# House Price Prediction using Linear, Ridge, and Polynomial Regression

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import load\_boston

from sklearn.linear\_model import LinearRegression, Ridge

from sklearn.preprocessing import PolynomialFeatures

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

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

# Load dataset

boston = load\_boston()

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

y = boston.target

# Split 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)

# 1️⃣ **Linear Regression**

lin\_reg = LinearRegression()

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

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

lin\_rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_lin))

lin\_r2 = r2\_score(y\_test, y\_pred\_lin)

# 2️⃣ **Ridge Regression**

ridge\_reg = Ridge(alpha=1.0)

ridge\_reg.fit(X\_train, y\_train)

y\_pred\_ridge = ridge\_reg.predict(X\_test)

ridge\_rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_ridge))

ridge\_r2 = r2\_score(y\_test, y\_pred\_ridge)

# 3️⃣ **Polynomial Regression**

poly = PolynomialFeatures(degree=2)

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

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

poly\_reg = LinearRegression()

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

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

poly\_rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred\_poly))

poly\_r2 = r2\_score(y\_test, y\_pred\_poly)

# 📊 Model Performance

print("Model Performance Comparison:")

print(f"Linear Regression -> RMSE: {lin\_rmse:.2f}, R²: {lin\_r2:.2f}")

print(f"Ridge Regression -> RMSE: {ridge\_rmse:.2f}, R²: {ridge\_r2:.2f}")

print(f"Polynomial Regression -> RMSE: {poly\_rmse:.2f}, R²: {poly\_r2:.2f}")

# 📈 Visualization

plt.figure(figsize=(14, 5))

# Linear Regression Plot

plt.subplot(1, 3, 1)

plt.scatter(y\_test, y\_pred\_lin, color='blue', alpha=0.6)

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

plt.title('Linear Regression')

plt.xlabel('Actual Prices')

plt.ylabel('Predicted Prices')

# Ridge Regression Plot

plt.subplot(1, 3, 2)

plt.scatter(y\_test, y\_pred\_ridge, color='green', alpha=0.6)

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

plt.title('Ridge Regression')

plt.xlabel('Actual Prices')

plt.ylabel('Predicted Prices')

# Polynomial Regression Plot

plt.subplot(1, 3, 3)

plt.scatter(y\_test, y\_pred\_poly, color='purple', alpha=0.6)

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

plt.title('Polynomial Regression')

plt.xlabel('Actual Prices')

plt.ylabel('Predicted Prices')

plt.tight\_layout()

plt.show()

Model Performance Comparison:

Linear Regression -> RMSE: 4.93, R²: 0.71

Ridge Regression -> RMSE: 4.90, R²: 0.72

Polynomial Regression -> RMSE: 3.75, R²: 0.83

