Car Dheko –

Used Car Price Prediction

Project Report

Submitted By:

Tharanipriyadevi G

Table of contents	
1.	Summary
2.	Introduction
	Problem Statement
	Objective
	Project scope
3.	Data preprocessing
	Data overview
	Data cleaning and preprocessing
4.	Exploratory Data Analysis (EDA)
	Objective of EDA
	Key Insights
	Impact of EDA on model development
5.	Model Development
	Methodology
	Algorithms i. Linear Regression ii. Decision Tree Regressor iii. Random Forest Regressor iv. Gradient Boosting Regressor (GBR)
	Model Evaluation
6.	Model Deployment (streamlit application)

1. Summary:

At Car Dheko, we developed a machine learning model to predict used car prices, aimed at enhancing customer experience and improving pricing accuracy. The project involved data preprocessing, exploratory data analysis (EDA), and the application of advanced machine learning algorithms to build a highly accurate model. The result is a Streamlit-based web application that provides users with an intuitive platform for car price predictions, streamlining decision-making and transactions in the used car market.

2. Introduction:

2.1. Problem Statement

Determining the accurate value of used cars is a complex task due to the wide range of variables that influence pricing, such as age, condition, and market demand. Car Dheko aims to overcome this challenge by creating a machine learning model capable of delivering precise price predictions. By integrating this model into an intuitive web application, the solution will simplify the pricing process for both customers and sales teams, improving decision-making and overall efficiency.

2.2. Objective

The main goal is to develop and deploy a machine learning model that can predict used car prices based on various input features, including make, model, year, fuel type, transmission, kilometres driven, seats, engine displacement and more. This model will be integrated into a Streamlit application, offering users quick and accurate price estimates at their fingertips.

2.3. Project scope

➢ Creation of a machine learning model to predict used car prices.

➢ Integration of the model into a Streamlit web application for deployment.

➢ Design of an intuitive interface for easy use by both customers and sales teams.

3. Data preprocessing

3.1. Data overview:

The dataset for this project was sourced from Car Dheko and includes detailed records of used car prices. It features attributes such as make, model, year, fuel type, transmission, kilometres driven, ownership, city, and more. This dataset, originally in an unstructured format, has been converted into a structured format for analysis.

3.2. Data Cleaning and Preprocessing:

3.2.1 Price Conversion:

➢ Converted price values from various formats (e.g., "₹ 4 Lakh" and "₹ 1,50,000") into a consistent numeric format by removing non-numeric characters and converting terms like "Lakh" and "Crore" into numeric values.

3.2.2 Feature Engineering:

➢ **Categorical Features:**
  o Checked and corrected spelling errors in categorical columns.
  o Encoded features like fuel type, body type, transmission and more using label encoding.
  o Applied one-hot encoding to certain categorical features to handle non-ordinal categories.
  o Some car features columns converted one-hot encoding.

➢ **Numerical Features:**
  o Cleaned features like kilometres driven (km), cargo volume and more by replacing commas with nothing, removing terms such as "kg," "litres," and "rpm," and eliminating unnecessary characters like '@' and ','.
  o Converted numerical features to integers where necessary.

3.2.3 Unrelated Columns:

➢ Dropped columns that were irrelevant to the analysis, such as image URLs, links, and key columns.
➢ Removed any duplicated columns.

4. Exploratory Data Analysis (EDA):

4.1. Objective of EDA

Exploratory Data Analysis (EDA) is a crucial step in data preprocessing aimed at understanding the underlying patterns, relationships, and structures in a dataset. The goal is to summarize the main characteristics of the data and identify any potential issues or insights that may influence further analysis or model development.

4.2. Key Insights

4.2.1 Check for Null Values:

➢ **Objective:** Identify missing values in the dataset.

➢ **Methods:** Analyze the extent and patterns of missing data, and choose appropriate strategies for handling them (e.g., filling with mean, median, mode, or using imputation techniques). Identify missing values in the dataset and address them by filling with mean, median, or mode as appropriate.

4.2.2. Remove Outliers:

➢ **Objective:** Detect and handle outliers that may skew the analysis or affect model performance.

➢ **Methods:** Use statistical methods like the Interquartile Range (IQR) to identify and remove outliers.

4.2.3. Visualization:

➢ **Univariate Analysis:** Examine the distribution of individual features using histograms, box plots, and density plots.

➢ **Bivariate Analysis:** Explore relationships between pairs of features using scatter plots, correlation coefficients, and cross-tabulations.

➢ **Multivariate Analysis:** Investigate interactions between multiple features through techniques such as pair plots and 3D scatter plots.

➢ **Correlation Heatmap:** Visualize the strength and direction of relationships between features using a correlation matrix.

4.2.4. Categorical Columns Encoding:

➢ **Objective**: Prepare categorical features for modelling by converting them into numerical formats.

> **Methods**: Use label encoding, one-hot encoding, or ordinal encoding based on the data's nature and the model requirements.

4.2.5. Scaling:

> **Objective:** Normalize numerical features to ensure they contribute equally to the model.

> **Methods:** Apply scaling techniques like Min-Max Scaler to standardize feature ranges.

4.2.6. Correlation Matrix:

> **Objective:** Identify and analyze relationships between features.

> **Methods:** Generate a correlation matrix to assess how features correlate with each other and with the target variable.

4.3. Impact of EDA on Model Development

> **Objective:** Refine the dataset by removing features with minimal impact on the target variable.

> **Methods:** Evaluate feature importance and relevance to reduce dimensionality and improve model efficiency.

EDA is an iterative process that helps in making informed decisions about data cleaning, feature engineering, and model selection. It provides a comprehensive understanding of the dataset, paving the way for effective data analysis and model building.

5. Model Development:

5.1. Methodology:

Different regression models were evaluated, including Linear Regression, Gradient Boosting, Decision Tree, and Random Forest, to determine the most accurate and dependable model for predicting used car prices.

5.2. Algorithms:

i. Linear Regression:

> **Overview:** Linear Regression was utilized as the initial model due to its straightforward nature and interpretability.

> **Cross-Validation:** A 5-fold cross-validation was implemented to evaluate the model's effectiveness.

> **Regularization:** Ridge and Lasso techniques were employed to mitigate the risk of overfitting.

ii. Gradient Boosting Regressor (GBR):

> **Overview:** GBR was chosen for its capacity to capture complex, non-linear patterns in the data.

> **Hyperparameter Tuning:** A Randomized Search approach was employed to fine-tune parameters such as n_estimators, learning_rate, and max_depth.

iii. Decision Tree Regressor:

> **Overview:** Decision Trees were selected for their clear interpretability and ability to model intricate non-linear relationships.

> **Pruning:** The tree was pruned to avoid overfitting by restricting its depth.

iv. Random Forest Regressor:

> **Overview:** Random Forest, an ensemble technique, was adopted for its high accuracy and resilience.

> **Hyperparameter Tuning:** Randomized Search was used to optimize parameters like n_estimators and max_depth.

5.3. Model Evaluation:

The models were assessed using the following metrics:

> **Mean Squared Error (MSE):** Evaluates the average of the squared differences between the actual and predicted values.

> **Mean Absolute Error (MAE):** Provides an average of the absolute differences between predicted and actual values, offering a straightforward measure of prediction accuracy.

> **R^2 Score:** Reflects the proportion of variance in the dependent variable that is explained by the independent variables.

Results:

> **Random Forest Regressor:** Delivered the highest performance, with the top R^2 score and the lowest MSE and MAE, making it the selected model for deployment.

6. Model Deployment (Streamlit Application):

6.1. Overview of the Streamlit Application:

Streamlit is an open-source Python library designed for swiftly building custom web apps tailored for data science and machine learning. Its ease of use and versatility make it an excellent tool for deploying machine learning models as interactive web applications.

6.2. Features of the Application:

➢ **User Input Interface:** The app offers a user-friendly interface allowing users to enter car details, such as make, model, year, fuel type, transmission, mileage, number of owners, and city. It utilizes drop-down menus and sliders to simplify the input process and minimize errors.

➢ **Price Prediction:** After collecting user inputs, the app employs the trained Random Forest model to estimate the car's price. The predicted value is promptly displayed, enhancing user interaction.

➢ **Visualizations:** The app features visualizations that illustrate the effects of various attributes on car pricing, aiding users in understanding how different factors influence the price.

6.3. Backend Implementation:

➢ **Model Loading:** The trained Random Forest model is integrated into the application using the pickle library, ensuring it is available for making predictions.

➢ **Data Preprocessing:** User inputs are processed in the same manner as the training data to maintain consistency and accuracy in the predictions.

6.4. Deployment Process:

The application was hosted on a cloud platform, allowing it to be easily accessed through a web browser by both customers and sales representatives.

7. Reason behind the selection of the model:

7.1. Random Forest Regressor

- ➢ **Robustness**: Due to its ensemble nature, Random Forest is less prone to overfitting and offers greater robustness compared to individual decision trees.
- ➢ **Accuracy**: The model consistently delivered the most accurate predictions across all metrics (MSE, MAE, R^2).
- ➢ **Versatility**: It effectively manages both numerical and categorical data, making it well-suited for the diverse features in this dataset.

8. Conclusion

8.1. Project Impact

The integration of the predictive model through the Streamlit application revolutionizes the customer experience at Car Dheko by delivering swift and precise price estimates. This advancement not only enhances the decision-making process for both customers and sales representatives but also paves the way for future improvements in predictive analytics.

8.2. Future Work

- ➢ **Enhanced Features:** Adding elements such as insurance information and seller feedback could further sharpen the accuracy of predictions.
- ➢ **Localized Models:** Creating specialized models for various cities could better accommodate regional pricing differences.
- ➢ **Ongoing Model Refinement:** Continuously updating the model with fresh data will maintain its accuracy and relevance over time.

9. Appendices:

9.1. Model Performance Metrics:

The Random Forest model was selected for deployment due to its superior performance, achieving the highest R^2 score and the lowest MSE/MAE, making it the most accurate and reliable model for predictions.

Model	MSE	MAE	R^2
Linear Regression	384822.41	260609487387.97	0.67
Gradient Boosting	202191.26	115615244060.34	0.85

Decision Tree Regressor	222684.27	133799043589.36	0.83
Random Forest Regressor	203524.62	111503215079.77	0.86

References:

- ➢ [Pandas Documentation](#)
- ➢ [Data visualization Documentation](#)
- ➢ [Scikit learn Documentation](#)
- ➢ [Streamlit Documentation](#)

Software:

- ➢ Anaconda Navigator – Jupyter Notebook
- ➢ Visual studio code for streamlit