**7. Develop a program to demonstrate the working of Linear Regression and Polynomial Regression. Use Boston Housing Dataset for Linear Regression and Auto MPG Dataset (for vehicle fuel efficiency prediction) for Polynomial Regression.**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

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

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

# -----------------------------

# PART 1: Linear Regression - Boston Housing Dataset

# -----------------------------

from sklearn.datasets import load\_boston

# Load Boston Housing dataset

boston = load\_boston()

boston\_df = pd.DataFrame(boston.data, columns=boston.feature\_names)

boston\_df['MEDV'] = boston.target

# Features and target

X\_boston = boston\_df[['RM']] # Avg. number of rooms per dwelling

y\_boston = boston\_df['MEDV']

# Train-Test Split

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

# Linear Regression

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train\_b, y\_train\_b)

y\_pred\_b = lin\_reg.predict(X\_test\_b)

# Plotting

plt.figure(figsize=(8, 5))

plt.scatter(X\_test\_b, y\_test\_b, color='blue', label='Actual')

plt.plot(X\_test\_b, y\_pred\_b, color='red', linewidth=2, label='Prediction')

plt.title('Linear Regression - Boston Housing (RM vs MEDV)')

plt.xlabel('Average Number of Rooms per Dwelling (RM)')

plt.ylabel('Median Value of Homes (MEDV)')

plt.legend()

plt.grid(True)

plt.show()

# Metrics

print("Boston Housing Linear Regression:")

print("MSE:", mean\_squared\_error(y\_test\_b, y\_pred\_b))

print("R^2 Score:", r2\_score(y\_test\_b, y\_pred\_b))

# -----------------------------

# PART 2: Polynomial Regression - Auto MPG Dataset

# -----------------------------

# Load Auto MPG dataset

auto\_df = pd.read\_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data",

delim\_whitespace=True, header=None,

names=["mpg", "cylinders", "displacement", "horsepower", "weight",

"acceleration", "model\_year", "origin", "car\_name"])

# Data Cleaning

auto\_df = auto\_df.replace("?", np.nan)

auto\_df['horsepower'] = auto\_df['horsepower'].astype('float')

auto\_df.dropna(inplace=True)

# Feature and target

X\_auto = auto\_df[['horsepower']].values

y\_auto = auto\_df['mpg'].values

# Train-Test Split

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

# Polynomial Transformation

poly = PolynomialFeatures(degree=2)

X\_train\_poly = poly.fit\_transform(X\_train\_a)

X\_test\_poly = poly.transform(X\_test\_a)

# Polynomial Regression

poly\_reg = LinearRegression()

poly\_reg.fit(X\_train\_poly, y\_train\_a)

y\_pred\_a = poly\_reg.predict(X\_test\_poly)

# Plotting

plt.figure(figsize=(8, 5))

plt.scatter(X\_test\_a, y\_test\_a, color='green', label='Actual')

sorted\_indices = X\_test\_a[:, 0].argsort()

plt.plot(X\_test\_a[sorted\_indices], y\_pred\_a[sorted\_indices], color='orange', linewidth=2, label='Polynomial Fit')

plt.title('Polynomial Regression - Auto MPG (Horsepower vs MPG)')

plt.xlabel('Horsepower')

plt.ylabel('Miles per Gallon (MPG)')

plt.legend()

plt.grid(True)

plt.show()

# Metrics

print("\nAuto MPG Polynomial Regression:")

print("MSE:", mean\_squared\_error(y\_test\_a, y\_pred\_a))

print("R^2 Score:", r2\_score(y\_test\_a, y\_pred\_a))