**Used Car Price Prediction Application**

**Objective**

To build a machine learning regression model that predicts the selling price of used cars based on various features like fuel type, kilometers driven, transmission type, etc. The model is deployed using the Flask framework on Heroku.

**Dataset Information**

* **Dataset Source:** [Kaggle - Vehicle Dataset from CarDekho](https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho?select=CAR+DETAILS+FROM+CAR+DEKHO.csv)
* **Features Used:**
  + Car\_Name: Name of the car (dropped as it's not relevant for prediction)
  + Year: Year of purchase (transformed into age of the car)
  + Selling Price (Target Variable): Selling price of the car in lakhs
  + Present Price: Current price of the car in lakhs
  + Kms Driven: Total kilometers driven
  + Fuel Type: Petrol, Diesel, or CNG (encoded)
  + Seller Type: Dealer or Individual (dropped as not essential)
  + Transmission: Manual or Automatic (encoded)
  + Owner: First, second, or third owner

**Approach & Methodology**

**1. Data Preprocessing**

* Checked for duplicate records and dropped them.
* Dropped redundant columns (Car\_Name and Seller\_Type).
* Created a new feature age\_of\_the\_car by subtracting Year from 2022, then dropped the Year column.
* Encoded categorical columns (Fuel\_Type and Transmission).
* Separated the target variable (Selling\_Price) and independent features.

**2. Data Splitting**

* Split the dataset into training (70%) and testing (30%) using train\_test\_split.

**3. Model Selection & Training**

* Used a **Random Forest Regressor** to train the model.
* Evaluated performance using **R-squared score (90.1%)**.

**4. Model Deployment**

* Saved the trained model using **Pickle (model.pkl)**.
* Developed a Flask API (app.py) to load the model and serve predictions.
* Built a simple frontend using **HTML and Flask**.
* Deployed the application on **Heroku**.

**Code Implementation Overview**

**Backend (Flask API - app.py)**

* Loads the trained model.
* Accepts input data via a GET request.
* Returns predicted car prices in JSON format.

**Frontend (index.html)**

* Simple form for users to input car details.
* Sends data to the Flask API for price prediction.
* Displays the predicted price.

**Key Learnings**

* Feature engineering improves model performance (e.g., transforming Year to age\_of\_the\_car).
* Encoding categorical variables is crucial for numerical model inputs.
* Flask simplifies model deployment and API creation.
* Heroku provides an easy way to deploy machine learning models for public use.

**Challenges Faced**

* Handling categorical encoding properly to prevent model bias.
* Tuning the Random Forest model to optimize accuracy.
* Deploying Flask API smoothly on Heroku with environment dependencies.

**Results & Conclusion**

* Successfully built and deployed a machine learning model for used car price prediction.
* Achieved **90.1% R-squared accuracy** using Random Forest Regressor.
* Created a fully functional API and a simple user interface.

**GitHub Repository / Deployment Link**

* **GitHub:** https://github.com/yashzob/CarPrediction