Question33:

33. Scenario: You work as a data scientist for an automobile company that sells various car

models. The company has collected data on different car attributes, such as engine size,

horsepower, fuel efficiency, and more, along with their corresponding prices. The marketing team

wants to build a predictive model to estimate the price of cars based on their features.

Question: Your task is write a Python program that perform linear regression modeling to predict

car prices based on a selected set of features from the dataset. Additionally, you need to evaluate

the model's performance and provide insights to the marketing team to understand the most

influential factors affecting car prices.

Answer:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LinearRegression

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

# Load the car price dataset

file\_path =r"C:\Users\jampa\Downloads\car\_price\_data.csv"

df = pd.read\_csv(file\_path)

# Perform bivariate analysis (Horsepower vs. Price)

plt.figure(figsize=(8,4))

sns.scatterplot(data=df, x='Horsepower', y='Price\_USD', alpha=0.7)

plt.title('Horsepower vs. Price')

plt.xlabel('Horsepower')

plt.ylabel('Price (USD)')

plt.show()

# Prepare data for linear regression

features = ['Engine\_Size\_L', 'Horsepower', 'Fuel\_Efficiency\_MPG', 'Weight\_lbs', 'Cylinders', 'Year']

X = df[features]

y = df['Price\_USD']

# Split data into training and testing sets

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

# Train the linear regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error (MSE): {mse:.2f}")

print(f"R-Squared (R2): {r2:.2f}")

# Plot the regression line for a single feature (e.g., Horsepower)

plt.figure(figsize=(8,4))

sns.scatterplot(x=X\_test['Horsepower'], y=y\_test, alpha=0.7, label='Actual Prices')

plt.scatter(X\_test['Horsepower'], y\_pred, color='red', s=20, alpha=0.6, label='Predicted Prices')

plt.title('Horsepower vs. Price (Predictions)')

plt.xlabel('Horsepower')

plt.ylabel('Price (USD)')

plt.legend()

plt.show()

Output:



