**Boston Housing Price Prediction**

**ABSTRACT**

This project focuses on predicting housing prices using machine learning techniques. By analyzing the relationships between features like crime rate, number of rooms, and location, the project aims to build an accurate model for price prediction. The study involves Linear Regression and Ridge Regression models, using Python and libraries such as scikit-learn, Pandas, and Matplotlib for data processing, model development, and visualization. The findings provide valuable insights into the factors affecting housing prices, making the project useful for real estate stakeholders and researchers.

**INTRODUCTION**

The Boston Housing Price Prediction project aims to develop a machine learning model to predict housing prices based on various factors like crime rate, number of rooms, and proximity to amenities. Accurate price prediction is essential for decision-making in real estate and urban planning. This project uses regression models to understand the relationships between housing features and prices, evaluating model performance using metrics like Mean Squared Error (MSE) and R-squared score.

**TECHNIQUES AND LIBRARIES**

### Techniques Used:

### Linear Regression: A basic regression model that establishes a linear relationship between features and prices.

### Ridge Regression: A regularized version of Linear Regression that minimizes overfitting for more accurate predictions.

### Libraries Used:

### scikit-learn: For implementing and evaluating regression models.

### Pandas & NumPy: For data cleaning, manipulation, and preprocessing.

### Matplotlib & Seaborn: For creating visualizations like heatmaps and scatter plots.

### FEATURES

* **Crime Rate:** A measure of safety in the neighborhood.
* **Number of Rooms:** The average number of rooms in a house.
* **Location:** Proximity to amenities and urban centers.
* **Property Age:** Age of the house or building.
* **Tax Rates:** Local property tax levels.
* **Proximity to Employment Centers:** Distance to job opportunities.

**CODING:**

**Install package**

pip install flask numpy pandas scikit-learn

**Code**

from sklearn.datasets import fetch\_california\_housing

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

from sklearn.linear\_model import LinearRegression, Ridge

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

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Load the California Housing dataset

data = fetch\_california\_housing(as\_frame=True)

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

df['Price'] = data.target

# Display dataset information

print(df.head())

print(df.info())

# Save dataset to a .txt file

df.to\_csv('california\_housing\_data.txt', sep='\t', index=False)

# Preprocess Data

X = df.drop('Price', axis=1)

y = df['Price']

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

# Train Models

lr = LinearRegression()

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

ridge = Ridge(alpha=1.0)

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

# Evaluate Models

models = {'Linear Regression': lr, 'Ridge Regression': ridge}

for name, model in models.items():

    y\_pred = model.predict(X\_test)

    print(f"\n{name}")

    print(f"Mean Squared Error: {mean\_squared\_error(y\_test, y\_pred):.4f}")

    print(f"R-squared: {r2\_score(y\_test, y\_pred):.4f}")

# Visualize Predictions

plt.figure(figsize=(10, 6))

plt.scatter(y\_test, lr.predict(X\_test), alpha=0.7, label="Linear Regression")

plt.scatter(y\_test, ridge.predict(X\_test), alpha=0.7, label="Ridge Regression")

plt.xlabel("Actual Prices")

plt.ylabel("Predicted Prices")

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

plt.legend()

plt.show()

# Correlation Heatmap

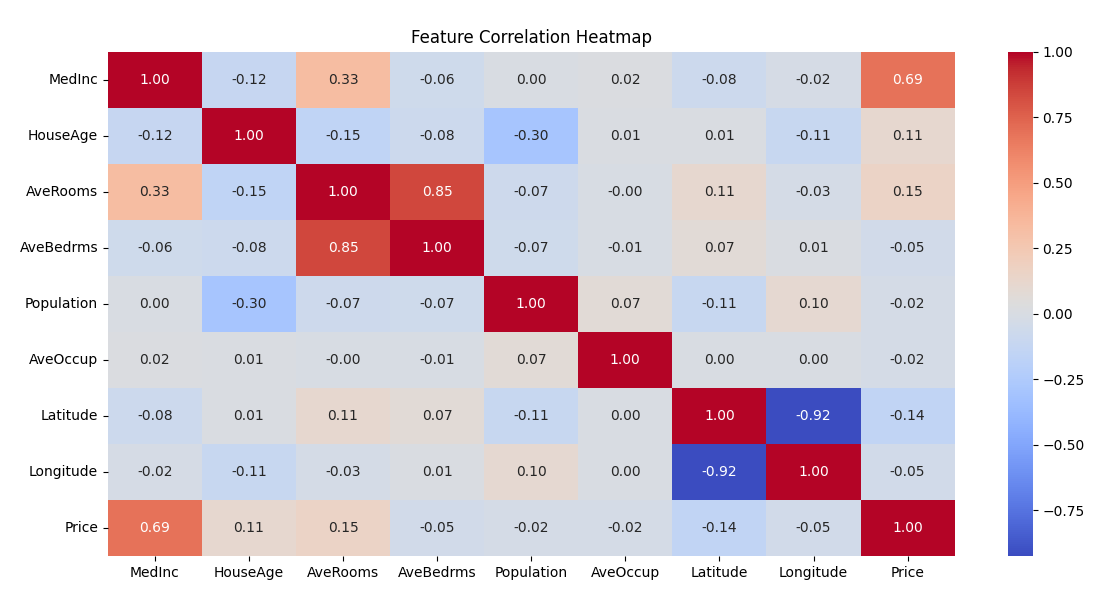
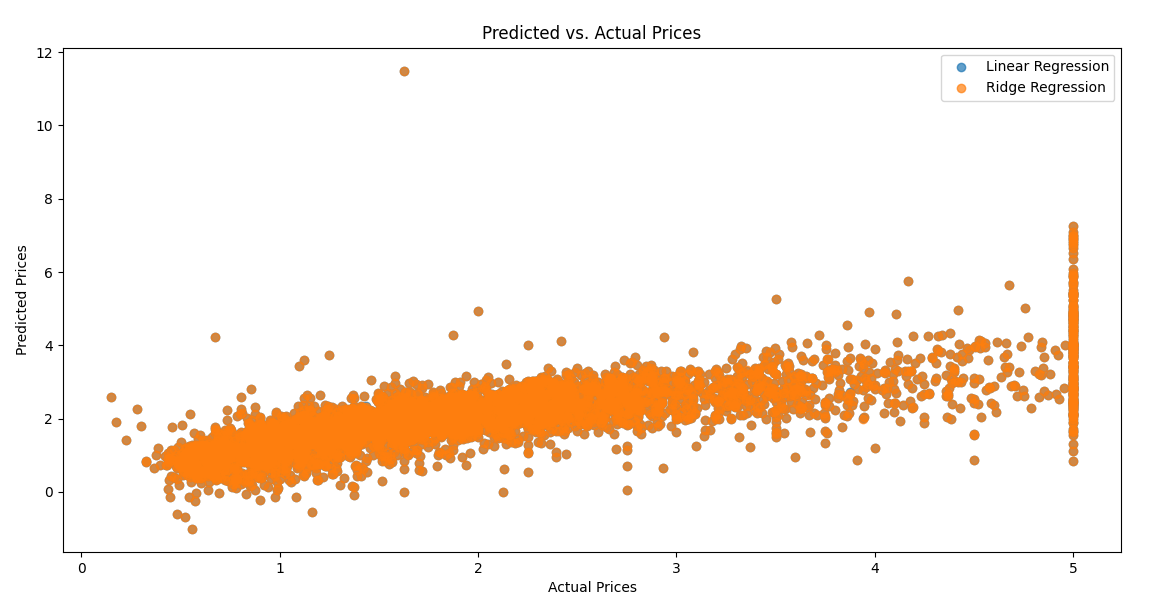
plt.figure(figsize=(12, 8))

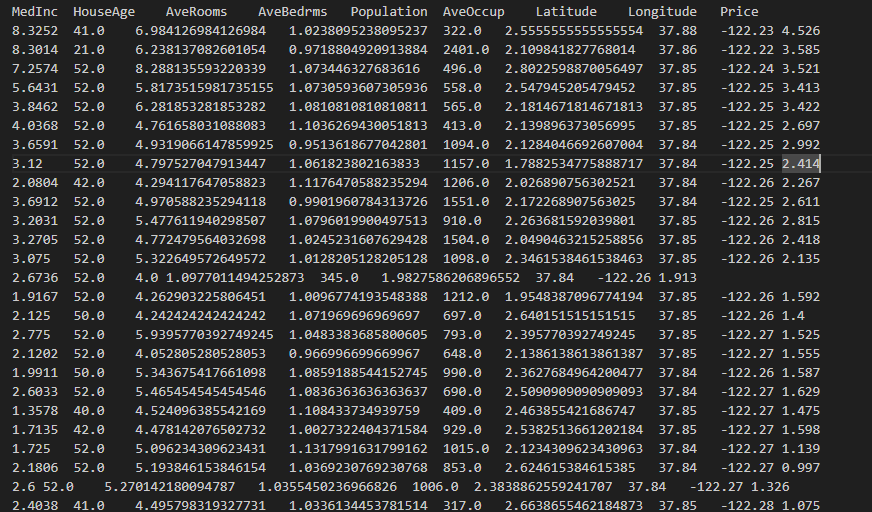
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")

plt.title("Feature Correlation Heatmap")

plt.show()

print("\nDataset saved to 'california\_housing\_data.txt'.")

**OUTPUT: **

****

**CONCLUSION:**

This project successfully demonstrates how machine learning can predict housing prices by analyzing various features. The Ridge Regression model performs better than Linear Regression, as it reduces overfitting. The visualizations and feature analysis provide actionable insights for real estate professionals and researchers. This study highlights the importance of data preprocessing, feature engineering, and model evaluation in creating reliable predictive systems.