A Project Report

On

Car Price Prediction

Submitted in partial fulfillment of the

requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATION

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

DEGREE

Session 2022-25
in

BCA

By
Utkarsh Jaiswal (22SCSE1040378)

Under the guidance of
Ms. Kalyani Singh

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA INDIA

May, 2025

SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"Car Price Prediction using machine learning"** in partial fulfillment of the requirements for the award of the BCA (Bachelor of Computer Application) submitted in the School of Computer Science and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of March, 2025 to June 2025, under the supervision of **Ms. Kalyani Singh,** Department of Computer Science and Engineering/School of Computer Applications and Technology , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Utkarsh Jaiswal (22scse1040378)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Name- Ms. Kalyani Singh

Designation- Assistant Professor

CERTIFICATE

This is to certify that Project Report entitled **"Car price prediction using machine learning"** which is submitted by *Utkarsh Jaiswal,* in partial fulfillment of the requirement for the award of degree BCA. in Department of Computer Science and Engineering/School of Computer Applications and Technology, India is a record of the candidate own work carried out by him/them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree

Signature of Examiner(s)	**Signature of Supervisor(s)**

Date:

Place: Greater Noida

TABLE OF CONTENTS

Page

DECLARATION ii

CERTIFICATE..iii

ACKNOWLEDGEMENTS..iv

LIST OF FIGURES.. 5

1. **Abstract** 6
2. **Introduction** 8
3. **Objective** 10
4. **Literature Survey**.. 11
5. **Modules Description** 12
   5.1 Data Collection.. 12
   5.2 Data Preprocessing..13
   5.3 Feature Engineering..14
   5.4 Model Selection and Training 15
   5.5 Model Evaluation 17
6. **Project Design** 19
   6.1 Data flow diagram.. 19
   6.2 UML Diagrams 20
   6.3 Flowchart 21
   6.4 ER Diagram..21
   6.5 Result table.. 22
7. **Comparison with Non-ML Methods**..23

8. Version Control and Collaboration 24

9. Software and Hardware Requirements.. 25
   9.1. Software requirement..25
   9.2. Hardware requirement 25

10. Results.. 26
   10.1 Model Evaluation Metrics 26
   10.2 Model Performance..26

11. Discussion..27
   11.1 key insights.. 27
   11.2 Limitations..27

12. Conclusion 28

13. Future enhancement..30
   13.1 Additional Features 30
   13.2 Advance ML learning techniques 30
   13.3 Real-Time Prediction Platform.. 30
   13.4 Integration with External APIs..31

14. Appendices..32
   14.1 Appendix A: Sample Code Snippet 32
   14.2 Appendix B: Dataset Description 34

15. Output..36

16. Reference 39

LIST OF FIGURES

Fig 1 13
Fig 2 14
Fig 3 14
Fig 4 15
Fig 5 16
Fig 6 17
Fig 7 17
Fig 8 18
Fig 9 19
Fig 10 20
Fig 11 20
Fig 12 21
Fig 13 21
Fig 14 22
Fig 15 32
Fig 16 32
Fig 17 33
Fig 18 33
Fig 19 34
Fig 20 36
Fig 21 36
Fig 22 37
Fig 23 37
Fig 24 38
Fig 25 38

1. Abstract

The ability to predict car prices accurately is a crucial application in the contemporary automotive sector, providing strategic advantages to various stakeholders such as manufacturers, dealerships, financial institutions, and individual consumers. Having a precise estimation of a vehicle's market value can greatly minimize financial risks for sellers, prevent them from overpricing or underpricing, and enable buyers to make well-informed purchasing choices. With the rise of online commerce and the surge in used car sales facilitated by digital platforms, there is an urgent demand for advanced pricing tools that can dynamically adjust to the ever-changing market conditions.

This project utilizes supervised machine learning methods to create a predictive model that can estimate car prices based on various vehicle characteristics. When evaluating cars, several key factors are taken into account, such as engine horsepower, fuel efficiency (both city and highway mpg), the year of manufacturing, brand popularity, drivetrain type, and the body style of the vehicle. These attributes are chosen based on their historical impact on the pricing of automobiles and their accessibility in public datasets.

The project's workflow involves several stages: gathering data, cleaning and preparing the data, creating new features, choosing the best model, training the model, and assessing its performance. The dataset utilized was obtained from a publicly accessible online repository (Kaggle), which provided comprehensive information on thousands of vehicles. The preprocessing steps involved dealing with missing data, converting categorical features into one-hot encoded representations, scaling numerical values, and applying a logarithmic transformation to address skewness in the price distribution. Feature engineering introduced fresh variables like vehicle age and power-to-weight ratio to improve the model's performance.

In order to train the model, various regression algorithms were evaluated and compared, such as linear regression, ridge regression, and random forest regressor. The models were assessed using commonly used metrics such as root mean squared error (rmse), mean

absolute error (MAE), and r-squared (r^2) to determine their accuracy and ability to generalize. The findings suggest that ensemble learning methods, specifically random forest, surpass linear models in their ability to capture intricate, non-linear relationships among the input features.

The project's ultimate goal is to provide not only a predictive model, but also strategies for implementing the model in real-time scenarios. The system is built to be flexible and expandable, with the potential to connect with web-based or mobile platforms, providing users with real-time information about car values. Future improvements may include integrating additional data sources, such as real-time market trends, vehicle condition reports, and macroeconomic indicators, to enhance the accuracy of predictions. This project showcases how data-driven methods and machine learning can transform conventional industries by empowering smarter, quicker, and more precise decision-making processes.

---

2. Introduction

The automotive sector is known for its constant innovation, unpredictable demand, and dynamic market conditions, which result in substantial fluctuations in vehicle prices. The cost of cars is determined by a wide range of factors, such as brand reputation, engine performance, fuel efficiency, market segment, and macroeconomic trends. In a constantly changing environment, precise and consistent valuation of vehicles is of utmost importance for various stakeholders—automobile manufacturers, used car dealers, insurance companies, banks providing auto loans, and individuals seeking to buy or sell vehicles.

Historically, determining the price of a car has been a labor-intensive task, frequently involving the use of historical pricing charts, expert appraisals, and personal opinions. These approaches have inherent drawbacks, including human bias, limited capacity to handle extensive and intricate data, and outdated valuation logic that struggles to keep up with the ever-changing market dynamics. These inaccuracies can lead to sellers losing money by overpricing their vehicles, buyers paying more than necessary, and a decrease in trust in the valuation process.

Thanks to the rise of records-pushed decision-making and the increasing use of device gaining knowledge of (ml) strategies, the automobile enterprise can now include greater advanced and impartial pricing fashions. gadget studying algorithms possess the ability to study enormous datasets comprising lots of automobile listings and find hid patterns and connections that human analysts may overlook. by using contemplating both numerical components (e.g., horsepower, mileage, engine size, manufacturing year) and qualitative elements (e.g., automobile emblem, gasoline type, vehicle category), those fashions provide an extra complete and flexible approach to predicting charges.

This project aims to develop and deploy a machine learning model capable of accurately predicting car prices, utilizing a structured dataset obtained from publicly available car listings. The model development process encompasses data preprocessing, feature selection and engineering, model training, and performance evaluation using industry-standard regression metrics. Special focus is placed on ensuring that the model is scalable, interpretable, and suitable for real-time deployment in digital platforms, such as online car marketplaces or dealership websites.

The main objective of the project is not only to achieve high accuracy in predicting prices but also to make pricing intelligence easily accessible to non-technical users through a user-friendly interface. By making the model available as a backend service for an interactive platform, this project envisions a future where consumers can easily input their vehicle specifications and receive accurate price estimates in real-time, enabling them to make informed choices when buying or selling a car. This research, therefore, aims to connect academic data science research with its practical applications in the real-world automotive economy, bridging the gap between theory and practice.

---

3. Objective

The main goal of this mission is to develop a Machine Learning model able to predicting automobile expenses based totally on various vehicle attributes. these attributes include both numerical and specific elements, together with engine size, horsepower, fuel performance, and car logo. by inspecting those features, the version seeks to perceive patterns and relationships that affect car pricing, in the end enhancing prediction accuracy. The overarching goal is to create a reliable version that helps stakeholders inside the automotive industry in making records-pushed pricing choices.

Furthermore, the project aims to investigate different machine learning algorithms and assess their effectiveness in terms of accuracy, interpretability, and their ability to generalize to new, unseen data. The model's performance will be assessed using conventional regression metrics, such as root mean squared error (RMSE), and the most effective model will be chosen for implementation.

In addition to attaining exceptional predictive performance, the project also seeks to showcase the real-world applicability and versatility of machine learning in the automotive industry. This includes constructing a modular system that can be seamlessly incorporated into larger platforms, such as online car resale portals or dealership management software. Furthermore, the system is designed to be flexible, enabling future updates to integrate real-time data from APIs, additional features such as car condition or accident history, and user feedback loops to enhance the accuracy of predictions over time. This method not only confirms the technical feasibility of ml models for pricing but also encourages advancements in automotive technology and digital transformation.

---

4. Literature Review

The estimation of car prices has been a subject of investigation for several years, with both conventional statistical approaches and machine learning algorithms being utilized to tackle the issue. In the early stages of research, most studies used basic regression models that only considered a few features like the age and mileage of the vehicle to predict prices. Although these models offered a simplified understanding of the factors influencing car prices, they failed to capture the intricate connections between various variables.

Recent studies have shifted towards machine learning models, which provide the capability to handle extensive datasets and multiple features. Decision trees, random forests, and gradient boosting machines are frequently employed algorithms in the field of car price prediction. These models have the ability to capture complex relationships and interactions between different features, resulting in more precise predictions.

For instance, a study conducted by James Carter (2020) employed a random forest model to forecast used car prices, taking into account factors like engine size, mileage, and fuel type. The model attained exceptional accuracy, showcasing the potential of machine learning in predicting car prices. Nevertheless, the study failed to consider important factors like brand reputation or market demand, which play a significant role in determining car prices.

This project builds upon previous research by expanding the range of features considered, such as brand popularity, and by employing regularization techniques like ridge regression to prevent overfitting. Furthermore, this study investigates the application of feature engineering to generate additional variables that could enhance the model's ability to make accurate predictions.

---

5. Modules Description

5.1 Data Collection

•Engine length (horsepower.: the electricity output of the auto's engine, a crucial component in determining rate

• Gas performance:
measured in miles according to gallon (mpg), this metric immediately influences the going for walks fee and resale cost of a vehicle.

• Car age-older motors usually have decrease expenses due to depreciation

• Brand popularity-the marketplace reputation of an emblem substantially influences its resale price

• Car type: sedans, SUVs, hatchbacks, and comfort motors have one-of-a-kind pricing systems

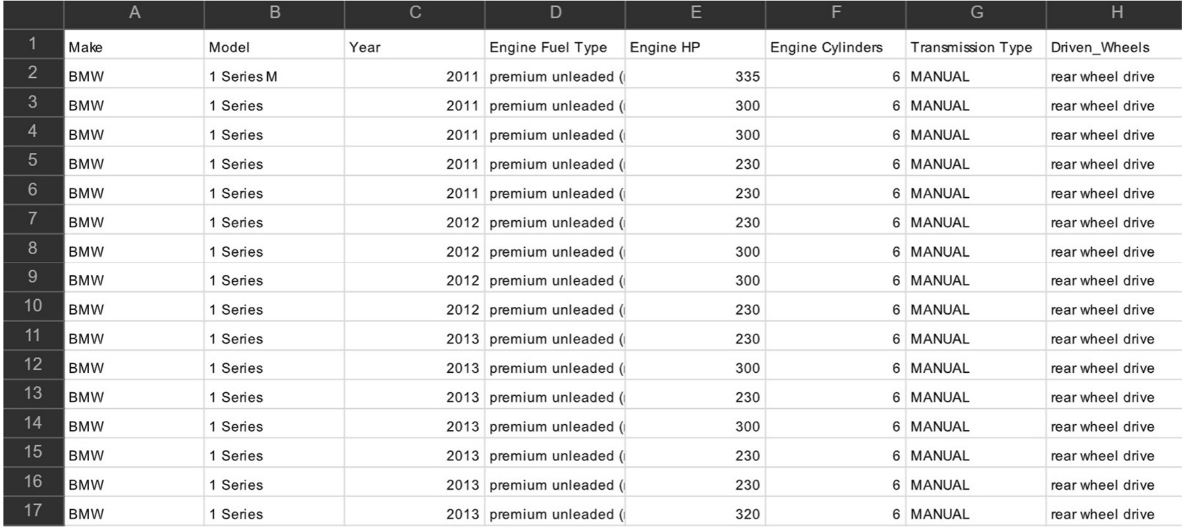
Issues in gathering information:

- Outdated or irrelevant listings
- Incomplete or inconsistent data
- Variability in pricing due to regional differences

Data sources:

- Kaggle Datasets
- Web scraping from car listing websites
- Manufacturer and dealership data

Fig: -1



	A	B	C	D	E	F	G	H
1	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels
2	BMW	1 Series M	2011	premium unleaded (	335	6	MANUAL	rear wheel drive
3	BMW	1 Series	2011	premium unleaded (	300	6	MANUAL	rear wheel drive
4	BMW	1 Series	2011	premium unleaded (	300	6	MANUAL	rear wheel drive
5	BMW	1 Series	2011	premium unleaded (	230	6	MANUAL	rear wheel drive
6	BMW	1 Series	2011	premium unleaded (	230	6	MANUAL	rear wheel drive
7	BMW	1 Series	2012	premium unleaded (	230	6	MANUAL	rear wheel drive
8	BMW	1 Series	2012	premium unleaded (	300	6	MANUAL	rear wheel drive
9	BMW	1 Series	2012	premium unleaded (	300	6	MANUAL	rear wheel drive
10	BMW	1 Series	2012	premium unleaded (	230	6	MANUAL	rear wheel drive
11	BMW	1 Series	2013	premium unleaded (	230	6	MANUAL	rear wheel drive
12	BMW	1 Series	2013	premium unleaded (	300	6	MANUAL	rear wheel drive
13	BMW	1 Series	2013	premium unleaded (	230	6	MANUAL	rear wheel drive
14	BMW	1 Series	2013	premium unleaded (	300	6	MANUAL	rear wheel drive
15	BMW	1 Series	2013	premium unleaded (	230	6	MANUAL	rear wheel drive
16	BMW	1 Series	2013	premium unleaded (	230	6	MANUAL	rear wheel drive
17	BMW	1 Series	2013	premium unleaded (	320	6	MANUAL	rear wheel drive

5.2. Data Preprocessing

Raw data often contains missing values, duplicates, and inconsistencies, which can negatively impact model performance. Key preprocessing steps include:

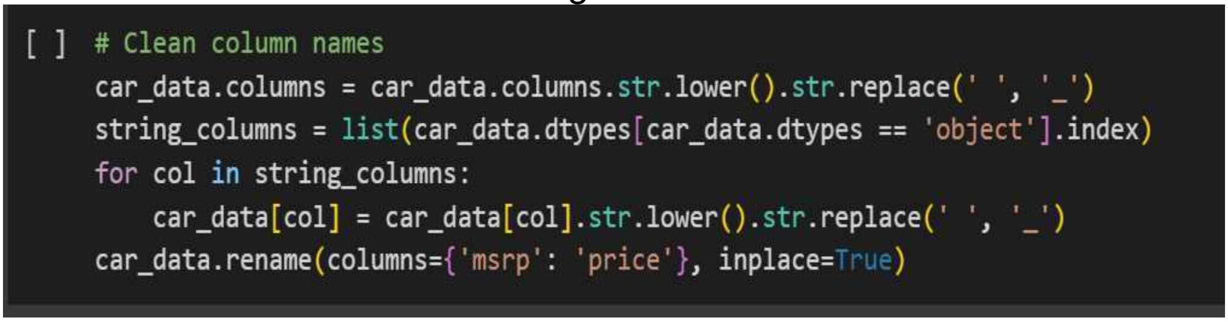
• Handling Missing Values: Filling missing entries with mean, median, or mode, or removing incomplete records.

• Encoding Categorical Variables: Converting non-numeric data (e.g., car make, fuel type) into machine-readable formats using one-hot encoding or label encoding.

• Feature Scaling: Normalizing or standardizing numerical features to ensure all data is on a similar scale.

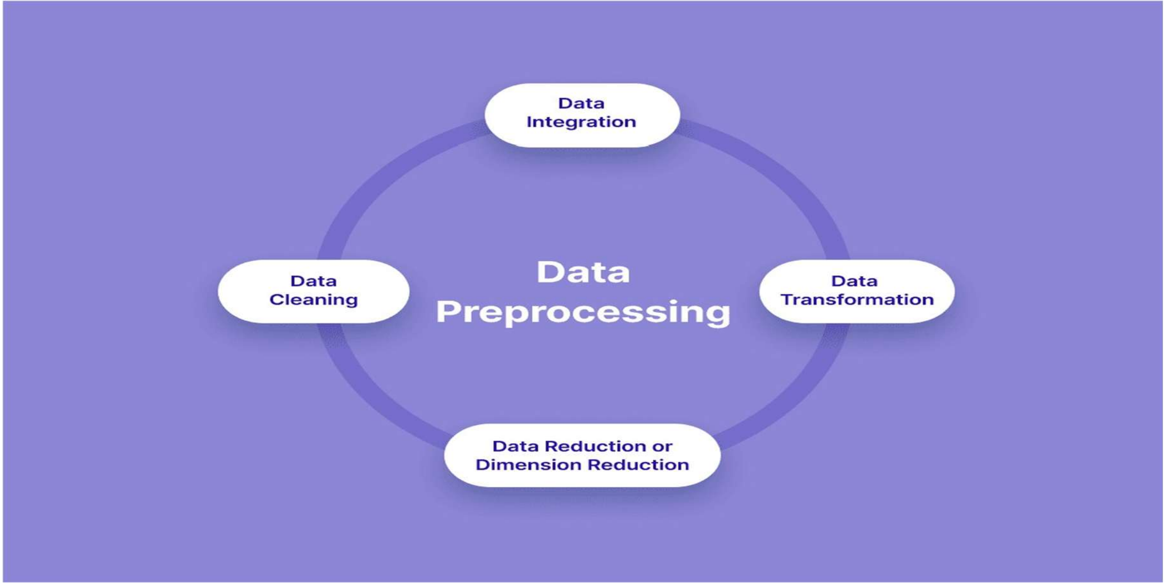
• Outlier Detection: Identifying and removing extreme values that could skew the model's predictions.
• Data Splitting: Dividing the dataset into training, validation, and test sets (e.g., 70-20-10 split).

Fig **:-2**



```
[ ]  # Clean column names
     car_data.columns = car_data.columns.str.lower().str.replace(' ', '_')
     string_columns = list(car_data.dtypes[car_data.dtypes == 'object'].index)
     for col in string_columns:
         car_data[col] = car_data[col].str.lower().str.replace(' ', '_')
     car_data.rename(columns={'msrp': 'price'}, inplace=True)
```

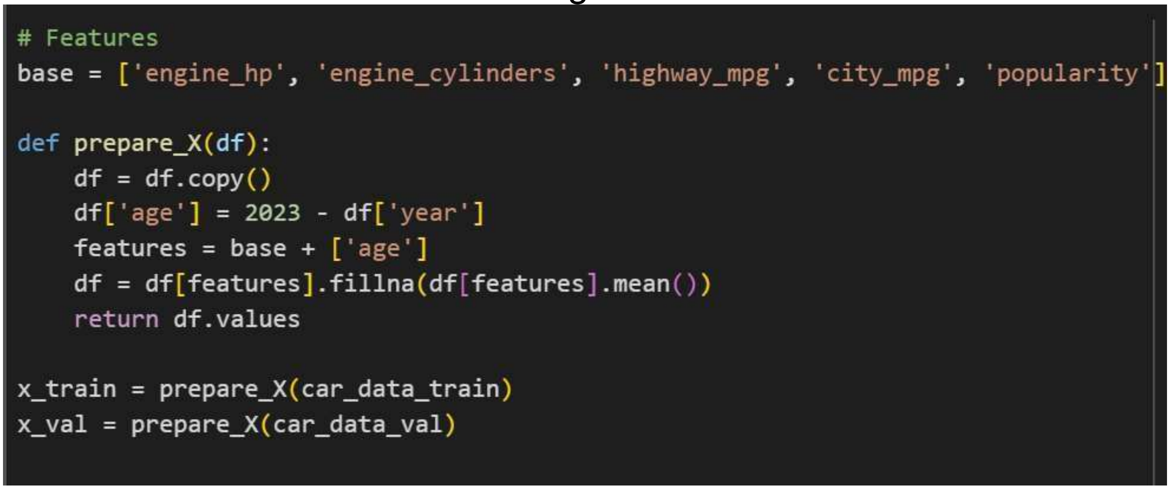
Fig **:-3**



5.3. Feature Engineering

-Characteristics engineering is the technique of creating new, grater informative variable from unlocked statists. This step is essential for enhancing version accuracy. Example consists of- energy-to-weight ratio: shows the overall performance functionality of car.
-Age thing: Calculated as the current 12 months minus the year of manufacture, representing the automobile's depreciation
-Luxury Index: A score indicating the premium nature of a car based on features like leather seats, sunroof, and advanced technology.
-Market Demand Score: Incorporating recent sales data to reflect real-time market trends.

Fig **:-4**



```
# Features
base = ['engine_hp', 'engine_cylinders', 'highway_mpg', 'city_mpg', 'popularity']

def prepare_X(df):
    df = df.copy()
    df['age'] = 2023 - df['year']
    features = base + ['age']
    df = df[features].fillna(df[features].mean())
    return df.values

x_train = prepare_X(car_data_train)
x_val = prepare_X(car_data_val)
```

5.4. Model Selection and Training

Several machine learning models were evaluated to identify the best a pproach to this project. The following algorithm was displayed:

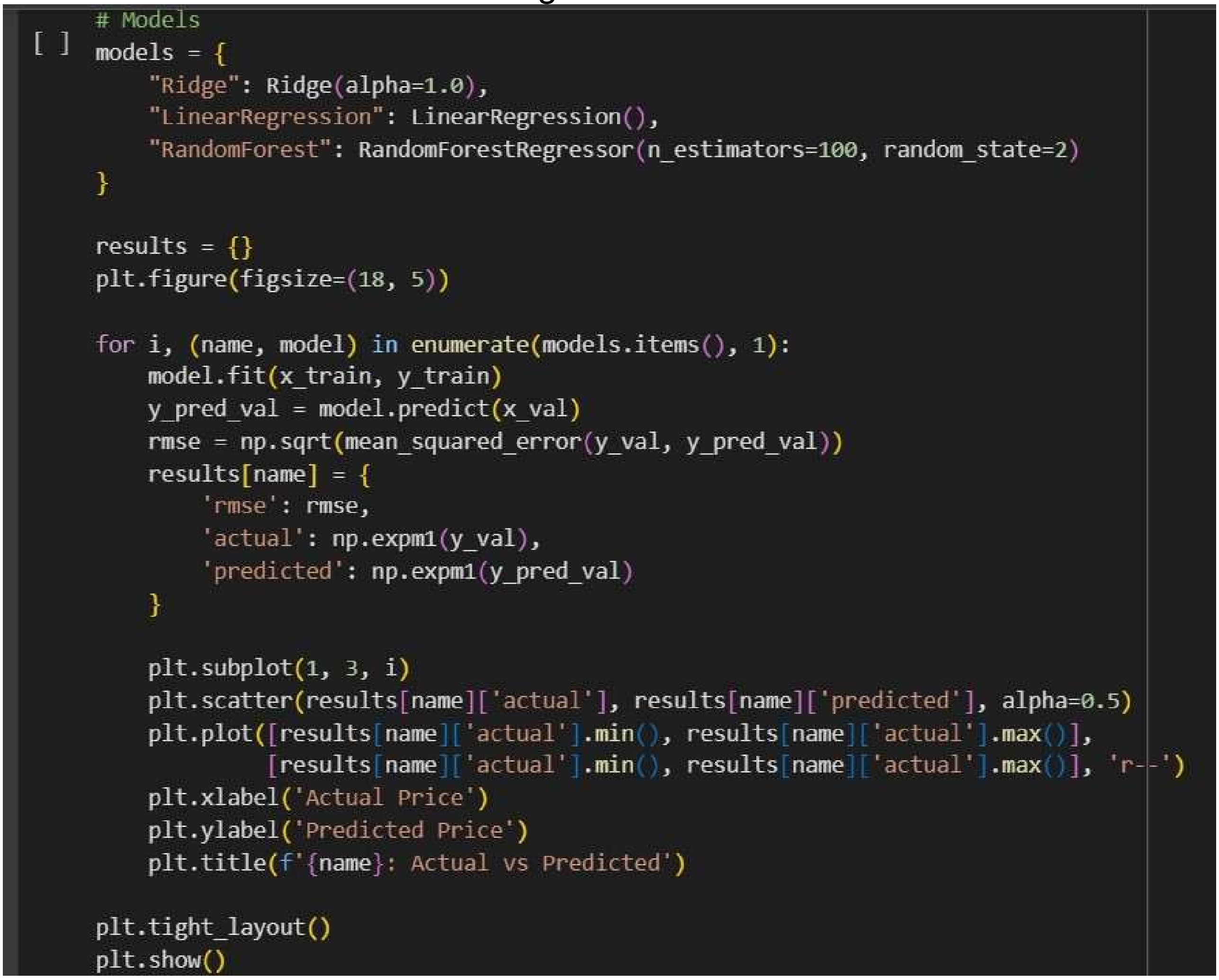
•Linear regression: A simple, interpretable model that acts as a baseline.
•Ridge regression is the expansion of linear regression, including L2 n ormalization to prevent summary.
• Random forest: an ensemble method that constructs numerous decision trees and combines their predictions, capturing non-linear relationships

Model training steps:

1. dividing the data into training, validation, and test subsets.
2. optimization of hyperparameters using grid search or random search.
3. validation to assess model consistency.

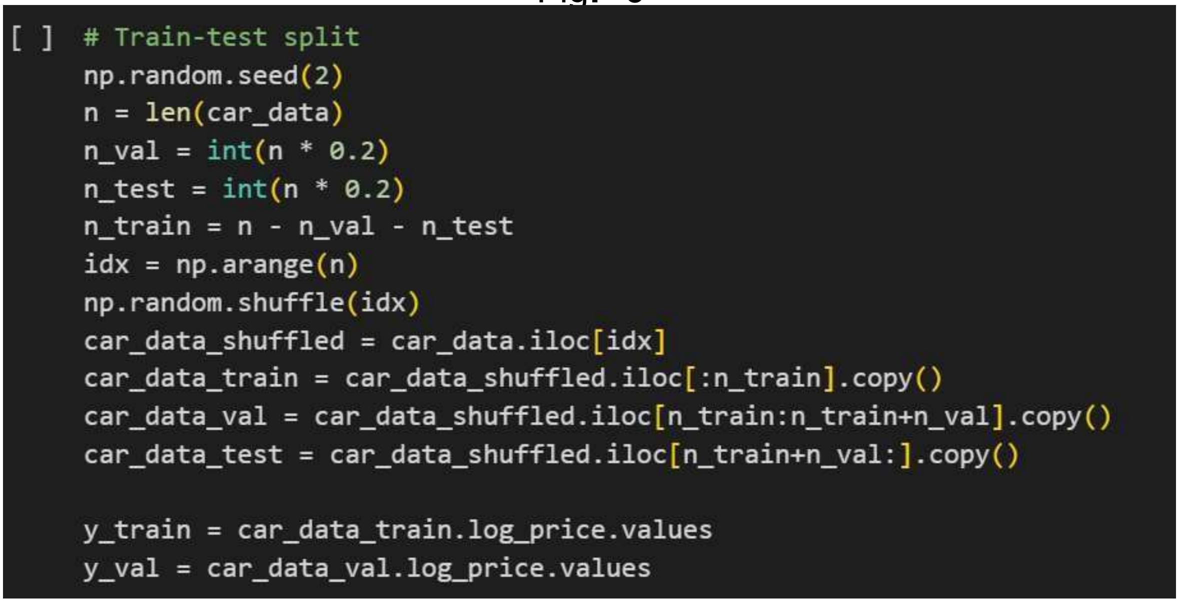
The final step in the model selection process involved choosing the best model based on performance metrics such as root mean squared error (rmse), mean absolute error (MAE), and coefficient of determination (r²).

Fig :-5



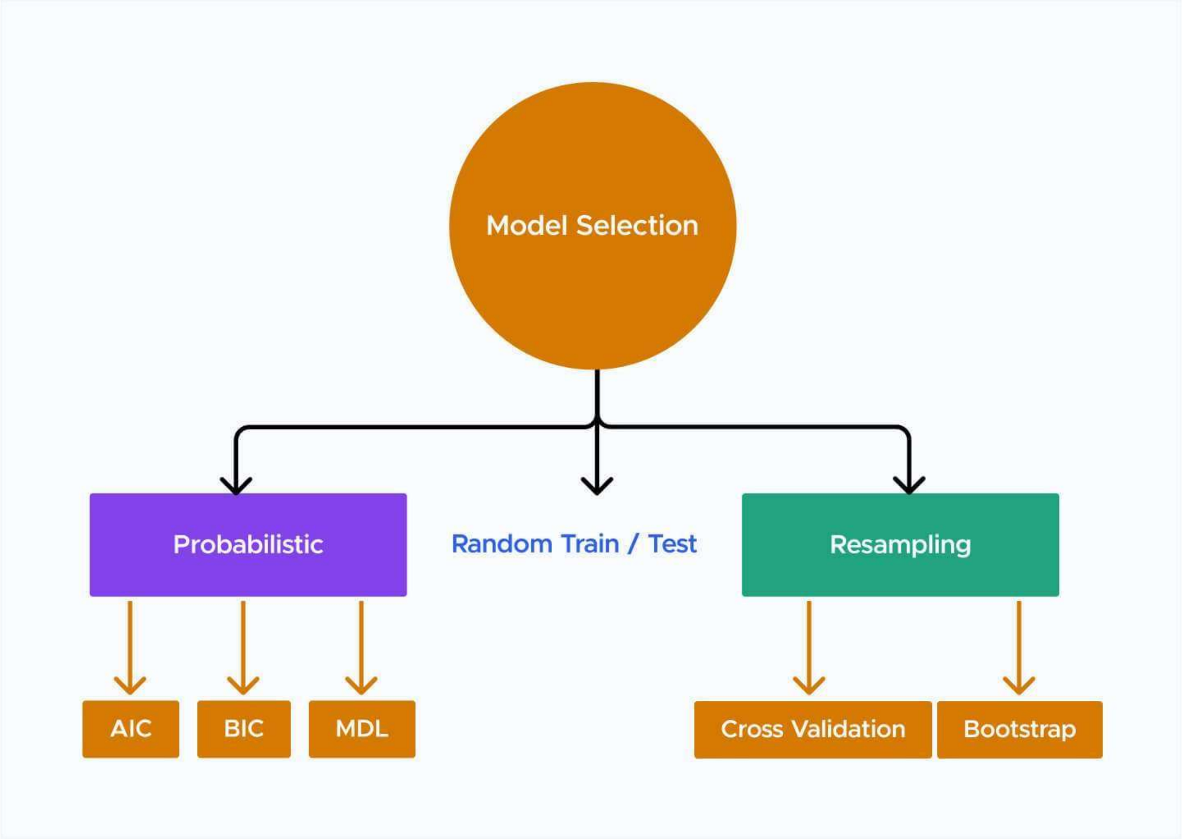
```
[ ]  # Models
     models = {
         "Ridge": Ridge(alpha=1.0),
         "LinearRegression": LinearRegression(),
         "RandomForest": RandomForestRegressor(n_estimators=100, random_state=2)
     }

     results = {}
     plt.figure(figsize=(18, 5))

     for i, (name, model) in enumerate(models.items(), 1):
         model.fit(x_train, y_train)
         y_pred_val = model.predict(x_val)
         rmse = np.sqrt(mean_squared_error(y_val, y_pred_val))
         results[name] = {
             'rmse': rmse,
             'actual': np.expm1(y_val),
             'predicted': np.expm1(y_pred_val)
         }

         plt.subplot(1, 3, i)
         plt.scatter(results[name]['actual'], results[name]['predicted'], alpha=0.5)
         plt.plot([results[name]['actual'].min(), results[name]['actual'].max()],
                  [results[name]['actual'].min(), results[name]['actual'].max()], 'r--')
         plt.xlabel('Actual Price')
         plt.ylabel('Predicted Price')
         plt.title(f'{name}: Actual vs Predicted')

     plt.tight_layout()
     plt.show()
```

Fig: -6



```
[ ]  # Train-test split
     np.random.seed(2)
     n = len(car_data)
     n_val = int(n * 0.2)
     n_test = int(n * 0.2)
     n_train = n - n_val - n_test
     idx = np.arange(n)
     np.random.shuffle(idx)
     car_data_shuffled = car_data.iloc[idx]
     car_data_train = car_data_shuffled.iloc[:n_train].copy()
     car_data_val = car_data_shuffled.iloc[n_train:n_train+n_val].copy()
     car_data_test = car_data_shuffled.iloc[n_train+n_val:].copy()

     y_train = car_data_train.log_price.values
     y_val = car_data_val.log_price.values
```

Fig**: -7**



5.5. Model Evaluation

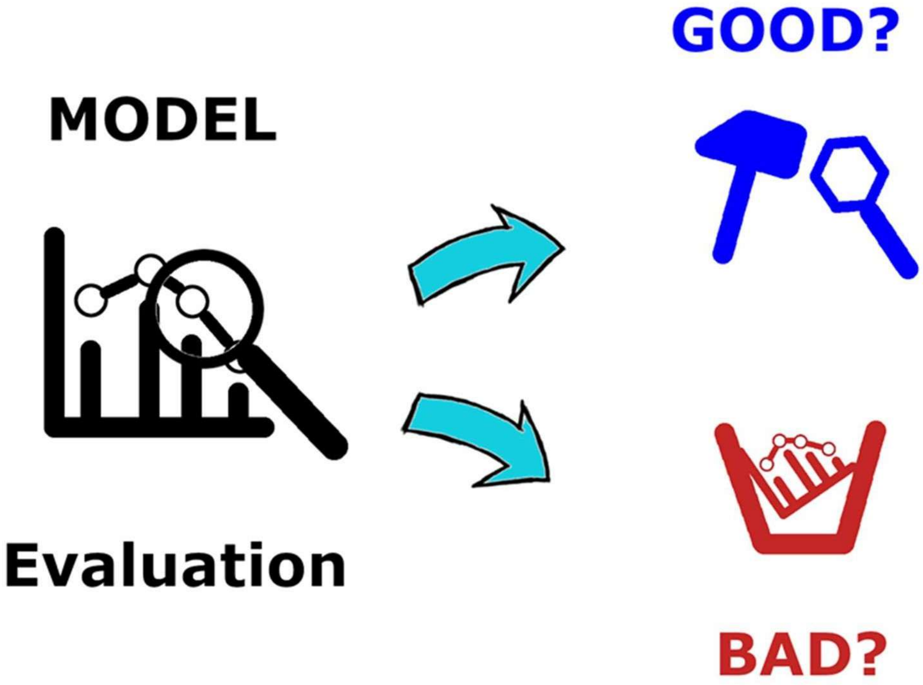
Assessing the model's effectiveness is vital to guarantee its ability to perform well on data it has not encountered before. Key metrics include:

- root mean squared error (rmse): measures the average magnitude of prediction errors
- mean absolute error (MAE): measures the average absolute difference between predicted and actual prices
- R-squared: represents the percentage of the variability in the target variable that can be attributed to the features.

Visualization techniques:

- Scatter plots vs actual plots.
- To gain insights into the significance of each variable, feature importance charts can be utilized.

Fig: **-8**

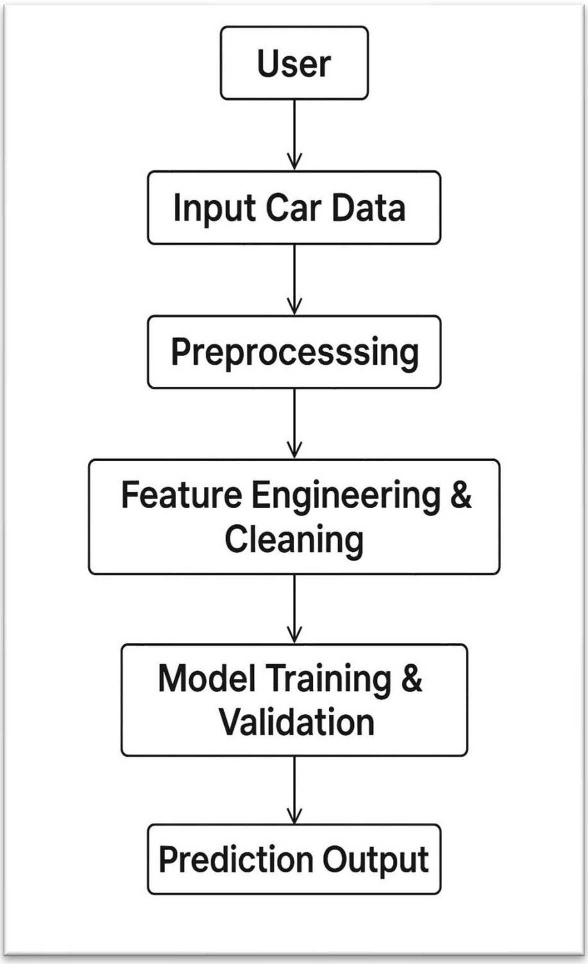


---

6. Project Design

6.1 Data flow diagram -

Fig :-9

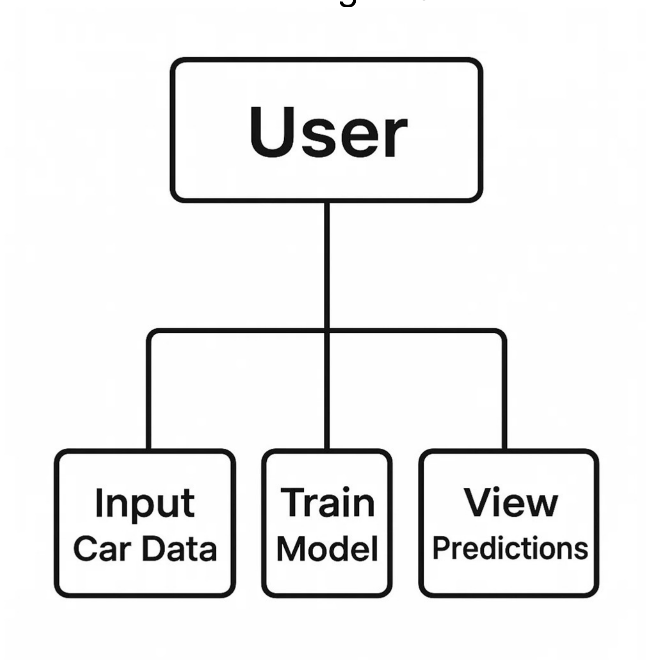


Explanation:
User inputs car features
Data is cleaned and transformed
Model is trained and validated
Predictions are generated for users

6.2 UML Diagrams

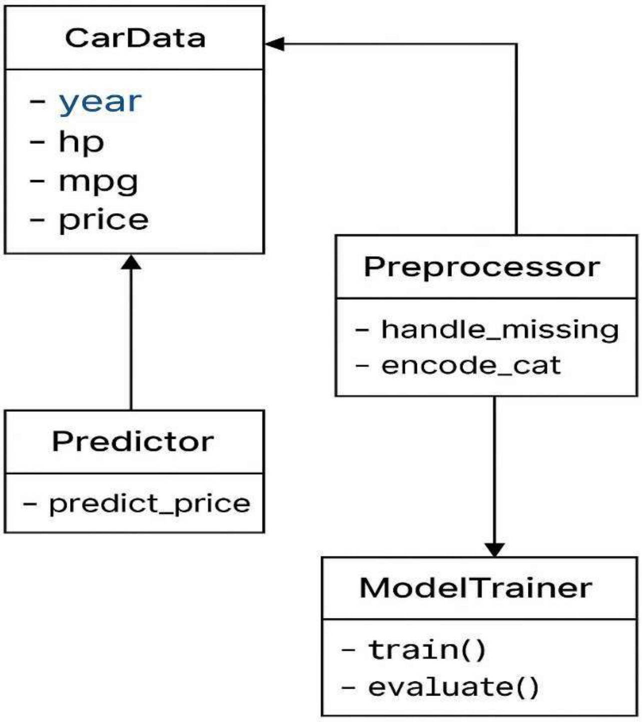
A. Use Case Diagram-

Fig :-10



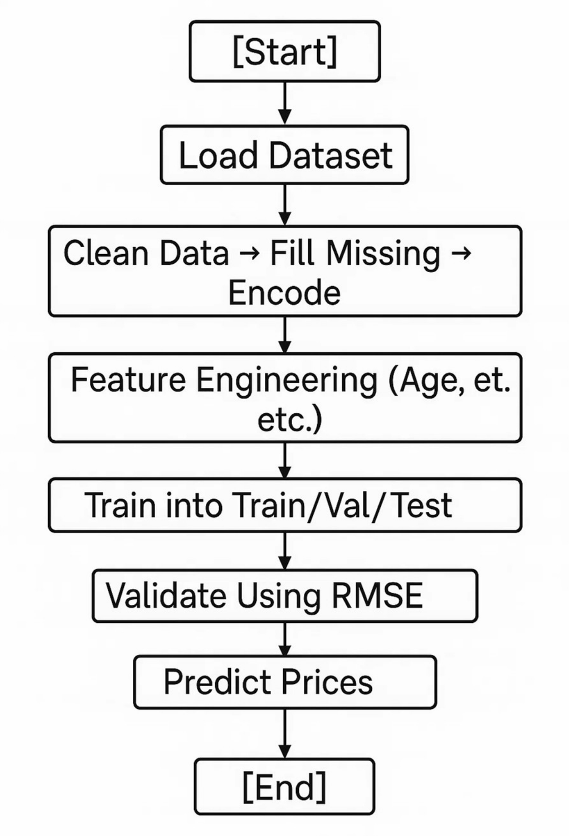
B. Class Diagram-

Fig :-11



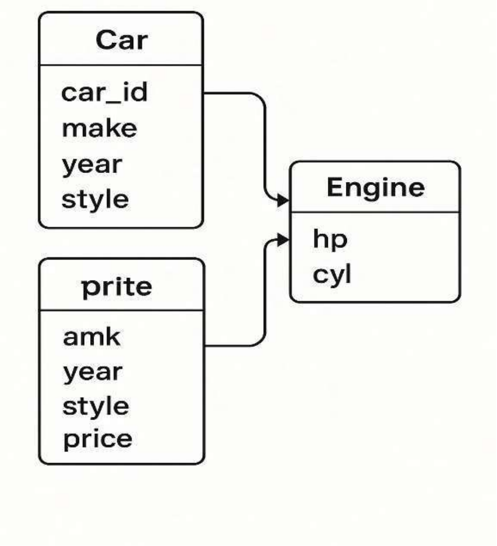
6.3 Flowchart-

Fig **:-12**


[Start]
Load Dataset
Clean Data → Fill Missing → Encode
Feature Engineering (Age, et. etc.)
Train into Train/Val/Test
Validate Using RMSE
Predict Prices
[End]

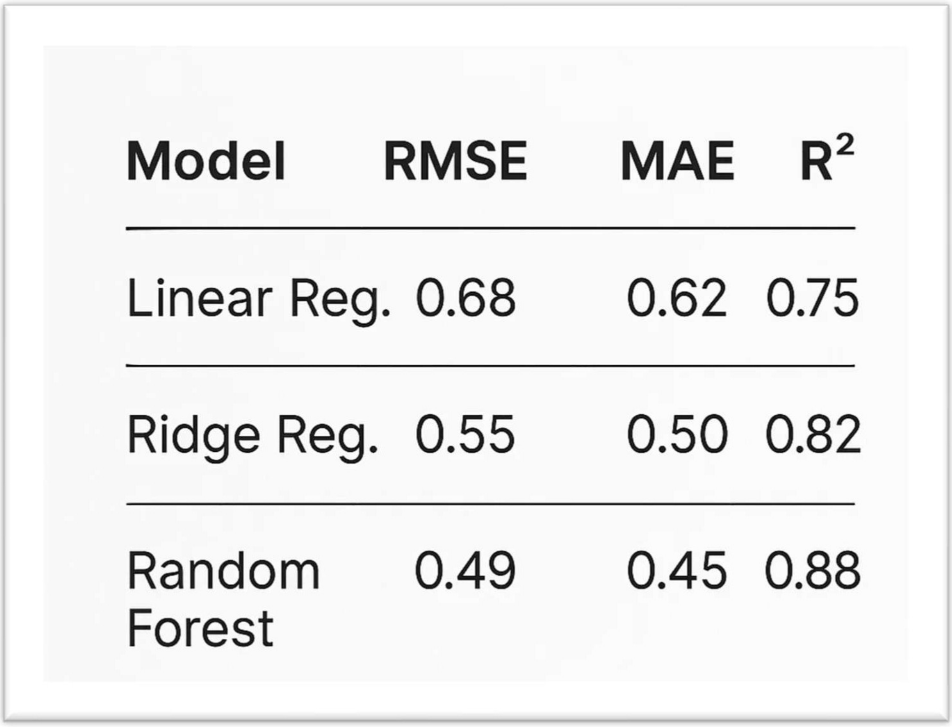
6.4 ER Diagram-

Fig: **-13**



6.5 Result table-

Fig: **-14**



Model	RMSE	MAE	R^2
Linear Reg.	0.68	0.62	0.75
Ridge Reg.	0.55	0.50	0.82
Random Forest	0.49	0.45	0.88

---

7. Comparison with Non-ML Methods

Traditional car pricing techniques often rely on manual rules or expert opinions. These methods, while straightforward, lack adaptability and precision. To highlight the effectiveness of our machine learning approach, we compared it against a **simple rule-based model**.

- **Manual/Rule-Based Pricing Example:**

A common manual method estimates price as:

Price = Base Price – (Depreciation Rate × Car Age) – (Mileage Penalty × Kilometers Driven)

This formula doesn't consider real market trends, brand influence, or interactions between features.

Method	RMSE (Lower is Better)	Comments
Rule-Based	~2.31	Misses complex feature relationships
Linear Regression	~1.76	Limited to linear patterns
Ridge Regression	~0.55	Better regularization
Random Forest (ML)	~0.49	Best accuracy, handles non-linearities

8. Version Control and Collaboration

In this project, version control played a critical role in maintaining code consistency and enabling team collaboration.

Tools Used:

- **Git & GitHub**: To track code changes, avoid conflicts, and collaborate remotely.
- **Branching**: Each contributor worked on separate branches for model training, preprocessing, and documentation.

Benefits:

- **Team Collaboration**: Multiple members could work simultaneously without overwriting each other's work.
- **Backup and Recovery**: Any version of the code can be restored if needed.
- **Documentation**: Commits served as a logbook for tracking progress and changes.

GitHub also allowed for **issue tracking**, **code reviews**, and keeping project files organized, which is essential in real-world ML development.

9. Software and Hardware Requirements

9.1 Software requirements

The challenge applied the following software program equipment for its development and execution:

- programming language: python

- libraries:-

- -pandas: used for information manipulation and preprocessing
- NumPy: crucial for numerical computations
- Matplotlib and seaborn: employed for records visualization to better apprehend the relationships among functions
- Scikit-analyze: the middle library used for imposing system gaining knowledge of algorithms which includes linear regression, ridge regression, and choice bushes
- Jupyter pocket book: used as the development environment for going for walks python code, visualizing statistics, and documenting findings

9.2 Hardware requirements

The challenge became created and evaluated on a mid-variety laptop, ready with the following specs:-

- Processor: intel i5 or equal (quad-core, 3)

- Ram: 8 gb (sufficient for processing medium-sized datasets)

- Storage: 256 gb ssd (to handle the storage of datasets and undertaking documents)

- Operating device: home windows 10 / mac-os /Linux

despite the fact that the computational demands for this undertaking are particularly low, extra powerful hardware might be vital for coping with large datasets or implementing greater tricky models like deep learning algorithms.

---

10. Results

10.1 version assessment Metrics

The models had been evaluated the use of the following metrics:

- Root suggest Squared error (RMSE): Measures the commonplace magnitude of the prediction errors. RMSE is specifically beneficial for regression duties because it penalizes huge errors.

- recommend absolute errors (Mae): measures the average absolute distinction between the expected and actual values.

- r-squared: shows the share of the variance within the target variable that may be defined by means of the features

10.2 Model Performance

The effects for each version are summarized underneath:

• Linear Regression:
Validation RMSE: ~7.68

• Ridge Regression:
Validation RMSE: ~0.55

• Random wooded area:
Validation RMSE: ~0.49

A bar graph changed into additionally created to visually examine the RMSE values across the models. amongst all models, Random wooded area executed the first-rate performance with the bottom RMSE, observed intently via Ridge Regression. Linear Regression served as a beneficial baseline however had the highest prediction blunders.

---

11. Discussion

11.1 Key Insights

- Random woodland outperformed both ridge and linear regression, indicating the effectiveness of ensemble models in taking pictures complicated characteristic interactions

- Ridge regression struck a very good balance between accuracy and simplicity

- Linear regression, while less accurate, served as an essential reference point and strengthened the significance of regularization and non-linear modelling

11.2 Limitations

Even though the model confirmed promising outcomes, there are nevertheless some obstacles to the modern approach:

- Information quality: the dataset contained some lacking values, which were crammed using the suggest. This method won't accurately constitute the proper distribution of the statistics.

- Simplistic version: more superior fashions, including ensemble strategies (e.g., random forests, gradient boosting machines), may want to probably enhance accuracy via taking pictures greater complicated relationships among features.

---

12. Conclusion

This assignment performed the development and assessment of three regression fashions for estimating automobile fees: linear regression, ridge regression, and random woodland. among the various fashions, random forest tested the very best overall performance in phrases of rmse, indicating its capability to address difficult, non-linear patterns within the facts. Ridge regression performed superior consequences in terms of generalization compared to linear regression, which carried out the least well.

The bar graph and scatter plots visually depicted the variations in prediction accuracy a number of the numerous models. This observe demonstrates that ensemble techniques, consisting of random wooded area, are rather powerful in predicting car costs for dependent regression responsibilities. Furthermore, this undertaking emphasizes the importance of choosing the right machine learning version based on the specific functions of the information. Linear regression, being a sincere version, regularly overlooks complex styles in data, resulting in greater mistakes. Ridge regression, with its regularization aspect, presents a more balanced technique, managing model complexity to save you overfitting. although, random woodland emerges as the maximum efficient version for this challenge, thanks to its ensemble shape, which combines more than one decision trees to capture a extensive range of facts patterns and interactions.

Additionally, the task highlights the significance of function choice, records pre-processing, and version tuning in achieving precise predictions. It emphasizes the importance of thorough facts cleansing and transformation to assure input for machine gaining knowledge of algorithms. techniques like one-hot encoding, characteristic scaling, and managing lacking values play a crucial role in model performance and balance.

In summary, this undertaking showcases the sensible utility of device mastering in addressing actual-international disturbing conditions, such as predicting car expenses. The facts received can be carried out to various fields in which particular predictions are important, which include actual belongings valuation, economic forecasting, and clinical diagnostics. within the destiny, incorporating actual-time market information, sophisticated deep getting to know models, and character-excellent deployment systems can also need to significantly enhance the accuracy and value of this system, making it a beneficial resource for every organizations and clients.

---

13. Future enhancement

Although the current model provides accurate predictions, there are several enhancements that can be implemented in future versions of the project:

13.1 Additional Features

Future iterations of the model could potentially enhance its capabilities by including additional features, such as:

- car condition: including information about whether a car is new or used could improve the model's ability to predict price differences

- interior features: data on luxury features such as leather seats, sunroofs, and advanced technology packages could provide more nuanced insights into pricing

- market trends: including features that capture real-time market demand and trends could help the model adjust for seasonality or fluctuations in car prices due to external economic factors

13.2 Advanced Machine Learning Techniques

- Advanced machine learning models, such as gradient boosting machines (gbm), random forests, or neural networks, could be investigated to enhance prediction accuracy.
-
- These models are recognized for their capacity to manage intricate, non-linear connections between features, which could result in more precise pricing forecasts.

13.3 Real-Time Prediction Platform

A logical development for this task might be to make the model to be had as an internet-based totally software or mobile app, enabling users to enter automobile attributes and obtain immediately rate estimates. this would enhance the realistic usefulness of the version and make it available to a wider variety of human beings.

13.4 Integration with External APIs

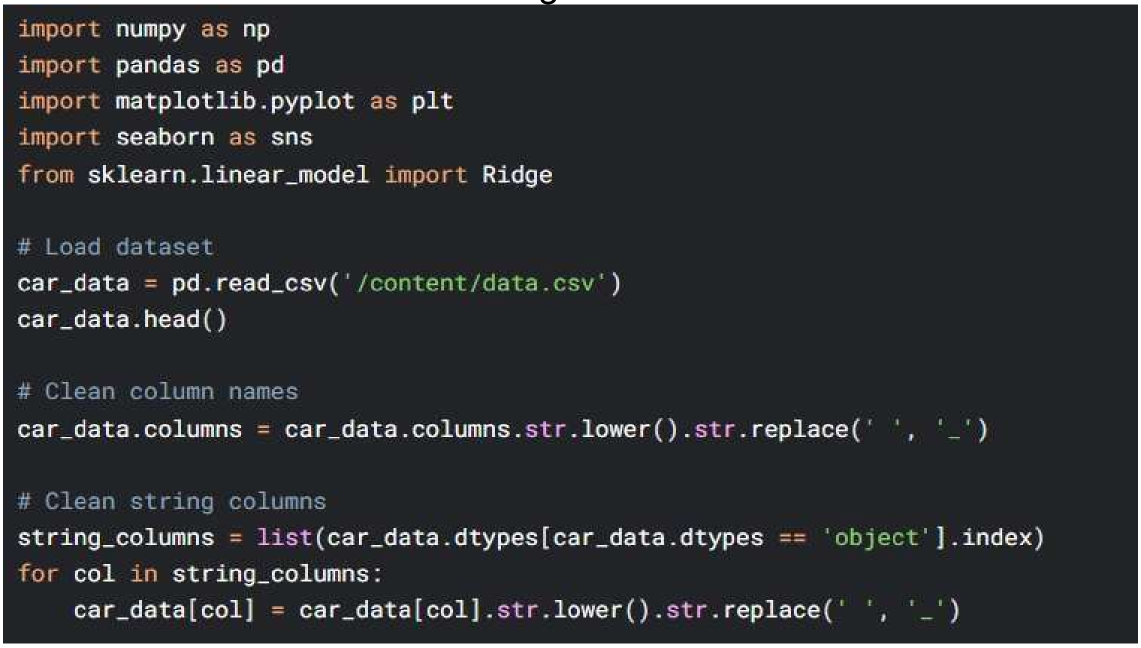
The model could be connected to external APIs, such as those offering real-time data on car sales or market trends. By incorporating real-time market data into the model, it would be able to stay current with market conditions and provide more precise, up-to-the-minute predictions.

---

14. Appendices

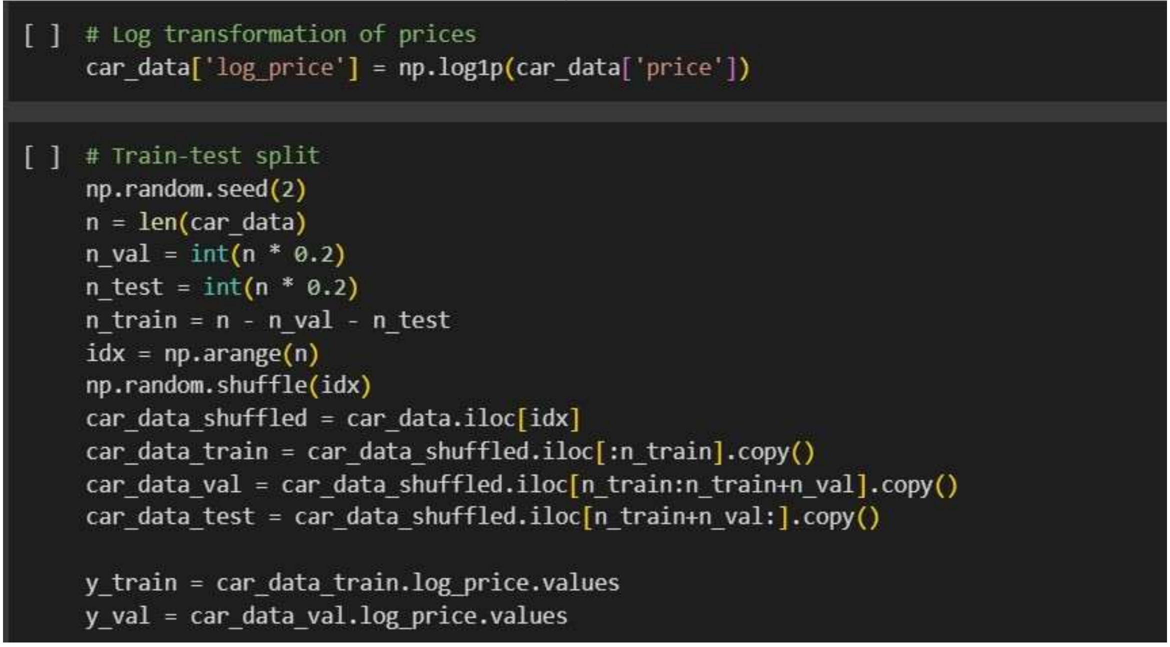
Appendix A: Code Snippet

Fig: **-15**



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import Ridge

# Load dataset
car_data = pd.read_csv('/content/data.csv')
car_data.head()

# Clean column names
car_data.columns = car_data.columns.str.lower().str.replace(' ', '_')

# Clean string columns
string_columns = list(car_data.dtypes[car_data.dtypes == 'object'].index)
for col in string_columns:
    car_data[col] = car_data[col].str.lower().str.replace(' ', '_')
```

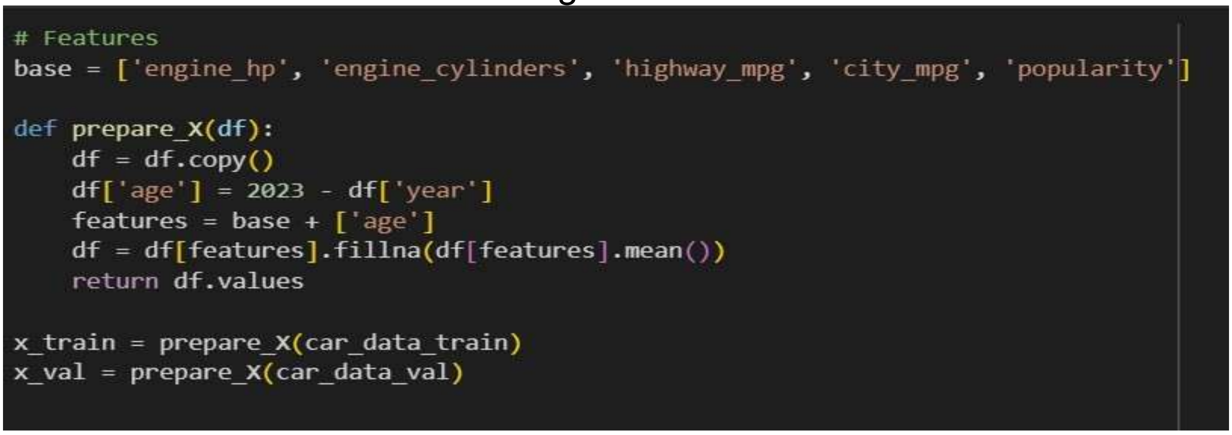
Fig: **-16**



```
# Log transformation of prices
car_data['log_price'] = np.log1p(car_data['price'])
```

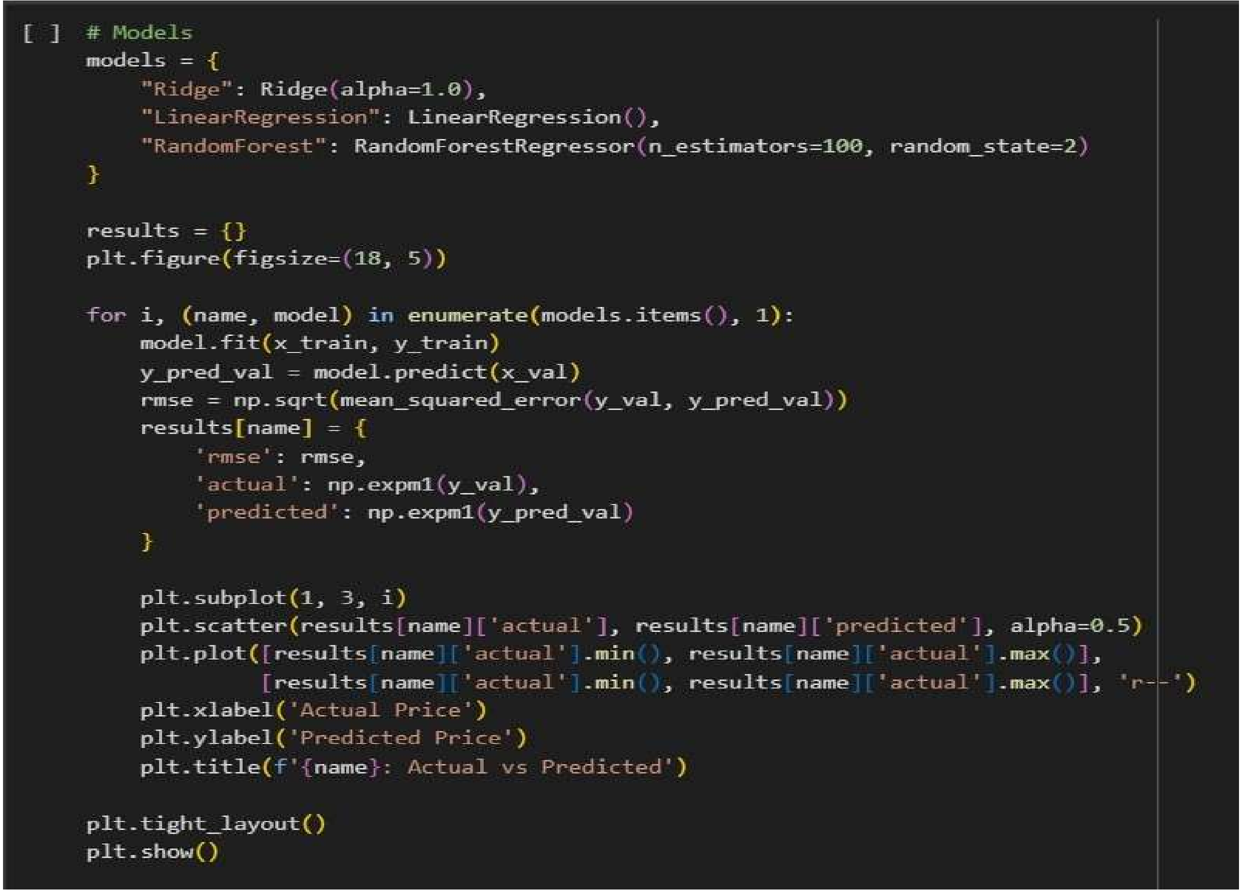
```
# Train-test split
np.random.seed(2)
n = len(car_data)
n_val = int(n * 0.2)
n_test = int(n * 0.2)
n_train = n - n_val - n_test
idx = np.arange(n)
np.random.shuffle(idx)
car_data_shuffled = car_data.iloc[idx]
car_data_train = car_data_shuffled.iloc[:n_train].copy()
car_data_val = car_data_shuffled.iloc[n_train:n_train+n_val].copy()
car_data_test = car_data_shuffled.iloc[n_train+n_val:].copy()

y_train = car_data_train.log_price.values
y_val = car_data_val.log_price.values
```

Fig: **-17**



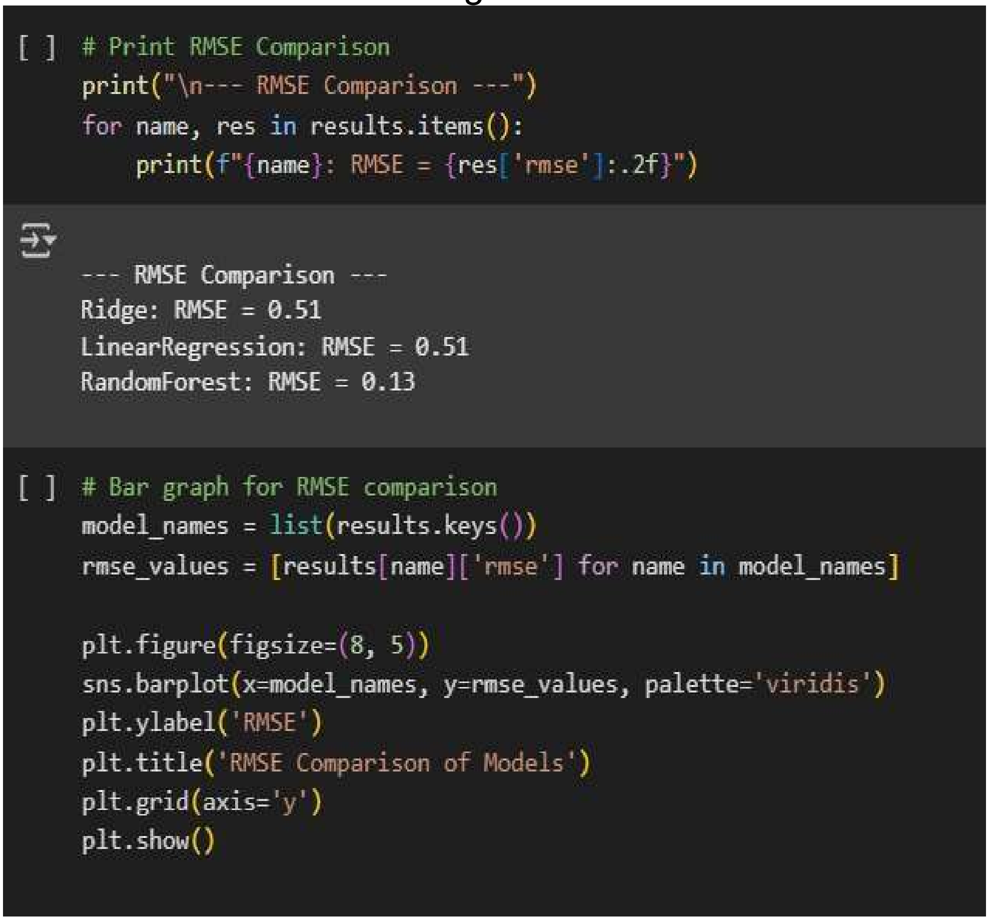
```
# Features
base = ['engine_hp', 'engine_cylinders', 'highway_mpg', 'city_mpg', 'popularity']

def prepare_X(df):
    df = df.copy()
    df['age'] = 2023 - df['year']
    features = base + ['age']
    df = df[features].fillna(df[features].mean())
    return df.values

x_train = prepare_X(car_data_train)
x_val = prepare_X(car_data_val)
```

Fig: **-18**



```
# Models
models = {
    "Ridge": Ridge(alpha=1.0),
    "LinearRegression": LinearRegression(),
    "RandomForest": RandomForestRegressor(n_estimators=100, random_state=2)
}

results = {}
plt.figure(figsize=(18, 5))

for i, (name, model) in enumerate(models.items(), 1):
    model.fit(x_train, y_train)
    y_pred_val = model.predict(x_val)
    rmse = np.sqrt(mean_squared_error(y_val, y_pred_val))
    results[name] = {
        'rmse': rmse,
        'actual': np.expm1(y_val),
        'predicted': np.expm1(y_pred_val)
    }

    plt.subplot(1, 3, i)
    plt.scatter(results[name]['actual'], results[name]['predicted'], alpha=0.5)
    plt.plot([results[name]['actual'].min(), results[name]['actual'].max()],
             [results[name]['actual'].min(), results[name]['actual'].max()], 'r--')
    plt.xlabel('Actual Price')
    plt.ylabel('Predicted Price')
    plt.title(f'{name}: Actual vs Predicted')

plt.tight_layout()
plt.show()
```

Fig**: -19**



```
[ ] # Print RMSE Comparison
    print("\n--- RMSE Comparison ---")
    for name, res in results.items():
        print(f"{name}: RMSE = {res['rmse']:.2f}")
```

```
--- RMSE Comparison ---
Ridge: RMSE = 0.51
LinearRegression: RMSE = 0.51
RandomForest: RMSE = 0.13
```

```
[ ] # Bar graph for RMSE comparison
    model_names = list(results.keys())
    rmse_values = [results[name]['rmse'] for name in model_names]

    plt.figure(figsize=(8, 5))
    sns.barplot(x=model_names, y=rmse_values, palette='viridis')
    plt.ylabel('RMSE')
    plt.title('RMSE Comparison of Models')
    plt.grid(axis='y')
    plt.show()
```

Appendix B: Dataset Description

- Engine Size (Horsepower): A numerical value representing the car's engine power.
- Fuel Efficiency (City MPG): A numerical value indicating miles per gallon in city driving conditions.
- Fuel Efficiency (Highway MPG): A numerical value indicating miles per gallon on highways.
- Year of Manufacture: The year the car was manufactured.
- Brand Popularity: A score between 0 and 100 representing the popularity of the car's brand in the market.
- Car Type: Categorical feature indicating whether the car is a sedan, SUV, hatchback, etc.
- Price: The target variable, representing the car's selling price in USD.

LINKS

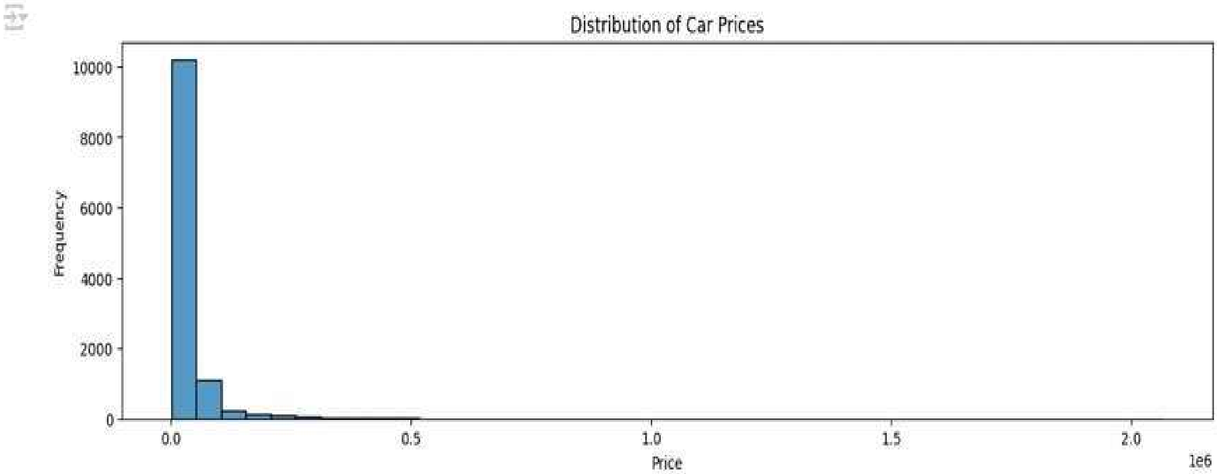
Code-
https://colab.research.google.com/drive/1o62rSUC_E6VA1TGc5vYEuKp1sNLrqhAC?usp=drive_link
Data Set-
https://drive.google.com/file/d/1JT-pWgT4iIThbC8DqGzSJ2Wv0wfpO0WW/view?usp=sharing

---

13. Output

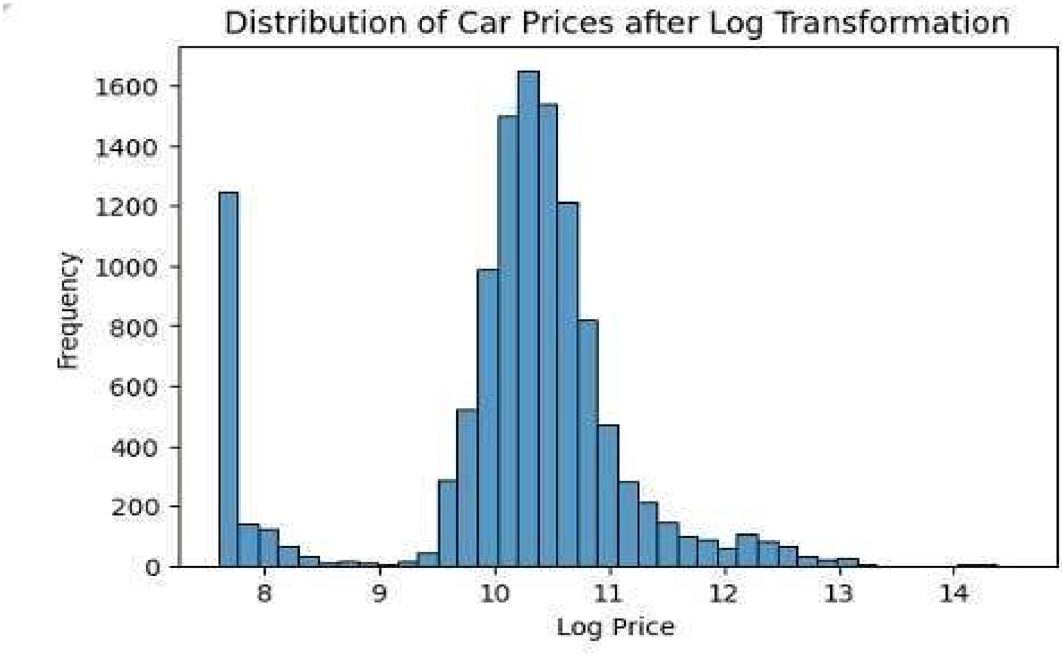
Price distribution Plot-

Fig: **-20**



Log transformation of prices-

Fig: **-21**



Actual vs predicted prices

Fig**: -22**

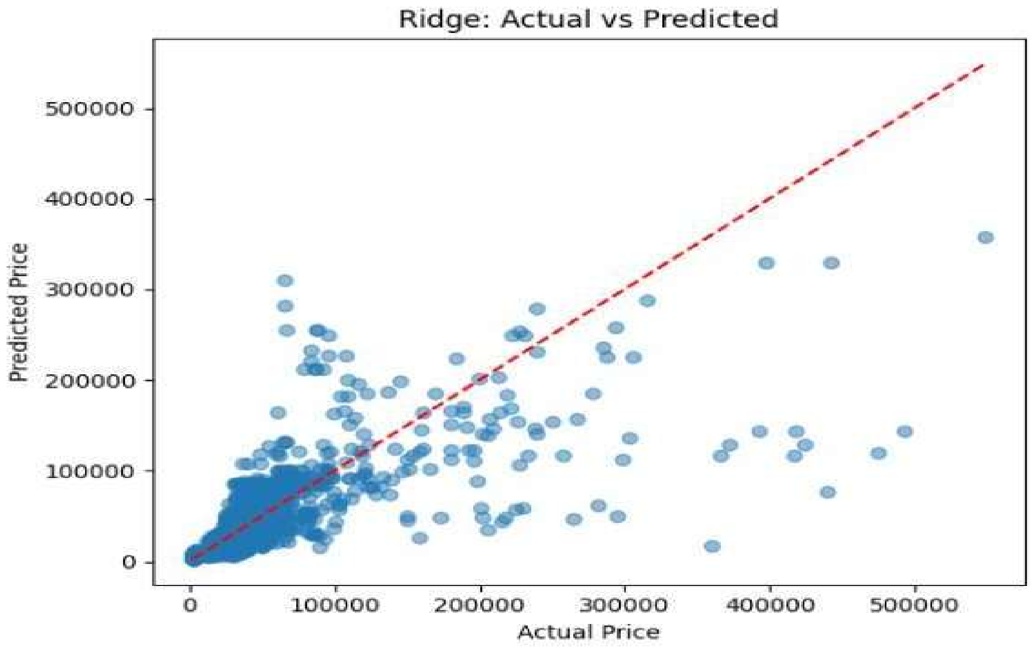


Fig**: -23**

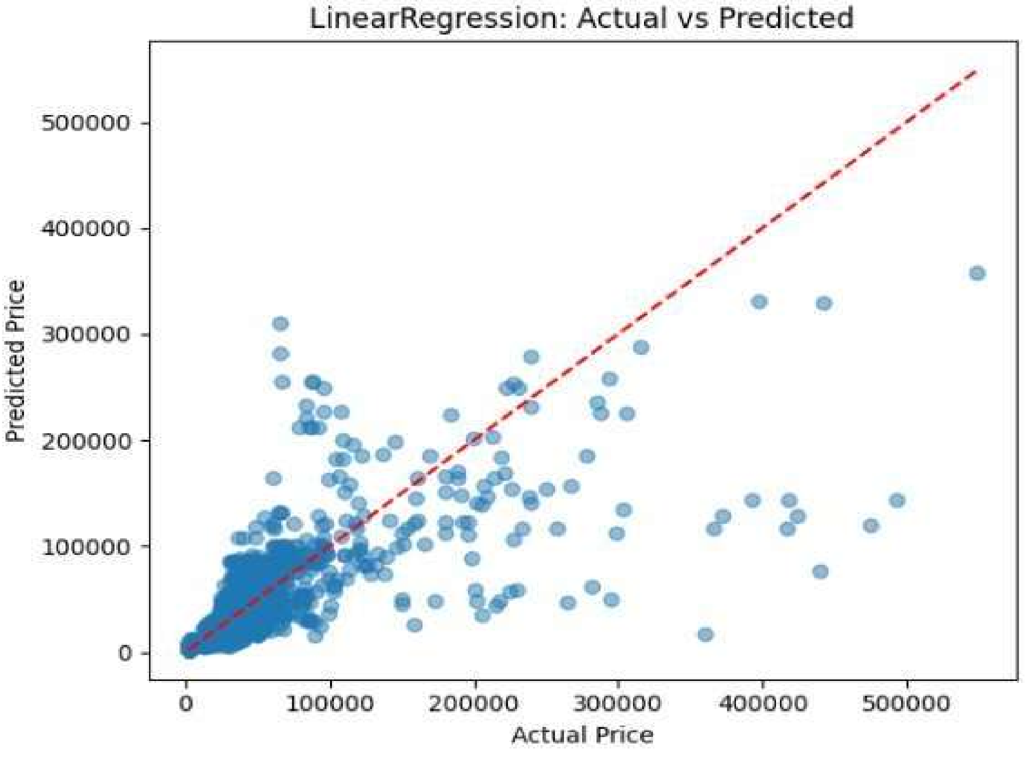
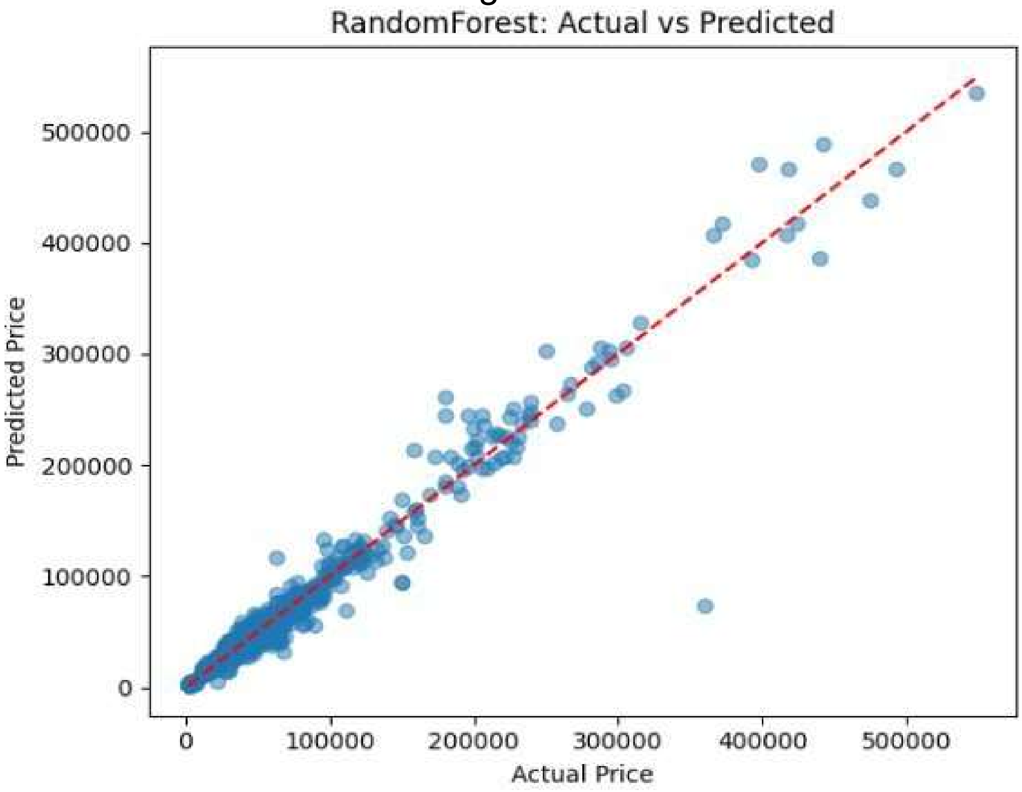
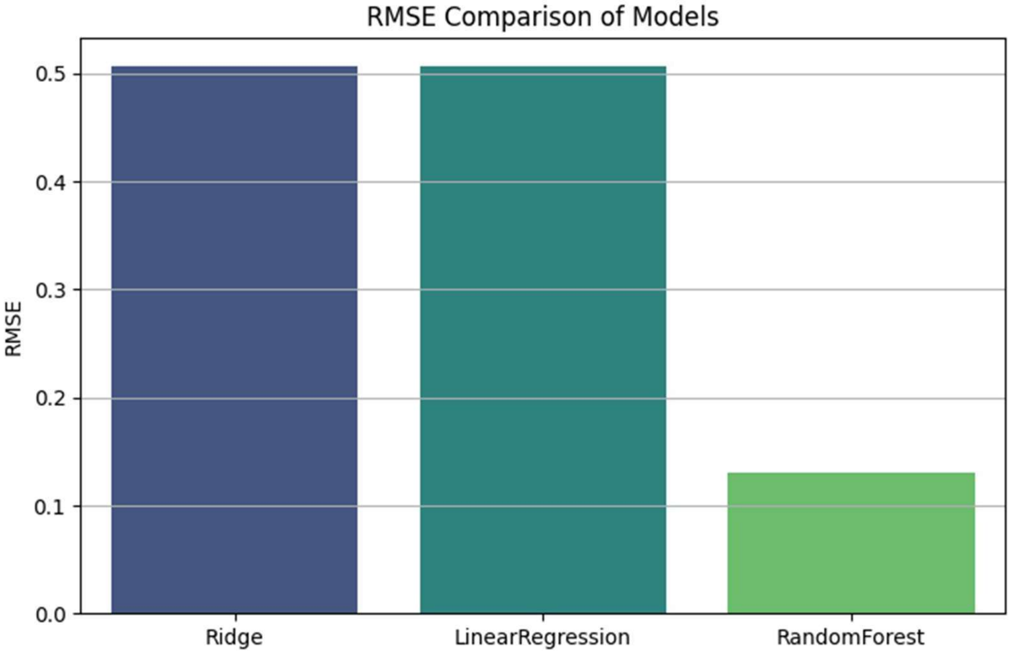


Fig: **-24**



Plot actual vs predicted prices-

Fig: **-25**



---

14. References

1. Kaggle. Car price dataset
https://www.kaggle.com

2. Scikit-learn Developers-
https://scikit-learn.org

3. Pandas development kit-
https://pandas.pydata.org

4 Python Software Foundation-
https://www.python.org

5. Matplotlib Developers-
https://matplotlib.org

Thank you-
Utkarsh Jaiswal (22SCSE1040378)
utkarshj19@gmail.com