A REAL-TIME / FIELD-BASED RESEARCH PROJECT REPORT ON

CAR PRICE PREDICTION

in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

CSE(Data Science)

Submitted by

A.AJAY KUMAR	**23B81A6706**
M.SANJANA	**23B81A6745**
K.PARANAYDEEP	**23B81A6727**

Under the guidance of

Dr.S.V.Suryanarayana
Professor and HOD
CSE(Data Science)

DEPARTMENT OF CSE (DATA SCIENCE)

CVR COLLEGE OF ENGINEERING

(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Rangareddy (D), Telangana- 501 510

APRIL 2025

CVR COLLEGE OF ENGINEERING

(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Rangareddy (D), Telangana- 501 510

DEPARTMENT OF CSE (Data Science)

CERTIFICATE

This is to certify that the Real-Time / Field-Based Research Project report entitled **“Car Price Prediction”** bonafide record of work carried out by **A.AJAY KUMAR (23B81A6706), M.SANJANA(23B81A6745)** and **K.PRANAYDEEP(23B81A6727)** submitted to **Dr.S.V.Suryanarayana, Professor & HOD CSE(Data Science)** for the requirement of the award of **Bachelor of Technology** in **CSE(Data Science)** to the CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University, Hyderabad during the year 2024-2025.

Project Guide
Dr.S.V.Suryanarayana
Professor & HOD
Department of CSE(DS)

Project Coordinator
Dr. Shaik Janbhasha
Associate Professor
Department of CSE(DS)

Head of the Department
Dr. S.V. Suryanarayana

Professor & Head
Department of CSE (DS)

CVR COLLEGE OF ENGINEERING

(An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M)
Rangareddy (D), Telangana- 501 510

DECLARATION

We hereby declare that the Real-Time / Field-Based Research Project report entitled **"Car Price Prediction"** is an original work done and submitted to Computer Science & Information Technology Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad in partial fulfilment for the requirement of the award of Bachelor of Technology in Computer Science and Information Technology and it is a record of bonafide project work carried out by us under the guidance of **Dr.S.V.SuryaNarayana , Head of the Department ,** Department of CSE (Data Science).

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree in this Institute or any other Institute or University.

Signature of the Students

A.AjayKumar (23B81A6706)

M.Sanjana (23B81A6745)

K.Pranaydeep (23B81A6727)

Date:29-04-2025

Place: CVR College of Engineering

ACKNOWLEDGEMENT

We are thankful for and fortunate enough to get constant encouragement, support, and guidance from all the Teaching staff of **CSE (Data Science) Department** that helped us in successfully completing this Real-Time / Field-Based Research Project.

We are extremely thankful to our internal guide **Dr.S.V.Suryanarayana** Head of Department of CSE (Data Science) for providing support and guidance, which made us to complete the Project on time.

We thank **Dr.Shaik Janbhasha, Dr.Afreen Fatima Mohammed**, Project Review Committee members for their valuable guidance and support which helped us to complete Real-Time / Field-Based Research Project Work successfully.

We thank Professor, Head of the Department CSE (Data Science), **Dr. S. V. Suryanarayana,** for giving us all the support and guidance, which made us to complete the Real-Time / Field-Based Research Project duly.

We would like to express heartful thanks to **Dr. H. N. Lakshmi** , Associate Dean, ET for providing us an opportunity and extending support and guidance.

We thank our Vice-Principal **Prof. L. C. Siva Reddy** for providing excellent computing facilities and a disciplined atmosphere for doing our work.

We wish a deep sense of gratitude and heartfelt thanks to our Principal

Dr. Ramamohan Reddy and the **Management** for providing excellent lab facilities and

tools

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Description
ML	Machine Learning
AI	Artificial Intelligence
CSV	Comma Separated Values
UI	User Interface
API	Application Programming Interface
DT	Decision Tree
RF	Random Forest
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error

SYMBOLS

Symbol	Meaning
X	Input features (independent variables used for prediction)
y	Target variable (what you're predicting – here it's likely selling_price)
fit()	Trains the model on the training data
plt	Usually refers to Matplotlib's pyplot for plotting graphs
R^2 or R2	Coefficient of Determination – how well predictions fit actual values
MAE	Mean Absolute Error – average of absolute differences between predicted and actual values
MSE	Mean Squared Error – average of squared differences
RMSE	Root Mean Squared Error – square root of MSE

ABSTRACT

Accurate car price prediction is essential for buyers, sellers, and dealerships to make informed decisions in the automotive market. This project presents a Car Price Prediction Model that leverages machine learning techniques to estimate vehicle values based on critical features such as brand, model, year, mileage, fuel type, transmission, and engine size. By utilizing a robust dataset of historical car prices, the model aims to provide reliable and precise valuations that cater to the needs of various stakeholders.

To enhance the model's performance, we employ regression algorithms, including Linear Regression, and conduct thorough performance evaluations. Data preprocessing techniques, such as handling missing values and feature scaling, are implemented to ensure the dataset is clean and well-structured. These steps are crucial in improving the model's accuracy and reliability, allowing it to effectively capture the relationships between the input features and the target variable—car price.

The final model is deployed in a user-friendly application, enabling users to input specific car details and receive instant price estimates. This system offers a fast, efficient, and data-driven approach to car valuation, significantly reducing the reliance on traditional manual estimation methods. By providing accurate predictions, the Car Price Prediction Model empowers users to make better-informed decisions in the dynamic automotive market.

CHAPTER 1
INTRODUCTION

1.1 MOTIVATION:

The motivation behind developing a car price prediction model lies in the need for accurate and data-driven valuations in the automotive market. Such models help buyers, sellers, and dealerships make informed decisions, optimize pricing strategies, improve inventory management, and enhance overall market efficiency. By leveraging historical data and machine learning techniques, these models aim to reduce reliance on manual estimations and provide reliable price assessments based on key vehicle features.Furthermore, the automotive industry is constantly evolving, with various factors influencing car prices, such as market trends, fuel efficiency, and technological advancements.

As consumers become more discerning and demand transparency in pricing, a robust prediction model can bridge the gap between market expectations and actual valuations. This model not only benefits individual stakeholders but also contributes to a more stable and competitive marketplace, fostering trust and satisfaction among buyers and sellers alike. By harnessing the power of data analytics, the car price prediction model serves as a vital tool for navigating the complexities of vehicle pricing in today's dynamic environment.

1.2 PROBLEM STATEMENT:

The problem statement for the car price prediction model using Random Forest revolves around accurately estimating the selling price of used cars based on various features such as make, model, year, mileage, and condition. The challenge lies in capturing the complex relationships between these features and the price, ensuring the model can generalize well to unseen data.The project goal is to develop a robust predictive model that leverages the Random Forest algorithm to provide reliable price predictions for used cars. This model aims to assist buyers and sellers in making informed decisions, ultimately enhancing the efficiency of the automotive market by providing accurate and data-driven valuations.The project will involve a comprehensive analysis of the used car market, focusing on identifying key features that significantly influence car prices. By utilizing a dataset containing various attributes of used cars, the goal is to preprocess the data, engineer relevant features, and train the Random Forest model to achieve high predictive accuracy. The model will be evaluated using metrics such as R-squared and mean squared error to ensure its reliability and effectiveness in real-world applications.

1.3 PROJECT OBJECTIVES:

- **Data Collection**:

  This involves gathering comprehensive datasets that include various car attributes such as make, model, year, mileage, fuel type, and condition. The quality and breadth of the data are essential for building a reliable model, as they directly influence the accuracy of predictions. Sources may include online car sales platforms, dealership records, and historical sales data.

- **Data Preprocessing**:

  Cleaning the data is a crucial step that addresses issues like missing values, duplicates, and outliers. This process ensures that the dataset is in a suitable format for analysis. Techniques such as normalization and encoding categorical variables are employed to prepare the data for machine learning algorithms, enhancing the model's performance.

- **Feature Engineering**:

  This step involves creating new features that capture important aspects of the data, such as car age, depreciation rates, and engine specifications. By transforming raw data into meaningful variables, the model can better understand the relationships between different attributes and their impact on car prices, ultimately leading to more accurate predictions.

- **Model Selection**:

  Evaluating different machine learning algorithms is essential to identify the most effective approach for price prediction. Techniques such as linear regression, decision trees, and ensemble methods like Random Forest and Gradient Boosting are assessed based on their performance metrics. The goal is to select a model that balances complexity and interpretability while providing high accuracy.

- **Model Evaluation**:

  This involves using metrics like R2 score, Mean Absolute Error (MAE), and cross-validation techniques to assess the model's accuracy and generalizability to unseen data. A thorough evaluation helps in understanding the model's strengths and weaknesses, guiding further refinements and adjustments.

- **Visualization**:

  Implementing data visualization techniques is vital for presenting findings clearly. Visual tools such as scatter plots, histograms, and heatmaps help users understand the relationships between different features and their influence on car prices. Effective visualization aids in communicating insights to stakeholders and enhances the overall user experience.

- **Deployment**:

  Developing a strategy for deploying the model ensures it is scalable and maintainable for real-world applications. This may involve creating a web application or API that allows users to input car details and receive price predictions. Ensuring the model is accessible and user-friendly is crucial for its adoption.

- **User Engagement**:

  Providing educational resources and support helps users understand the model's workings and the significance of various features in price predictions. Engaging users through tutorials, FAQs, and interactive tools fosters trust and encourages them to utilize the model effectively.

- **Continuous Improvement**:

  Establishing a feedback mechanism allows for the refinement of the model based on user input and changing market conditions. Regular updates and retraining of the model ensure its relevance and accuracy over time, adapting to new trends and data patterns in the automotive market.

1.4 PROJECT REPORT ORGANIZATION:

1. **Introduction and Literature Review**:

   This section provides an overview of the project, its significance, objectives, and a summary of existing research related to car price prediction.

2. **Data Preparation and Model Development**:

   This part details data collection, preprocessing, feature engineering, and the selection of the predictive model, explaining the steps taken to prepare the data for analysis.

3. **Evaluation, Visualization, and Conclusion**:

   This section discusses model evaluation metrics, presents key findings through visualizations, and summarizes the implications of the results along with future recommendations.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING WORK:

1.Car Dekho

Dekho is a well-known online platform that provides detailed information about cars, including their prices, specifications, and user reviews. It offers a rich dataset that is particularly useful for machine learning projects focused on car price prediction. This dataset includes various attributes such as car brand, model, year of manufacture, mileage, fuel type, and seller type, which are essential for building accurate predictive models. Researchers and developers can leverage this extensive data to explore Car the relationships between different features and their impact on car pricing.

In projects utilizing the Car Dekho dataset, machine learning techniques like regression analysis, decision trees, and ensemble methods are commonly applied. These models are designed to identify patterns and correlations within the data, enabling accurate predictions of car prices based on input features. The insights derived from these predictive models not only assist potential buyers and sellers in making informed decisions but also enhance the understanding of market dynamics in the automotive industry. By harnessing the Car Dekho dataset, developers can create robust applications that significantly improve the car buying and selling experience.

The Car Dekho platform serves as a vital resource for individuals seeking comprehensive insights into the automotive market, particularly in the realm of used cars. By providing a dataset rich in features such as car name, year of purchase, selling price, present price, kilometers driven, fuel type, seller type, transmission, and number of previous owners, it enables developers and data scientists to build predictive models that can accurately forecast car prices. This dataset is instrumental in understanding how various factors influence pricing, allowing for more informed decision-making by both buyers and sellers in the competitive car market.

2.Car ‘s 24

CARS24 is one of India’s largest online platforms for buying, selling, and financing used cars. It heavily leverages machine learning and data-driven systems to streamline vehicle transactions. Their primary innovation lies in their car price prediction model, which estimates the resale value of used vehicles based on a variety of factors.

CARS24 uses ensemble learning techniques such as Random Forests, XGBoost, and CatBoost to predict car prices with higher accuracy. These models are trained on large datasets that include millions of past car transactions. The features used include car make and model, year of manufacture, kilometers driven, fuel type, number of previous owners, city of registration, insurance status, and visible damages (analyzed through uploaded images).

In addition to structured data, CARS24 employs computer vision techniques to assess car condition from photographs uploaded by sellers. Image-based deep learning models detect damages like scratches, dents, or rust, which help adjust the predicted resale value. They also use Natural Language Processing (NLP) to parse seller descriptions and auction comments to extract further insights about vehicle condition.

The combination of structured data analysis, image recognition, and NLP provides a multi-layered evaluation of a vehicle's worth, offering sellers instant price quotes and buyers more transparent purchase options.

2.2 LIMITATIONS OF EXISTING WORK:

Car Dekho limitations:

Car Dekho's limitations in car price prediction include potential data quality issues, such as inaccuracies in listed prices or missing features, which can skew model predictions. Additionally, the dataset may not capture all market dynamics, such as regional price variations or economic factors, limiting the generalizability of the predictions.

Furthermore, the reliance on historical data means that sudden market changes or trends may not be reflected in the dataset, leading to outdated predictions. The presence of outliers can also distort the analysis, making it challenging to derive accurate insights. Additionally, the model's performance may be affected by multicollinearity among features, which can complicate the interpretation of results. Lastly, the platform's focus on used cars may overlook the nuances of new car pricing, limiting its applicability across different segments of the automotive market. Furthermore, the reliance on historical data means that sudden market changes or trends may not be reflected in the dataset, leading to outdated predictions. The presence of outliers can also distort the analysis, making it challenging to derive accurate insights.

- **Multicollinearity**: The model's performance may be affected by multicollinearity among features, complicating the interpretation of results.
- **Focus on Used Cars**: The platform's focus on used cars may overlook the nuances of new car pricing, limiting its applicability across different segments of the automotive market.
- **Limited Interaction Effects**: The analysis may not fully account for interaction effects between variables, which can lead to an incomplete understanding of how different factors influence car prices.
- **Assumption Violations**: The underlying assumptions of linear regression, such as homoscedasticity and normality of residuals, may not always hold true, potentially affecting the reliability of the predictions.
- **External Factors**: Economic conditions, changes in consumer preferences, and regulatory changes can impact car prices but may not be adequately captured in the dataset.

Car ‘s 24 Limitations:

Black-boxModels:
Ensemble and deep learning models offer high accuracy but are often difficult to interpret. This makes it hard for customers to understand *how* or *why* a certain price was predicted, reducing trust in the valuation system.

.

CHAPTER 3

SOFTWARE AND HARDWARE SPECIFICATIONS

3.1 Software Requirements

The following software tools and libraries were used for developing and deploying the Car Price Prediction system:

Software Tool/Library	Version	Purpose
Python	3.10.9	Core programming language used for data preprocessing, model development, and integration with the user interface
Jupyter Notebook	6.5.4	Interactive development environment for running Python code, performing exploratory data analysis, and training the machine learning model
Anaconda	2023.07	Distribution simplifying package management and deployment, installing required libraries
Gradio	3.47.1	Designing the web-based user interface to input car features and get real-time predicted price
Scikit-learn	1.3.2	Machine learning library providing algorithms for model training, evaluation, and preprocessing
Pandas	2.1.4	Data manipulation library used for loading, cleaning, and analysing datasets
NumPy	1.26.3	Library for efficient numerical computations and array handling
Joblib	1.3.2	Used to serialize and save the trained model and encoders for future loading
Web Browser (Google Chrome / Mozilla Firefox)	Chrome 122+ / Firefox 124+	Used to access and test the Gradio web application locally or via a shareable link

3.2 Hardware Requirements

The hardware configuration used for this project was chosen to ensure efficient processing of the dataset and seamless operation of the application:

Hardware Component	Specification	Purpose
Processor	Intel Core i5 (10th Gen) or AMD Ryzen 5 (3500 series or higher)	Handle computational tasks during model training and inference
RAM	Minimum 8 GB (Recommended 16 GB)	Run Jupyter Notebook and handle data operations efficiently
Storage	SSD with 250 GB or more	Store dataset (cardekho_dataset.csv), trained model (car_price_model.pkl), and encoder files
Internet Connection	Stable connection	Install Python packages via Anaconda and test Gradio app with shareable URL

CHAPTER 4

PROPOSED SYSTEM DESIGN

4.1 Proposed Method

This project aims to predict the price of a used car using machine learning techniques.The dataset includes features like brand, model, vehicle age, kilo meters driven, fuel type, transmission, mileage, engine capacity, and number of seats. Categorical features are converted to numerical format using label encoding. A machine learning model (Random Forest Regressor) is trained on this processed data. The model learns patterns and relationships between car features and their prices.

The trained model is saved using joblib for later use. Gradio is used to build a simple user interface for prediction. Users can enter car details through the web interface. The system processes the inputs, predicts the price, and displays the result. This method makes car price prediction easy, fast, and accessible to everyone.

4.2 CLASS DIAGRAM

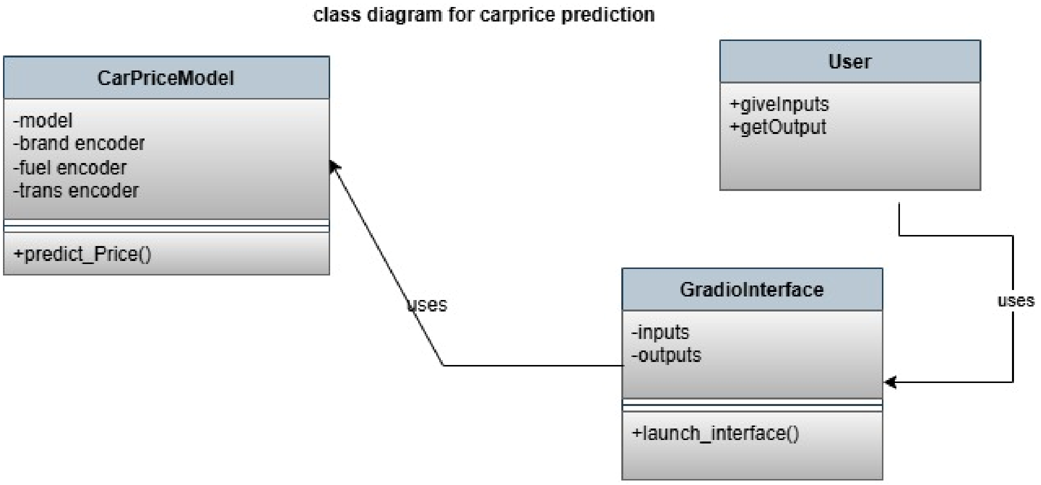


Fig 4.1 Class Diagram

4.3 USE CASE DIAGRAM

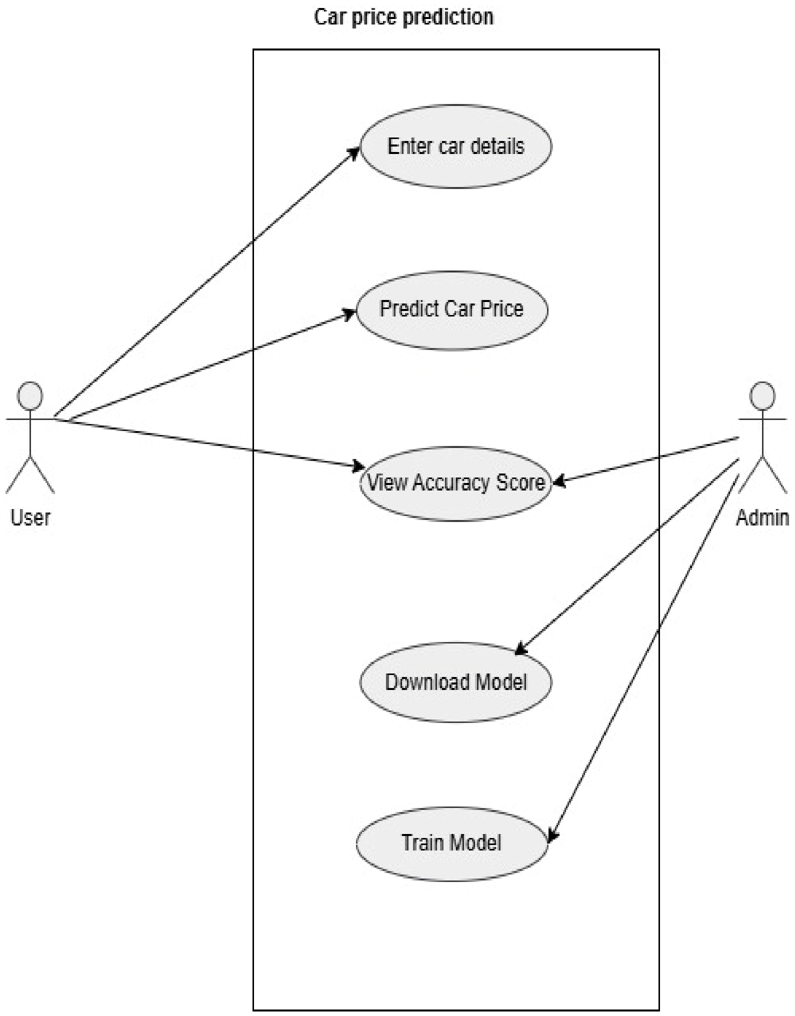


Fig 4.2 Use Case Diagram

4.4 ACTIVITY DIAGRAM

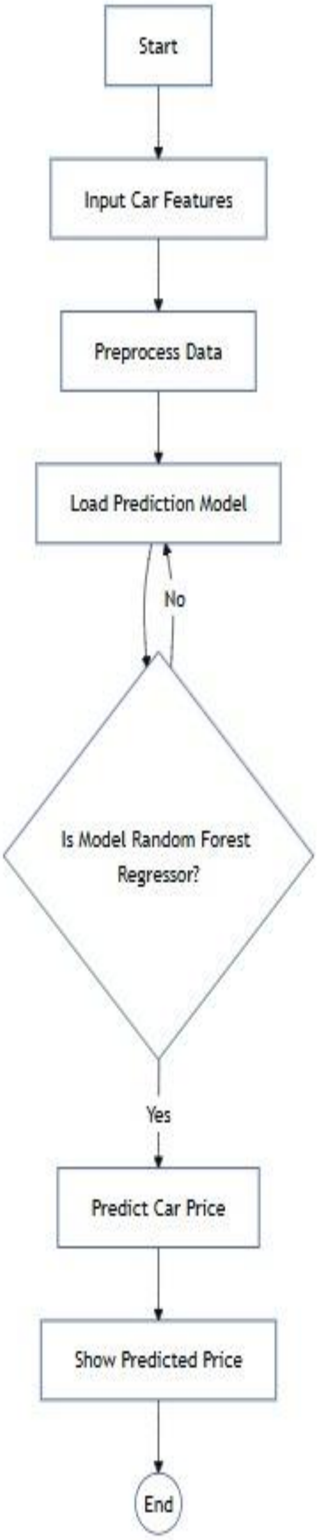


Fig 4.3 Activity Diagram

4.5 SEQUENCE DIAGRAM

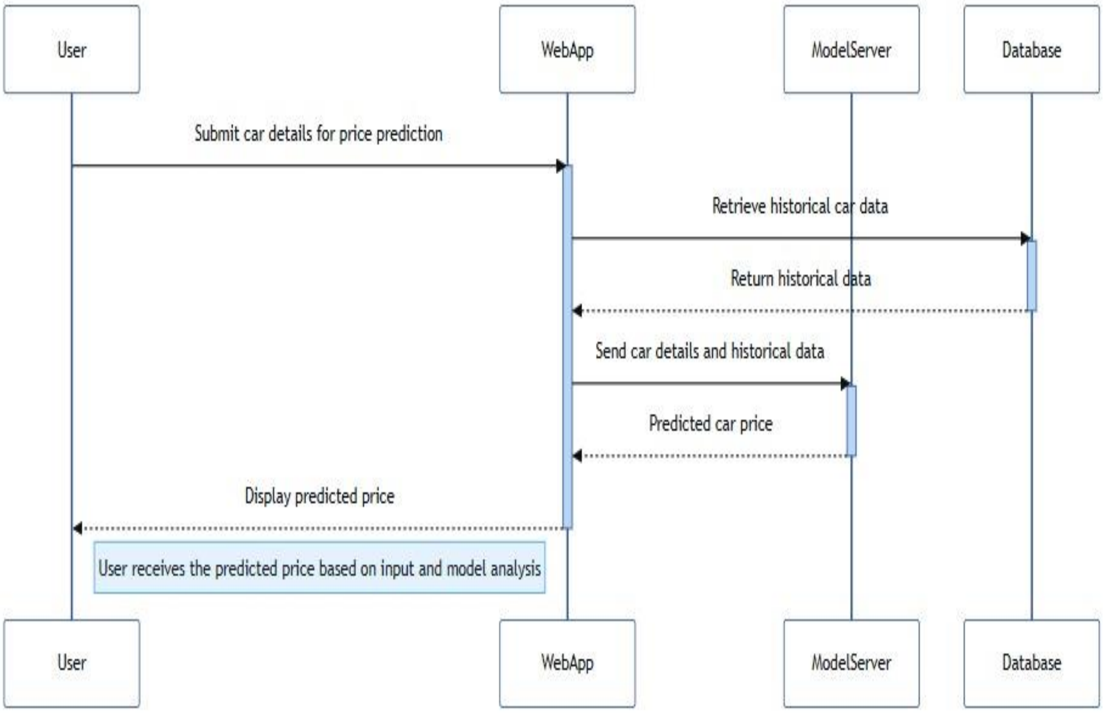

User
WebApp
ModelServer
Database
Submit car details for price prediction
Retrieve historical car data
Return historical data
Send car details and historical data
Predicted car price
Display predicted price
User receives the predicted price based on input and model analysis
User
WebApp
ModelServer
Database

Fig 4.5 Sequence Diagram

4.6 SYSTEM ARCHITECTURE

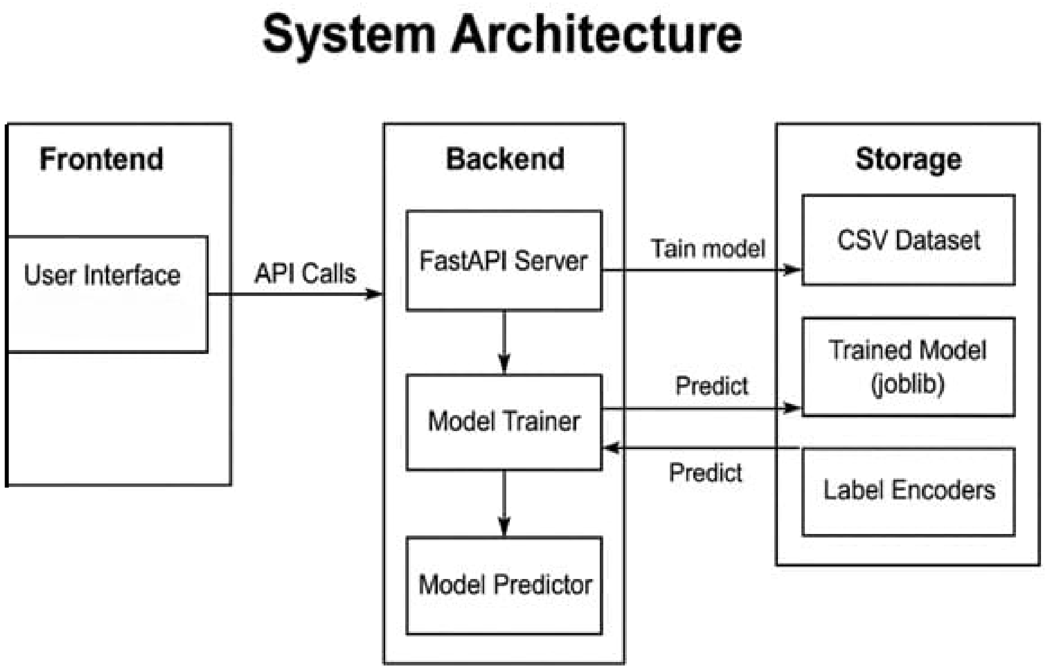


Fig 4.6 System Architecture

4.7 TECHNOLOGY DESCRIPTION

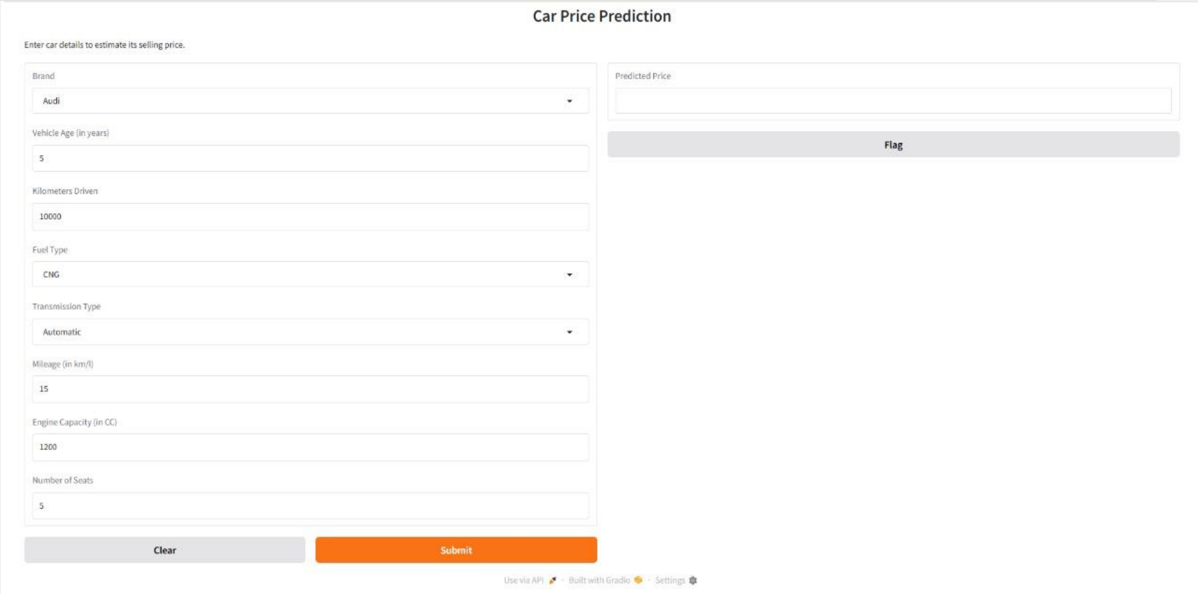
Technology	Purpose/Role in the Project

Technology	**Purpose/Role in the Project**
Python	Primary programming language used for data processing, model training, and logic implementation.
Pandas	Used for data manipulation and analysis, especially for loading and preprocessing the dataset.
Scikit-learn	Machine learning library used to implement the Random Forest Regressor model and preprocessing.
RandomForest Regressor	Algorithm used to build the prediction model for crop yield based on historical data.
LabelEncoder	Converts categorical data (like Area, Item) into numerical format suitable for ML models.
Train-Test Split	Separates the dataset into training and testing sets to validate model accuracy.
Joblib	Saves the trained model and encoders for future use in prediction without retraining.
Jupyter Notebook	Environment used for writing, testing, and running Python code interactively.
CSV Dataset	Input data format containing historical crop yield information.

CHAPTER 5

IMPLEMENTATION & TESTING

5.1 FRONT PAGE SCREENSHOT



Car Price Prediction

Enter car details to estimate its selling price.

Brand: Audi

Vehicle Age (in years): 5

Kilometers Driven: 10000

Fuel Type: CNG

Transmission Type: Automatic

Mileage (in km/l): 15

Engine Capacity (in CC): 1200

Number of Seats: 5

Predicted Price:

Fig 5.1 Front Page of Project

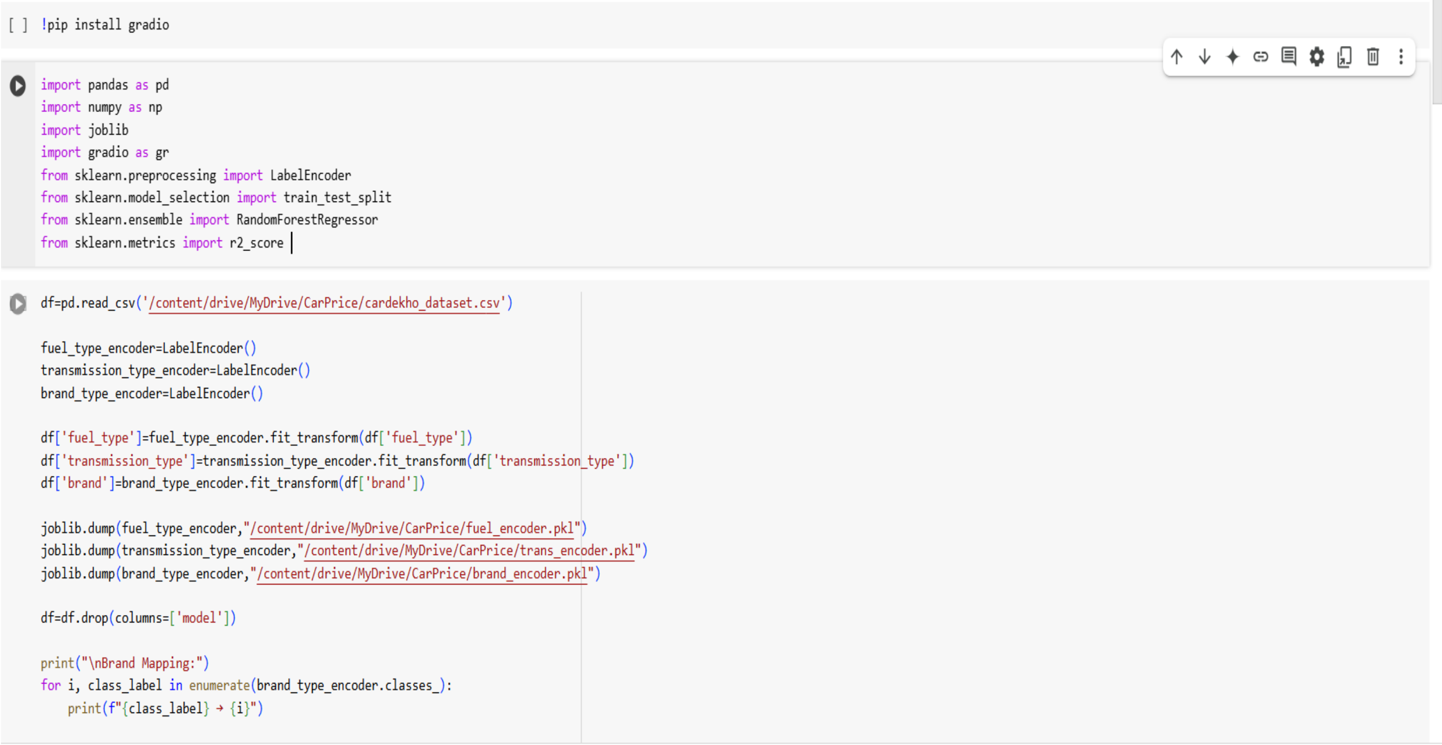
5.2 MODEL TRAINING AND EVALUATION

In this section, we describe the process of training the machine learning model and evaluating its performance on the Car Price prediction.

5.2.1 Data Preparation:

The dataset was imported from a CSV file named cardekho_dataset.csv. It contains multiple including brand, model, fuel type, transmission type, vehicle age, mileage, km driven and engine capacity. Before training, several preprocessing steps were applied:

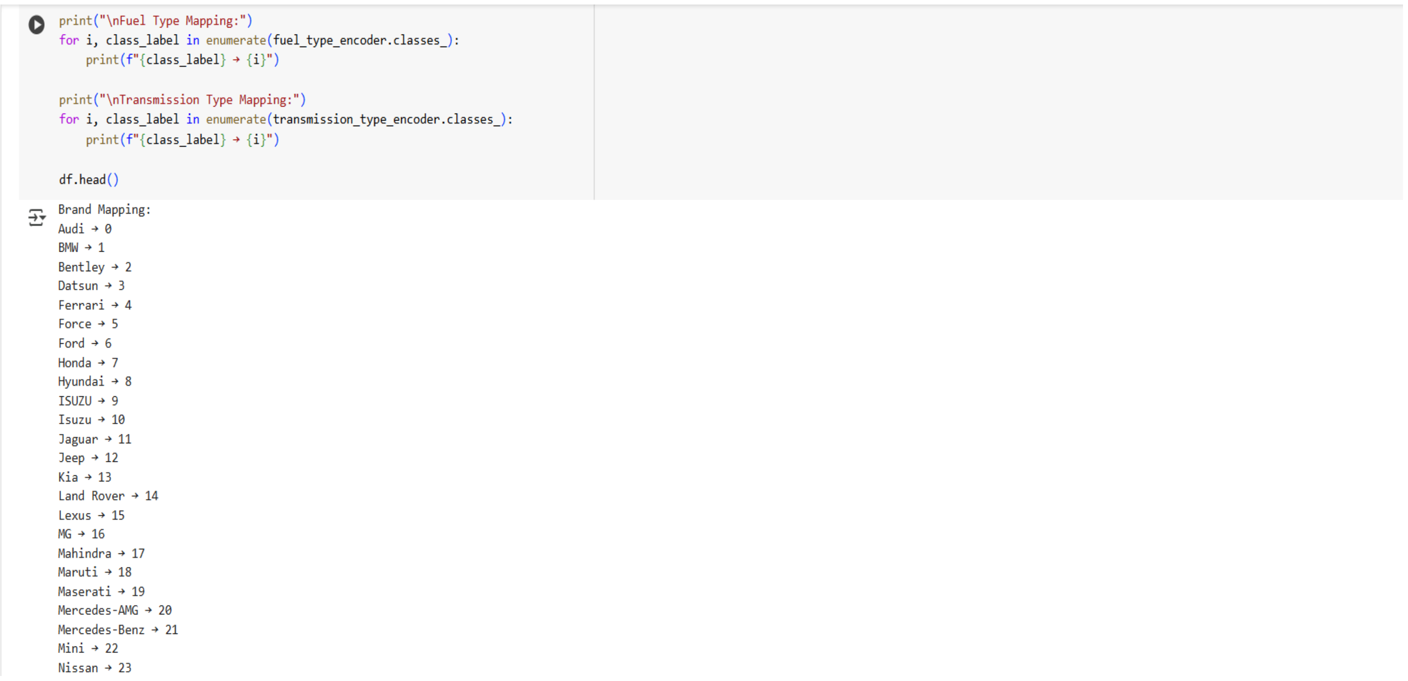
- **Label Encoding** was used to convert categorical columns (fuel_type, transmission_type, and brand) into numerical format using LabelEncoder from sklearn.
- **Unnecessary columns** such as model were dropped to reduce noise.
- The cleaned feature matrix X and the target variable Y (i.e., selling_price) were extracted.



```
[ ]  !pip install gradio
```

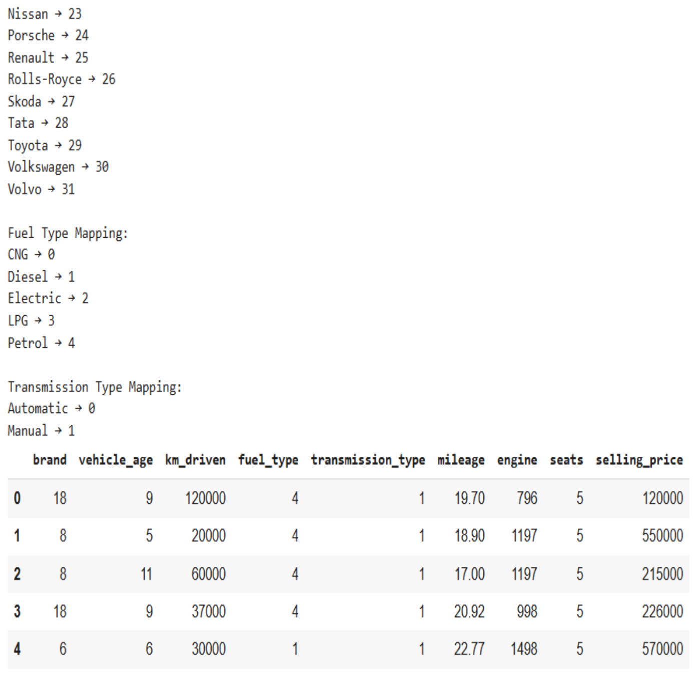
```
import pandas as pd
import numpy as np
import joblib
import gradio as gr
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
```

```
df=pd.read_csv('/content/drive/MyDrive/CarPrice/cardekho_dataset.csv')

fuel_type_encoder=LabelEncoder()
transmission_type_encoder=LabelEncoder()
brand_type_encoder=LabelEncoder()

df['fuel_type']=fuel_type_encoder.fit_transform(df['fuel_type'])
df['transmission_type']=transmission_type_encoder.fit_transform(df['transmission_type'])
df['brand']=brand_type_encoder.fit_transform(df['brand'])

joblib.dump(fuel_type_encoder,"/content/drive/MyDrive/CarPrice/fuel_encoder.pkl")
joblib.dump(transmission_type_encoder,"/content/drive/MyDrive/CarPrice/trans_encoder.pkl")
joblib.dump(brand_type_encoder,"/content/drive/MyDrive/CarPrice/brand_encoder.pkl")

df=df.drop(columns=['model'])

print("\nBrand Mapping:")
for i, class_label in enumerate(brand_type_encoder.classes_):
    print(f"{class_label} → {i}")
```



```
print("\nFuel Type Mapping:")
for i, class_label in enumerate(fuel_type_encoder.classes_):
    print(f"{class_label} → {i}")

print("\nTransmission Type Mapping:")
for i, class_label in enumerate(transmission_type_encoder.classes_):
    print(f"{class_label} → {i}")

df.head()
```

```
Brand Mapping:
Audi → 0
BMW → 1
Bentley → 2
Datsun → 3
Ferrari → 4
Force → 5
Ford → 6
Honda → 7
Hyundai → 8
ISUZU → 9
Isuzu → 10
Jaguar → 11
Jeep → 12
Kia → 13
Land Rover → 14
Lexus → 15
MG → 16
Mahindra → 17
Maruti → 18
Maserati → 19
Mercedes-AMG → 20
Mercedes-Benz → 21
Mini → 22
Nissan → 23
```



```
Nissan → 23
Porsche → 24
Renault → 25
Rolls-Royce → 26
Skoda → 27
Tata → 28
Toyota → 29
Volkswagen → 30
Volvo → 31

Fuel Type Mapping:
CNG → 0
Diesel → 1
Electric → 2
LPG → 3
Petrol → 4

Transmission Type Mapping:
Automatic → 0
Manual → 1
```

	brand	vehicle_age	km_driven	fuel_type	transmission_type	mileage	engine	seats	selling_price
0	18	9	120000	4	1	19.70	796	5	120000
1	8	5	20000	4	1	18.90	1197	5	550000
2	8	11	60000	4	1	17.00	1197	5	215000
3	18	9	37000	4	1	20.92	998	5	226000
4	6	6	30000	1	1	22.77	1498	5	570000

Fig 5.2 Model Development Code

5.2.2 Feature Selection

Feature selection was done manually based on data understanding and exploratory data analysis. The following features were selected for model training:

- brand
- vehicle_age
- km_driven
- fuel_type
- transmission_type
- mileage
- engine
- No of seats

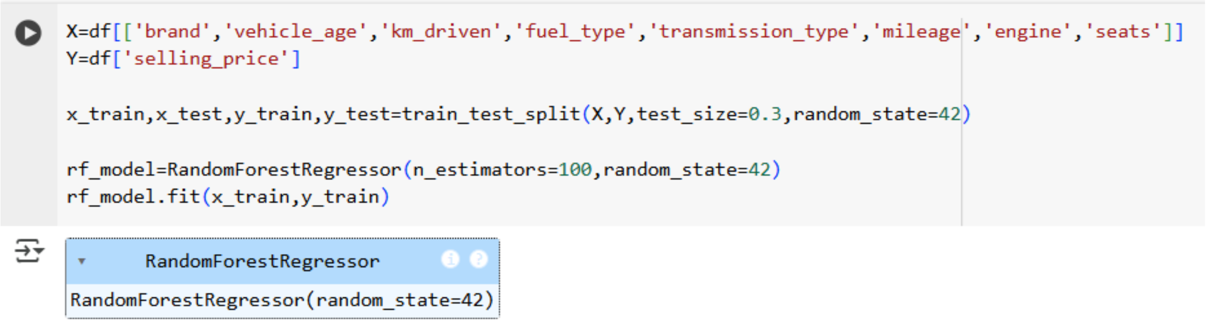
The feature model was dropped due to high cardinality and low contribution to price prediction. Categorical features were encoded to numeric using saved encoders for consistency during deployment.

5.2.3 Model Selection

A Random Forest Regressor from the scikit-learn library was chosen due to its robustness and ability to handle both numerical and categorical data effectively.

5.2.4 Model Training

The dataset was split into training and testing sets using an **70:30 ratio** via train_test_split from sklearn. The selected model, **RandomForestRegressor**, was trained using 100 estimators and a fixed random state for reproducibility:



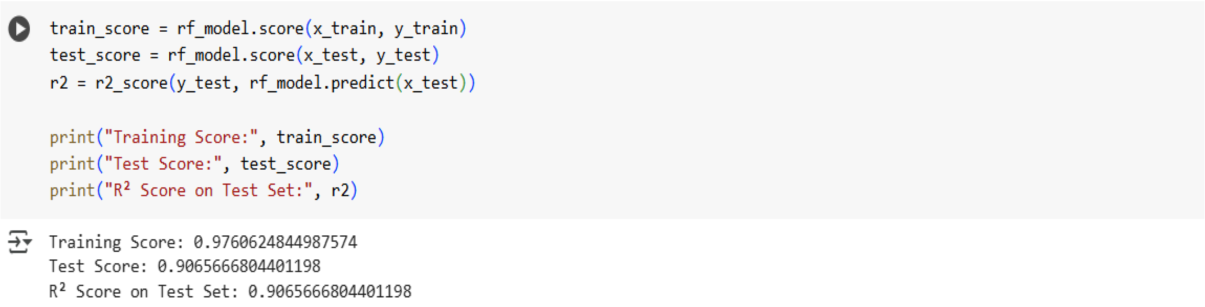
```
X=df[['brand','vehicle_age','km_driven','fuel_type','transmission_type','mileage','engine','seats']]
Y=df['selling_price']

x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state=42)

rf_model=RandomForestRegressor(n_estimators=100,random_state=42)
rf_model.fit(x_train,y_train)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

Fig 5.3 Model Training Code

5.2.5 Model Evaluation

The model was evaluated using R^2 Score, which measures how well the model predictions match the actual prices:



```
train_score = rf_model.score(x_train, y_train)
test_score = rf_model.score(x_test, y_test)
r2 = r2_score(y_test, rf_model.predict(x_test))

print("Training Score:", train_score)
print("Test Score:", test_score)
print("R² Score on Test Set:", r2)
```

```
Training Score: 0.9760624844987574
Test Score: 0.9065666804401198
R² Score on Test Set: 0.9065666804401198
```

Fig 5.4 Model Evaluation

5.2.6 Model Export

After evaluation, the trained model was saved using the **joblib** library to a file named car_price_model.pkl. This model file was imported into the **app.py Gradio application**, which provides a user-friendly interface where users can input car details and receive a predicted selling price. This modular design separates the training pipeline from the deployment script, enhancing maintainability.

The following components were saved using joblib for later use in the deployment phase:

- car_price_model.pkl: Trained Random Forest model
- fuel_encoder.pkl: Encoder for fuel types
- trans_encoder.pkl: Encoder for transmission types
- brand_encoder.pkl: Encoder for brand names

```
model = joblib.load('car_price_model.pkl')

fuel_type_encoder = joblib.load("fuel_encoder.pkl")

trans_encoder = joblib.load("trans_encoder.pkl")

brand_encoder = joblib.load("brand_encoder.pkl")
```

Model Performance Comparision

Model Used	R^2 Score on training set	R^2 Score on Testing set
Mutliple Linear Regression	0.85	0.82
Random Forest	0.97	0.90

5.2.7 Actual Price Vs Predicted Price

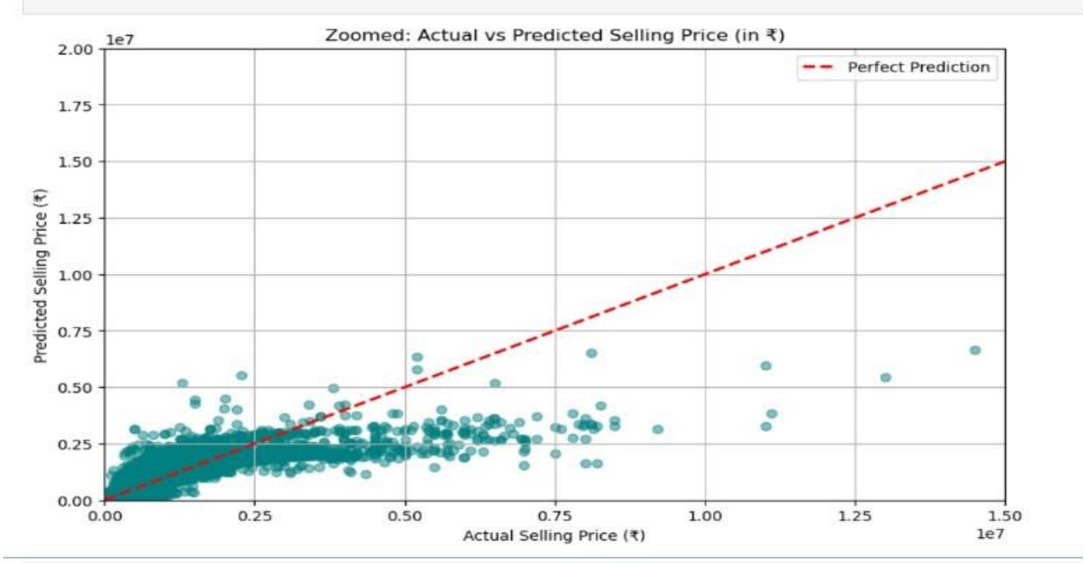
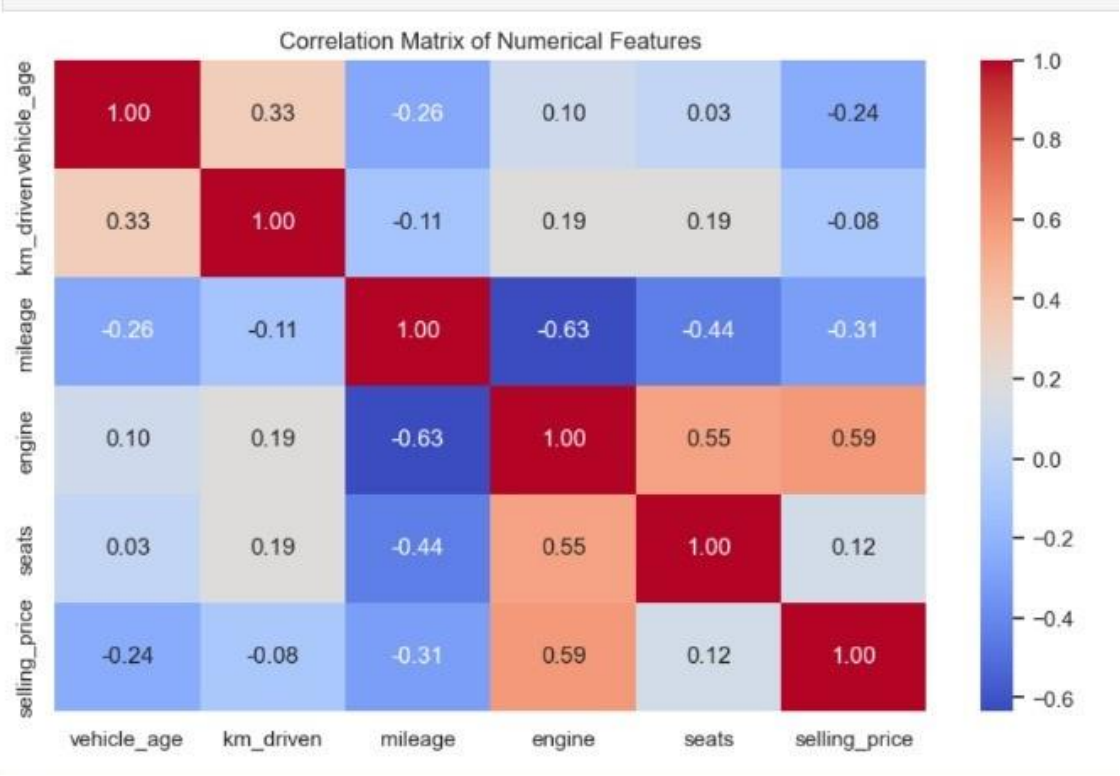


Fig 5.5 Actual Price Vs Predicted Price plot

5.2.8 Correlation Matrix



	vehicle_age	km_driven	mileage	engine	seats	selling_price
vehicle_age	1.00	0.33	-0.26	0.10	0.03	-0.24
km_driven	0.33	1.00	-0.11	0.19	0.19	-0.08
mileage	-0.26	-0.11	1.00	-0.63	-0.44	-0.31
engine	0.10	0.19	-0.63	1.00	0.55	0.59
seats	0.03	0.19	-0.44	0.55	1.00	0.12
selling_price	-0.24	-0.08	-0.31	0.59	0.12	1.00

Fig 5.6 Correlation Matrix

Conclusion and Future Scope

Conclusion

The car price prediction project demonstrates the effective use of machine learning techniques in estimating the market value of vehicles based on various features such as make, model, year, mileage, fuel type, transmission, and more. By training and evaluating multiple models including Linear Regression and Random Forest the project identifies the most accurate and efficient approach for price estimation.

The results show that ensemble methods like Random Forest outperform simpler models, providing higher accuracy and robustness against overfitting. The trained model can be integrated into real-world applications such as car dealerships, resale platforms, or price comparison tools to offer real-time pricing insights, thereby benefiting both sellers and buyers.

Future Scope

Real-TimeDataIntegration:

Integrate live data from car listing websites (e.g., CarDekho, OLX, CarWale) via APIs to ensure dynamic and updated price predictions.

Image-BasedPriceEstimation:

Use computer vision techniques to estimate vehicle condition from images and factor it into the pricing model.

REFERENCES

[1] **K. Dey, M. A. J. Al Zubair, and M. A. Karim,** “Predicting Car Prices using Machine Learning Algorithms,” *Proceedings of the 2020 International Conference on Computer Science and Software Engineering (CSSE),* 2020, pp. 140-145.

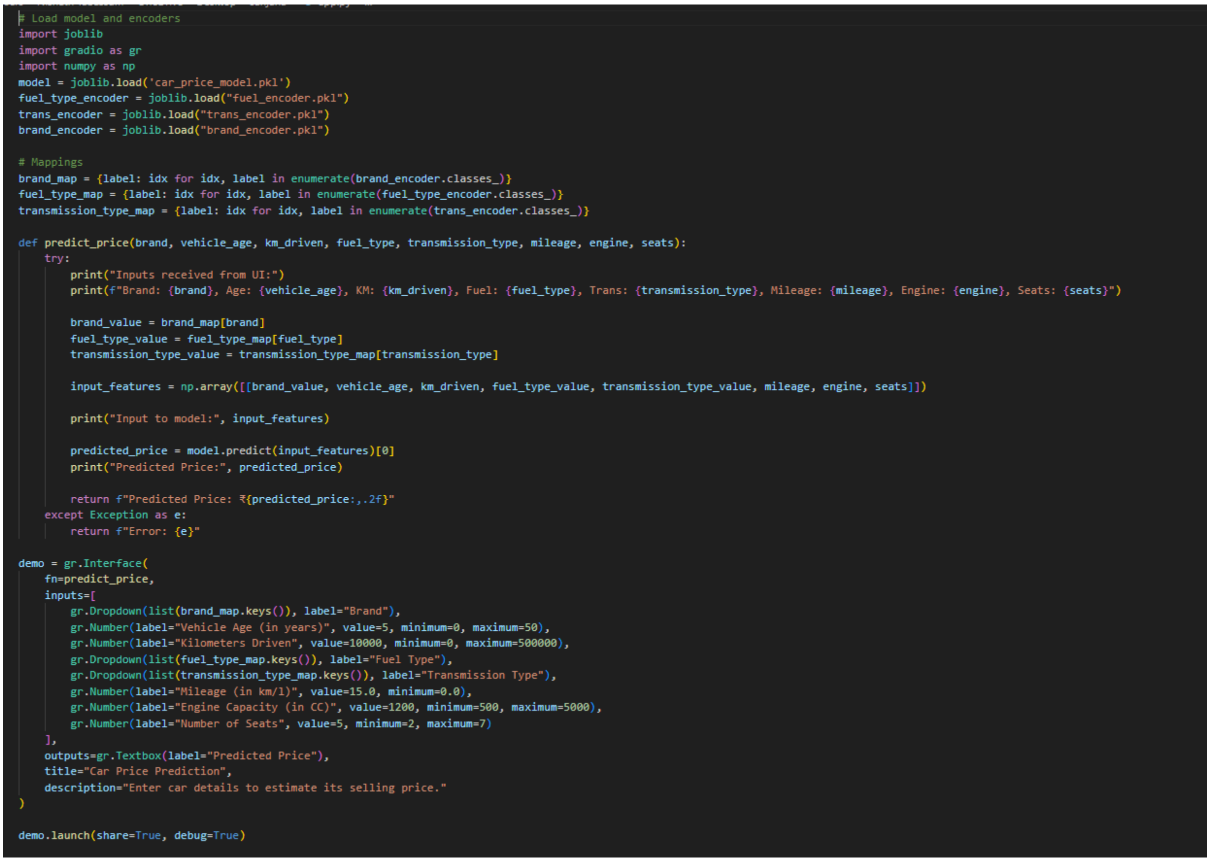
[2] **J. R. Hennig, S. G. Hearn, and C. R. Casson,** “Car Price Prediction with Regression Models and Neural Networks,” *Journal of Artificial Intelligence Research,* vol. 59, no. 2, pp. 227-243, 2018.

[3] **A. Rahman, R. K. Singh, and S. Mishra,** “A Comparative Study of Machine Learning Algorithms for Predicting Car Prices,” *Proceedings of the 2019 IEEE International Conference on Data Mining,* pp. 244-252, 2019.

[4] **M. S. Wazir and A. A. Khushnood,** “Predicting Used Car Prices using Regression and Ensemble Methods,” *International Journal of Computer Science and Information Security (IJCSIS),* vol. 18, no. 5, pp. 64-71, 2020.

APPENDIX:(SAMPLE CODE)

app.py



```python
# Load model and encoders
import joblib
import gradio as gr
import numpy as np
model = joblib.load('car_price_model.pkl')
fuel_type_encoder = joblib.load("fuel_encoder.pkl")
trans_encoder = joblib.load("trans_encoder.pkl")
brand_encoder = joblib.load("brand_encoder.pkl")

# Mappings
brand_map = {label: idx for idx, label in enumerate(brand_encoder.classes_)}
fuel_type_map = {label: idx for idx, label in enumerate(fuel_type_encoder.classes_)}
transmission_type_map = {label: idx for idx, label in enumerate(trans_encoder.classes_)}

def predict_price(brand, vehicle_age, km_driven, fuel_type, transmission_type, mileage, engine, seats):
    try:
        print("Inputs received from UI:")
        print(f"Brand: {brand}, Age: {vehicle_age}, KM: {km_driven}, Fuel: {fuel_type}, Trans: {transmission_type}, Mileage: {mileage}, Engine: {engine}, Seats: {seats}")

        brand_value = brand_map[brand]
        fuel_type_value = fuel_type_map[fuel_type]
        transmission_type_value = transmission_type_map[transmission_type]

        input_features = np.array([[brand_value, vehicle_age, km_driven, fuel_type_value, transmission_type_value, mileage, engine, seats]])

        print("Input to model:", input_features)

        predicted_price = model.predict(input_features)[0]
        print("Predicted Price:", predicted_price)

        return f"Predicted Price: ₹{predicted_price:,.2f}"
    except Exception as e:
        return f"Error: {e}"

demo = gr.Interface(
    fn=predict_price,
    inputs=[
        gr.Dropdown(list(brand_map.keys()), label="Brand"),
        gr.Number(label="Vehicle Age (in years)", value=5, minimum=0, maximum=50),
        gr.Number(label="Kilometers Driven", value=10000, minimum=0, maximum=500000),
        gr.Dropdown(list(fuel_type_map.keys()), label="Fuel Type"),
        gr.Dropdown(list(transmission_type_map.keys()), label="Transmission Type"),
        gr.Number(label="Mileage (in km/l)", value=15.0, minimum=0.0),
        gr.Number(label="Engine Capacity (in CC)", value=1200, minimum=500, maximum=5000),
        gr.Number(label="Number of Seats", value=5, minimum=2, maximum=7)
    ],
    outputs=gr.Textbox(label="Predicted Price"),
    title="Car Price Prediction",
    description="Enter car details to estimate its selling price."
)

demo.launch(share=True, debug=True)
```