Source code:

# EBPL-DS - Smart House Price Forecasting using Regression

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

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

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_california\_housing

# Load dataset (California housing dataset as example)

data = fetch\_california\_housing()

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

df['Price'] = data.target

# Features and target

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

y = df['Price']

# 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)

# Model 1: Linear Regression

lr\_model = LinearRegression()

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

lr\_pred = lr\_model.predict(X\_test)

# Model 2: Random Forest Regressor

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

rf\_pred = rf\_model.predict(X\_test)

# Evaluation

print("Linear Regression R2 Score:", r2\_score(y\_test, lr\_pred))

print("Random Forest R2 Score:", r2\_score(y\_test, rf\_pred))

# Plotting actual vs predicted

plt.figure(figsize=(10, 5))

plt.plot(y\_test.values[:50], label="Actual Prices")

plt.plot(rf\_pred[:50], label="Predicted Prices (RF)")

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

plt.xlabel("Sample")

plt.ylabel("Price")

plt.legend()

plt.show()