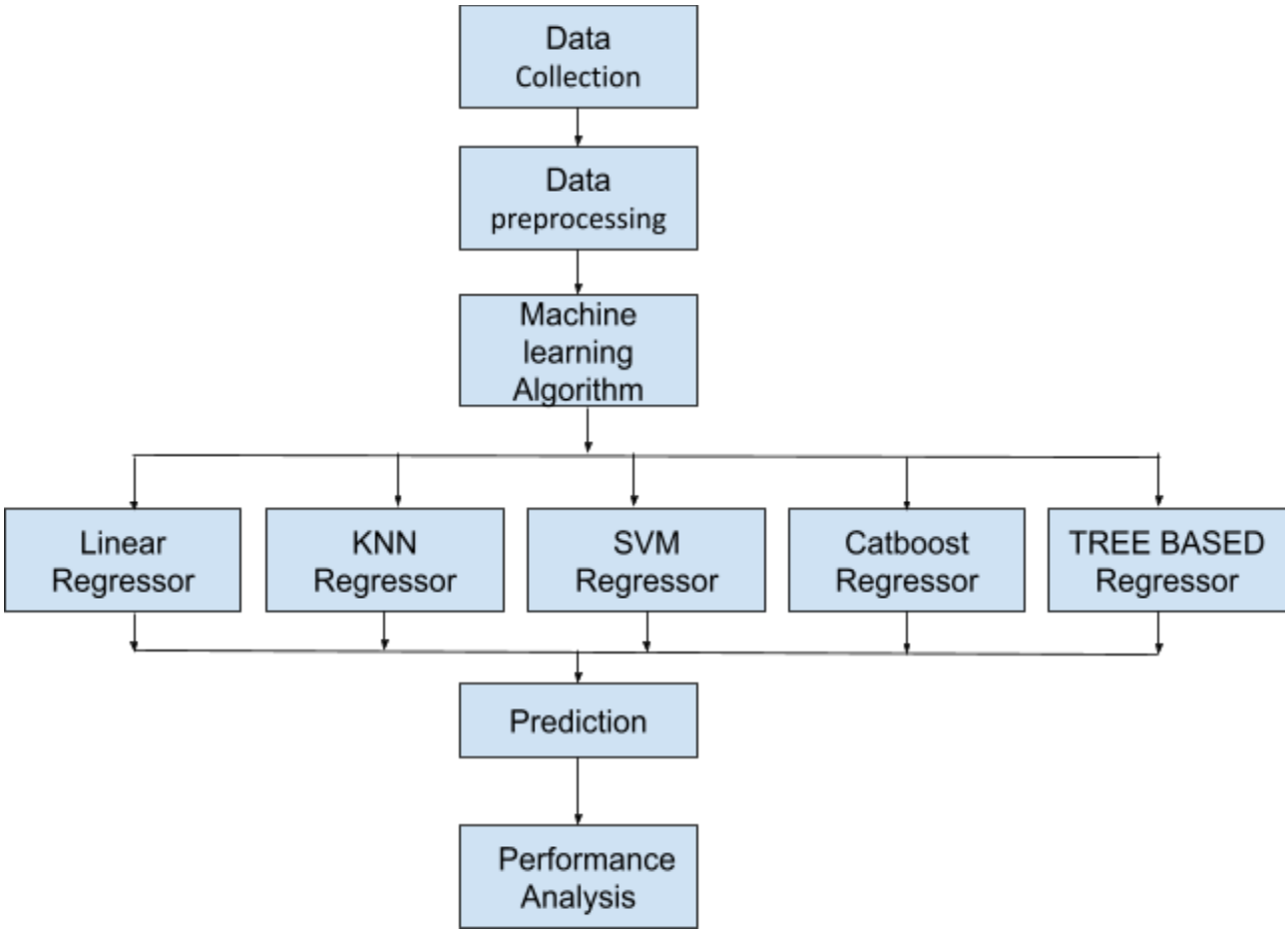


WORKING OF SYSTEM

SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system.

The working of this system is described as follows: -
Dataset collection is collecting data which contains wine details. Data is further cleaned, made into the desired form. Different regression techniques as stated will be applied on preprocessed data to predict the performance of models. r-squared shows how well the data fit the regression model , Other performance metrics are also considered for models.



DATASET DESCRIPTION

Dataset contains following features:-

- Brand:- Brand of car
- model :- Model name of car
- year :- Year of manufacturing
- Transmission:- Transmission type
- Mileage:- Mileage of car

- fuelType:- Type of Fuel used
- Tax
- Mpg:- Car moves miles per gallon
- engineSize:- Size of engine

  Price:- Price of car

```
RangeIndex: 7632 entries, 0 to 7631
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   brand         7632 non-null   object
 1   model         7632 non-null   object
 2   year          7632 non-null   int64
 3   transmission  7632 non-null   object
 4   mileage       7632 non-null   int64
 5   fuelType      7632 non-null   object
 6   tax           7632 non-null   int64
 7   mpg           7632 non-null   float64
 8   engineSize    7632 non-null   float64
 9   price         7632 non-null   int64
dtypes: float64(2), int64(4), object(4)
```

From the dataset we can see we have 7632 rows and 10 features as we remove the car id feature from the dataset. Now the dataset has 4 object type features, 4 integer type dataset and 2 float type features.

LIBRARIES USED

Python libraries make it easy for us to handle the data and perform typical and complex tasks with a single line of code.

Libraries used in this case study are as follow:-

- **Pandas**– Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

- **NumPy**–NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.

- **Matplotlib**– Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

- **Seaborn**– Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

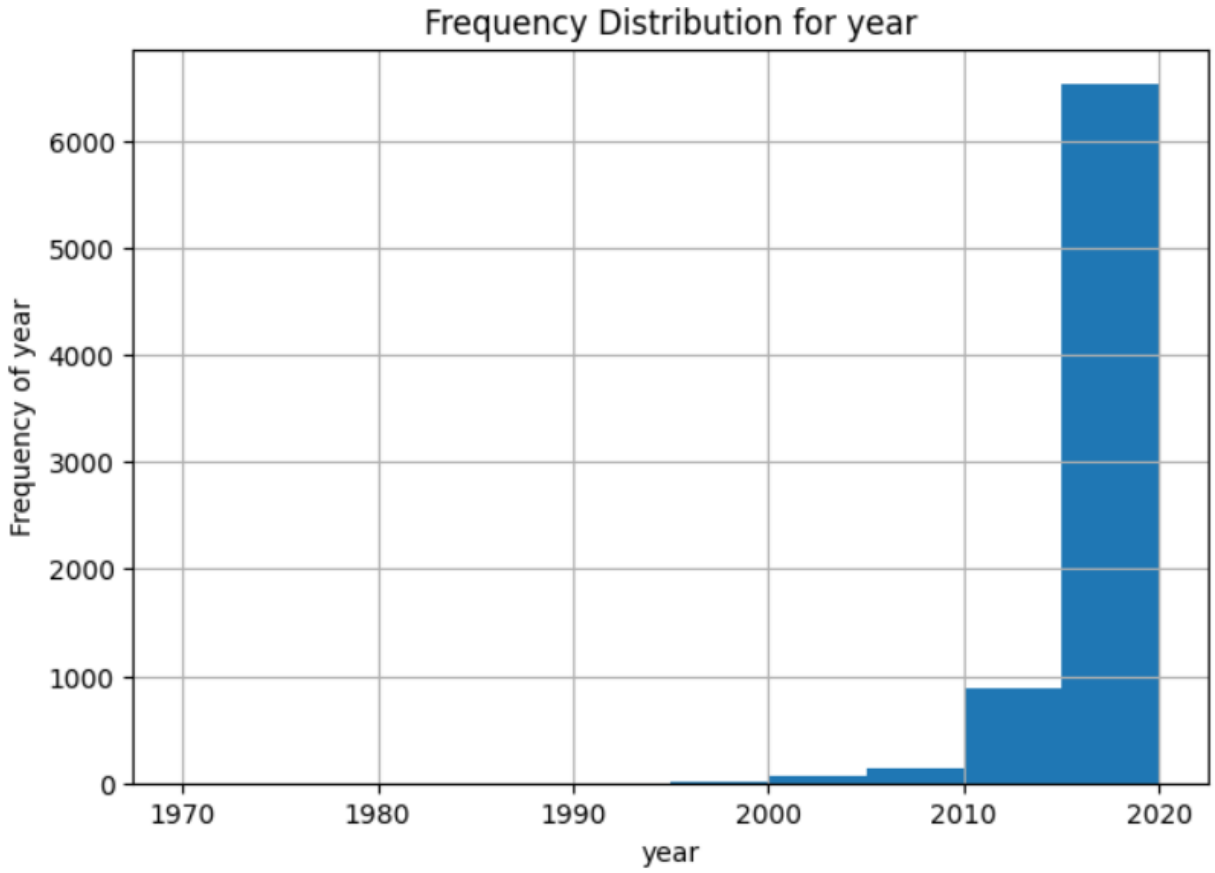
- **Sklearn**– This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.

EXPLORATORY DATA ANALYSIS

Univariate Analysis

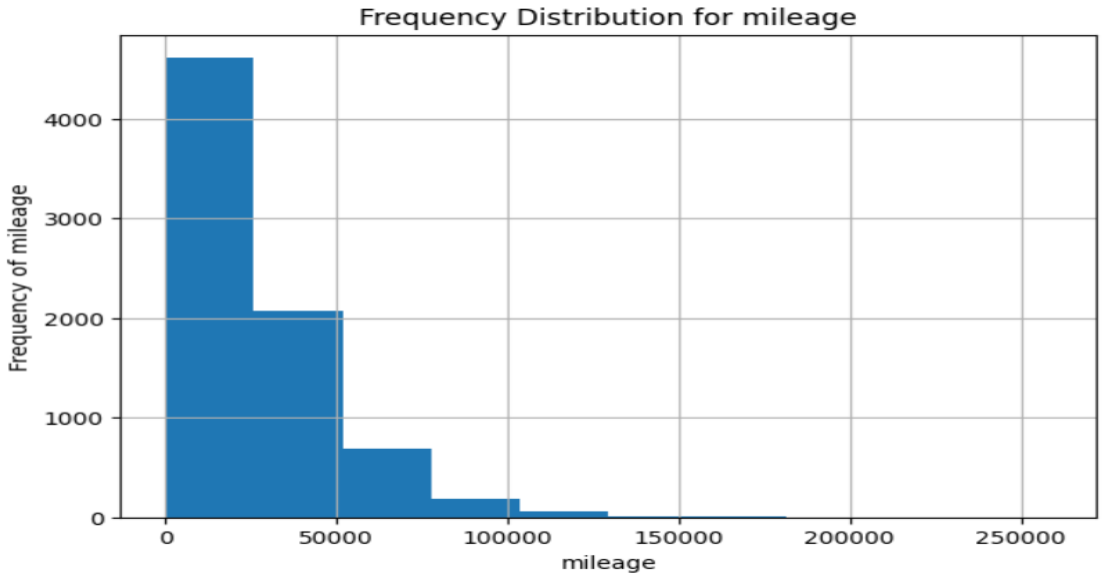
YEAR

From the histogram we can see most of the cars are manufactured between 2015 to 2020. While there are some outliers in data.

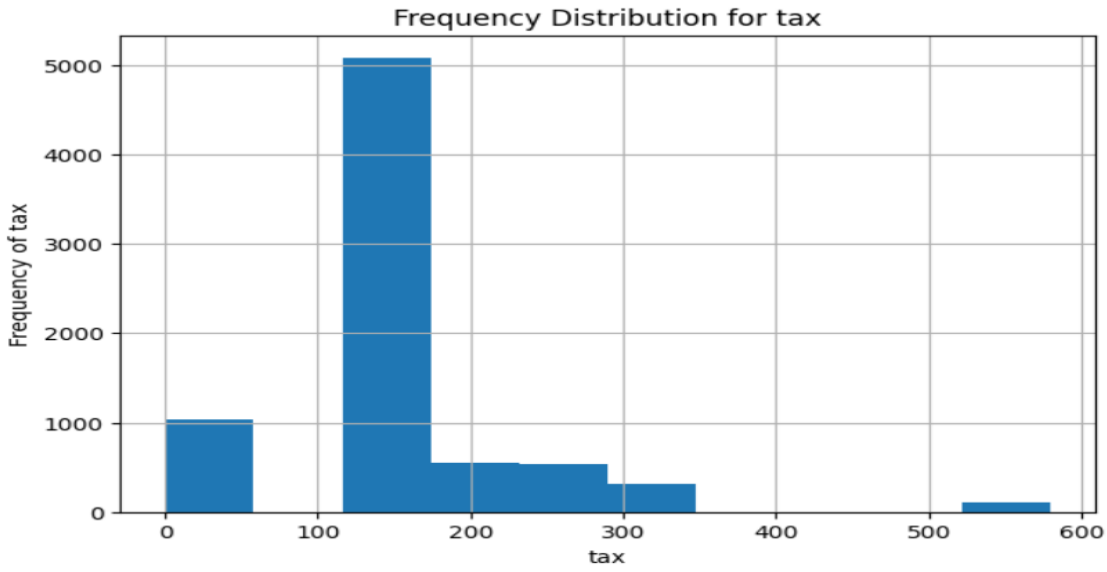


MILEAGE

Mileage lies up to 50000 for Most of the cars with outliers up to 250000.

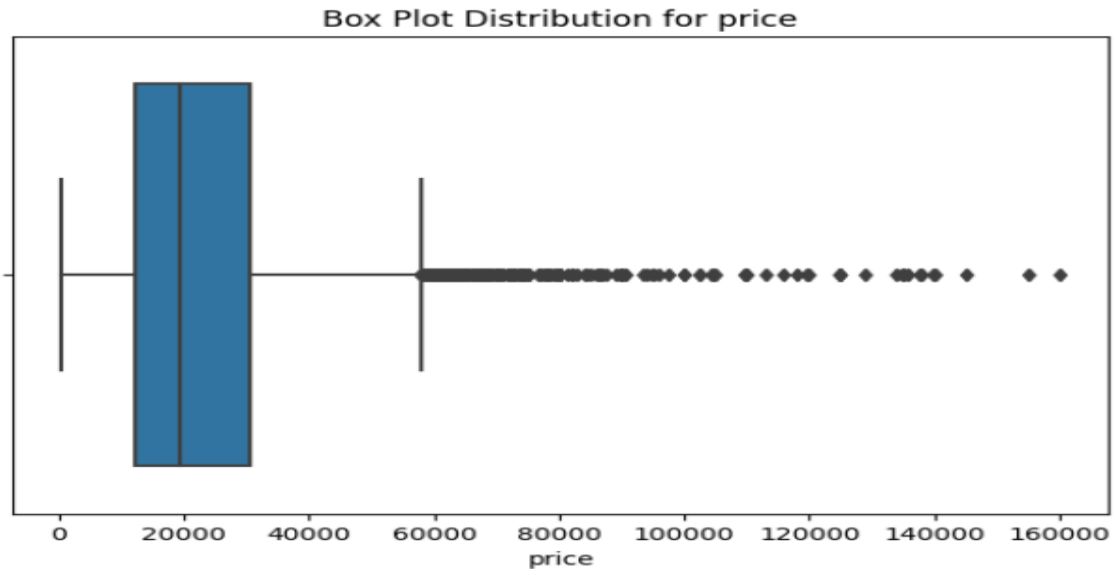


TAX



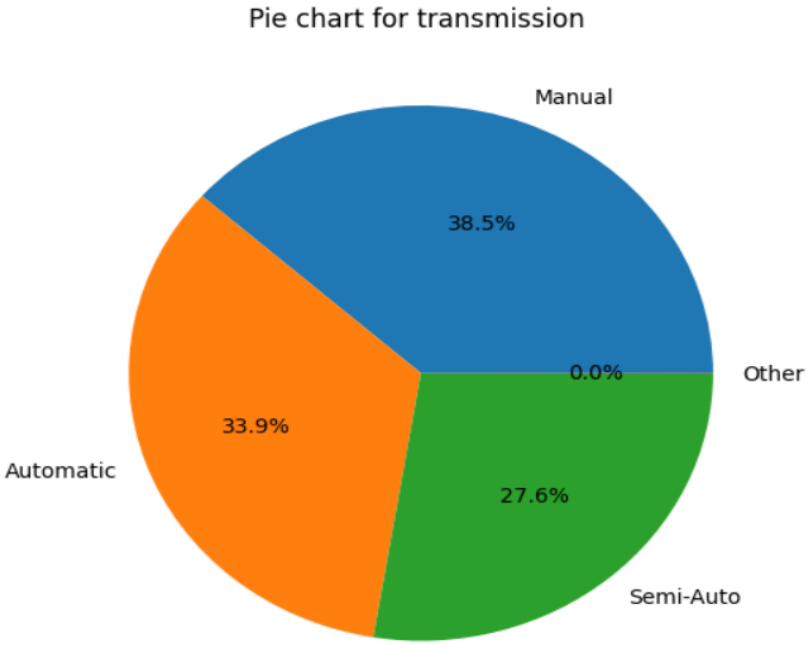
PRICE

Mostly cars range between 13k to 30k dollars. While there are many outliers, it may depend on other features.



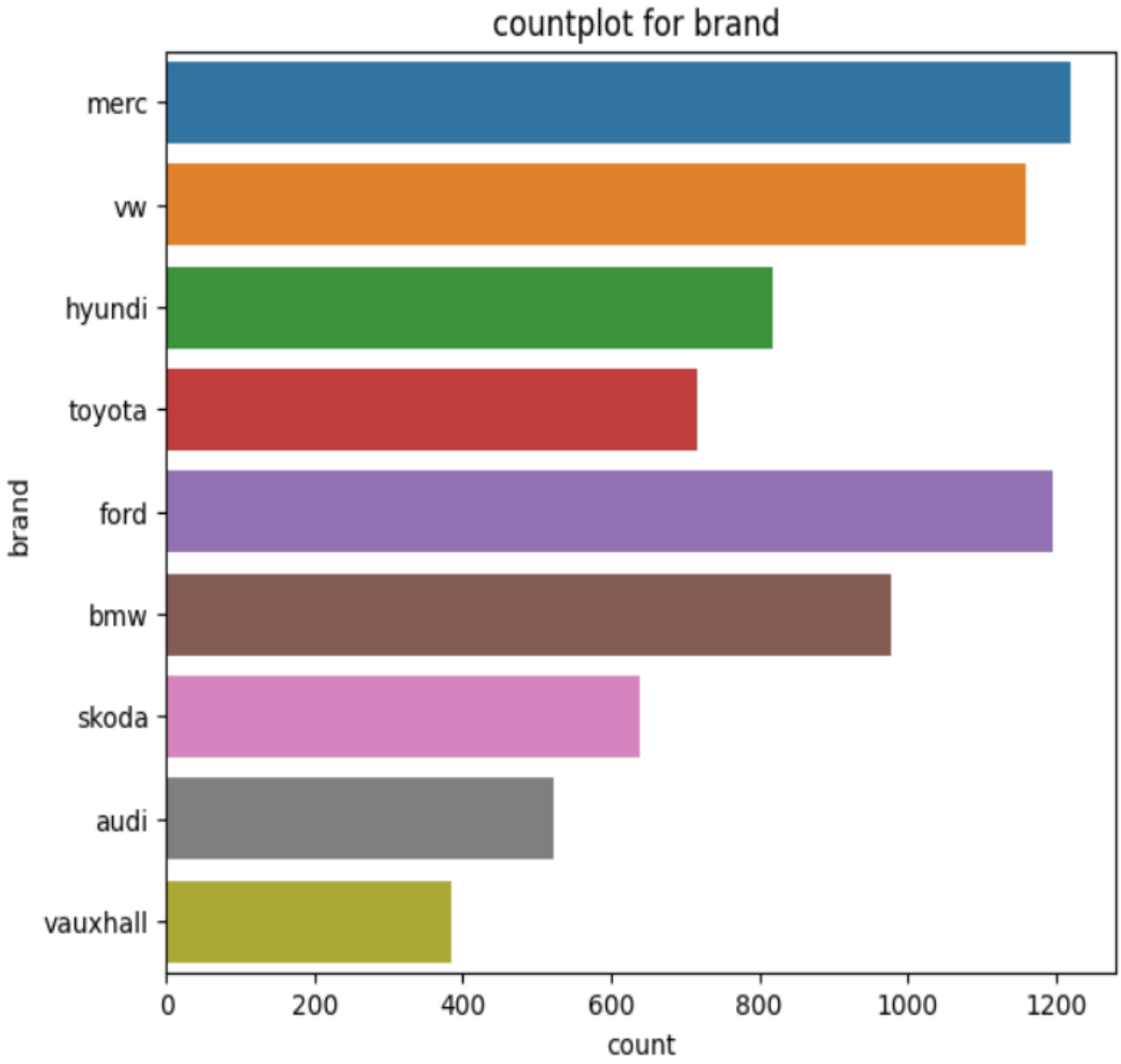
TRANSMISSION

Pie chart showing the distribution of transmission type of cars.



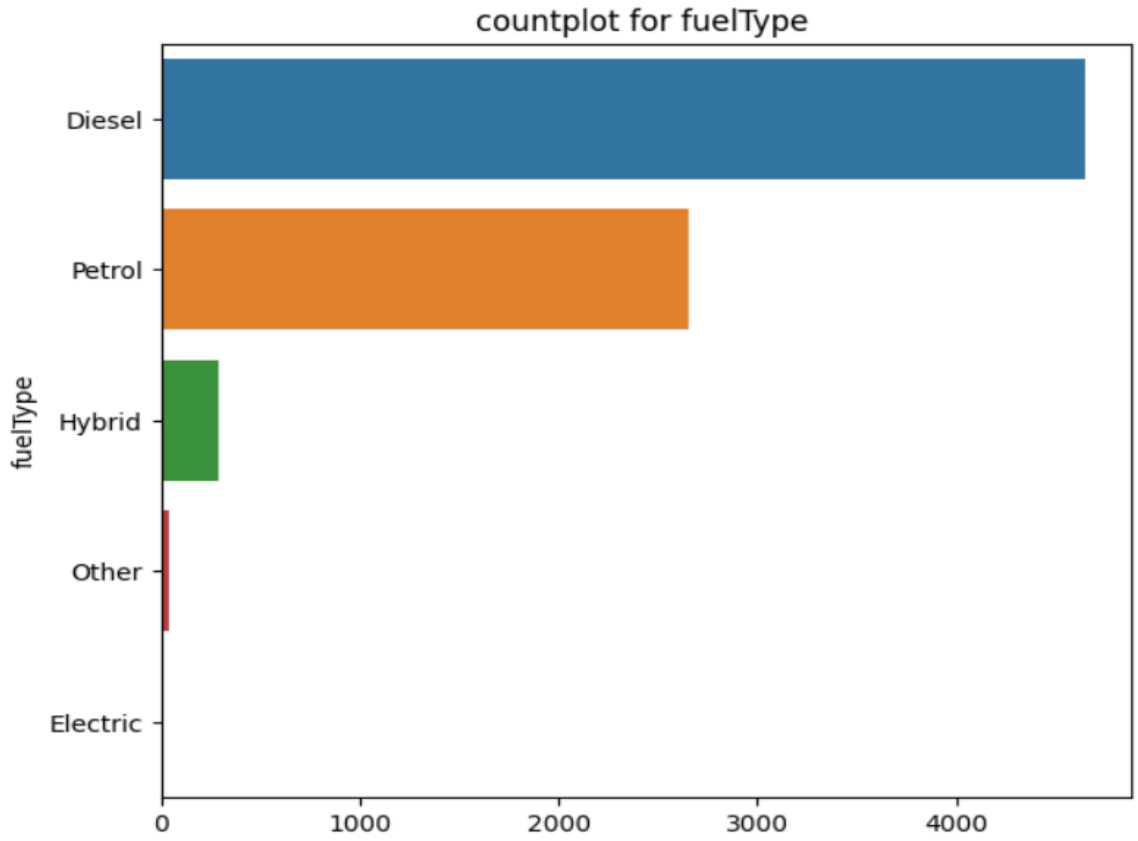
BRAND

Countplot for brand distribution showing no. of cars of each brand in dataset.



FUEL TYPE

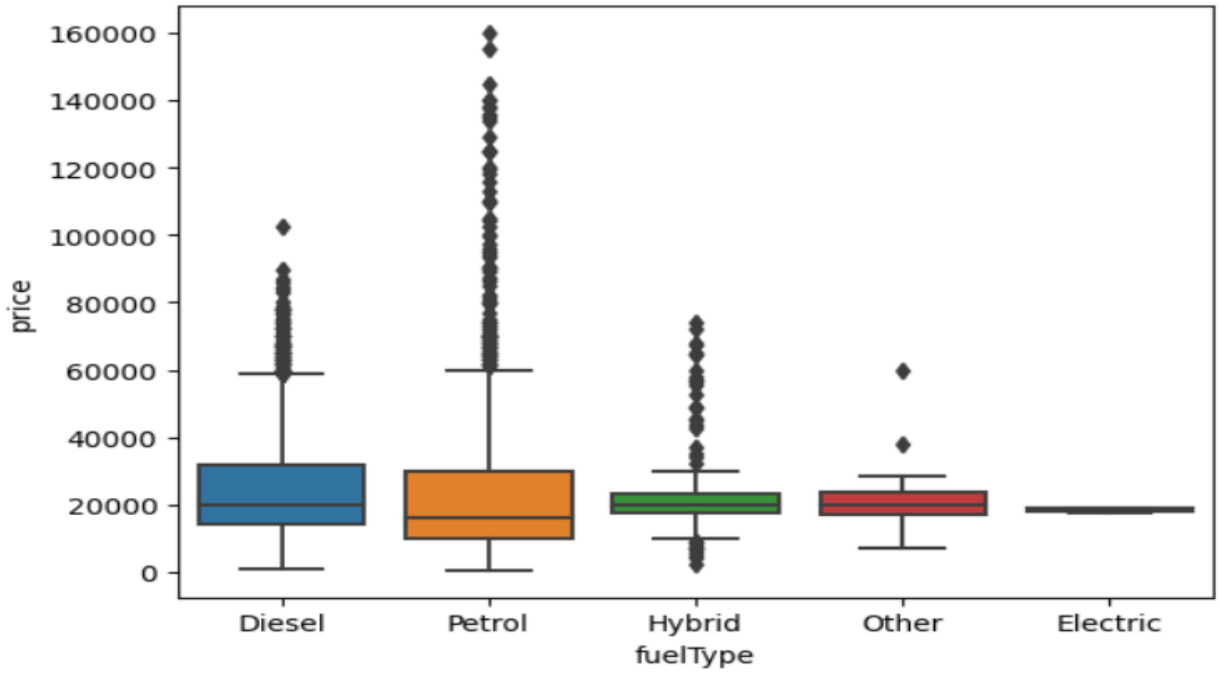
The Countplot of Fuel Type is showing that there are mostly diesel cars and negligible Electric cars.



PRICE VS FUEL TYPE

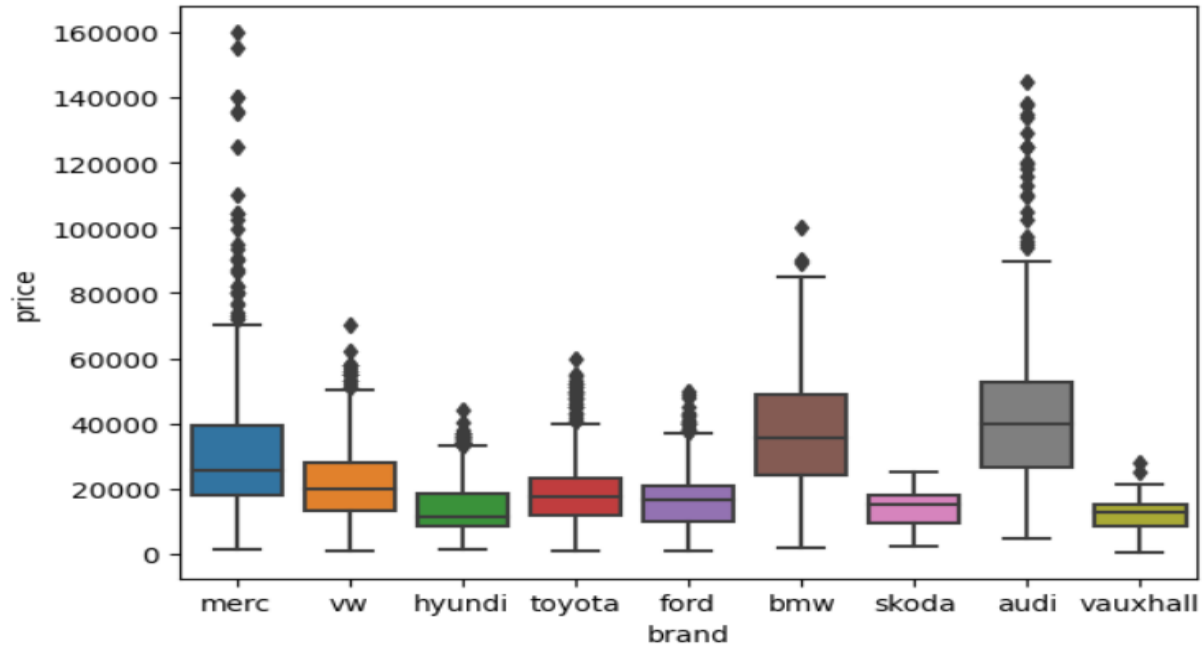
Now we will compare how the fuel Type is affecting our target Price .

From the box plot distribution we can see electric cars have less price while diesel and patrol have more variations in price.



PRICE VS BRAND

From the box plot distribution we can see Audi is most expensive with some outliers and vauxhall is the most affordable brand.



MODEL BUILDING AND EVALUATION

We will use this car dataset for training and testing in the ratio of 80-20. The training data is also split into training and validation data. The training dataset is used for model learning while validation and testing data is used for evaluating the prediction model. Finally, 64% of the total data is used for training, 16% used for validation and 20% of data is used for testing.

We will evaluate model performance with different algorithms.

EVALUATION

To evaluate our model we will use R- squared score and Root Mean squared error as performance metrics.

LINEAR REGRESSION MODEL

R^2 SCORE

- R^2 score for training data is : 0.669922
- R^2 score for validation data is : 0.675631
- R^2 score for test data is : 0.677090

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 9387.767710
- R-Mean squared error for validation data is : 9204.617791
- R-Mean squared error for test data is : 9734.476914

RIDGE

R^2 SCORE

- R^2 score for training data is : 0.669922
- R^2 score for validation data is : 0.675649
- R^2 score for test data is : 0.677083

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 9387.768176
- R-Mean squared error for validation data is : 9204.370450
- R-Mean squared error for test data is : 9734.577687

LASSO

R^2 SCORE

- R^2 score for training data is : 0.669922
- R^2 score for validation data is : 0.675648
- R^2 score for test data is : 0.677079

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 9387.768217
- R-Mean squared error for validation data is : 9204.378970
- R-Mean squared error for test data is : 9734.651960

SUPPORT VECTOR REGRESSOR

R^2 SCORE

- R^2 score for training data is : 0.287931
- R^2 score for validation data is : 0.304732
- R^2 score for test data is : 0.279274

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 13788.440854
- R-Mean squared error for validation data is : 13476.047956
- R-Mean squared error for test data is : 14543.103809

KNN REGRESSOR

R^2 SCORE

- R^2 score for training data is : 0.944935
- R^2 score for validation data is : 0.915256
- R^2 score for test data is : 0.908207

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 3834.362463
- R-Mean squared error for validation data is : 4704.808805
- R-Mean squared error for test data is : 5190.125673

DECISION TREE REGRESSOR

R^2 SCORE

- R^2 score for training data is : 0.999864
- R^2 score for validation data is : 0.924058
- R^2 score for test data is : 0.921804

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 190.849070
- R-Mean squared error for validation data is : 4453.751879
- R-Mean squared error for test data is : 4790.328270

RANDOM FOREST REGRESSOR

R^2 SCORE

- R^2 score for training data is : 0.993703
- R^2 score for validation data is : 0.950772
- R^2 score for test data is : 0.948878

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 1296.617206
- R-Mean squared error for validation data is : 3585.834988
- R-Mean squared error for test data is : 3873.248070

CATBOOST REGRESSOR

R^2 SCORE

- R^2 score for training data is : 0.985439
- R^2 score for validation data is : 0.962919
- R^2 score for test data is : 0.965815

R-MEAN SQUARED ERROR

- R-Mean squared error for training data is : 1971.711032
- R-Mean squared error for validation data is : 3112.174318
- R-Mean squared error for test data is : 3167.285383

CONCLUSION

We started with the data preprocessing where we checked for missing data,removing the unnecessary feature. Afterward, we perform EDA on the dataset to explore all the features . Then after EDA started training on different machine learning models, and for each model, after that we conclude with the resultant R^2 score and RMSE.
Based on prediction we conclude -

Based on the R-squared (R2) scores and root mean squared error (RMSE) for the various machine learning regression models, we can draw the following conclusions to identify the best model:

R-squared (R2) Scores:

- The "CatBoost Regressor" achieved the highest R2 score on the test dataset, indicating that it provides the best fit to the data, with an R2 score of 0.965815.

- The Random Forest regressor had a relatively high R2 score of 0.948878 on the test dataset.
- The "Decision Tree Regressor" also had a high R2 score on the test dataset, with a value of 0.921804.
- The R2 scores of "Lasso Regression" and "Ridge Regression" are lower than those of the Decision Tree models.

Root Mean Squared Error (RMSE):

- The "CatBoost Regressor" had the lowest RMSE on the test dataset, suggesting it provides the most accurate predictions, with an RMSE of 3167.29.
- The Random Forest Regressor and Decision Tree Regressor had relatively low RMSE values on the test dataset as well, with RMSE values of 3873.25 and 4790.33, respectively.

Considering both R2 scores and RMSE values, the "CatBoost Regressor" consistently outperforms other models, making it the best model among the ones evaluated. It provides the highest goodness-of-fit (R2) and the lowest prediction errors (RMSE) on the test dataset.

FUTURE SCOPE OF IMPROVEMENTS

As we can see the prediction on models , Some models are performing very well with training data but not able to retain the same performance with test/unseen data.While training, loss also occurs in some models. Hence, The model is overfitted.

To improve the performance in future we can do some more preprocessing and select only relevant features for model training and can improve by tuning, we can compare more models as well to get better results.

REFERENCES

https://scikit-learn.org/stable/modules/linear_model.html

https://catboost.ai/en/docs/concepts/python-reference_catboostregressor

https://medium.com/analytics-vidhya/evaluation-metrics-for-regression-algorithms-along-with-their-implementation-in-python-9ec502729dad

https://matplotlib.org/stable/plot_types/index.html