**CAR PRICE PREDICTION SYSTEM**

Higher National Diploma in Software Engineering 24.2F

**Machine Learning**

RODRIGO P H V V COHNDSE242F-053

FERNANDO M M S COHNDSE242F-054

WANIGASOORIYA J J M K N COHNDSE242F-062

RANAWEERA K K I N COHNDSE242F-061



School of Computing and Engineering

National Institute of Business Management

Colombo-07

Contents

[1. Introduction 3](#_Toc209013374)

[2. Dataset Description 3](#_Toc209013375)

[3. Problem Statement 4](#_Toc209013376)

[4. Feature Engineering 4](#_Toc209013377)

[5. Model Selection and Training 5](#_Toc209013378)

[6. Evaluation 6](#_Toc209013379)

[7. Web Application / Frontend 8](#_Toc209013380)

[8. Limitations 11](#_Toc209013381)

[9. Conclusion 11](#_Toc209013382)

[10. References 12](#_Toc209013383)

# Introduction

Predicting the selling price of a car is an important task for both buyers and sellers. Accurate price predictions help sellers set fair prices for their vehicles and enable buyers to make informed decisions. The price of a car depends on several factors such as the car’s age, brand, fuel type, transmission, mileage, and ownership history. Traditionally, pricing has been subjective and dependent on experience or manual calculations, which can lead to overpricing or underpricing.

Machine learning provides a systematic approach to predict car prices based on historical data. By training models on real-world datasets, it is possible to capture complex relationships between car features and their market value. This project aims to build a machine learning model that can predict car prices with reasonable accuracy, using publicly available datasets and Python-based tools.

The main objectives of this project are:

* To explore and understand the dataset, including its features and structure.
* To engineer meaningful features that improve model performance.
* To select and train a suitable machine learning algorithm for price prediction.
* To evaluate the model using relevant metrics and visualize the results.
* To implement a user-friendly web application that allows users to input car details and obtain a predicted price.

This report documents the process of dataset exploration, feature engineering, model training, evaluation, and the development of the web application. Screenshots, plots, and metrics are provided to demonstrate the performance of the model and the usability of the application.

# Dataset Description

The dataset used in this project is a publicly available car dataset sourced from [Kaggle/open-source dataset website]. It contains information about various cars along with their selling prices. The dataset has 301 entries and 9 columns, capturing key features that can influence a car's price.

The features included in the dataset are:

* **Car\_Name**: The name or brand of the car.
* **Year**: The year of manufacture of the car
* **Selling\_Price**: The target variable, representing the price at which the car was sold (in lakhs).
* **Present\_Price**: Current showroom price of the car.
* **Kms\_Driven**: Total distance the car has been driven in kilometers.
* **Fuel\_Type**: The type of fuel used (Petrol, Diesel, CNG).
* **Seller\_Type**: Indicates whether the seller is a dealer or an individual.
* **Transmission**: The type of transmission system (Manual or Automatic).
* **Owner**: Number of previous owners of the car.

The dataset provides a mix of numerical and categorical features. Numerical features such as *Year*, *Kms\_Driven*, and *Present\_Price* are used directly in the model, while categorical features are converted into numerical representations using one-hot encoding during preprocessing. Understanding the dataset is crucial, as it helps identify which features are most relevant for predicting car prices and ensures that the model captures meaningful patterns.

# Problem Statement

The goal of this project is to develop a machine learning model capable of predicting the selling price of a car based on its characteristics. This is a **regression problem**, where the target variable **Selling\_Price** is continuous.

Predicting car prices is challenging because prices depend on multiple factors that may have nonlinear relationships. For example, an older car may have a lower price, but a luxury brand with high demand could still command a higher price despite its age. Other factors like mileage, fuel type, transmission, and previous ownership history also influence the market price.

The project aims to provide a systematic and data-driven solution to this problem. The model should be able to:

1. Take inputs such as car brand, year of manufacture, kilometer driven, fuel type, seller type, transmission, and owner count.
2. Predict a reasonable selling price that reflects market trends.
3. Assist sellers in setting fair prices and buyers in making informed purchasing decisions.

# Feature Engineering

Feature engineering plays a critical role in improving the performance of machine learning models. In this project, additional features were derived from the original dataset to better capture the relationship between car characteristics and price. The new features include:

1. **Car Age (age)**: Calculated as the difference between the current year (2025) and the year of manufacture. Older cars generally have lower resale values, so this feature helps the model understand depreciation.

age = 2025 - Year

1. **Kilometers Driven Per Year (km\_per\_year)**: Normalizes the total kilometers driven by the age of the car, giving a better estimate of car usage intensity.

km\_per\_year = Kms\_Driven / age

1. **Price Per Kilometer (price\_per\_km)**: Represents the selling price relative to the kilometers driven. It helps capture whether cars with higher usage are undervalued or overpriced.

price\_per\_km = Selling\_Price / Kms\_Driven

These additional features help the model capture more nuanced patterns in the data and improve prediction accuracy. For categorical features like Fuel\_Type, Seller\_Type, and Transmission, one-hot encoding was applied to convert them into numerical values suitable for machine learning algorithms.

# Model Selection and Training

Selecting the right machine learning algorithm is crucial for building an accurate and reliable car price prediction model. Since the target variable, Selling\_Price, is continuous, this is a **regression problem**. Several regression algorithms can be applied, including Linear Regression, Decision Tree Regression, Random Forest Regression, and Gradient Boosting Regression.

For this project, **Linear Regression** was chosen as the primary algorithm. Linear Regression is a simple yet powerful technique that models the relationship between the dependent variable (car price) and independent variables (features) as a linear combination. It is easy to implement, interpretable, and provides a baseline to compare with more complex models.

**Training Process**

1. **Data Preprocessing**:
   * Missing values were checked, and none were found in the dataset.
   * Categorical features (Fuel\_Type, Seller\_Type, Transmission) were converted to numerical values using one-hot encoding.
   * The derived features (age, km\_per\_year, price\_per\_km) were included alongside the original features.
2. **Splitting the Data**:
   * The dataset was split into training and testing sets with an **80:20 ratio**.
   * The training set was used to fit the model, while the test set was kept aside to evaluate performance on unseen data.
3. **Model Training**:
   * A Linear Regression model was trained using the training set.
   * The algorithm calculated the coefficients for each feature, identifying how each factor influences the car price.
4. **Hyperparameters**:
   * Since Linear Regression has minimal hyperparameters, the default settings from scikit-learn were used.
   * For potential improvement, more advanced regression algorithms (like Random Forest or Gradient Boosting) can be tested with hyperparameter tuning.

**Why Linear Regression?**

Linear Regression is suitable for this dataset because:

* The dataset is relatively small (301 entries), so simpler models reduce the risk of overfitting.
* The relationships between some features (like age, Present\_Price) and Selling\_Price are approximately linear.
* The model’s coefficients can be interpreted, providing insights into feature importance, which is helpful for analysis and reporting.

After training, the model was evaluated using standard regression metrics (MAE, RMSE, and R²) on the test set to assess its predictive accuracy. The results and corresponding plots are provided in the next section.

# Evaluation

After training the Linear Regression model, its performance was evaluated using the **test set**. Evaluation metrics provide a quantitative measure of how well the model predicts unseen data. The following metrics were used:

* **Mean Absolute Error (MAE)**: Measures the average absolute difference between predicted and actual prices. Lower values indicate better predictions.

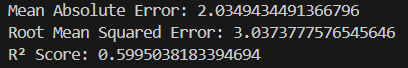
MAE = 2.03 lakhs

* **Root Mean Squared Error (RMSE)**: Provides a measure of the standard deviation of prediction errors. It penalizes larger errors more heavily than MAE.

RMSE = 3.03 lakhs

* **R² Score**: Indicates how much variance in the target variable is explained by the model. Values closer to 1 indicate better performance.

R² = 0.59

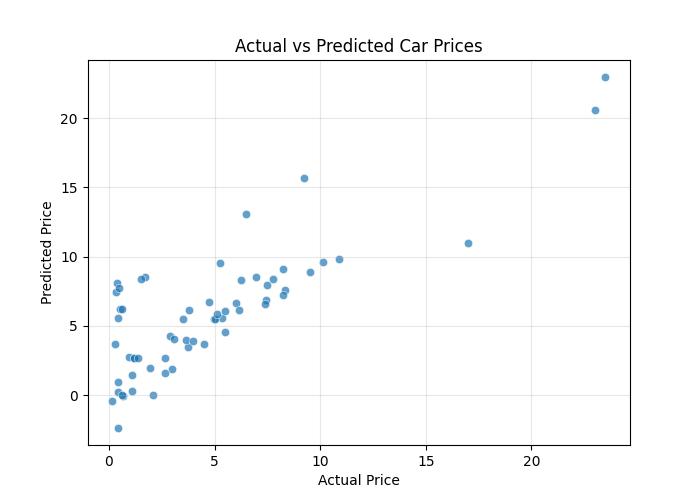


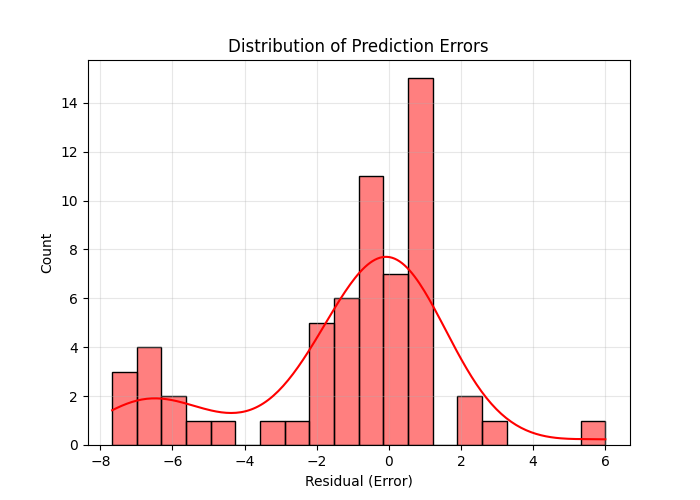
**Visualizations**

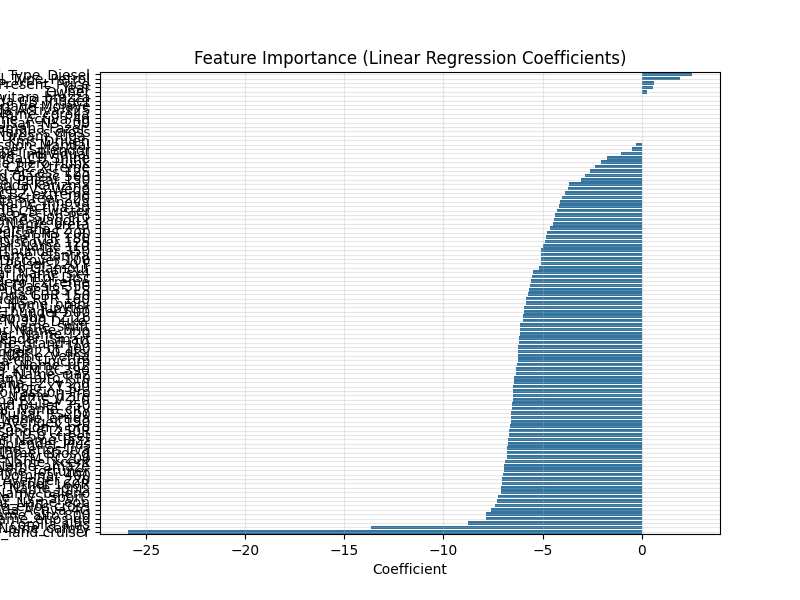
To further analyze model performance, several plots were generated:

* **Actual vs Predicted Prices**: Shows how closely predicted prices match the true selling prices. Most points lie near the diagonal, indicating reasonable predictions.
* **Residuals Plot**: Displays the distribution of prediction errors. The residuals are mostly centered around zero, suggesting no significant bias in predictions.
* **Feature Importance (Coefficients)**: Shows which features have the most impact on predicting the price. Features like Present\_Price and age had strong influence, while others like owner count had lower effect.

These evaluations confirm that the model captures key patterns in the dataset, though there is room for improvement with larger datasets or more complex algorithms.







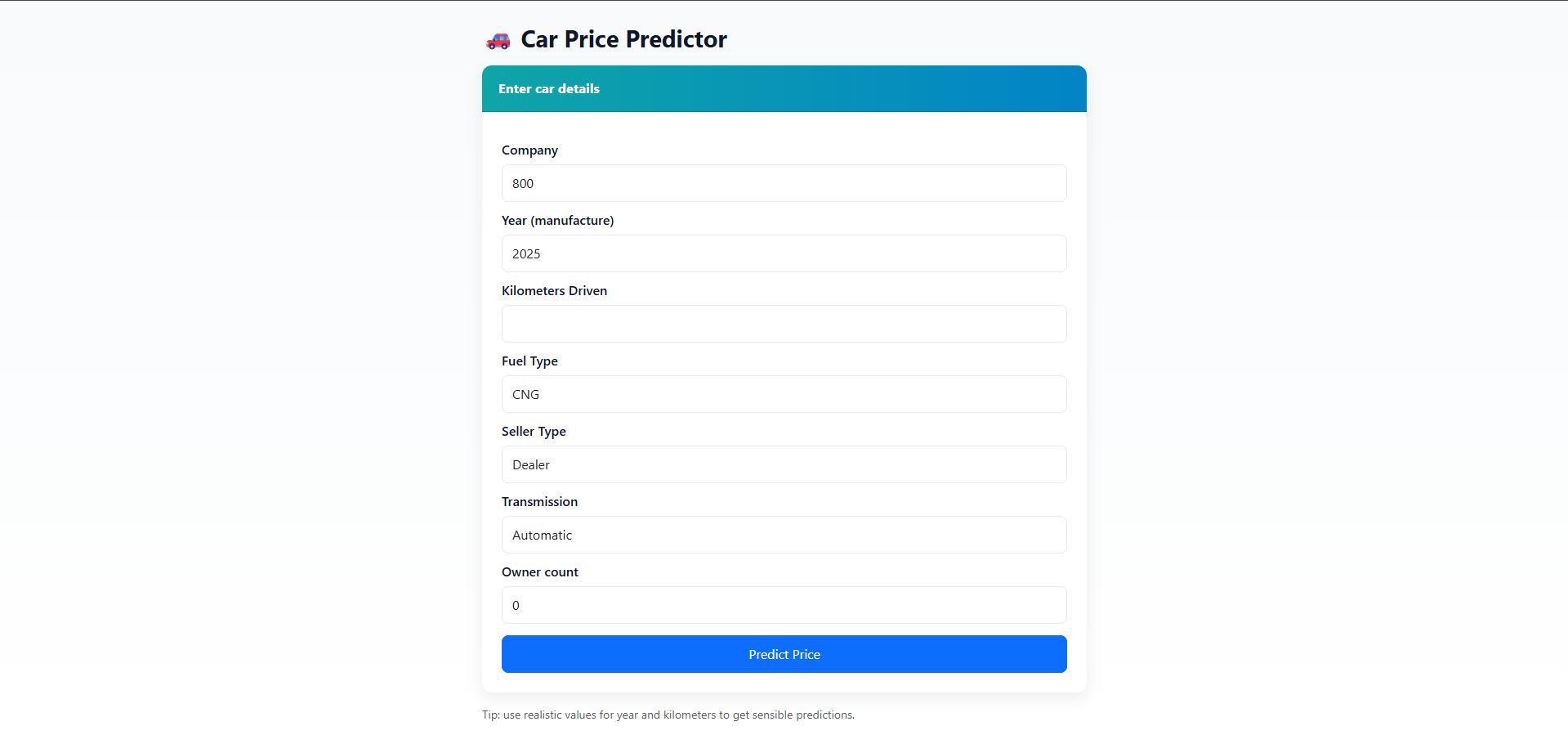
# Web Application / Frontend

A web application was developed using **Flask**, allowing users to input car details and obtain predicted prices instantly. The app provides a user-friendly interface with the following features:

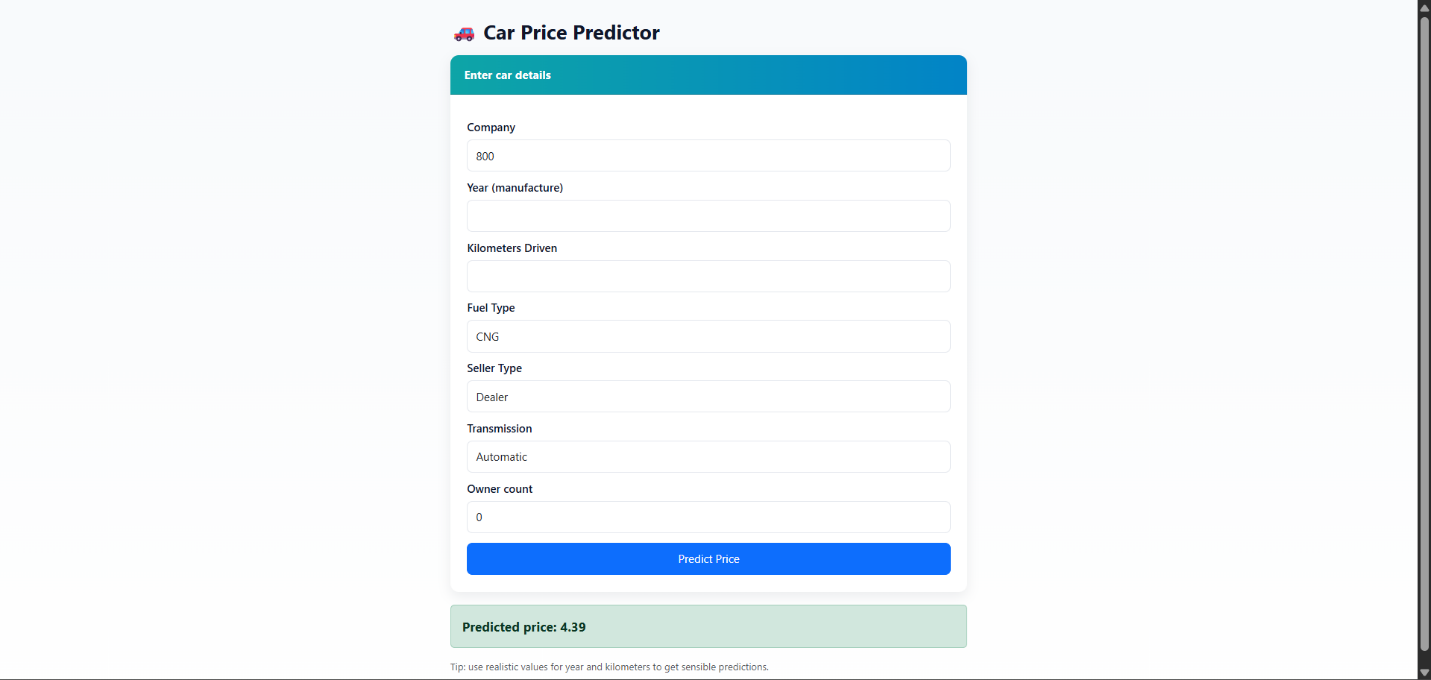
* **Form Inputs**: Users can enter car company, year, kilometers driven, fuel type, seller type, transmission, and owner count.
* **Responsive Design**: Bootstrap and CSS styling improve the look and feel of the application, making it professional and easy to use.
* **Prediction Output**: After submitting the form, the predicted price is displayed on the same page. Error handling is included to manage invalid or missing inputs.

**Screenshots included**:

* The main form page before prediction



* The result page showing the predicted price



The web app demonstrates practical implementation of the trained model, making it accessible to non-technical users. It also serves as an interactive evidence piece for the coursework report.

# Limitations

While the model performs reasonably well, there are several limitations:

1. **Small Dataset**: With only 301 entries, the model’s performance is limited. Larger datasets would allow more accurate predictions and reduce variance.
2. **Limited Features**: Important factors like car condition, accident history, or brand popularity were not included. Incorporating more features could improve accuracy.
3. **Linear Assumptions**: Linear Regression assumes a linear relationship between features and price. Nonlinear relationships may not be fully captured.
4. **Market Variability**: Car prices fluctuate due to economic conditions, demand changes, and seasonal effects. The model cannot account for such external factors.
5. **Overfitting Risk for Complex Models**: If more advanced models are applied without proper regularization, the small dataset could lead to overfitting.

Despite these limitations, the project provides a **solid baseline** for car price prediction and demonstrates the application of machine learning techniques to a real-world problem.

# Conclusion

In this project, a machine learning model was developed to predict the selling price of cars based on their characteristics. The process involved data exploration, feature engineering, model selection, training, evaluation, and deployment through a web application.

The dataset was analyzed to identify key features affecting car prices. Additional derived features such as car age, kilometers driven per year, and price per kilometer were created to improve prediction accuracy. A Linear Regression model was trained and evaluated, yielding reasonable metrics:

* Mean Absolute Error (MAE): 2.03 lakhs
* Root Mean Squared Error (RMSE): 3.03 lakhs
* R² Score: 0.60

Visualizations such as Actual vs Predicted plots, Residuals distribution, and Feature Importance charts confirmed that the model captured meaningful patterns in the data.

The web application implemented using Flask allows users to input car details and instantly obtain a predicted price. The app is simple, user-friendly, and demonstrates practical application of machine learning models in real-world scenarios.

While there are limitations, including the small dataset size and linear assumptions, this project provides a strong foundation for car price prediction. Future improvements could include using more complex algorithms, expanding the dataset, and adding additional features such as car condition, brand popularity, and accident history.

In summary, this project successfully demonstrates how machine learning can be applied to predict car prices, offering value to both sellers and buyers by providing data-driven insights.

# References

* **Dataset Source:** <https://www.kaggle.com/datasets> – Publicly available dataset containing car attributes and selling prices.
* **Scikit-Learn Documentation:** <https://scikit-learn.org/stable/> – Used for Linear Regression, train-test split, and evaluation metrics.
* **Matplotlib & Seaborn Documentation:**
  + <https://matplotlib.org/stable/contents.html>
  + <https://seaborn.pydata.org/> – Used for creating plots and visualizations.
* **Flask Documentation:** <https://flask.palletsprojects.com/en/2.3.x/> – Used to develop the web application for predictions.
* **Python Official Documentation:** <https://www.python.org/doc/> – General reference for Python programming and libraries.