**Car Price Prediction Analysis**

**Define problem statement**

The goal is to build a predictive model that accurately estimates the price of a car based on various features like make, model, year, mileage, fuel type, etc. This model could assist potential car buyers and sellers by providing a reliable estimate of a car’s price given its attributes.

**Select Suitable Dataset**

For this task, a dataset containing car price information and related

attributes is needed. Common sources for car datasets include:

* **Kaggle** (e.g., "Car Price Prediction" dataset)
* **UCI Machine Learning Repository**
* **Open-source Automotive websites** like Edmunds, or scraped data (if allowed)

The dataset should have features such as:

* **Car attributes**: make, model, year, mileage, transmission, fuel type, etc.
* **Price**: the target variable for prediction.

**Implement Project using Python**

# Import Libraries

import pandas as pd, numpy as np, matplotlib.pyplot as plt, seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

from sklearn.preprocessing import StandardScaler

# Load and Preprocess Data

df = pd.read\_csv('car\_data.csv').dropna(subset=['price']).fillna(df.mean())

df = pd.get\_dummies(df, columns=['make', 'model', 'fuel\_type', 'transmission'], drop\_first=True)

scaler = StandardScaler(); df[['mileage', 'engine\_size', 'year']] = scaler.fit\_transform(df[['mileage', 'engine\_size', 'year']])

X, y = df.drop('price', axis=1), df['price']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Visualizations

plt.figure(figsize=(14, 6))

plt.subplot(1, 3, 1); sns.boxplot(x='fuel\_type', y='price', data=df); plt.title("Price by Fuel Type")

plt.subplot(1, 3, 2); sns.histplot(df['price'], bins=30, kde=True); plt.title("Price Distribution")

plt.subplot(1, 3, 3); sns.scatterplot(x='mileage', y='price', data=df); plt.title("Mileage vs Price")

plt.show()

# Pearson Correlation Matrix

plt.figure(figsize=(10, 6)); sns.heatmap(df.corr(), annot=True, cmap="coolwarm"); plt.title("Correlation Matrix"); plt.show()

# Model Training and Evaluation

lr, rf = LinearRegression(), RandomForestRegressor(n\_estimators=100, random\_state=42)

lr.fit(X\_train, y\_train); rf.fit(X\_train, y\_train)

y\_pred\_lr, y\_pred\_rf = lr.predict(X\_test), rf.predict(X\_test)

# Evaluation Metrics

print("Linear Regression - MAE:", mean\_absolute\_error(y\_test, y\_pred\_lr), "MSE:", mean\_squared\_error(y\_test, y\_pred\_lr), "R2:", r2\_score(y\_test, y\_pred\_lr))

print("Random Forest - MAE:", mean\_absolute\_error(y\_test, y\_pred\_rf), "MSE:", mean\_squared\_error(y\_test, y\_pred\_rf), "R2:", r2\_score(y\_test, y\_pred\_rf))

**Visualize Data with Box Plot, Histogram, Scatter Plot**

 **Box Plot**: Useful for identifying outliers in numeric features like price, mileage, or age.

 **Histogram**: Display the distribution of features, such as price, to understand skewness and spread.

 **Scatter Plot**: Plot features (like age or mileage) against price to observe trends and possible correlations.

**Plot Pearson Correlation and Explain Relationship**

The Pearson correlation coefficient measures the linear correlation between variables. A correlation matrix can help in identifying which features have a significant impact on price.

# Correlation Matrix

plt.figure(figsize=(12, 8))

correlation\_matrix = df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm")

plt.title("Correlation Matrix of Features")

plt.show()

**Explanation**: Look for features with high correlation to price. For instance, if mileage has a strong negative correlation with price, it indicates that higher mileage typically results in a lower price.

**Identify Dependent and Independent Features**

 **Dependent Variable**: The target feature we're trying to predict, i.e., price.

 **Independent Variables**: Features used for prediction, such as make, model, year, mileage, fuel\_type, etc.

**Analysis / Prediction as per Problem Statement**

After training the model, use it to make predictions on a test dataset or unseen data. Evaluate the results based on how accurately the model predicts the car prices. You can further optimize the model by feature selection, hyperparameter tuning, or using ensemble methods.

**Github Link**

[Priyanshu87571/Car-Price-Prediction-Analysis: This is My first repository](https://github.com/Priyanshu87571/Car-Price-Prediction-Analysis)