USED CAR PRICE PREDICTION

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

submitted by

Archana T Dharan

Sneha K S

Sulaikha Nazrin

ICT ACADEMY OF KERALA
THIRUVANANTHAPURAM, KERALA, INDIA
Nov 2022

List of Figures

List of Abbreviations

Abbreviations	Definition
EDA	Exploratory Data Analysis
csv	Comma-Separated Values
pd	Pandas library in Python
np	NumPy library in Python
XGBoost	eXtreme Gradient Boosting
R^2	R-squared. It refers to the coefficient of determination
sklearn	scikit-learn
RMSE	Root mean squared error
MSE	Mean Square Error
MAE	Mean Absolute Error
INR	Indian Rupee

Table of Contents

ABSTRACT

The automobile industry has witnessed a significant surge in the use of online platforms for buying and selling vehicles. With the increasing volume of transactions and the wide variety of car models, predicting the price of a used car based on various attributes is crucial for both sellers and buyers. This project focuses on developing a car price prediction model using machine learning techniques. The model is trained to predict the price of a used car based on several key features such as the car's name, age, kilometers driven, mileage, engine capacity, maximum power, fuel type, transmission type, seller type, and the number of seats. The dataset used for training the model consists of historical data on used cars, where each record includes these features along with the corresponding price. The project utilizes the Gradient Boosting Regressor algorithm, which is known for its efficiency in regression tasks, to predict the price of a car based on the input features. The model is then deployed on a Stream lit web application, enabling users to input the required data and obtain an estimated price for a used car. This web application serves as a practical tool for individuals looking to buy or sell used cars, providing a reliable price estimation based on the car's attributes. The interface ensures ease of use with input fields for car details, and the prediction result is displayed dynamically once the user submits their inputs. This solution addresses the need for quick and accurate price estimations in the used car market and leverages machine learning to optimize decision-making for both buyers and sellers.

1. Problem Definition

The goal of this project is to develop a predictive model to estimate car prices based on various attributes, such as make, model, year, mileage, and fuel type. The project aims to use machine learning algorithms to predict the selling price of a car and perform Exploratory Data Analysis (EDA) to understand the factors influencing car prices, such as mileage, brand, and manufacturing year. Additionally, the project will identify the key features that significantly affect car prices. Finally, the trained model will be integrated into a web application, allowing users to input car details and receive price predictions.

1.1 Overview

The project focuses on developing a machine learning model to predict car prices based on various attributes such as make, model, year of manufacture, mileage, fuel type, and other relevant features. Accurate car price predictions are essential for both buyers and sellers to make informed decisions in the automobile market. This project involves a comprehensive analysis of the dataset through Exploratory Data Analysis (EDA) to identify key factors influencing car prices. The insights gained from the analysis are used to train predictive models, ensuring high accuracy and reliability. Furthermore, the project aims to deploy the model in a user-friendly web application, enabling users to input car details and receive instant price predictions. This solution can streamline the car-buying and selling process, making it more efficient and transparent.

1.2 Problem Statement

The problem is to develop a reliable model that can accurately predict the selling price of a car based on its features, such as brand, model, year of manufacture, mileage, fuel type, and other specifications. Determining car prices is a challenging task due to the influence of multiple factors, market fluctuations, and subjective elements like brand value and customer preferences. The aim is to address this complexity by analyzing historical data and leveraging machine learning techniques to create a system that helps sellers and buyers make informed decisions.

2. Introduction

Accurate pricing of used cars is critical in today's automotive market, as it facilitates informed decision-making for buyers, sellers, and dealerships. The goal of this project is to develop a machine-learning model that can predict the price of used cars based on a variety of vehicle attributes. By leveraging historical data and applying predictive analytics, this model aims to provide a reliable and efficient solution to estimating car prices. The project follows a structured pipeline, starting with data preprocessing and culminating in the deployment of a functional web application. The dataset comprises diverse features, including car specifications, seller details, and other key attributes, making it a valuable resource for building a robust predictive model. The core of the project is the Random Forest algorithm, chosen for its ability to handle complex relationships in data and provide high accuracy without extensive preprocessing. The deployed model offers real-time predictions through a user-friendly interface, making it accessible to a broad audience.

This document serves as a comprehensive guide, detailing each step of the project, from data preparation to deployment, and outlining areas for future improvement to enhance the model's accuracy and applicability.

3. Literature Survey

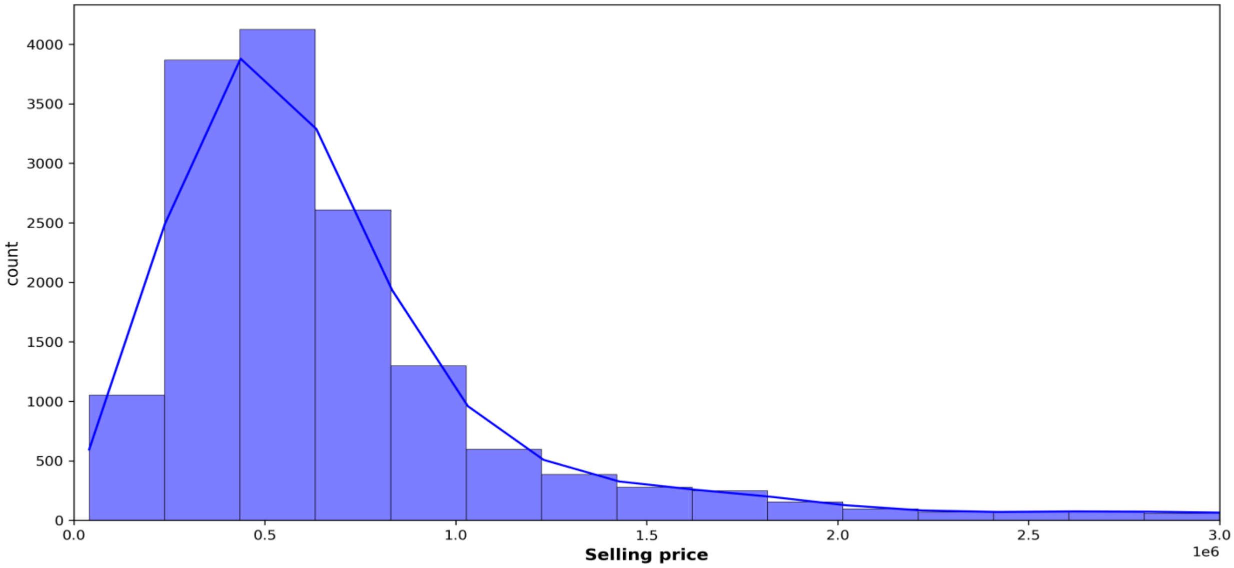
Car price prediction is a critical area of research in machine learning and data analytics, with applications in the automotive resale market, insurance valuation, and e-commerce platforms. Various machine learning models, including Linear Regression, Decision Trees, Random Forest, and advanced techniques like Gradient Boosting and XGBoost, have been explored to predict car prices accurately. These models leverage key features such as vehicle age, mileage, engine specifications, fuel type, transmission type, and brand to estimate prices. Effective preprocessing, such as handling missing data, eliminating redundant features, and encoding categorical variables using techniques like one-hot encoding, significantly impacts model performance. Ensemble learning methods have gained popularity due to their ability to combine multiple models and improve accuracy, while hyperparameter tuning further refines these models for optimal results. Recent studies also focus on feature importance analysis and advanced models like neural networks to capture complex relationships in the data. These approaches underscore the importance of robust methodologies in building reliable and accurate car price prediction systems.

For instance, a study titled **"How much is my car worth? A methodology for predicting used car prices using Random Forest"** investigates the application of Random Forest algorithms to forecast used car prices. The research highlights the effectiveness of this model in predicting transaction prices, achieving a training accuracy of 95.82% and a testing accuracy of 83.63%. The study emphasizes the importance of selecting the most correlated features to enhance prediction accuracy.

arXiv:1711.06970

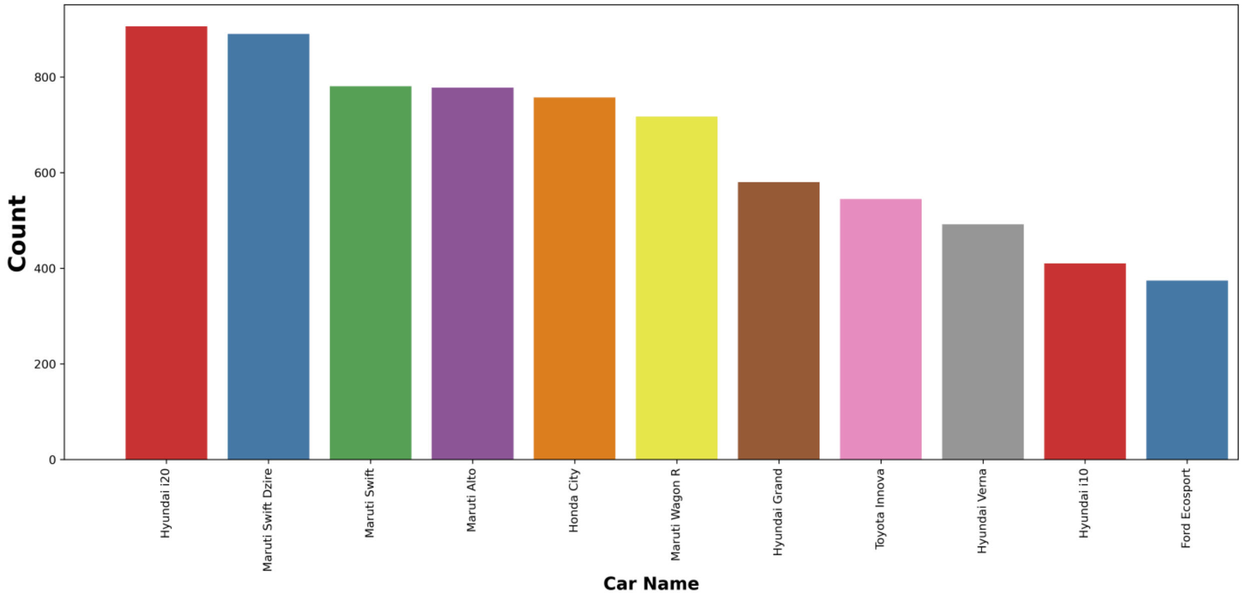
4. Exploratory Data Analysis

4.1 Selling Price Distribution



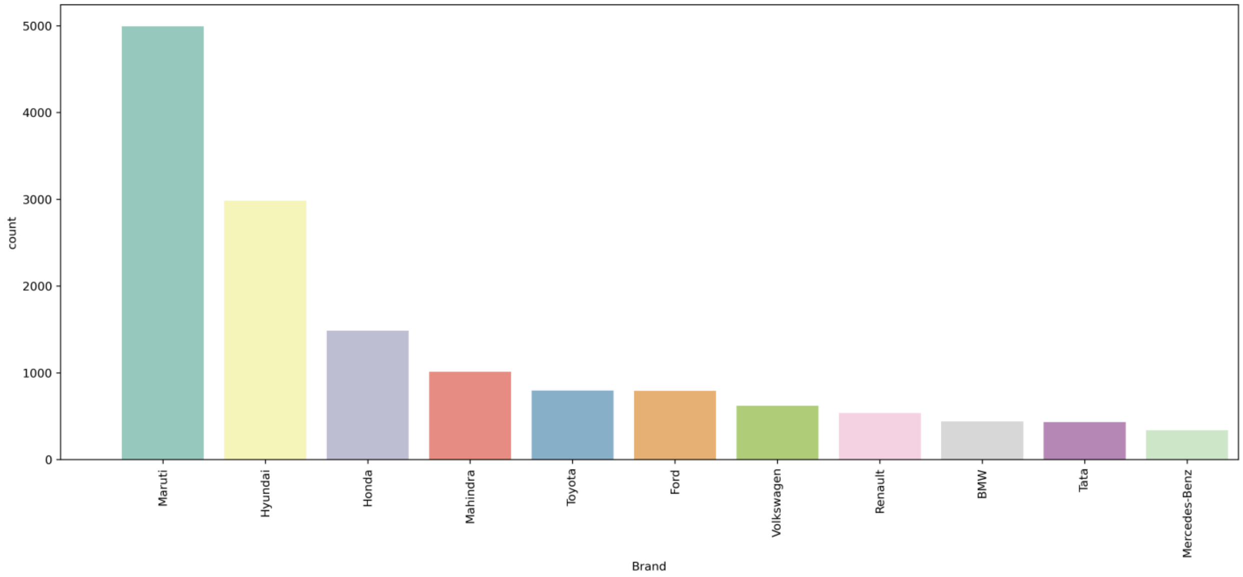
The histogram of the Target column shows a **right-skewed distribution**, meaning that most cars are priced on the lower end of the scale, while a smaller number of cars have very high selling prices.

4.2 Top 10 Most Sold Cars



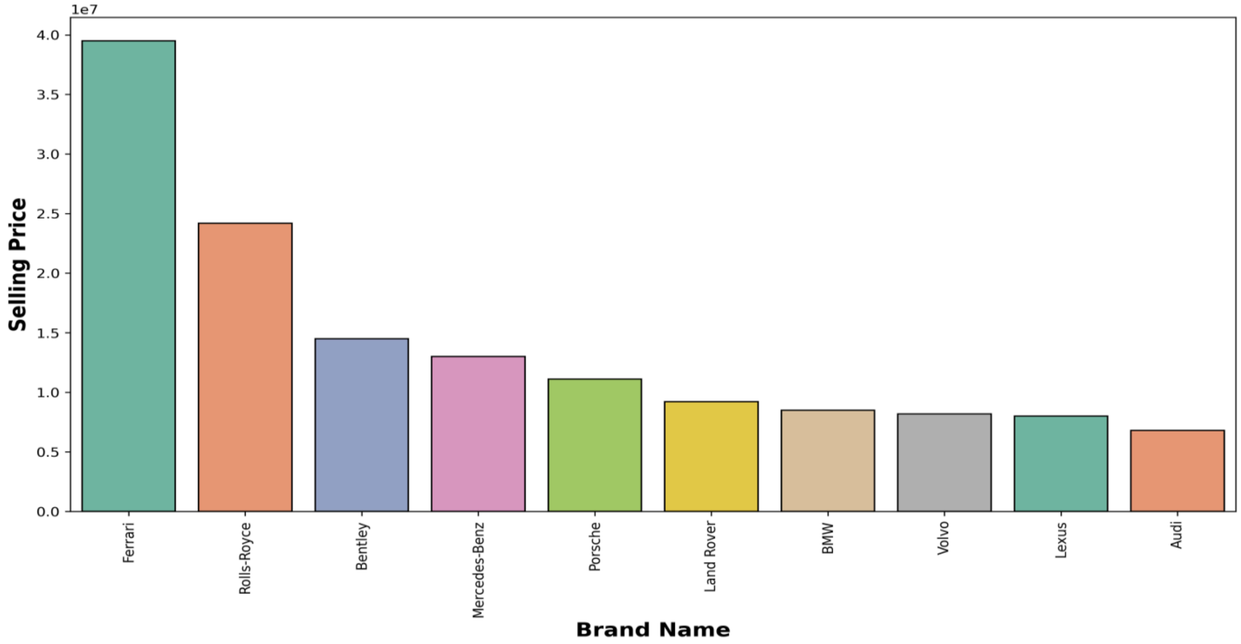
The figure shows that Hyundai i20 and Maruti Swift Dzire are the most sold cars, followed by Maruti Swift, Maruti Alto, and Honda City.

4.3 Top 10 Most Sold Brands



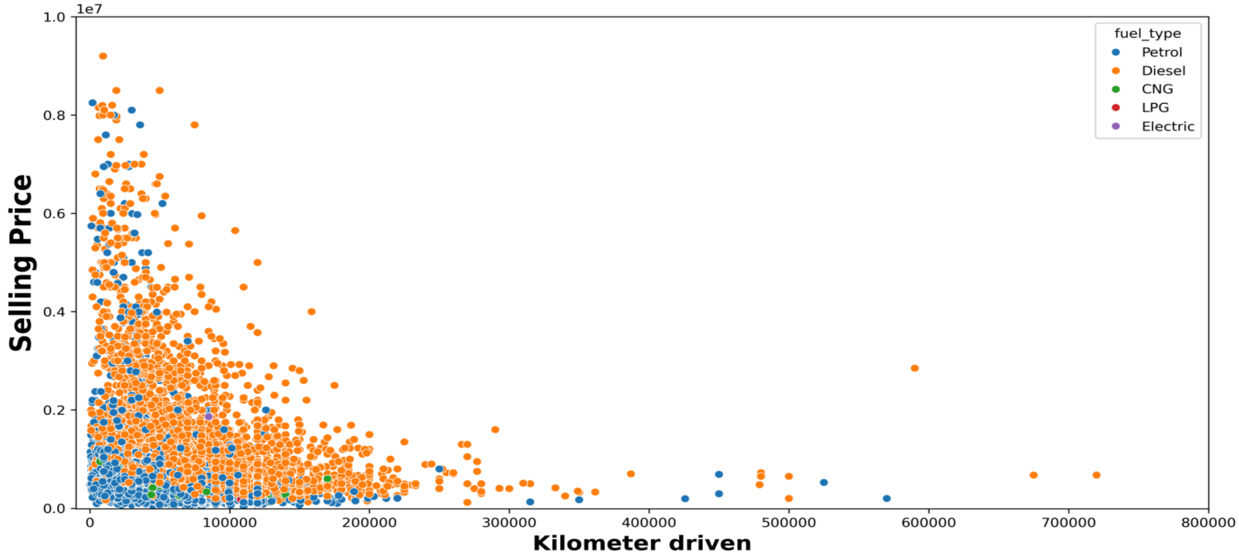
This figure shows that Maruti is the most popular car brand, significantly leading in sales, followed by Hyundai and Honda. Other brands like Mahindra, Toyota, and Ford also show notable sales figures.

4.4 Brand VS Highest Selling Price



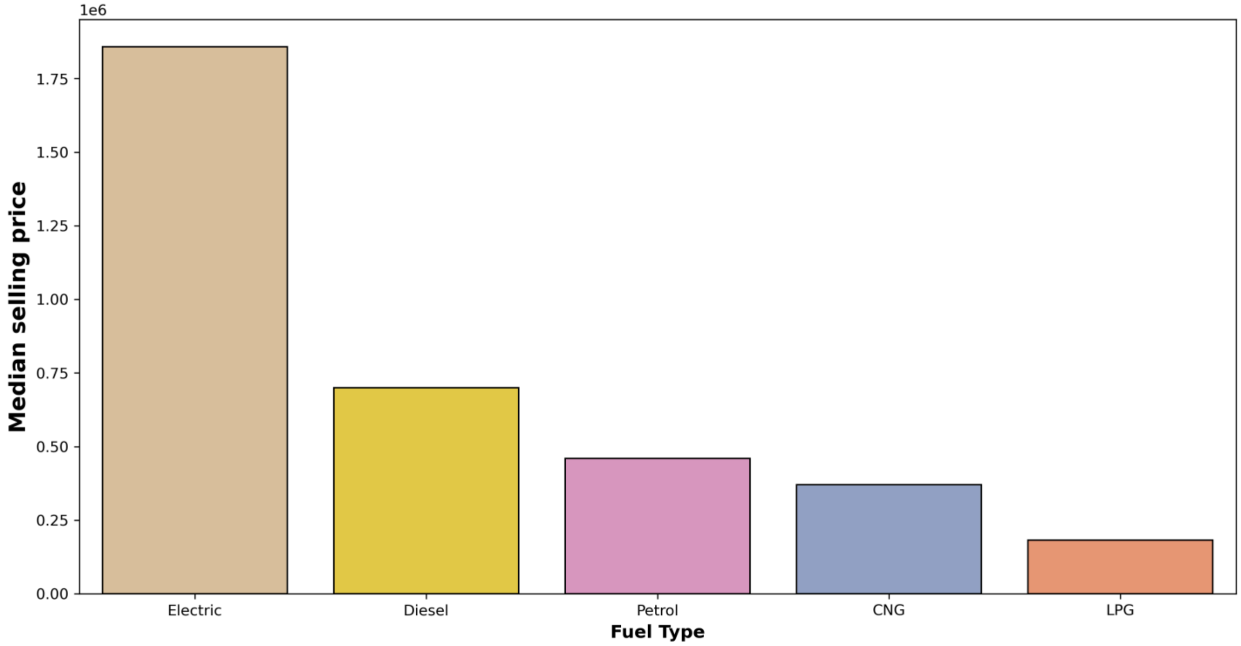
The costliest brand sold is Ferrari at 3.95 Crores, followed by Rolls-Royce at 2.42 Crores, highlighting the significant impact of brand name on the selling price of cars.

4.5 KM Driven VS Selling Price



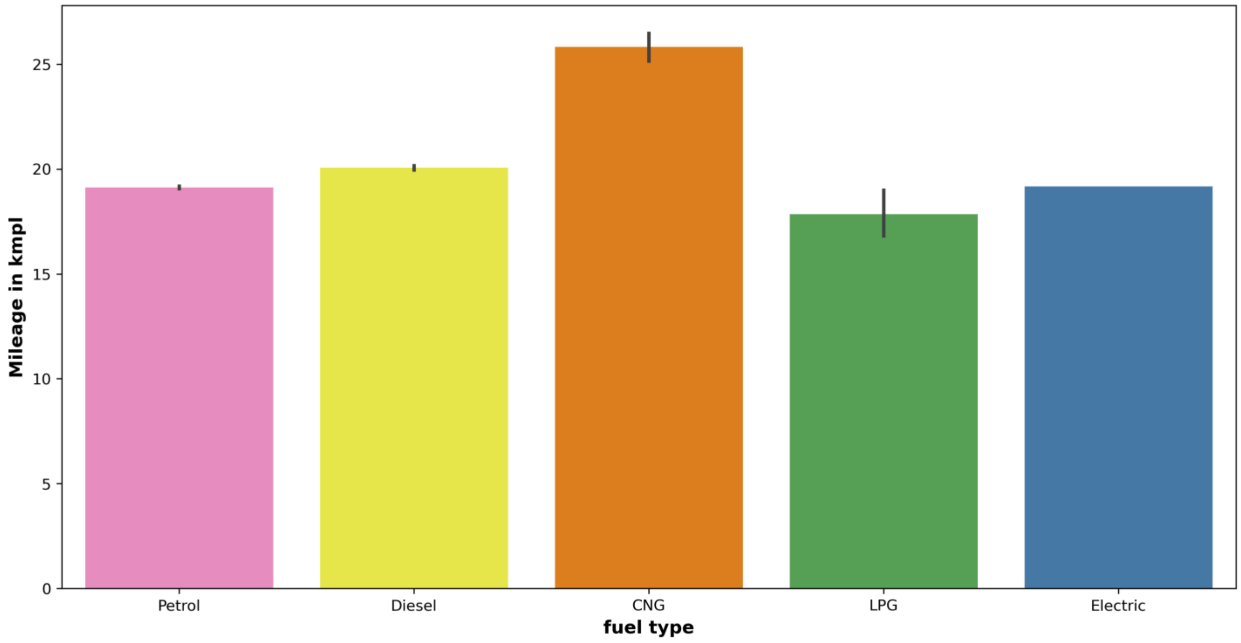
The scatter plot shows that as the kilometers driven increase, the selling price decreases, indicating a drop in value over time.

4.6 Fuel Type VS Selling Price

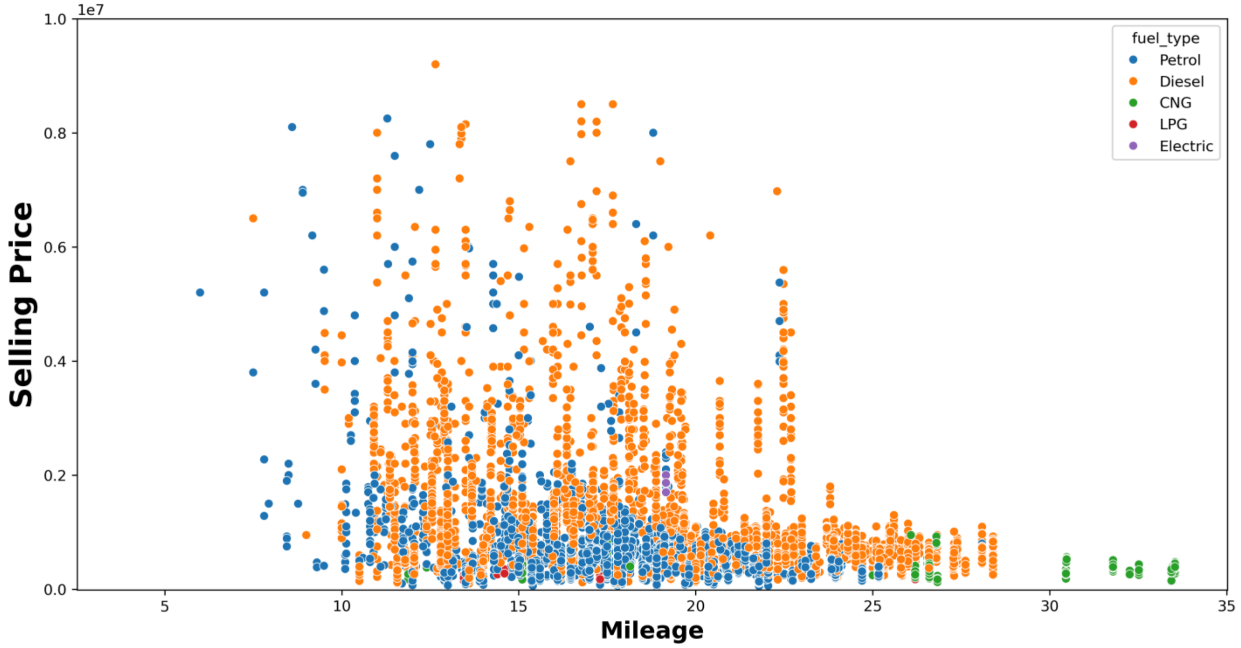


The figure shows that electric cars have the highest average selling price. This highlights that fuel type is an important factor influencing the target variable.

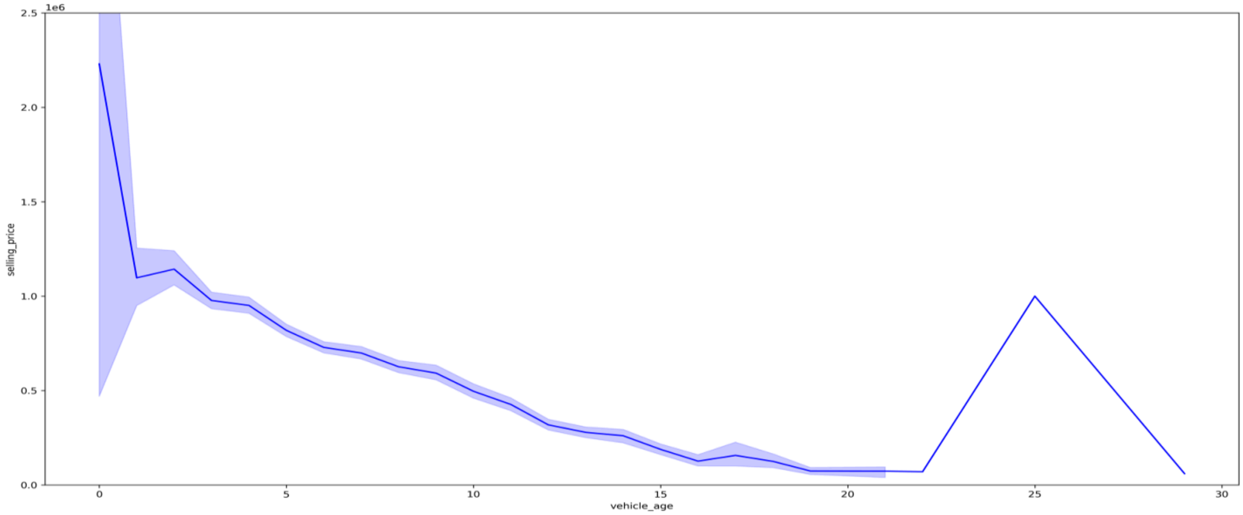
4.7 Fuel Type VS Mileage



4.8 Mileage VS Selling Price

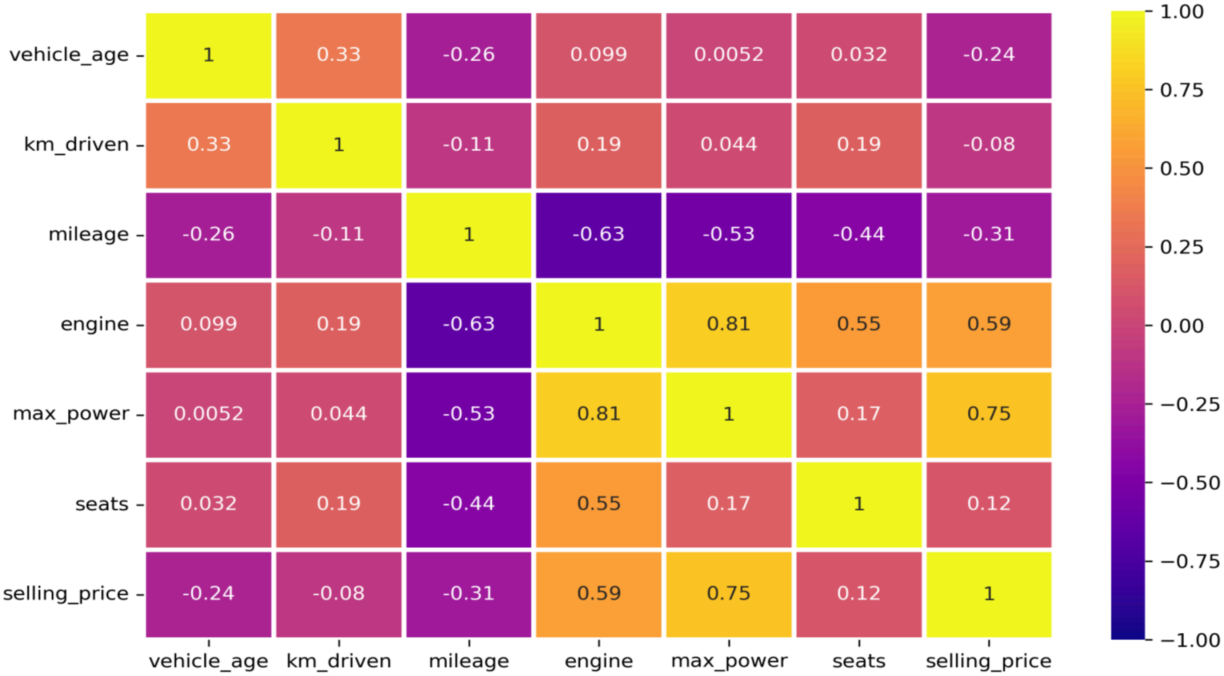


4.9 Vehicle Age VS Selling Price



The figure shows that the Vehicle age has a negative impact on Selling Price.

4.10 Heatmap



	vehicle_age	km_driven	mileage	engine	max_power	seats	selling_price
vehicle_age	1	0.33	-0.26	0.099	0.0052	0.032	-0.24
km_driven	0.33	1	-0.11	0.19	0.044	0.19	-0.08
mileage	-0.26	-0.11	1	-0.63	-0.53	-0.44	-0.31
engine	0.099	0.19	-0.63	1	0.81	0.55	0.59
max_power	0.0052	0.044	-0.53	0.81	1	0.17	0.75
seats	0.032	0.19	-0.44	0.55	0.17	1	0.12
selling_price	-0.24	-0.08	-0.31	0.59	0.75	0.12	1

Here Maximum power and Engine shows a High correlation.

5. Car Price Prediction Project - Code Explanation

5.1. Introduction

In this section, we will explain the steps and logic followed to predict car prices using machine learning. The process includes data importation, preprocessing, exploratory data analysis (EDA), and model building.

5.2. Importing Libraries

To begin, essential libraries are imported for data analysis and machine learning tasks. We use **pandas** for data manipulation and analysis, **numpy** for numerical operations, **matplotlib** and **seaborn** for visualization, and **plotly** for interactive charts. Additionally, **sklearn** provides the necessary tools for machine learning, such as preprocessing, model building, and evaluation. **Warnings** are used to suppress unnecessary warnings during execution, while `%matplotlib inline` ensures that visualizations are displayed directly within the Jupyter notebook.

5.3. Data Loading and Inspection

The dataset is loaded using `pd.read_csv()`, which contains 15411 entries and 13 columns. These columns represent different car attributes such as `car_name`, `brand`, `model`, `vehicle_age`, `km_driven`, `seller_type`, `fuel_type`, `transmission_type`, `mileage`, `engine`, `max_power`, `seats`, and `selling_price`. After loading the data, it is observed that there are no missing values across any of the columns. The data types include a mix of integers, floats, and object types, and the dataset occupies a memory space of 1.6MB.

5.4. Exploratory Data Analysis (EDA)

Through the exploratory data analysis, we examine the distribution and range of numerical features such as `vehicle_age`, `km_driven`, `mileage`, `engine`, `max_power`, `seats`, and `selling_price`. The selling price ranges from ₹40,000 to ₹39,50,000. For categorical variables like `car_name`, `brand`, `model`, `seller_type`, `fuel_type`, and `transmission_type`, we investigate the unique categories and their frequencies. The analysis provides a deeper understanding of the data, identifying key trends and distributions that will be useful for further analysis and model building.

5.5. Handling Special Cases

A special case is handled in the `seats` column, where some records have a value of 0, likely due to data entry errors. Specifically, models like `Honda City` and `Nissan Kicks` had 0 as the number of seats. These values are replaced with 5, which is considered a reasonable default. This correction ensures that the data is accurate and consistent.

5.6. Category Frequency Analysis

The `car_name` column consists of 121 unique categories, with the `Hyundai i20` being the most frequent, accounting for 5.88% of the data. The `brand` column contains 32 unique brands, with `Maruti` being the most common, representing 32.39% of the dataset. The `model` column has 120 unique categories, with `i20` being the most frequent. The `seller_type` column has three categories: `Individual`, `Dealer`, and `Trustmark Dealer`, with `Dealer` being the most common (61.90%). Regarding fuel types, the dataset shows that 49.59% of cars use `Petrol`, while `Electric` cars are the least frequent, making up just 0.03% of the records. The `transmission_type` column has two categories: `Manual` and `Automatic`, with `Manual` being the dominant category, appearing in 79.33% of the data.

5.7 Converting Categorical Columns into Numerical Using OneHotEncoder in Pipeline

In the car price prediction project, we use a **Pipeline** to simplify the process of converting categorical variables into numerical values. Specifically, **OneHotEncoder** is applied within the pipeline to handle the encoding of categorical features automatically. This process transforms categorical columns into binary columns, each representing a distinct category. By incorporating the encoding step directly into the pipeline, we ensure that it is consistently applied across both training and testing datasets, eliminating the need for manual handling. This approach also minimizes the risk of data leakage, ensuring that our model is trained with clean, well-processed data. The pipeline approach makes the workflow more streamlined and efficient, particularly when working with multiple categorical variables such as `car_name`, `seller_type`, `fuel_type`, `transmission_type`, and `seats`. Additionally, a **ColumnTransformer** is used to handle numerical features differently from categorical ones. For numerical features like `vehicle_age`, `km_driven`, `mileage`, `engine`, and `max_power`, we apply **StandardScaler** to standardize the values, ensuring uniform scaling across these features.

5.8 Model Building & Evaluation

We evaluated multiple models for car price prediction, including **Linear Regression**, **Decision Tree Regressor**, and **Random Forest Regressor**. The **Linear Regression** model gave an R² score of 0.793, while the **Decision Tree Regressor** performed slightly better with an R² score of 0.889 and an RMSE of 288,791.39. However, the **Random Forest Regressor** outperformed both with an R² score of 0.939 and a significantly lower RMSE of 215,034.12. It also showed impressive performance on the training set, with an MSE of 15,282,068,022.25, RMSE of 123,620.66, and an R² score of 0.9812, indicating a strong fit. Given its superior performance, we plan to proceed with hyperparameter tuning for the Random Forest model to further optimize its accuracy.

Here are the evaluation metrics for all the models:

1. **Linear Regression**:
    - **Mean Squared Error (MSE)**: 155,553,927,566.92
    - **Root Mean Squared Error (RMSE)**: 394,403.26
    - **R² Score**: 0.793
2. **Decision Tree Regressor**:
    - **Mean Squared Error (MSE)**: 155,553,927,566.92
    - **Root Mean Squared Error (RMSE)**: 288,791.39
    - **R² Score**: 0.889
3. **Random Forest Regressor**:
    - **Mean Squared Error (MSE)**: 46,239,674,645.17
    - **Root Mean Squared Error (RMSE)**: 215,034.12
    - **R² Score**: 0.939
    - **Training Set**:
        - **Mean Squared Error (MSE)**: 15,282,068,022.25
        - **Root Mean Squared Error (RMSE)**: 123,620.66
        - **R² Score**: 0.9812

The **Random Forest Regressor** outperforms the other models in terms of MSE, RMSE, and R² score, both on the test set and training set.

5.9 Random Forest with Hyperparameter Tuning

After performing **Hyperparameter Tuning** on the **Random Forest Regressor**, we achieved the following results: The **Mean Squared Error (MSE)** is 43,454,328,157.27, the **Root Mean Squared Error (RMSE)** is 208,457.02, and the **R² Score** is 0.942. Additionally, on the **training set**, the model performs with an **RMSE** of 23,684,057,440.51, a **Mean Absolute Error (MAE)** of 153,896.26, and an **R² Score** of 0.9708. These improved results suggest that the Random Forest model is performing well and we are now moving towards further enhancing it into a more advanced model.

5.10 Advanced Models Comparison

The **Gradient Boosting Regressor** stands out as the best-performing model in terms of both **Root Mean Squared Error (RMSE)** and **R² Score**. With an **RMSE** of 208,457.02 and an **R² Score** of 0.942, it outperforms all other models. The **Random Forest Regressor** follows closely with an **RMSE** of 215,034.12 and an **R² Score** of 0.939. The **XGBoost** model, while competitive, has a slightly higher **RMSE** of 249,401.21 and **R² Score** of 0.917. The **Decision Tree Regressor** and **Linear Regression** perform the least, with **RMSE** values of 288,791.39 and 394,403.26, respectively, along with significantly lower **R² Scores**.

Conclusion:

Based on the above evaluation, the **Gradient Boosting Regressor** is selected as the final model due to its superior performance with an **RMSE** of 208,457.02 and an **R² Score** of 0.9423. While other models like **Random Forest** and **XGBoost** showed competitive results, Gradient Boosting proved to be the most balanced model for both training and testing. The **Linear Regression** and **Decision Tree** models, however, demonstrated relatively poor performance. Therefore, the **Gradient Boosting Regressor** will be saved for further testing of the data using the **Pickle** library for deployment.

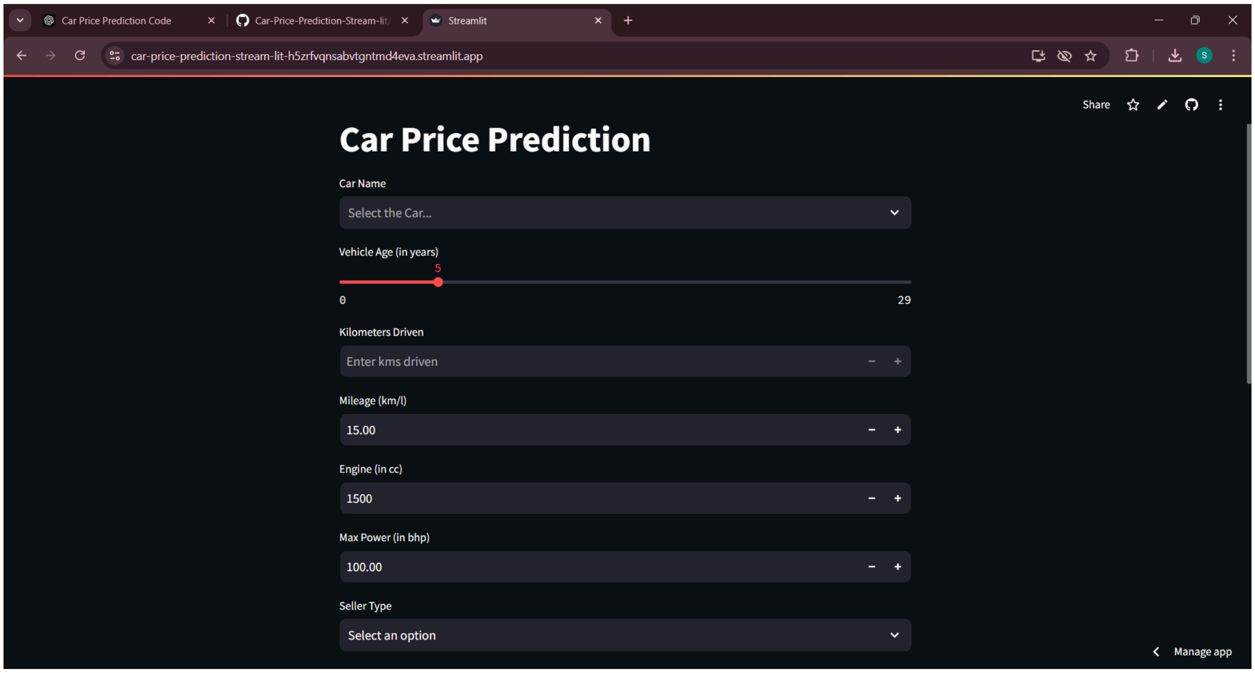
6. Car Price Prediction Web App Using Streamlit and Gradient Boosting Model

This Stream lit application is designed to predict the price of a car based on user inputs. First, it allows users to select various car features such as car name, vehicle age, kilometers driven, mileage, engine size, maximum power, seller type, fuel type, transmission type, and the number of seats through interactive input fields. The application includes dropdowns and sliders for each of these features, ensuring that the user selects valid options.

Once the user provides all the required details and clicks the "Predict Price" button, the data is structured into a DataFrame and passed to the pre-trained Gradient Boosting model, which was previously saved using pickle. This model, which has been tuned for optimal performance, then generates a prediction based on the input data. The predicted car price is displayed to the user, helping them get an estimate of the car's value based on the provided attributes.

Throughout the process, validation checks ensure that all selections are made correctly, and warnings are shown if any field is missing or incorrectly filled. Once all inputs are valid, a success message confirms that the data is ready for prediction.

7. Result


Car Price Prediction Code
Car-Price-Prediction-Stream-lit
Streamlit
car-price-prediction-stream-lit-h5zrfvqnsabvtgntmd4eva.streamlit.app
Share
Manage app

Car Price Prediction

Car Name: Select the Car...

Vehicle Age (in years): 5 (0–29)

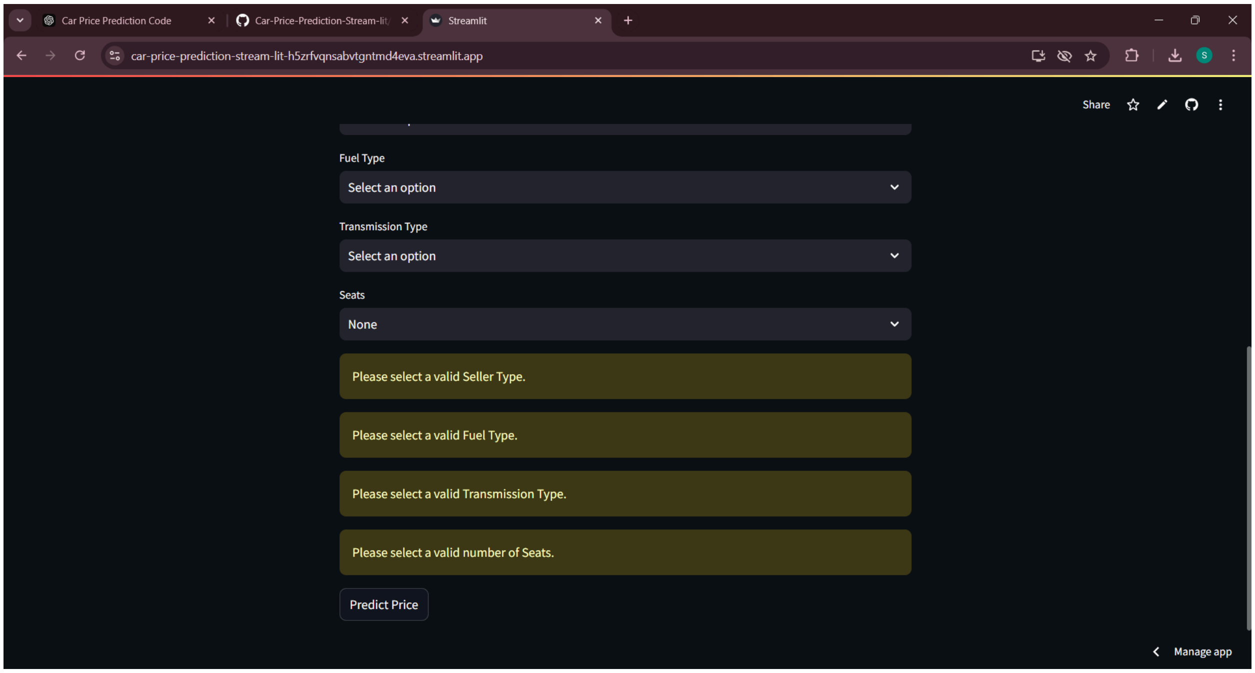
Kilometers Driven: Enter kms driven

Mileage (km/l): 15.00

Engine (in cc): 1500

Max Power (in bhp): 100.00

Seller Type: Select an option


Car Price Prediction Code
Car-Price-Prediction-Stream-lit
Streamlit
car-price-prediction-stream-lit-h5zrfvqnsabvtgntmd4eva.streamlit.app
Share
Predict Price
Manage app

Fuel Type: Select an option

Transmission Type: Select an option

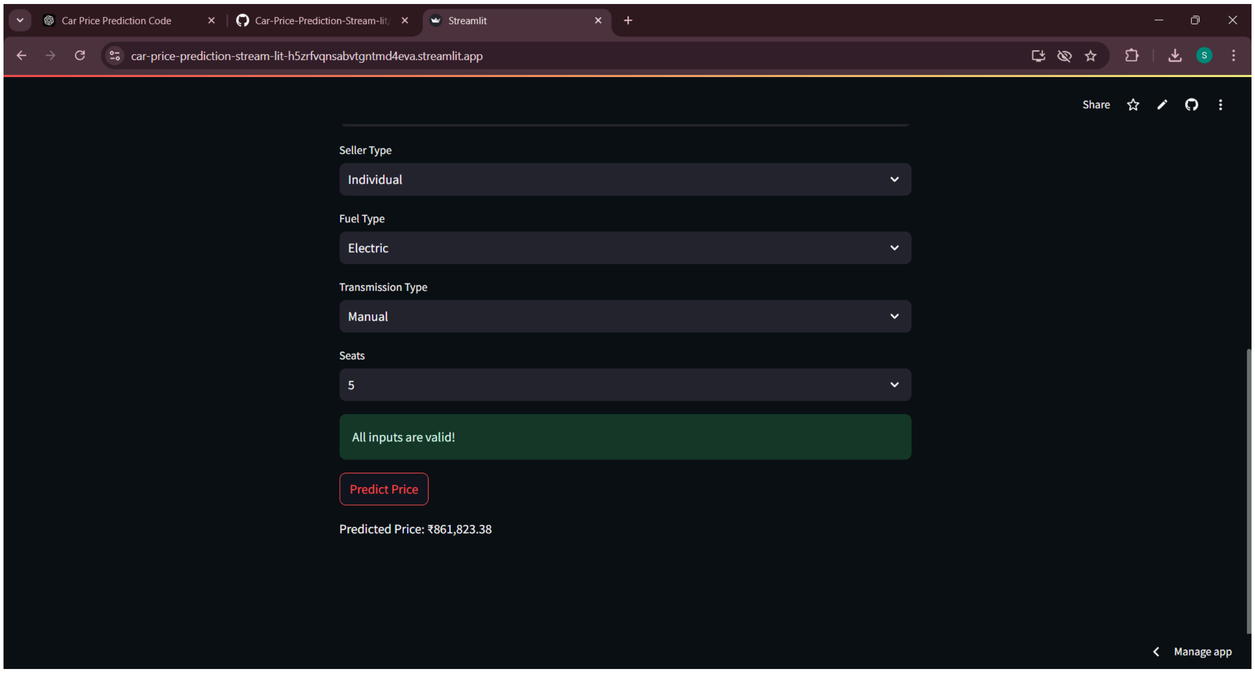
Seats: None

Please select a valid Seller Type.

Please select a valid Fuel Type.

Please select a valid Transmission Type.

Please select a valid number of Seats.


Car Price Prediction Code
Car-Price-Prediction-Stream-lit/
Streamlit
car-price-prediction-stream-lit-h5zrfvqnsabvtgntmd4eva.streamlit.app
Share
Manage app

Seller Type

Individual

Fuel Type

Electric

Transmission Type

Manual

Seats

5

All inputs are valid!

Predict Price

Predicted Price: ₹861,823.38

The screenshot showcases the user interface of the **Car Price Prediction Web App** built with Stream lit. It highlights the various input fields where users can select or enter car details such as the car's name, age, kilometers driven, mileage, engine capacity, max power, seller type, fuel type, transmission type, and number of seats. The app ensures that all necessary information is provided before allowing the user to proceed.

Once all inputs are filled, the user can click on the "Predict Price" button to view the predicted price of the car. The result is displayed clearly on the screen, showcasing the predicted car price in INR (Indian Rupees), providing an easy and intuitive way for users to estimate the price of a car based on its features.

8. Conclusion

In this project, we developed a **Car Price Prediction Model** using various machine learning techniques and deployed it through a **Streamlit web interface** for user-friendly interaction.

The **Jupyter Notebook** served as the foundation for model training and evaluation. We explored multiple machine learning models, including **Linear Regression**, **Decision Tree Regressor**, **Random Forest Regressor**, and **Gradient Boosting Regressor**, and evaluated their performance using key metrics like RMSE, R^2 score, and Mean Absolute Error (MAE). Among these, the **Gradient Boosting Regressor** emerged as the best-performing model, with the lowest RMSE and the highest R^2 score, making it the most suitable choice for car price prediction.

After hyperparameter tuning, **Random Forest Regressor** also showed promising results but was ultimately overshadowed by Gradient Boosting. The model's performance for both training and testing sets demonstrated robust predictive accuracy.

To make the model more accessible, we built a **Streamlit web interface** that allows users to input car details and receive an instant prediction of the car's price. The web app ensures an intuitive experience for users by validating inputs and displaying the predicted price after model execution.

In summary, the **Gradient Boosting Regressor** is the final selected model, with exceptional performance for car price prediction, and the **Streamlit web interface** allows for easy and interactive access to the model's predictions, making it practical for real-world use.

References

- **Car Price Prediction Using Machine Learning: A Survey**
  A review of machine learning models for car price prediction, focusing on feature selection and model evaluation.
  Source: *https://www.researchgate.net/publication/336222794_Car_Price_Prediction_Using_Machine_Learning_A_Survey*.

- **Prediction of Car Price Using Machine Learning Algorithms**
  A study on applying machine learning algorithms for car price prediction and model comparisons.
  Source: *https://www.sciencedirect.com/science/article/pii/S1877056820303656*.

- **Machine Learning Algorithms for Predicting Car Prices**
  A comparison of various machine learning algorithms for car price prediction.
  Source: *https://www.springerprofessional.de/en/machine-learning-algorithms-for-predicting-car-prices/18368974*.

- **Building an Interactive Car Price Prediction Web App with Streamlit**
  A tutorial on creating a car price prediction app using Python, Streamlit, and machine learning.
  Source: *https://medium.com/@yozizicoding/building-an-interactive-car-price-prediction-web-app-with-streamlit-9c212b6f40be*.

- **Car Price Prediction Using Machine Learning**
  An article on implementing machine learning models for car price prediction.
  Source: *https://towardsdatascience.com/car-price-prediction-using-machine-learning-b624f8a8a746*

- **GitHub Repository: Car Price Prediction Project**
  The official GitHub repository with code for car price prediction and deployment using Streamlit.
  Source: *https://github.com/sulaikhanazrin/Car-Price-Prediction-Stream-lit*.
  https://github.com/Arc-1327/Car-Price-Prediction.git