Car Dheko - Used Car Price Prediction

(RandomForest Model)

Project Report

By

Sakthi K

Introduction:

This project aims to enhance the customer experience and optimize the pricing process for used cars by developing a machine learning model. The model uses historical car price data and incorporates various features such as the car's make, model, year, fuel type, and transmission type. The goal is to predict used car prices accurately and integrate this model into an interactive web application built with Streamlit. This app will allow customers and sales representatives to input car details and receive real-time price predictions, streamlining the car buying and selling process.

A. Approach:

1. **Importing Files and Data Wrangling:**

    - Load datasets from multiple cities, which are initially in unstructured formats (e.g., messy data with text, JSON-like entries).
    - Use libraries like pandas to import data from Excel files and prepare it for processing.
    - Handle JSON-like data (e.g., dictionaries or lists embedded as strings) by using `ast. literal_eval` to safely convert them into Python objects.
    - Parse the JSON data and use `pandas. json_normalize()` to flatten nested JSON structures into a clean, structured data frame.
    - Add a new column called 'City' to each dataset, assigning the respective city name to all rows.
    - Merge the datasets from all cities using `pandas.concat()` to create a unified dataframe for model training, ensuring column alignment and handling duplicate entries.
    - Save the combined dataframe as a CSV file for further use.

2. **Handling Missing Values and Data Cleaning:**

    - Remove rows or columns containing missing data using `pandas.dropna()`.
    - Clean the data by removing unwanted symbols (e.g., ₹, Lakh, Crore, kmpl, CC) and eliminating commas and whitespace.
    - Convert values to a numerical format that is compatible with machine learning models.

3. **Data Visualization:**

    - Perform Exploratory Data Analysis (EDA) to understand the relationships between various features and the target variable (price).
    - Use a correlation matrix heatmap to identify significant relationships between features such as model year, kilometers driven, and price.
    - Detect and handle outliers in the price column using the Interquartile Range (IQR) method to prevent them from affecting model performance.

4. **Feature Selection:**

    - **Categorical Features:** These include attributes like fuel type, body type, brand, insurance validity, color, city, and transmission type.
    - **Numerical Features:** Features such as number of owners, model year, kilometers driven, mileage, engine capacity, and number of seats were cleaned and converted to integers for better usability in machine learning models.

5. **Encoding and Scaling:**

    - Apply One-Hot Encoding to convert categorical features into numerical values, a requirement for many machine learning models.
    - Use a Standard Scaler to normalize the numerical features, ensuring that all features are on the same scale and preventing dominance of certain features.

B. Model Development

1. **Train-Test Split:**

    - Split the dataset into training and testing sets to evaluate the model's performance.
    - Common split ratios used are 70-30 or 80-20, where 70% or 80% of the data is used for training, and the rest is used for testing.

2. **Model Selection:**
    i. **Random Forest Regressor:**

        - **Overview:** The Random Forest model was chosen as it is an ensemble learning method that works well for regression tasks. It builds multiple decision trees and aggregates their results to make predictions.
        - **Model Characteristics:**
            - Random Forest uses bagging, meaning each decision tree in the forest is trained on a random subset of the data.
            - Each decision tree considers a random subset of features when making decisions, which helps to reduce overfitting and improve model robustness.

    ii. **Preprocessing:**
        - **Numerical Features:** Features like kilometers driven, model year, and engine capacity are scaled using a StandardScaler to normalize them.
        - **Categorical Features:** Features such as fuel type, transmission type, and city are encoded using One-Hot Encoding to convert them into a format suitable for machine learning models.

iii. **Model Training and Integration:**
The final model is trained using the Random Forest regression algorithm. This model is integrated into a pipeline that includes data preprocessing steps, making it easier to apply the same transformations to new data.

iv. **Reproducibility and Consistency:**
By encapsulating the entire workflow in a single pipeline, we ensure that the same preprocessing steps are applied consistently, guaranteeing reliable predictions.

3. **Model Evaluation:** The performance of the Random Forest model was evaluated using the following metrics:
   i. **Mean Squared Error (MSE):** Measures the average squared difference between the actual and predicted values.
   ii. **Mean Absolute Error (MAE):** Provides an average of the absolute differences between predicted and actual values.
   iii. **R^2 Score:** Indicates how well the model explains the variance in the target variable (price).

Model Evaluation Table:

Model	MAE	MSE	RMSE	R^2
Linear Regression	1,935,936	5.75E+20	23,976,305	-4.83E+19
Decision Tree Regressor	1.06	2.87	1.70	0.76
Random Forest Regressor	0.83	1.73	1.31	0.85
Gradient Boosting Regressor	1.04	2.23	1.49	0.81
Ridge Regressor	2.58	1.12	1.61	0.78
Lasso Regressor	2.59	1.12	1.61	0.78

Results:

iv. The **Random Forest model** achieved the best performance, with the highest R^2 score and the lowest MSE and MAE. This makes it the most suitable model for deployment in this project.

v. **Hyperparameter Tuning:** Grid Search was used to tune the model's hyperparameters (e.g., `n_estimators` and `max_depth`) to find the optimal values that improve performance.

4. **Pipeline:**
   i. **Modular Structure:** The pipeline is designed to be modular, allowing clear separation between data preprocessing and model training. This ensures that each part of the workflow can be easily adjusted or replaced without affecting the entire process.
   ii. **Data Preprocessing:**
      - **Numerical Features:** StandardScaler is used to scale numerical features like kilometers driven, engine capacity, etc.

- **Categorical Features:** One-Hot Encoding is applied to categorical features such as fuel type, transmission, and city.

iii. **Integration with Model:** The Random Forest model is integrated into the pipeline, enabling seamless transformation of data and prediction generation.

C. Model Deployment - Streamlit

- **Streamlit** is used for deploying the trained Random Forest model in an interactive web application.
    1. **Features of the Application:**
        - **User Input Interface:**
            - Users can input car details like make, model, year, fuel type, transmission, kilometers driven, and city using user-friendly input options such as dropdown menus and sliders.
        - **Price Prediction:**
            - Once the user inputs the details, the app uses the trained Random Forest model to predict the car's price in real-time.
    2. **Backend Implementation:**
        - **Model Loading:** The trained Random Forest model, along with the necessary preprocessing components (e.g., StandardScaler, LabelEncoder), is loaded using the `pickle` library to ensure it's ready for predictions.
        - **Data Preprocessing:** The user inputs are preprocessed in the same way as the training data, ensuring consistency and accuracy in predictions.

D. Conclusion - Project Impact

- The deployment of the **Random Forest model** via the Streamlit application transforms the user experience at CarDekho by providing quick and reliable price estimates for used cars.
- Customers gain data-driven insights, helping them make informed decisions when buying or selling cars. Sales representatives benefit from a streamlined valuation process, saving time and ensuring consistent pricing.
- This deployment sets the stage for future innovations, including the possibility of integrating real-time market data, adding personalized recommendations, or implementing other advanced features to enhance the user experience further.