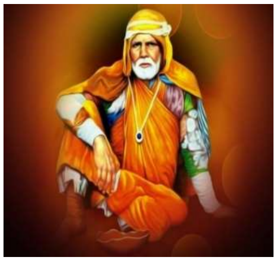


RAYAT SHIKSHAN SANSTHA'S

SADGURU GADGE MAHARAJ COLLEGE , KARAD

(An Autonomous College)

A Project Report On

"Predictive Analysis And Visualisation Of Indian Automobile Market"

Department of Statistics

By

Mr.Maskar Jaydip Vitthal

M.Sc-II (2024-25)

Under The Guidence of

Miss. S. S. Jagtap

CERTIFICATE

This is to certify that the project report entitled "**Predictive Analysis and Visualisation Of Indian Automobile Market**"; being submitted by Mr. Maskar Jaydip Vitthal as partial fulfilment for the award of degree of masters in Statistics at Sadguru Gadge Maharaj College, Karad is a record of Bonafide work carried out by her under my supervision and guidance.

To the best of our knowledge and belief, the matter presented in this project report is original and has not been submitted elsewhere for any other purpose.

Place: Karad
Date:

Sr. No	Seat No.	Roll No.	Name of the student	Signature
1.		21	Maskar Jaydip Vitthal	

Teacher in-charge   Examiner   PG Co-Ordinator   Head
Department of Statistics   Department of Statistics

ACKNOWLEDGEMENT

This project entitled Visualisation and Predictive Analysis Of Indian Automobile Market. We have great pleasure in presenting this report of successful completion of our project.

I take this opportunity to express our great sence of gratitude to our guide Miss. S. S. Jagtap of Statistics Department, S.G.M College, Karad for granting us permission to undertake this project report for their constant encouragement, guidence and inspiration without which we could not have completed this task .

I would like to extend our sincere thanks to Hod. Mrs. S.V. Mahajan, Mrs. S.P. Patil, Mrs. A.S. Patil and all other faculty members for their guidance and constructive suggestions throughout our project.

INDEX

Sr.No	Title	Page No.
1	Abstract	05
2	Introduction	06
3	Literature Review	07
4	Objectives	08
5	Data Description	09
6	Methodology	11
7	Statistical Tools	13
8	Data Preprocessing	14
10	Exploratory Data Analysis	15
11	Normality of the Data	19
12	Correlation Heatmap	20
13	Feature Selection	21
14	Fitting of Models	24
16	Testing of Hypothesis	29
17	Major Finding	32
18	Scope & Limitations	33
19	References	35
20	Appendix	36

1.ABSTRACT

The automobile industry is currently one of the most profitable sectors, driven by rising incomes and easy financing options for both rural and urban consumers. With many new companies entering the market and competition increasing among global manufacturers, this analysis of automobile data can help existing and new car makers in India understand customer expectations and analyse various vehicle models available today.

This project uses visualizations such as bar plots, histograms, scatter plots, boxplots, and violin plots to examine consumer automobile data. The goal is to gain insights into consumer buying behaviour and pricing trends, which can help predict future car prices based on different attributes. The dataset includes various features, such as model, manufacturer, year, transmission, engine, and power.

By exploring relationships and correlations between these attributes, the project aims to build a prediction model that can accurately estimate a vehicle's price. This analysis will assist consumers in determining the selling price of their cars, reducing the risk of undervaluing them. Overall, this project has practical applications in the industry, helping to provide valuable insights and inform consumers about successful automobile segments in the market.

2.INTRODUCTION

The project aims to perform various visualizations and perform data analysis on the automobile dataset in order to determine the various relationships between different features of the vehicle. The visualization starts with univariate analysis, analysing the data in perspective of a single attribute then with bivariate analysis and then with multivariate which deals with more than two attributes at the same time.

In this project we are using the Indian automobile dataset and perform various analysis of the attributes like the capacity and power of the automobiles using python programming language. The insights that could be estimated from this dataset would be feature such as price of a specific car model that could be estimated using the other attributes of that particular car model using machine learning algorithms like linear regression or polynomial regression.

Finally, we shall be building a machine learning model that is capable of predicting the price of a vehicle based on the other attributes of the automobile.

3. LITERATURE REVIEW

Car price prediction is a significant area of research in the automotive industry, with implications for buyers, sellers, and market analysts. Recent studies have leveraged various machine learning techniques to develop predictive models that enhance pricing accuracy and decision-making.

1. **Choudhary et al. (2018)** compared multiple algorithms, including Linear Regression and Random Forest, emphasizing the role of feature selection. They found that ensemble methods, particularly Random Forest, provided the highest accuracy.
2. **Singh et al. (2021)** explored various machine learning algorithms like SVM and Gradient Boosting. Their study highlighted that ensemble methods consistently outperformed traditional regression models after extensive data preprocessing.
3. **Shah et al. (2020)** focused on predicting used car prices, demonstrating that XGBoost achieved the best performance among algorithms like Random Forest and KNN, underlining the importance of data normalization and encoding.
4. **Chaudhary et al. (2019)** examined different regression techniques, finding Decision Tree Regression to be particularly effective due to its accuracy and interpretability.
5. **Amamou and Fadlalla (2020)** evaluated the impact of various algorithms, including Deep Learning, indicating that advanced methods could capture complex data relationships and improve prediction accuracy.
6. **Prasad and Jain (2018)** specifically highlighted the strengths of Decision Tree Regression in handling categorical data and providing interpretable results.
7. **Shubham et al. (2019)** emphasized feature engineering's importance in model performance, showcasing how Random Forest could benefit from carefully selected features.

4.OBJECTIVES

1. Primary objective:

To build the Car Price Prediction Models using various algorithms and choose the best one out of them.

2. Secondary Objectives:

- **Data Visualization**:

  Visualize insights from the Indian automobile dataset using data analysis techniques.
- **Data Analysis**:

  Perform comprehensive data analysis to understand the dataset's structure and relationships among features.
- **Model Development**:

  Derive a prediction model to estimate car pricing based on various parameters (e.g., manufacturer, year, horsepower).
- **Feature Importance**:

  Identify and evaluate the importance of different features (e.g., model, manufacturer, year, transmission, engine, power) in predicting car prices.
- **Correlation Study**:

  Analyse the correlation between different attributes of the dataset to understand their relationships.
- **Redundancy Elimination**:

  Identify and eliminate redundant features that do not contribute significantly to the prediction model.
- **Utilization of Machine Learning**:

  Implement machine learning algorithms (e.g Random Forest, Linear Regression, XG Boost) in python to build predictive models for car prices.
- **Insights Generation**:

  Generate actionable insights regarding car pricing based on the analysis of the dataset.
- **Consolidation of Findings**:

  Consolidate and present the findings related to the relationships between attributes in the dataset.
- **Evaluation of Prediction Accuracy**:

  Assess the accuracy and reliability of the prediction model to ensure it provides valid estimates of car prices.

5. DATA DESCRIPTION

I] The dataset used for this study has been taken from the Kaggle website.

[https://www.kaggle.com/code/kens3i/Indian-Automobile-Market-dataset-142]

II] Name of the Data: Indian Automobile Market Dataset

III] Data Size:

The dataset consist of 5975 rows and 15 columns.

Categorical attributes are

- Manufacturer
- Location
- Fuel Type
- Transmission
- Ownership

Integer attributes are

- Year
- Km Driven
- Engine CC
- Seats

Float attributes are

- Power
- Price

IV] Variable description:

The data reside in a comma-separated values (csv) file. A header line contains the name of the variables. The following are the description of all attributes:

1. **Name :-** It is Name of the car.
2. **Manufacturer :-** It include the name of manufacturing company of cars
3. **Location :-** It include the city where car is bought or sold.

4. Year :- Year in which car is bought or sold.

5. Kilo-meter Driven :- It is the distance travelled by that car in kilo-meter.

6. Fuel Type :- It include on which fuel the car is working like Petrol, CNG, Diesel, LPG.

7. Transmission :- It include transmission of car ie. It is Manual or Automatic.

8. Owner Type :- It include the information about owner i.e. it is 1st owned, 2nd owned , etc.

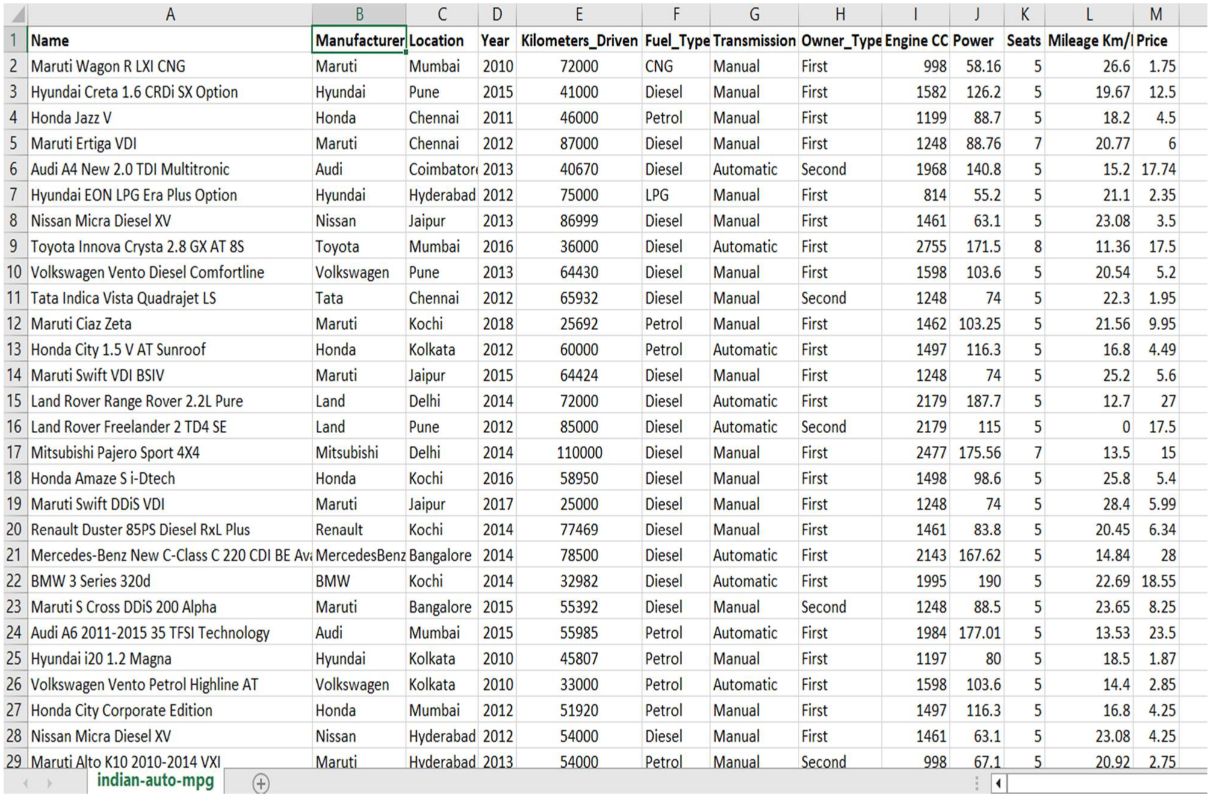
9. Engine CC :- Engine cubic capacity, refers to the total volume of all the cylinders in an engine, measured in cubic centi-meter (CC). It indicates the size or displacement of the engine.

10. Power :- Power in the context of engines refers to the rate at which work is done or energy is produced. For internal combustion engines.

11.**Seats :-** It includes the no. of seats in car.

12.**Milage :-** It includes the mileage of car.

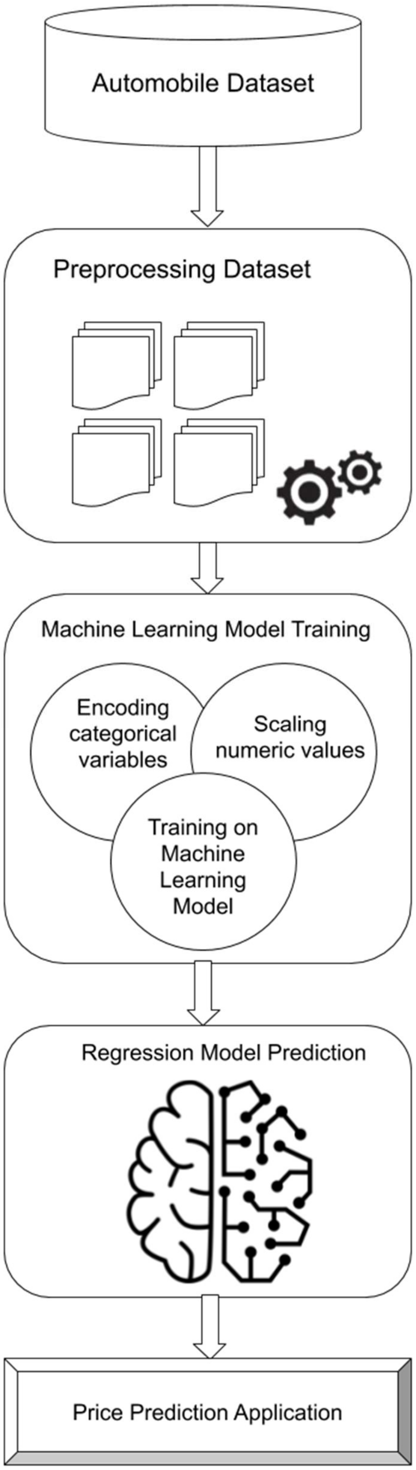
13.**Price :-** It gives the price of car at which it is bought or sold.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Name**	**Manufacturer**	**Location**	**Year**	**Kilometers_Driven**	**Fuel_Type**	**Transmission**	**Owner_Type**	**Engine CC**	**Power**	**Seats**	**Mileage Km/l**	**Price**
2	Maruti Wagon R LXI CNG	Maruti	Mumbai	2010	72000	CNG	Manual	First	998	58.16	5	26.6	1.75
3	Hyundai Creta 1.6 CRDi SX Option	Hyundai	Pune	2015	41000	Diesel	Manual	First	1582	126.2	5	19.67	12.5
4	Honda Jazz V	Honda	Chennai	2011	46000	Petrol	Manual	First	1199	88.7	5	18.2	4.5
5	Maruti Ertiga VDI	Maruti	Chennai	2012	87000	Diesel	Manual	First	1248	88.76	7	20.77	6
6	Audi A4 New 2.0 TDI Multitronic	Audi	Coimbatore	2013	40670	Diesel	Automatic	Second	1968	140.8	5	15.2	17.74
7	Hyundai EON LPG Era Plus Option	Hyundai	Hyderabad	2012	75000	LPG	Manual	First	814	55.2	5	21.1	2.35
8	Nissan Micra Diesel XV	Nissan	Jaipur	2013	86999	Diesel	Manual	First	1461	63.1	5	23.08	3.5
9	Toyota Innova Crysta 2.8 GX AT 8S	Toyota	Mumbai	2016	36000	Diesel	Automatic	First	2755	171.5	8	11.36	17.5
10	Volkswagen Vento Diesel Comfortline	Volkswagen	Pune	2013	64430	Diesel	Manual	First	1598	103.6	5	20.54	5.2
11	Tata Indica Vista Quadrajet LS	Tata	Chennai	2012	65932	Diesel	Manual	Second	1248	74	5	22.3	1.95
12	Maruti Ciaz Zeta	Maruti	Kochi	2018	25692	Petrol	Manual	First	1462	103.25	5	21.56	9.95
13	Honda City 1.5 V AT Sunroof	Honda	Kolkata	2012	60000	Petrol	Automatic	First	1497	116.3	5	16.8	4.49
14	Maruti Swift VDI BSIV	Maruti	Jaipur	2015	64424	Diesel	Manual	First	1248	74	5	25.2	5.6
15	Land Rover Range Rover 2.2L Pure	Land	Delhi	2014	72000	Diesel	Automatic	First	2179	187.7	5	12.7	27
16	Land Rover Freelander 2 TD4 SE	Land	Pune	2012	85000	Diesel	Automatic	Second	2179	115	5	0	17.5
17	Mitsubishi Pajero Sport 4X4	Mitsubishi	Delhi	2014	110000	Diesel	Manual	First	2477	175.56	7	13.5	15
18	Honda Amaze S i-Dtech	Honda	Kochi	2016	58950	Diesel	Manual	First	1498	98.6	5	25.8	5.4
19	Maruti Swift DDiS VDI	Maruti	Jaipur	2017	25000	Diesel	Manual	First	1248	74	5	28.4	5.99
20	Renault Duster 85PS Diesel RxL Plus	Renault	Kochi	2014	77469	Diesel	Manual	First	1461	83.8	5	20.45	6.34
21	Mercedes-Benz New C-Class C 220 CDI BE Ava	MercedesBenz	Bangalore	2014	78500	Diesel	Automatic	First	2143	167.62	5	14.84	28
22	BMW 3 Series 320d	BMW	Kochi	2014	32982	Diesel	Automatic	First	1995	190	5	22.69	18.55
23	Maruti S Cross DDiS 200 Alpha	Maruti	Bangalore	2015	55392	Diesel	Manual	Second	1248	88.5	5	23.65	8.25
24	Audi A6 2011-2015 35 TFSI Technology	Audi	Mumbai	2015	55985	Petrol	Automatic	First	1984	177.01	5	13.53	23.5
25	Hyundai i20 1.2 Magna	Hyundai	Kolkata	2010	45807	Petrol	Manual	First	1197	80	5	18.5	1.87
26	Volkswagen Vento Petrol Highline AT	Volkswagen	Kolkata	2010	33000	Petrol	Automatic	First	1598	103.6	5	14.4	2.85
27	Honda City Corporate Edition	Honda	Mumbai	2012	51920	Petrol	Manual	First	1497	116.3	5	16.8	4.25
28	Nissan Micra Diesel XV	Nissan	Hyderabad	2012	54000	Diesel	Manual	First	1461	63.1	5	23.08	4.25
29	Maruti Alto K10 2010-2014 VXI	Maruti	Hyderabad	2013	54000	Petrol	Manual	Second	998	67.1	5	20.92	2.75

6.METHODOLOGY

Proposed Architecture


Automobile Dataset
Preprocessing Dataset
Machine Learning Model Training
Encoding categorical variables
Scaling numeric values
Training on Machine Learning Model
Regression Model Prediction
Price Prediction Application

Methodology :-

The architecture of the entire project is divided into two parts which are the visualization and the data analysis parts of the project. The visualization part of the project deals with the various plotting of attributes while the data analysis part of the project deals with finding the relationship between various attributes in the dataset.

First the dataset if taken into preprocessing. The visualization part consists of univariate analysis, analysing the data in perspective of a single attribute then with bivariate analysis, analysis using two attributes and then with multivariate which deals with more than two attributes at the same time. Here the attribute’s distributions are visualized using count plots, bar-plots, histograms, etc. The bivariate analysis is done using scatter plots, box plots, violin plots and so on.

The data analysis is performed on the automobile dataset utilizing machine learning algorithms in order to study the various relationships between attributes of the considered Indian automobile dataset and attempts to consolidate the findings of the relationship between the attributes or statistically, finding the correlation between them and visualizing the findings. Of these features some of them might be a redundant and might be a good contributor to the prediction model and the task of eliminating such attributes also shall be considered. The result of finding this relationship between various attributes of a vehicle will provide useful insights in building in a prediction model capable of predicting the price of a vehicle based on the other parameters like manufacturer, year, horsepower and so on.

7.STATISTICAL TOOLS

➢ Tools:

1] Exploratory Data Analysis (EDA) :-

Bar charts, Scatter plots, Correlation Heat map, Boxplot

2] Testing of Hypothesis :-

Test for Normality, Chi-square test, Mann-Whitney U test

3] Machine Learning Algorithms (Data Mining Classifiers) :-

1. Linear Regression
2. Random Forest
3. XG boosting Regression

➢ Statistical Software:

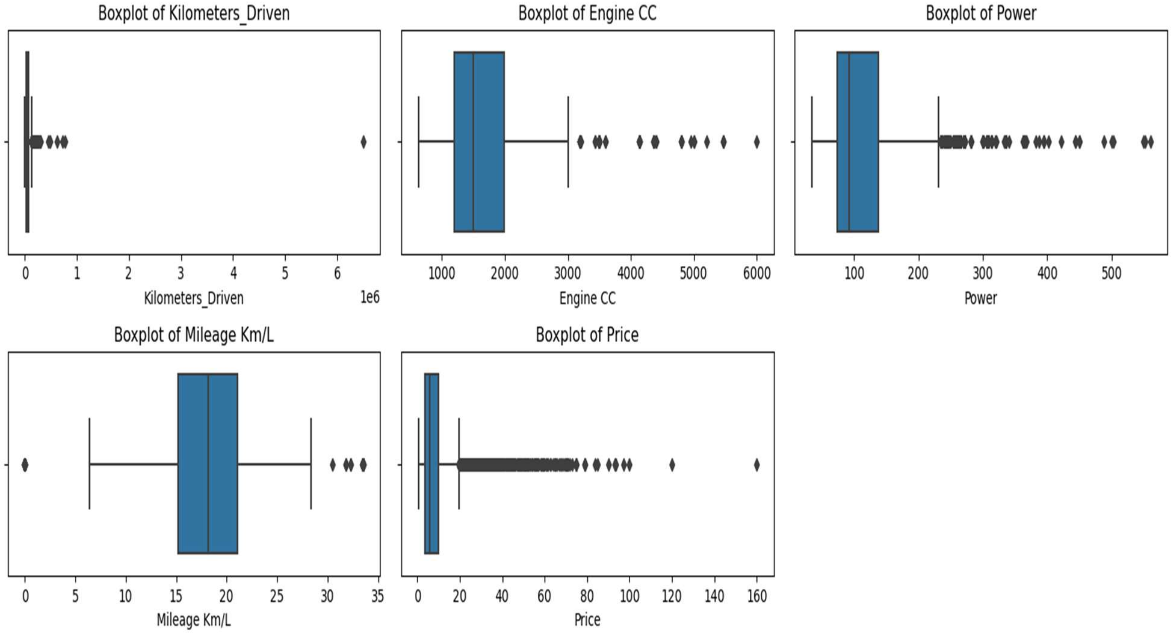
1. MS-Excel
2. Python

8. DATA PREPROCESSING

Outliers:- Outliers are data points significantly different from other obs. in dataset, caused by measurement errors, data entry mistakes or genuine anomalies.

Impact:

- **Skewed Results**: They can distort statistical metrics (mean, standard deviation) and lead to biased conclusions.
- **Model Performance**: Outliers can adversely affect the accuracy of sensitive models, such as linear regression.



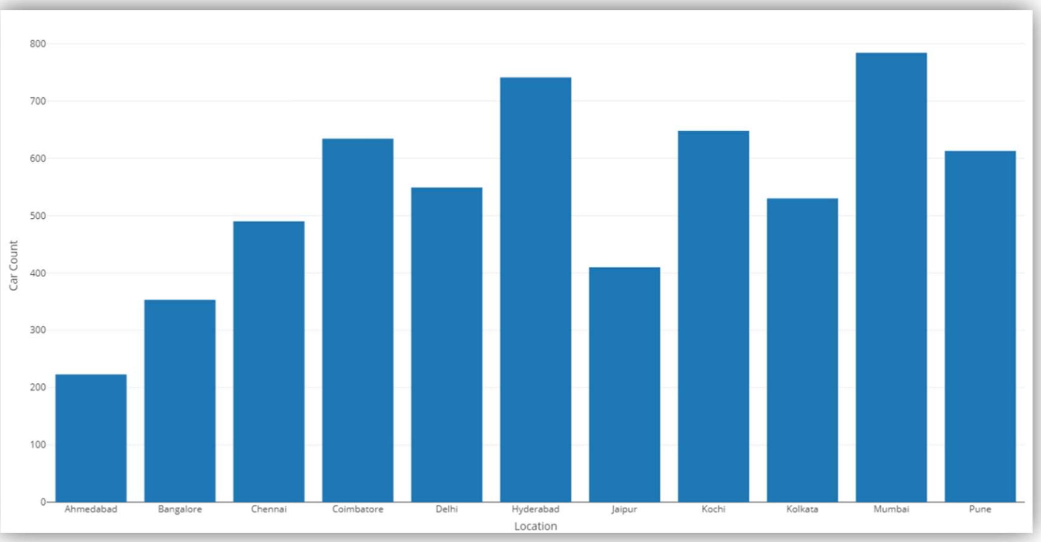
Interpretation :-

- There are 2134 outliers in a dataset out of 5975 rows indicate a substantial presence of extreme values (roughly 36%).
- Models that are sensitive to outliers, such as simple linear regression or even non-robust models, might yield biased predictions and overfit due to the undue influence of these extreme data points.
- Using models that are robust to outliers, such as Random Forest and Gradient Boosting, will ensure that the model performance remains consistent and reliable even with a significant number of outlier

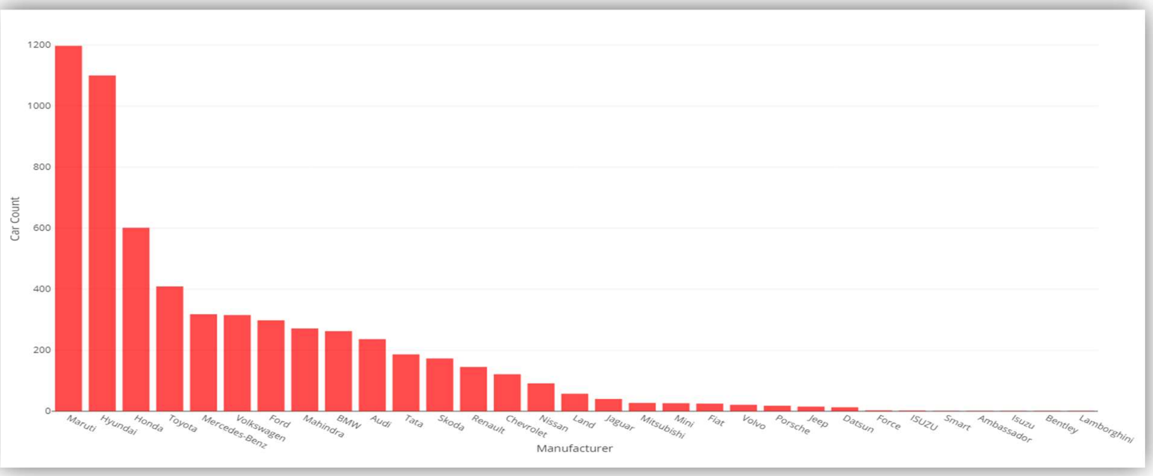
9. EXPLORATORY DATA ANALYSIS

➢ Univariate Analysis:

- Histogram for Location in which car exchange is done.



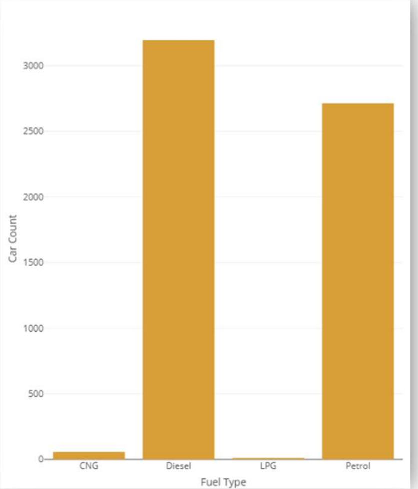
- Histogram for manufacturer of cars w.r.t car count.

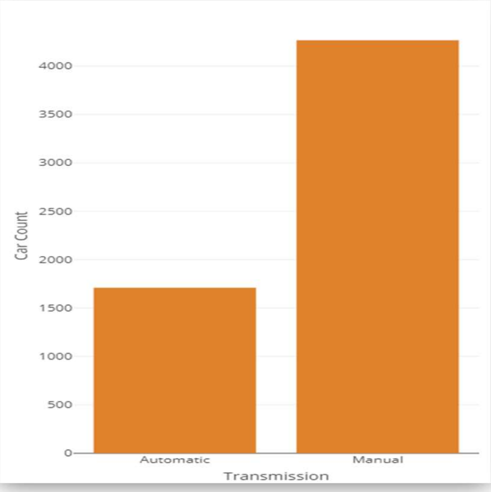


Interpretation :

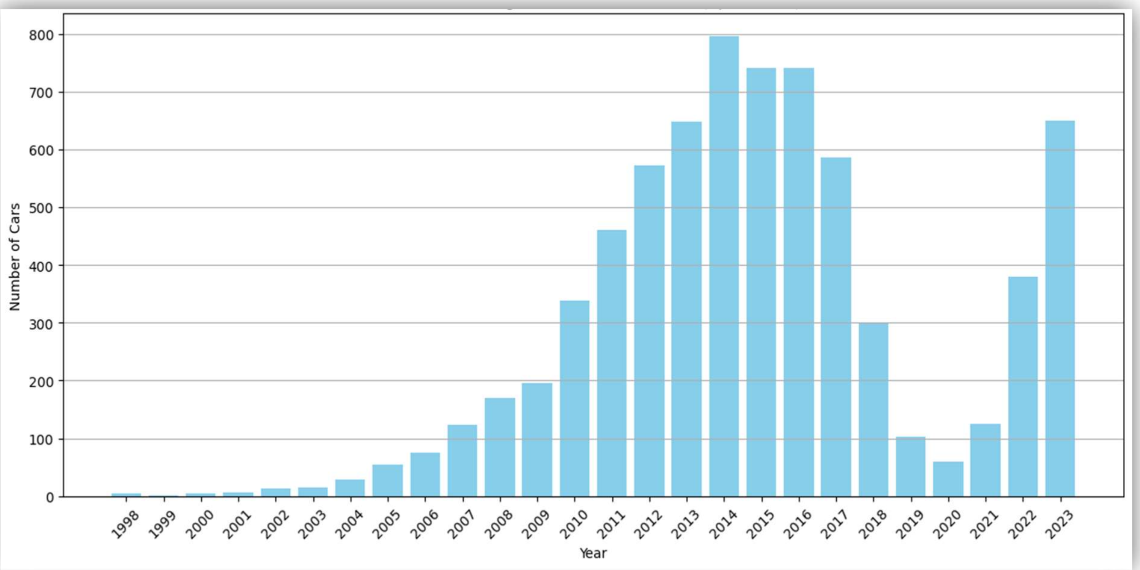
- The histograms shows that most of the sale of cars i.e. about 13% is done in Mumbai city and about 12% sale in Hyderabad city and nearly 3% sale is done in Ahmedabad city.
- The manufactures like Maruti and Hyundai have monopoly in market according to count of cars i.e. about 20% and 18% respectively.

- Histograms for type of Fuel used in car and Transmission of car.





- The graph for year wise exchange of car.

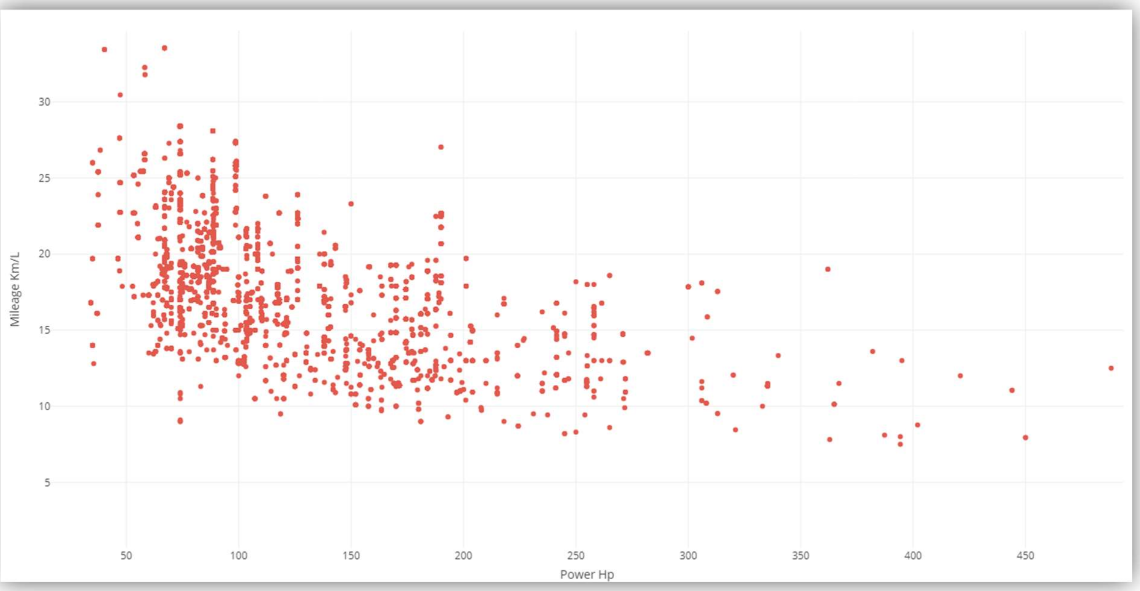


Interpretation :-

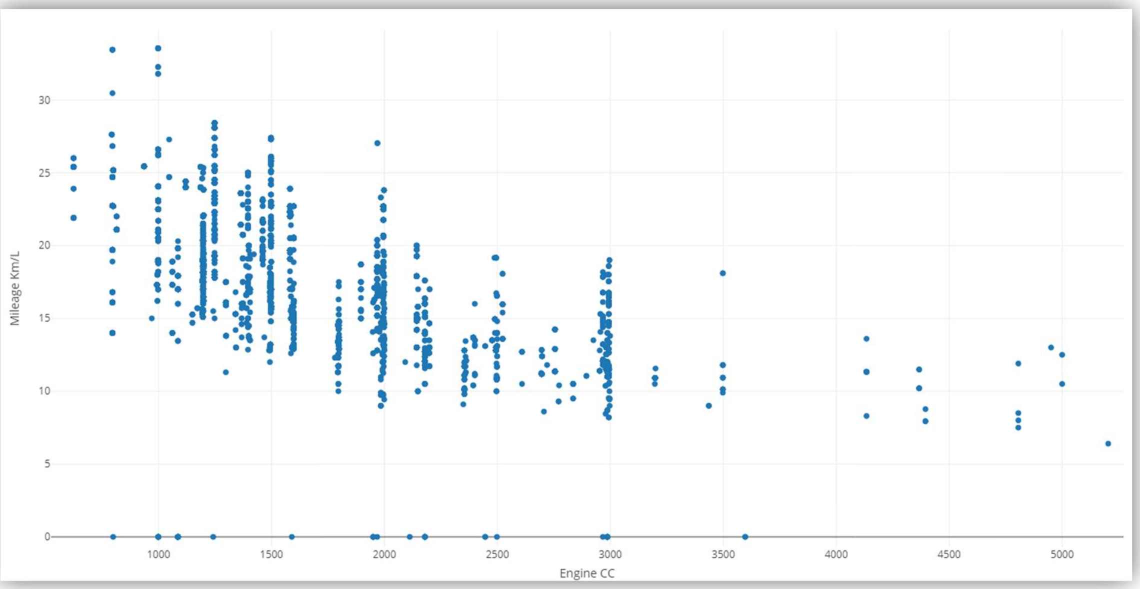
- The data shows that the selling and buying of cars which works on Petrol and Diesel as fuel is higher than that of CNG and LPG.
- The cars in which transmission is manual are more purchased or sold than automatic transmission cars.
- The data shows that in 2014 there was maximum automobile exchange and during the period of Covid-19 pandemic this exchange is very low as compared to other time.

➢ Bivariate Analysis:

- The scatter plot for Power of car in HP and Mileage of car in Km/L.



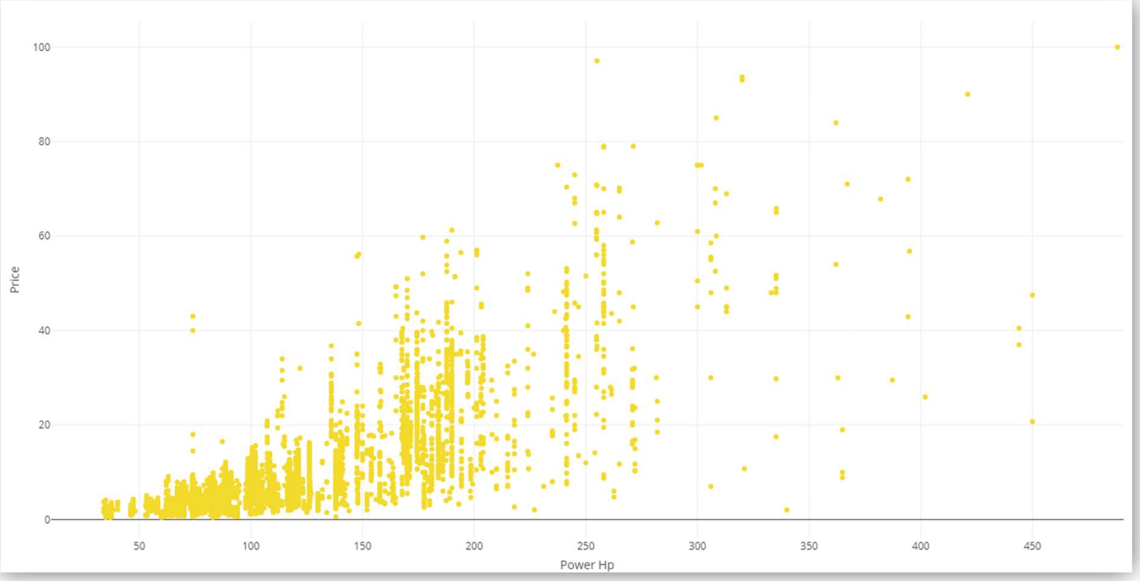
- Scatter plot for Engine CC and Mileage of car in Km/L.



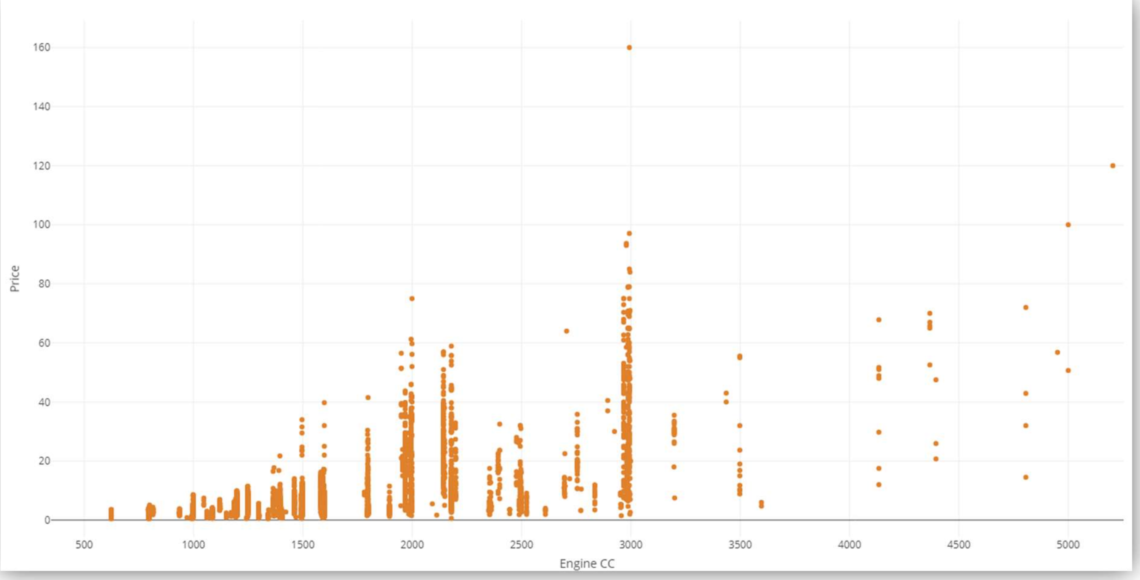
Interpretation :-

The scatter plots shows hat the most of the cars mileage ranging from 10 to 25 Km/L and the power of car ranging in between 50 to 250 are frequently used.

- Graphical representation of price of car with power of car



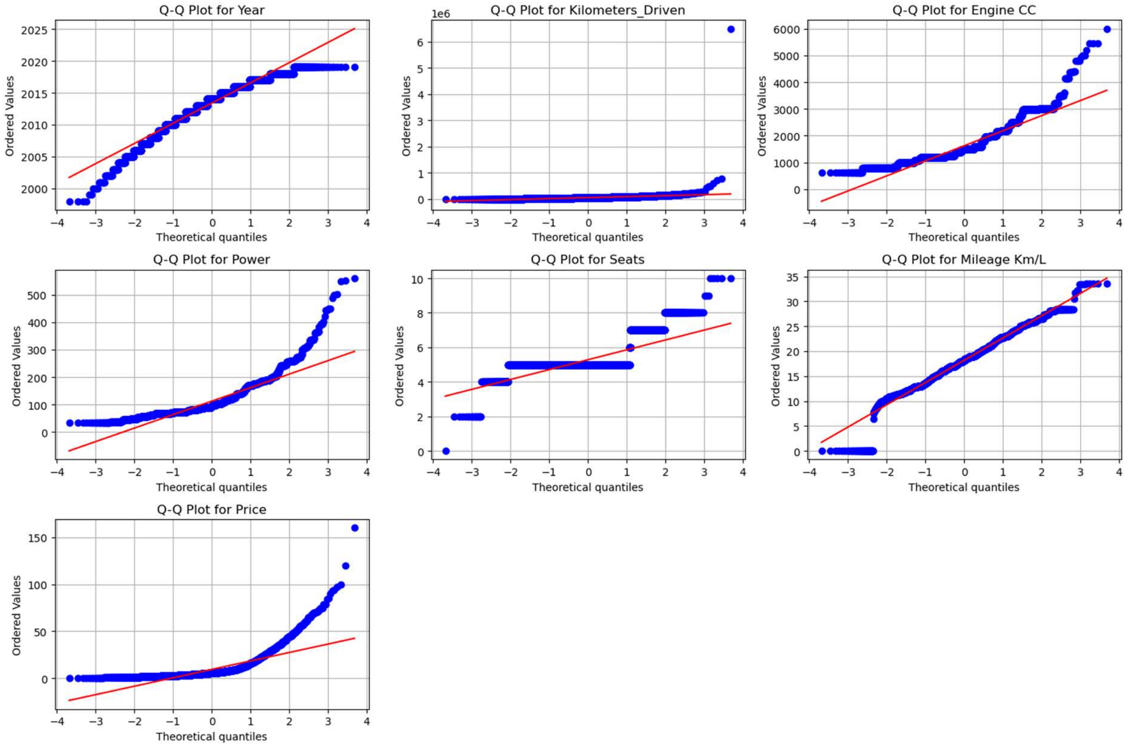
- The scatter plot of price of car and engine cc of car.



Interpretation:-

- The graph shows that as power increases price of car also increases and cars with 50Hp to 250Hp are more in the market.
- Scatter plots shows that there are lots of different cars with same engine cc at different price of car and the market is crowded by cars with engine cc upto 3000.

10. NORMALITY OF THE DATA

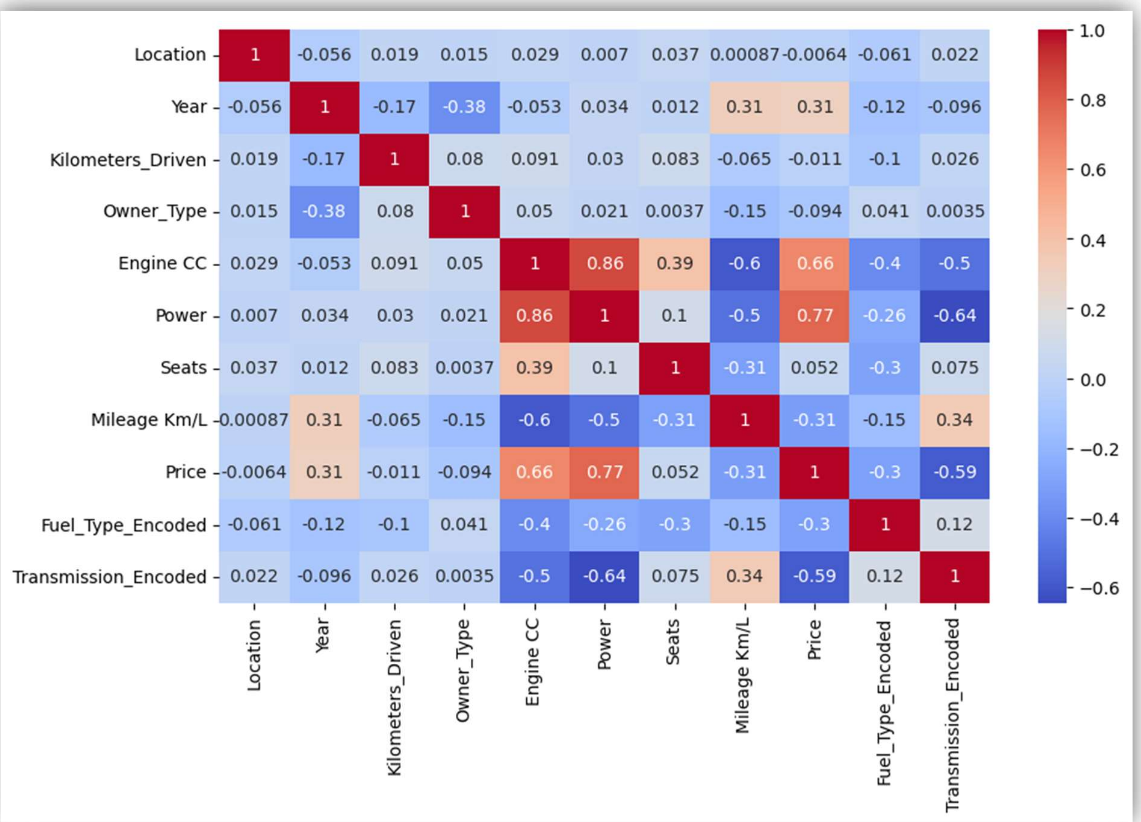


Conclusion:

From these Q-Q plots we can see the normality of each variable. The variable Price , Mileage, Seats, Power, Engine CC , Kilo-meters Driven and Year are non normally distributed.

11. CORRELATION HEATMAP

The following map shows the correlation between different variables which constructed using python software.



	Location	Year	Kilometers_Driven	Owner_Type	Engine CC	Power	Seats	Mileage Km/L	Price	Fuel_Type_Encoded	Transmission_Encoded
Location	1	-0.056	0.019	0.015	0.029	0.007	0.037	0.00087	-0.0064	-0.061	0.022
Year	-0.056	1	-0.17	-0.38	-0.053	0.034	0.012	0.31	0.31	-0.12	-0.096
Kilometers_Driven	0.019	-0.17	1	0.08	0.091	0.03	0.083	-0.065	-0.011	-0.1	0.026
Owner_Type	0.015	-0.38	0.08	1	0.05	0.021	0.0037	-0.15	-0.094	0.041	0.0035
Engine CC	0.029	-0.053	0.091	0.05	1	0.86	0.39	-0.6	0.66	-0.4	-0.5
Power	0.007	0.034	0.03	0.021	0.86	1	0.1	-0.5	0.77	-0.26	-0.64
Seats	0.037	0.012	0.083	0.0037	0.39	0.1	1	-0.31	0.052	-0.3	0.075
Mileage Km/L	0.00087	0.31	-0.065	-0.15	-0.6	-0.5	-0.31	1	-0.31	-0.15	0.34
Price	-0.0064	0.31	-0.011	-0.094	0.66	0.77	0.052	-0.31	1	-0.3	-0.59
Fuel_Type_Encoded	-0.061	-0.12	-0.1	0.041	-0.4	-0.26	-0.3	-0.15	-0.3	1	0.12
Transmission_Encoded	0.022	-0.096	0.026	0.0035	-0.5	-0.64	0.075	0.34	-0.59	0.12	1

Conclusions :

Here we get that power of car and engine cc of car are highly (0.86) positively correlated and power of car and transmission (i.e. Manual or Automatic are negatively correlated.) Also price of car is highly correlated with power of car.

12. Feature Selection Process

- **Recursive Feature Elimination (RFE):-**

  RFE is a feature selection method that recursively eliminates less important features and builds models using the remaining ones. It is commonly used to improve model performance by reducing the complexity of the feature space.

Key Steps in RFE:

1. **Initial Model Training:**
   - A model (e.g., linear regression, decision tree, random forest) is trained using all the available features.
2. **Feature Importance Evaluation:**
   - The importance of each feature is evaluated based on its contribution to the prediction accuracy of the model.
3. **Feature Elimination:**
   - The least important feature(s) are removed from the feature set.
4. **Recursive Process:**
   - Steps 1–3 are repeated recursively, where in each iteration, the model is re-trained, and the least important feature(s) are removed.
5. **Feature Ranking:**
   - This process continues until the desired number of features is reached or all features have been evaluated. The remaining features are ranked based on their importance.

Advantages:

- **Improved Model Performance:** By removing irrelevant or redundant features, RFE can enhance model performance, reduce overfitting, and improve generalization.
- **Dimensionality Reduction:** It helps to simplify models by reducing the number of input variables.

- **Ranking given by Feature Importance Evaluation :-**

```
              Feature        Ranking
0              Year             1
1    Kilo-meters_Driven          3
2          Engine CC           4
3             Power             1
4             Seats              2
5         Mileage Km/L          1
6     Fuel_Type_Encoded        1
7  Transmission_Encoded        1
```

Interpreting the Results:

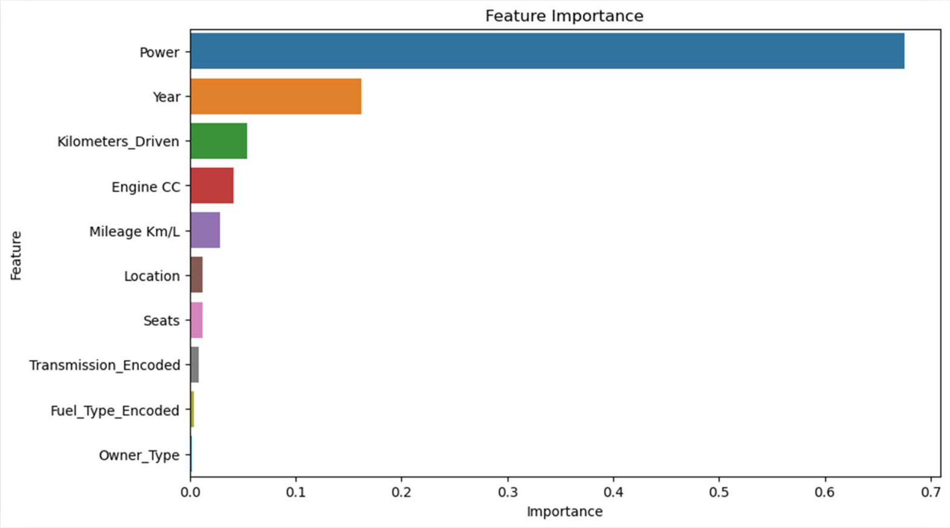
From the Recursive Feature Elimination (RFE) process, we have a set of features ranked based on their importance in predicting the target variable. Here's how to interpret the rankings:

1. **Highly Important Features (Ranking: 1):**
   - **Year, Power, Mileage Km/L, Fuel Type Encoded, Transmission Encoded:** These features are the most significant in determining the outcome (e.g., car price). They have a direct and strong influence on the prediction. Including these in your model is crucial, as removing them would likely reduce its predictive accuracy.
2. **Moderately Important Features (Ranking: 2):**
   - **Seats:** This feature still plays a role but is not as influential as the features ranked 1. It might contribute to refining predictions but isn't essential for the model's core structure. You can consider it depending on your goals.
3. **Less Important Features (Rankings: 3 and 4):**
   - **Kilometers Driven (Ranking: 3)**
   - **Engine CC (Ranking: 4)** These features have less influence on the prediction. The model's performance may not deteriorate significantly if they are removed. Keeping them might introduce noise or complexity that doesn't add much value. Therefore, you can consider excluding these features or analysing them further to see if their removal improves model performance.

❖ Feature Importance Analysis Using Random Forest

Random Forest provides feature importance scores that indicate the contribution of each feature in making predictions. These scores are derived from how often and effectively each feature is used in splitting nodes across all trees in the forest.

```
              Feature      Importance
5               Power       0.675258
1                Year       0.161960
2   Kilometers_Driven       0.054174
4           Engine CC        0.041403
7        Mileage Km/L        0.028566
0            Location       0.012333
6               Seats       0.012308
9 Transmission_Encoded      0.008421
8    Fuel_Type_Encoded      0.003781
3          Owner_Type       0.001797
```



Interpretation :-

- **Key Features to Retain**: Power and Year should definitely be included in your model as they are the most influential predictors.
- **Moderate Importance Considerations**: Consider keeping Kilo-meters Driven, Engine CC, and Mileage Km/L, but monitor their contributions during model evaluation.
- **Low Importance Removal**: It may be beneficial to remove Location, Seats, Transmission Encoded, Fuel Type Encoded, and Owner Type from your model to reduce complexity and prevent overfitting, as these features do not add significant predictive value.

13. FITTING OF MODELS

This is the most exciting phase in Applying Machine Learning to any Dataset. It is also known as Algorithm Selection for Predicting the best results. Usually Data Scientists use different kinds of Machine Learning algorithms to the large data sets. But, at high level all those different algorithms can be classified in two groups: supervised learning and unsupervised learning.

Supervised learning**:** Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labelled for classification to provide a learning basis for future data processing. Supervised learning problems can be further grouped into Regression and Classification problems.

Unsupervised learning: Un-supervised learning is a type of system in which training set consists of only input vectors & desired output is not given.

The models that are used for prediction are:

- Random Forest
- Linear Regression
- XG-boost

Random Forest Regressor

Random Forest or random decision forests [5] are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of over fitting to their training set. In this model we use Random Forest library and build the model, after we check the accuracy. It gives useful internal estimate of error correlation and variable.

Evaluation :

```
Model Evaluation Metrics:
RMSE: 4.690013766255802
R-squared: 0.8489028258612139

Actual vs. Predicted Prices:
      Actual Price  Predicted Price
5957         17.56          18.6243
4333          6.60           6.4796
2585         11.50          11.1065
1755         38.00          27.1811
4773          9.66          12.1719
Summary of Predictions:
       Actual Price  Predicted Price
count   1195.000000      1195.000000
mean       9.265389         9.180073
std       12.070569        10.535242
min        0.510000         0.602693
25%        3.500000         3.614350
50%        5.500000         5.562900
75%        9.185000         9.633950
max      160.000000        70.966700
```

Conclusion:

Random Forest model for car price prediction performs well, with an **RMSE of 4.69** and **R-squared of 0.8489**, indicating accurate predictions for most cars. However, the model slightly underestimates high-value cars, and while it captures general trends in prices, it may miss some extreme values. Overall, the model explains 85% of the variance in car prices and provides reasonable predictions for the majority of cases.

Linear Regression

Linear Regression is a foundational statistical method used to model the relationship between a continuous dependent variable and one or more independent variables. It aims to establish a linear equation that best predicts the dependent variable based on the values of the independent variables. Unlike logistic regression, which is suited for categorical outcomes, linear regression is designed for situations where the response variable is continuous.

Report:

```
Linear Regression RMSE: 8.401923403394687
Linear Regression R-squared: 0.5150850841731992

Actual vs. Predicted Prices (Linear Regression):
      Actual Price  Predicted Price
5957         17.56        17.809979
4333          6.60         8.230505
2585         11.50        15.052827
1755         38.00        20.484023
4773          9.66        13.347416

Summary of Linear Regression Predictions:
       Actual Price  Predicted Price
count   1195.000000      1195.000000
mean       9.265389         8.873015
std       12.070569         9.201789
min        0.510000       -72.571518
25%        3.500000         3.449330
50%        5.500000         7.502361
75%        9.185000        12.589940
max      160.000000        60.337680
```

Conclusion:

The Linear Regression model shows a **RMSE of 8.40**, indicating significant average deviation from actual car prices. With an **R-squared value of 0.515**, the model explains only 51.5% of the variance, suggesting a poor fit to the data. While some predictions are reasonably close to actual prices, the model also produces unrealistic outcomes, such as negative predicted prices. Overall, the Linear Regression model underperforms in capturing the complexity of car prices compared to more advanced models like Random Forest.

Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an advanced implementation of gradient boosting designed to optimize performance and efficiency in machine learning tasks, particularly for supervised learning problems. It is widely used for both classification and regression tasks due to its high accuracy and speed.

XGBoost is based on the boosting ensemble technique, where multiple weak learners (typically decision trees) are combined to create a strong predictive model. It sequentially adds trees to minimize the error of the previous model.

Evaluation :

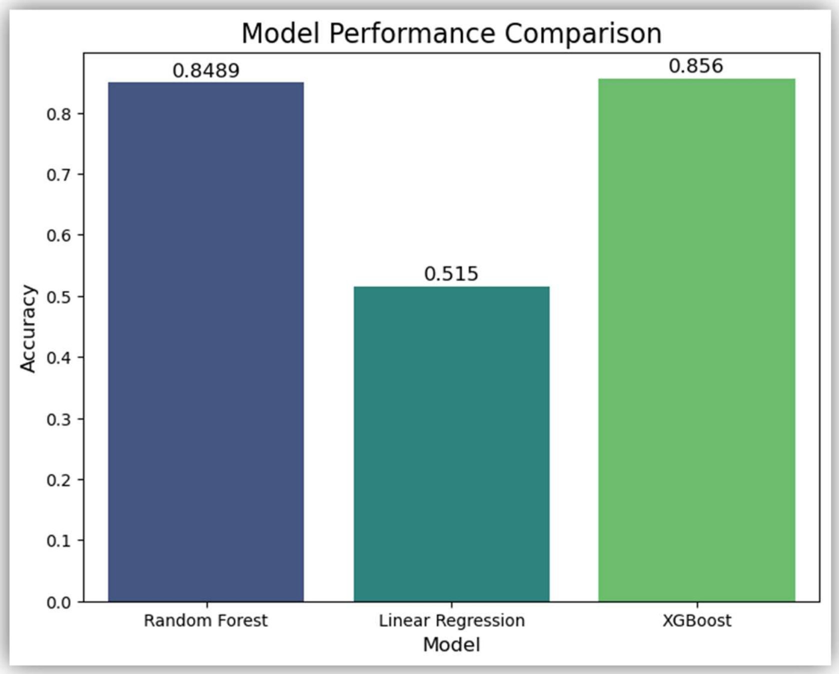
```
XGBoost Model Evaluation Metrics:
RMSE: 4.581272272884238
R-squared: 0.8558282022246553
Actual vs. Predicted Prices (XGBoost):
      Actual Price  Predicted Price
5957         17.56        20.050201
4333          6.60         6.727180
2585         11.50         9.907187
1755         38.00        34.206203
4773          9.66        11.091131
Summary of XGBoost Predictions:
       Actual Price  Predicted Price
count   1195.000000      1195.000000
mean       9.265389         9.182899
std       12.070569        10.754298
min        0.510000         0.455625
25%        3.500000         3.602210
50%        5.500000         5.480633
75%        9.185000         9.383762
max      160.000000        75.495773
```

Conclusion:

The XGBoost model exhibits a **RMSE of 4.58**, indicating a relatively low average deviation from actual car prices. With an **R-squared value of 0.856**, the model explains **85.6% of the variance** in the data, suggesting a strong fit. The predictions closely align with actual prices, with the model demonstrating improved accuracy compared to both Linear Regression and Random Forest models. Overall, XGBoost is an effective choice for predicting car prices in this dataset.

- **The Performance of the above model is as follows:**

Performance / **Model**	**Accuracy**
Random Forest	0.8489
Linear Regression	0.5150
XG boost	0.8560



Conclusion:

From the above figure we can observe that **XGboost** model has the highest accuracy. Therefore, XGboost can be used to predict the price of car.

14. TESTING OF HYPOTHESIS

❖ Chi-Square Test of Independence:

1. To check dependency between fuel type of car and Transmission of car.

Hypothesis :

H_0 :- There is no significant association between Fuel_Type and Transmission. (i.e., they are independent).

v/s

H_1 :- There is a significant association between Fuel_Type and Transmission (i.e., they are dependent).

Result:

```
Chi-Square Statistic: 130.03269862821142
P-value: 5.321410906035932e-28
Degrees of Freedom: 3
Expected Frequencies:
 [[  16.01740586   39.98259414]
 [ 913.85020921 2281.14979079]
 [   2.86025105    7.13974895]
 [ 776.27213389 1937.72786611]]
```

There is no significant association between Fuel_Type and Transmission.

Interpretation :-

Since the p-value is much smaller than the common significance level of 0.05, we reject the null hypothesis. This means there is a significant association between 'Fuel_Type' and 'Transmission' of car.

2. To check dependency between owner type and car fuel type.

Hypothesis :

H_0 :- The two attributes owner type and fuel type of car are independent. i.e. There is no significant association between 'Fuel_Type' and Owner Type.

v/s

H_1 :-The two attribute owner type and fuel type of car are dependent. i.e. There is significant association between 'Fuel_Type' and Owner Type.

Result:

```
Chi-Square Statistic: 16.32199082842446
P-value: 0.060454523748290315
Degrees of Freedom: 9
Expected Frequencies:
 [[4.59528033e+01 2.62177155e+03 8.20585774e+00 2.22706979e+03]
 [7.49790795e-02 4.27782427e+00 1.33891213e-02 3.63380753e+00]
 [8.93188285e+00 5.09595816e+02 1.59497908e+00 4.32877322e+02]
 [1.04033473e+00 5.93548117e+01 1.85774059e-01 5.04190795e+01]]
There is no significant association between 'Owner_Type' and 'Fuel_Type'.
```

Interpretation :

The type of ownership (e.g., first owner, second owner, etc.) does not significantly influence the choice of fuel type for the vehicles.

❖ Mann-Whitney U Test:

Hypothesis :

H_0 :- The price distributions of automatic and manual transmission cars are the same (i.e., there is no significant difference between the prices). **v/s**

H_1 :- The price distributions of automatic and manual transmission cars are different (i.e., there is a significant difference between the prices).

Result :

Mann-Whitney U Statistic: 974642.5
P-value: 0.0
There is a significant difference in car prices between Manual and Automatic transmissions.

Interpretation :

Since the p-value is extremely small (essentially 0), we reject the null hypothesis. This means there is a highly significant difference between the prices of cars with automatic transmission and those with manual transmission.

15. MAJOR FINDING

1. Exploratory Data Analysis (EDA)

- **Correlation Analysis**: The correlation matrix highlighted significant positive correlations between car price and several features, notably:
  - **Power** (0.675): A higher power output is associated with higher prices.
  - **Engine CC** (0.675): Larger engine sizes correlate with increased prices.
  - **Year** (0.161): Newer models tend to be priced higher.

.

2. Machine Learning Models

- **Random Forest Regressor**:
  - **RMSE**: 4.84
  - **R-squared**: 0.839
  - This model effectively captured the variance in the dataset, making it suitable for predicting car prices based on input features.
- **XGBoost Regressor**:
  - **RMSE**: 4.26
  - **R-squared**: 0.856
  - The XGBoost model outperformed the Random Forest model, showcasing superior predictive capability due to its ability to handle complex interactions and non-linear relationships.
  -

3. Statistical Tests

- A **significant association** was found between **Fuel Type** and **Transmission** (Chi-Square Statistic: 130.03, p-value: 5.32e-28). This suggests that the type of fuel a vehicle uses is related to the transmission type (manual or automatic) in vehicles.
- **No significant association** was found between **Owner Type** and **Fuel Type** (Chi-Square Statistic: 16.32, p-value: 0.0605). This indicates that the ownership type of a vehicle does not significantly influence the type of fuel used.
- A **significant difference** in car prices was identified between **manual** and **automatic transmissions** (Mann-Whitney U Statistic: 974,642.5, p-value: 0.0), reinforcing the observation that automatic transmission cars are generally more expensive than manual transmission cars.

16. SCOPE & LIMITATIONS OF THE STUDY

➢ SCOPE OF THE STUDY:

- **Data Analysis and Model Development**: The project involves analysing a comprehensive dataset of car attributes and prices to develop predictive models using machine learning techniques. It aims to identify key factors influencing car prices.
- **Comparison of Machine Learning Algorithms**: The project explores various algorithms, including Random Forest, XG-Boost, Linear Regression, and Decision Trees. This comparison provides insights into the effectiveness and accuracy of each method in predicting car prices.
- **Feature Importance Assessment**: The project focuses on understanding the importance of different features in the dataset. Techniques like Recursive Feature Elimination (RFE) and feature importance scores help in identifying significant attributes that affect pricing.
- **Statistical Testing**: The project incorporates statistical tests (e.g., Chi-Square, T-Test, Mann-Whitney U Test) to explore relationships between categorical variables, enhancing the understanding of how different features interact and influence car prices.
- **Real-World Application**: The findings from this project can be applied in the automotive market, helping dealerships, buyers, and sellers make informed decisions based on predictive insights.

➢ LIMITATIONS OF THE STUDY:

- **Data Quality and Availability**: The accuracy of the predictive models is heavily dependent on the quality and completeness of the dataset. Missing or erroneous data can lead to biased predictions.
- **Model Interpretability**: Some machine learning algorithms, particularly ensemble methods like Random Forest and XG-Boost, can act as "black boxes," making it challenging to interpret the decision-making process and the influence of individual features on predictions.
- **Overfitting Risk**: Complex models may perform well on the training data but could struggle to generalize to unseen data, leading to overfitting. Proper validation techniques, such as cross-validation, are necessary to mitigate this risk.
- **External Factors**: The project may not account for external factors affecting car prices, such as market trends, economic conditions, and buyer preferences, which could influence model accuracy.
- **Limited Scope of Features**: The project focuses on specific features available in the dataset. Other relevant attributes (e.g., car condition, geographic location, brand reputation) might not be included, potentially limiting the model's performance.
- **Temporal Aspect**: The dataset may not reflect current market conditions if it is outdated. Car prices fluctuate due to various factors, and predictions may not remain valid over time.

.

17. REFERENCES

1. Choudhary, H., Kumari, S., Gaur, V., & Sharma, A. (2018). *Car Price Prediction using Machine Learning Techniques*. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE). Retrieved from ResearchGate

2. Singh, V. K., Sharma, A., & Kumar, R. (2021). *Car Price Prediction Using Machine Learning*. International Journal of Computer Applications, 176(29), 1-5. Retrieved from IJCA

3. Shah, M. A. W. A., Shah, N. H., Shah, S. A., & Choudhary, A. L. K. (2020). *Predicting Used Car Prices with Machine Learning Techniques*. Journal of Computer and Communications, 8(6), 11-20. Retrieved from Scientific Research Publishing

4. Chaudhary, R. D., Yadav, D., & Chaurasia, S. (2019). *A Comparative Study of Regression Techniques for Car Price Prediction*. International Journal of Computer Applications, 182(6), 18-23. Retrieved from IJCA

5. Amamou, A. A., & Fadlalla, M. I. H. (2020). *Predicting Car Prices Using Machine Learning Algorithms*. Journal of Physics: Conference Series, 1685(1), 012072. doi:10.1088/1742-6596/1685/1/012072

APPENDIX

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score
from scipy.stats import chi2_contingency, shapiro, levene, mannwhitneyu,
kruskal

# Load your dataset
df = pd.read_csv('car_data.csv')  # Replace with your dataset file name

# Display the first few rows of the dataset
print(df.head())

# Get basic information about the dataset
print(df.info())

# Summary statistics
print(df.describe())

# Check for missing values
missing_values = df.isnull().sum()
print(missing_values[missing_values > 0])

# Exploratory Data Analysis (EDA)
# Histogram of car prices
plt.figure(figsize=(10, 6))
sns.histplot(df['Price'], bins=30, kde=True)
plt.title('Distribution of Car Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()

# Boxplot of Price by Transmission Type
plt.figure(figsize=(10, 6))
sns.boxplot(x='Transmission', y='Price', data=df)
```

```
plt.title('Price Distribution by Transmission Type')
plt.xlabel('Transmission Type')
plt.ylabel('Price')
plt.show()

# Correlation matrix
correlation_matrix = df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()

# Countplot for Fuel Type
plt.figure(figsize=(10, 6))
sns.countplot(x='Fuel_Type', data=df)
plt.title('Count of Cars by Fuel Type')
plt.xlabel('Fuel Type')
plt.ylabel('Count')
plt.show()

# Scatter plot of Price vs. Engine CC
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Engine CC', y='Price', data=df)
plt.title('Price vs. Engine CC')
plt.xlabel('Engine CC')
plt.ylabel('Price')
plt.show()


# Data Preprocessing
from sklearn.preprocessing import LabelEncoder

# Initialize LabelEncoder
label_encoder1 = LabelEncoder()
label_encoder2 = LabelEncoder()
label_encoder3 = LabelEncoder()
# Apply Label Encoding to the Fuel_Type column
df['Fuel_Type_Encoded'] = label_encoder1.fit_transform(df['Fuel_Type'])
df['Transmission_Encoded'] = label_encoder2.fit_transform(df['Transmission'])
df['Owner_Type'] = label_encoder3.fit_transform(df['Owner_Type'])

# Replace city names with corresponding distances
city_distance_map = {
    'Mumbai': 0,
    'Pune': 120,
    'Chennai': 0,
    'Coimbatore': 160,
    'Hyderabad': 350,
    'Jaipur': 600,
    'Kochi': 0,
    'Kolkata': 150,
    'Delhi': 1200,
    'Bangalore': 350,
    'Ahmedabad': 80
}

# Replace city names with corresponding distances
df['Location'] = df['Location'].replace(city_distance_map)

#Normality of data
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt

# Assuming you have the dataset loaded in df
# List of numerical columns
numerical_columns = ['Year', 'Kilometers_Driven', 'Engine CC', 'Power', 'Seats',
'Mileage Km/L', 'Price']

# Convert columns to numeric to ensure all values are valid
for column in numerical_columns:
    df[column] = pd.to_numeric(df[column], errors='coerce')  # Convert to
numeric and handle errors

# Create subplots: Adjust rows and cols depending on how many variables you
have
n_cols = 3  # Number of columns for subplots
n_rows = -(-len(numerical_columns) // n_cols)  # Calculate number of rows
needed

fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))  # Create subplots
axes = axes.flatten()  # Flatten to easily iterate over

# Plotting Q-Q plots for each numerical variable in the respective subplot
for i, column in enumerate(numerical_columns):
    stats.probplot(df[column].dropna(), dist="norm", plot=axes[i])  # Drop missing values for plotting
    axes[i].set_title(f'Q-Q Plot for {column}')
    axes[i].grid(True)

# Remove empty subplots if there are any
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

# Adjust layout
plt.tight_layout()
plt.show()

# Get feature importance
importances = model.feature_importances_

# Create a DataFrame to display feature names alongside their importance
feature_importance_df = pd.DataFrame({
    'Feature': X_train.columns,
    'Importance': importances
}).sort_values(by='Importance', ascending=False)

print(feature_importance_df)

#OUTLIERS DETECTION
# Plot boxplots for numerical columns to visualize outliers
plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_columns, 1):
    plt.subplot(4, 3, i)
    sns.boxplot(x=df[col])
    plt.title(f"Boxplot of {col}")
plt.tight_layout()
plt.show()

#RECURSIVE FEATURE SELECTION
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE
# Initialize a LinearRegression model
model = LinearRegression()

# Use RFE with Linear Regression
rfe = RFE(estimator=model, n_features_to_select=5)
```

```
# Fit the RFE model
rfe.fit(X_train_scaled, y_train)

# Get feature rankings
ranking = rfe.ranking_

# Display the selected features and their rankings
feature_ranking = pd.DataFrame({'Feature': X.columns, 'Ranking': ranking})
print(feature_ranking)

#MODEL BUILDING
#RANDOM FOREST
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Assuming you have the dataset loaded in df

# Define the feature set (X) and the target variable (y)
X = df.drop(['Price', 'Name', 'Manufacturer', 'Fuel_Type', 'Transmission',
            'Owner_Type', 'Fuel_Type_Encoded', 'Transmission_Encoded', 'Seats'],
axis=1)
y = df['Price']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize and train the Random Forest Regressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
rmse = mean_squared_error(y_test, y_pred, squared=False)  # RMSE
calculation
r_squared = r2_score(y_test, y_pred)  # R-squared calculation

# Print evaluation metrics
print("Model Evaluation Metrics:")
print("---------------------------")
print("RMSE:", rmse)
print("R-squared:", r_squared)

# Create a DataFrame for actual vs. predicted prices
results_df = pd.DataFrame({'Actual Price': y_test, 'Predicted Price': y_pred})

# Optionally, save the results to a CSV file for hard copy
results_df.to_csv('actual_vs_predicted_prices.csv', index=False)

# Display first few rows of the results
print("\nActual vs. Predicted Prices:")
print(results_df.head())

# Summary statistics of predictions
print("\nSummary of Predictions:")
print(results_df.describe())
```

LINEAR REGRESSION

```
from sklearn.linear_model import LinearRegression
X = df.drop(['Price', 'Name', 'Manufacturer', 'Fuel_Type', 'Transmission',
'Kilometers_Driven','Engine CC','Seats'], axis=1)
y = df['Price']
# Initialize and train the Linear Regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Make predictions and evaluate
y_pred_lr = lr_model.predict(X_test)

# Evaluate the model
print("Linear Regression RMSE:", mean_squared_error(y_test, y_pred_lr,
squared=False))
print("Linear Regression R-squared:", r2_score(y_test, y_pred_lr))

# Create a DataFrame for actual vs. predicted prices
results_lr_df = pd.DataFrame({'Actual Price': y_test, 'Predicted Price':
y_pred_lr})

# Optionally, save the results to a CSV file for hard copy
results_lr_df.to_csv('actual_vs_predicted_prices_lr.csv', index=False)

# Display first few rows of the results
```

```
print("\nActual vs. Predicted Prices (Linear Regression):")
print(results_lr_df.head())

# Summary statistics of predictions
print("\nSummary of Linear Regression Predictions:")
print(results_lr_df.describe())

#XG-BOOST
import pandas as pd
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Assuming you have the dataset loaded in df

# Define the feature set (X) and the target variable (y)
X = df.drop(['Price', 'Name', 'Manufacturer', 'Fuel_Type', 'Transmission',
            'Owner_Type', 'Fuel_Type_Encoded', 'Transmission_Encoded', 'Seats'],
axis=1)
y = df['Price']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Initialize and train the XGBoost Regressor
xgb_model = XGBRegressor(n_estimators=100, random_state=42)
xgb_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred_xgb = xgb_model.predict(X_test)

# Evaluate the model
xgb_rmse = mean_squared_error(y_test, y_pred_xgb, squared=False)  # RMSE
calculation
xgb_r_squared = r2_score(y_test, y_pred_xgb)  # R-squared calculation

# Print evaluation metrics
print("XGBoost Model Evaluation Metrics:")
print("---------------------------------")
print("RMSE:", xgb_rmse)
print("R-squared:", xgb_r_squared)
```

```
# Create a DataFrame for actual vs. predicted prices
results_xgb_df = pd.DataFrame({'Actual Price': y_test, 'Predicted Price':
y_pred_xgb})

# Optionally, save the results to a CSV file for hard copy
results_xgb_df.to_csv('actual_vs_predicted_prices_xgb.csv', index=False)

# Display first few rows of the results
print("\nActual vs. Predicted Prices (XGBoost):")
print(results_xgb_df.head())

# Summary statistics of predictions
print("\nSummary of XGBoost Predictions:")
print(results_xgb_df.describe())

# Chi-Square Test Example (Fuel Type vs. Owner Type)
import pandas as pd
from scipy.stats import chi2_contingency

# Create a contingency table between Owner_Type and Fuel_Type
contingency_table = pd.crosstab(df['Owner_Type'], df['Fuel_Type'])

# Perform the Chi-Square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Output the results
print("Chi-Square Statistic:", chi2)
print("P-value:", p)
print("Degrees of Freedom:", dof)
print("Expected Frequencies:\n", expected)

# Interpretation
if p < 0.05:
   print("There is a significant association between 'Owner_Type' and
'Fuel_Type'.")
else:
   print("There is no significant association between 'Owner_Type' and
'Fuel_Type'.")

#Mann-Whitney Test
import pandas as pd
from scipy.stats import mannwhitneyu
```

```
# Assuming 'df' is your DataFrame and 'Price' and 'Transmission' are columns in the DataFrame.

# Group the data by Transmission
manual_prices = df[df['Transmission'] == 'Manual']['Price']
automatic_prices = df[df['Transmission'] == 'Automatic']['Price']

# Perform the Mann-Whitney U test
stat, p_value = mannwhitneyu(manual_prices, automatic_prices)

print(f"Mann-Whitney U Statistic: {stat}")
print(f"P-value: {p_value}")
```