PROJECT TITLE:

CAR DHEKO – USED CAR PRICE PREDICTION

BATCH: MDTE13 (DS-WD-T-B10)

NAME: NANDHINI C

COURSE: DATA SCIENCE

TABLE OF CONTENTS

1. Summary	Page 3
2. Introduction	Page 3
2.1 Problem Statement	Page 3
2.2 Objective	Page 3
2.3 Project Scope	Page 4
3. Data Preprocessing	Page 4
3.1.Data Overview	Page 4
3.2. Data Cleaning and Preprocessing	Page 4
4. Exploratory Data Analysis (EDA)	Page 7
5. Model Development	Page 7
5.1. Methodology	Page 8
6. Model Deployment (Streamlit Application)	Page 10
7. Screenshots	Page 11
8. Reason Behind Model Selection	Page 13
9. Conclusion	Page 13
10. Appendices	Page 14
11. References	Page 15

SUMMARY:

This project focuses on developing a machine learning-based solution for predicting used car prices for Car Dheko. The aim is to enhance the pricing process by building an accurate predictive model and deploying it via a Streamlit web application. The solution incorporates data preprocessing, exploratory data analysis, model development, and deployment, ensuring a user-friendly interface for real-time price predictions.

INTRODUCTION:

2.1 Problem Statement

The used car market is influenced by multiple factors, making price prediction a complex task. Current manual methods can lead to inconsistencies and customer dissatisfaction. The need is to build a machine learning-based tool to streamline and enhance the pricing process. 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

To create an accurate, reliable, and user-friendly application that predicts used car prices based on input features such as make, model, year, fuel type, and transmission and more.

2.3 Project Scope

1. Utilize historical datasets of used car prices for analysis.
2. Develop machine learning models to predict car prices based on the key features.
3. Deploy the solution as a Streamlit application, ensuring accessibility for users.

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:

➢ Categorical Features:

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

➢ Numerical Features:

- 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 ','.
- Converted numerical features to integers where necessary.

➢ Unrelated Columns:

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

Source: Historical data from Car Dheko, collected across multiple cities.

Key Features:
'ft',

'bt',

'km',

'transmission',

'ownerNo',

'oem',

'model',

'modelYear',

'variantName',

'City',

'mileage',

'Seats',

'car_age',

'brand_popularity',

'mileage_normalized'

Step	**Description**
Handling Missing Values	Filled missing numerical values using mean/median, and created new categories for missing categorical data
Standardizing Formats	Converted inconsistent data types into uniform formats.
Encoding Categorical Variables	Used one-hot encoding for nominal data and label encoding for ordinal data.
Scaling Numerical Features	Applied Min-Max scaling for algorithms sensitive to data magnitude.
Outlier Removal	Used Interquartile Range (IQR) and Z-score methods to cap or remove extreme values.

4. Exploratory Data Analysis (EDA)

Objective:
1. Understand the distribution of features.
2. Identify correlations and patterns.
3. Determine key features influencing car prices.

Key Insights

1. Positive correlation observed between car price and engine size.
2. Older cars and cars with higher mileage have lower prices.
3. Certain brands hold higher resale value, irrespective of age.
4. City-based variations in pricing due to regional demand.

Impact of EDA on Model Development

1. Simplified feature selection by highlighting impactful variables.
2. Insights into relationships informed model algorithm choices.
3. Helped address multicollinearity through correlation heatmaps.

5. Model Development Methodology

1. Splitting the data into training (80%) and testing (20%) sets.
2. Evaluating models using cross-validation and hyperparameter tuning.

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

Algorithms

1. **Linear Regression:**
   - Simple and interpretable model.
   - Suitable for linear relationships.

   ➢ 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.

2. **Decision Tree Regressor:**
   - Handles non-linear data well.
   - Prone to overfitting.

   ➢ 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.

3. **Random Forest Regressor:**
    - ○ Reduces overfitting through ensemble learning.
    - ○ Robust to missing data.

    ➢ 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.

4. **Gradient Boosting Regressor (GBR):**
    - ○ Builds strong predictive models iteratively.
    - ○ Highly accurate but computationally expensive.

    ➢ 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.

Model Evaluation

- Metrics used:
    - ○ Mean Absolute Error (MAE)
    - ○ Mean Squared Error (MSE)
    - ○ R-squared Score

5. Comparison of model performance determined the selection of Random Forest Regressor as the final model.

➢ 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.

6. Model Deployment (Streamlit Application)

Overview of Streamlit Application

Streamlit was used to deploy the final machine learning model as a web application. The app provides an interactive interface for users to input car details and receive price predictions instantly.

Features of the Application

1. User-friendly input forms for car details.
2. Real-time prediction of car prices.
3. Clear error messages and input validation.
4. Visualizations of price trends based on input parameters.

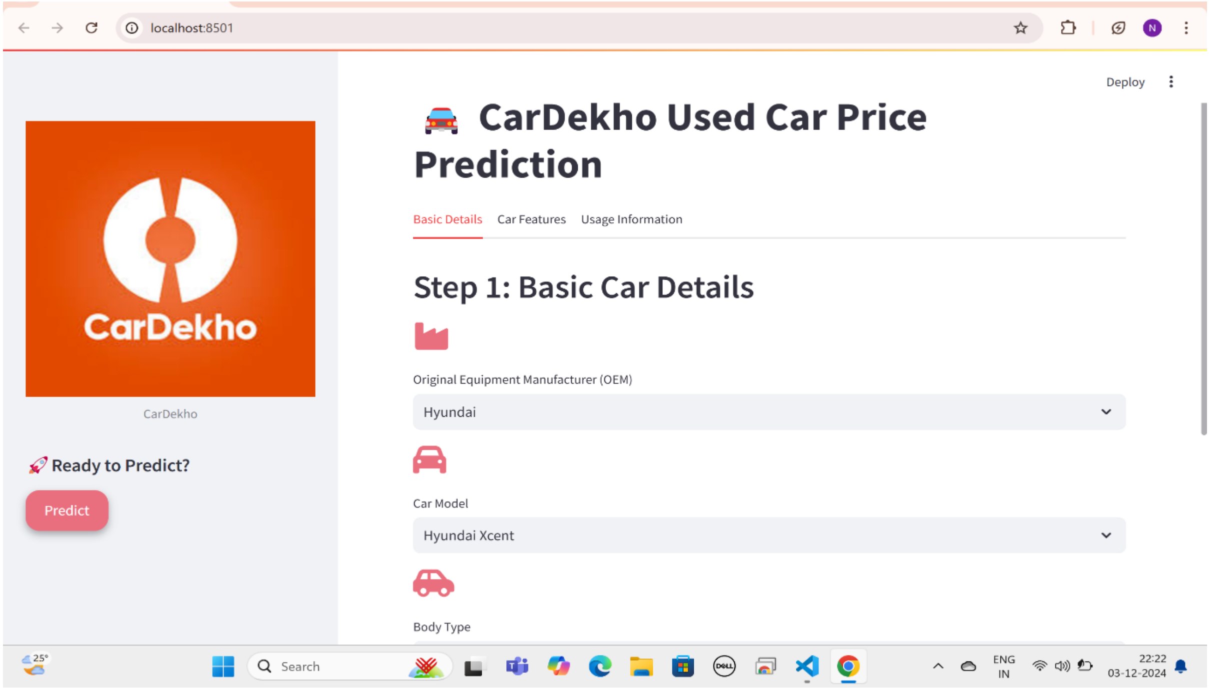
Backend Implementation

1. Pre-processed the input features to align with the model.
2. Loaded the trained model for real-time predictions.
3. Ensured quick response times for user queries.

Deployment Process

1. Exported the trained model using Python's joblib or pickle.
2. Integrated the model into the Streamlit app.

7.ScreenShots



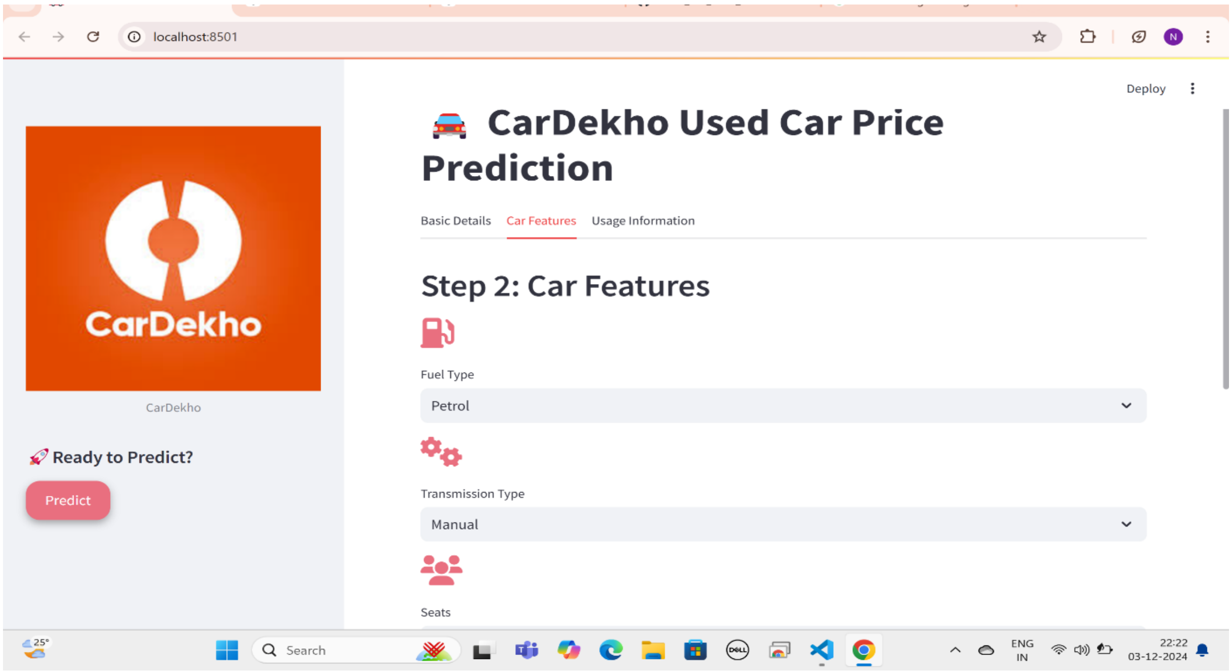
CarDekho Used Car Price Prediction

Step 1: Basic Car Details

Original Equipment Manufacturer (OEM): Hyundai

Car Model: Hyundai Xcent

Body Type



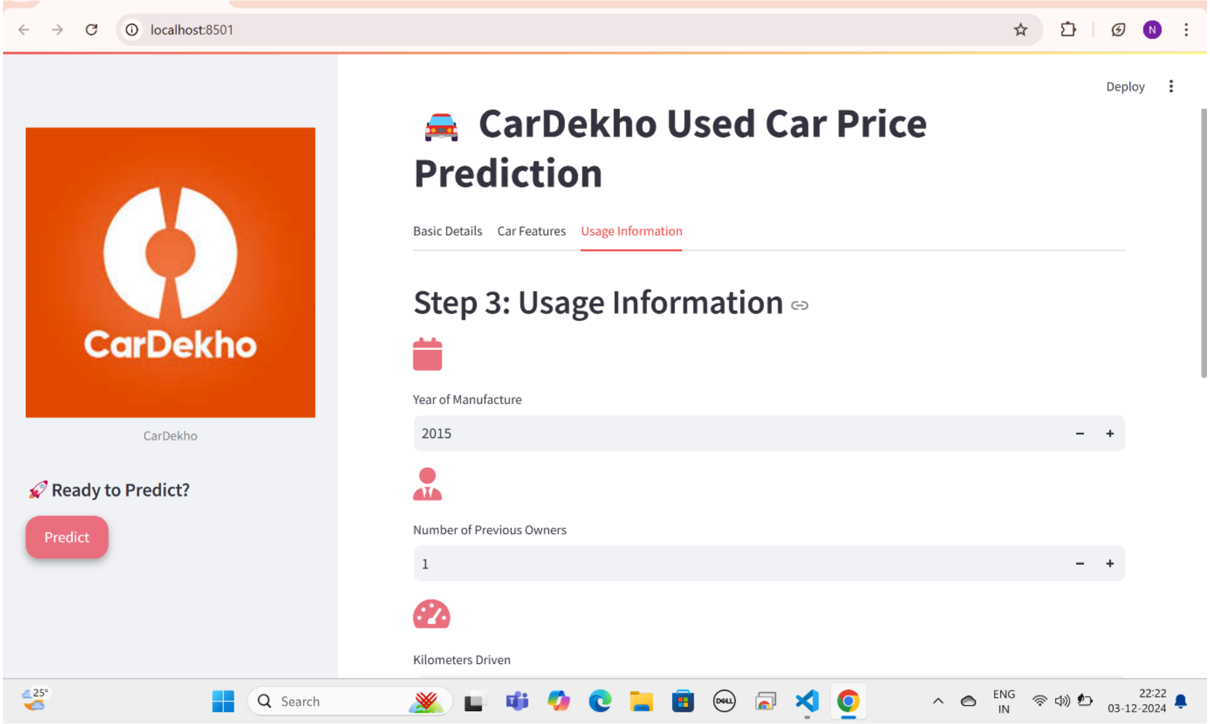
CarDekho Used Car Price Prediction

Step 2: Car Features

Fuel Type: Petrol

Transmission Type: Manual

Seats



localhost:8501

Deploy

CarDekho

🚀 Ready to Predict?

Predict

🚘 CarDekho Used Car Price Prediction

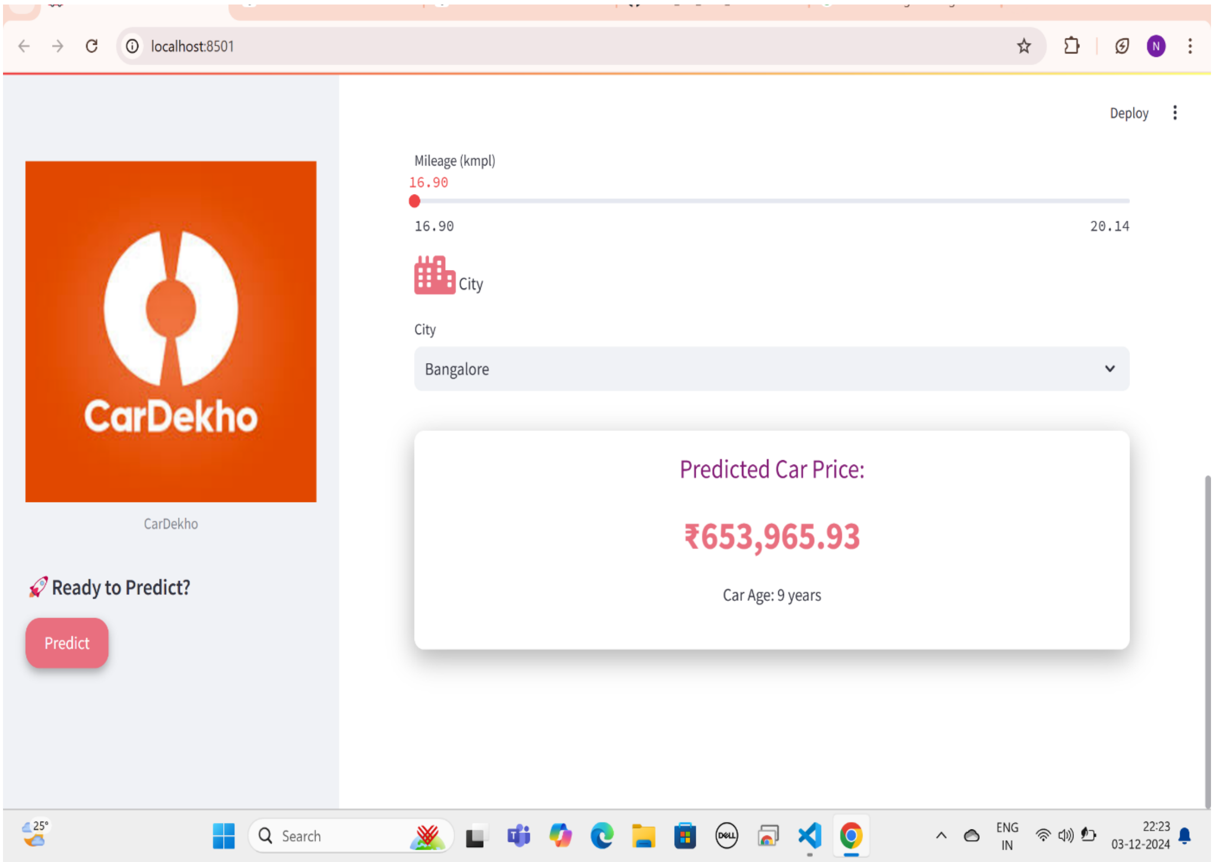
Basic Details | Car Features | Usage Information

Step 3: Usage Information

Year of Manufacture: 2015

Number of Previous Owners: 1

Kilometers Driven



localhost:8501

Deploy

CarDekho

🚀 Ready to Predict?

Predict

Mileage (kmpl): 16.90 (range 16.90 – 20.14)

City

City: Bangalore

Predicted Car Price:

₹653,965.93

Car Age: 9 years

8. Reason Behind the Selection of the Model

8.1. Random Forest Regressor:

Robustness

- Random Forest’s ensemble nature makes it less prone to overfitting and more robust compared to single decision trees.

Accuracy

- The model consistently provided the most accurate predictions across all metrics (MSE, MAE, R^2).

Versatility

- It effectively handles both numerical and categorical data, making it suitable for the diverse features in this dataset.

9. Conclusion

Project Impact

- Automated and improved accuracy of used car price estimation.
- Enhanced customer trust through consistent and transparent pricing.
- Improved sales representatives' efficiency.

Future Work

1. Expand the dataset to include features like car condition or insurance history.
2. Introduce AI-powered feedback loops to refine predictions.
3. Optimize application deployment for faster and more scalable user access.

10. Appendices

Model Performance Metrics

The Random Forest model was selected for deployment due to its superior performance, achieving the highest R² score and the lowest MSE/MAE, making it the most accurate and reliable model for predictions.

Metric	MAE	MSE	R-squared
Linear Regression	158699.20287340562	48351470014.415855	0.62
Decision Tree	152517.0616255033	43201186286.646324	0.64
Gradient boosting	108576.23916320053	25097389873.630695	0.80
Random Forest	**74234.3852167622**	**13448159595.57039**	**0.89**

11. References

Software

1. Python (NumPy, Pandas, Scikit-learn)
2. Streamlit
3. Matplotlib and Seaborn (for visualization)
4. Joblib/Pickle (for model saving)

Sources

1. Car Dheko historical datasets.
2. Python documentation and libraries.
3. Online tutorials on Streamlit and machine learning deployment.