PROJECT REPORT

COMPLIED BY

HUSNAIN BUKHARI (20i-0626)

ANUSHA ZUBAIR(20i-2454)

RABIA MUSTAFA (20i-1853)

Project: Car Price Prediction

Introduction

The goal of this project is to develop a machine learning model using data mining techniques that can predict the selling price of cars based on various features such as the car's year, present price, kilometers driven, fuel type, seller type, transmission, and owner. The project involves exploratory data analysis (EDA), preprocessing, model training, hyperparameter tuning, and evaluation. The trained model will be used to make price predictions, and a user-friendly front-end interface will be developed to display the results.

Data Description

The dataset used for this project is provided in a CSV file named car_data.csv. It contains information about different cars, including their features and selling prices. The columns in the dataset are as follows:

Car_Name: The name of the car
Year: The year of the car's manufacture
Selling_Price: The selling price of the car (target variable)
Present_Price: The current showroom price of the car
Kms_Driven: The total kilometers driven by the car
Fuel_Type: The type of fuel used by the car (Petrol, Diesel, CNG)
Seller_Type: The type of seller (Dealer or Individual)
Transmission: The type of transmission (Manual or Automatic)
Owner: The number of previous owners of the car

Methodology

1. Exploratory Data Analysis (EDA)
    a. Load the dataset using pandas.
    b. Perform descriptive statistics and visualizations to gain insights into the data.
    c. Identify missing values and handle them appropriately.
    d. Front-end Interface

2. Preprocessing
    a. Split the dataset into features (X) and the target variable (y).
    b. Identify categorical and numerical features.
    c. Define a preprocessor using scikit-learn's ColumnTransformer.

    d. Use StandardScaler to scale the numerical features.
    e. Use OneHotEncoder to one-hot encode the categorical features

3. Model Training and Evaluation
    a. Split the preprocessed data into training and testing sets.
    b. Define a random forest regression model.
    c. Create a pipeline that includes the preprocessor and the model.
    d. Use grid search with cross-validation to find the best hyperparameters for the model.
    e. Fit the model to the training data and evaluate its performance using root mean squared error (RMSE)

4. Front-end Interface
    a. Develop a user-friendly front-end interface using Flask.
    b. Create an HTML form that allows users to input car features.
    c. Retrieve the form data in the Flask application.
    d. Preprocess the user input using the preprocessor.
    e. Make predictions using the trained model.
    f. Display the predicted price to the user.

Results

The trained model achieved a reasonable level of performance in predicting car prices, as measured by RMSE which was 0.96 on the test data. The best hyperparameters for the random forest regression model were determined using grid search with cross-validation which were ({'model_max_depth': None, 'modelmin_samples_split': 2, 'model_n_estimators': 50}). The front-end interface provides a user-friendly way for users to input car features and obtain price predictions.

Future Improvements

- Include more advanced preprocessing techniques, such as handling outliers and imbalanced data.
- Explore other regression algorithms and compare their performance to the random forest model.
- Collect more data to improve the accuracy of the predictions.