# Import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# For model building

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

from sklearn.linear\_model import LinearRegression

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

# Load dataset

# Download 'train.csv' from https://www.kaggle.com/c/boston-housing/data and place it in your working directory

df = pd.read\_csv("housing.csv") # or use the full path if needed

# Display the first few rows of the dataset

print("Dataset preview:\n", df.head())

# Check for null values

print("\nMissing values:\n", df.isnull().sum())

# Features and target variable

X = df.drop("MEDV", axis=1) # MEDV is the target variable (Median value of owner-occupied homes)

y = df["MEDV"]

# Split the dataset 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)

# Create Linear Regression model

model = LinearRegression()

# Train the model

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

# Predict on test data

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

# Evaluate the model

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

rmse = np.sqrt(mse)

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

print(f"\nModel Evaluation:")

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

print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")

print(f"R^2 Score: {r2:.2f}")

# Optional: Plotting actual vs predicted values

plt.figure(figsize=(8, 6))

plt.scatter(y\_test, y\_pred, color='blue', edgecolor='k')

plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')

plt.xlabel("Actual MEDV")

plt.ylabel("Predicted MEDV")

plt.title("Actual vs Predicted Home Prices")

plt.grid(True)

plt.show()