AI\_PHASE-4: HOUSE PRICE PREDICTION

**DEVELOPMENT PART 2**

***Step 1: Data Preprocessing and* Exploration**

Before building the machine learning model, we need to prepare the data:

Import necessary libraries (e.g., pandas, numpy, scikit-learn).

Load and explore the dataset.

Handle missing data, if any.

Encode categorical variables (if present) using techniques like one-hot encoding.

Split the dataset into training and testing sets.

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

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

data = pd.read\_csv("USA\_Housing.csv")

data.dropna(inplace=True)

data = pd.get\_dummies(data, columns=['State'])

X = data.drop(['Price'], axis=1)

y = data['Price']

**Step 2: Feature Selection**

Select the features (independent variables) that are most relevant for predicting house prices. This can be done through techniques like feature importance analysis or domain knowledge.

selected\_features = X.columns

**Step 3: Model Selection** Choose a machine learning algorithm for regression (since you're predicting house prices). Common choices include:

* Linear Regression
* Random Forest Regressor
* Gradient Boosting Regressor
* Support Vector Regressor
* Neural Networks

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

model = LinearRegression()

**Step 4: Model Training**

* Instantiate the selected model.
* Train the model using the training data
* Tune hyperparameters for better performance

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

**Step 5:Model Evaluation:**

* Present the evaluation metrics used and the model's performance.
* Include cross-validation results.

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

mae = mean\_absolute\_error(y\_test, y\_pred)

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

rmse = mean\_squared\_error(y\_test, y\_pred, squared=False)

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

print("Mean Absolute Error:", mae)

print("Mean Squared Error:", mse)

print("Root Mean Squared Error:", rmse)

print("R-squared:", r2)

**Step 6: Model Interpretation** Examine the model to understand the feature importances, coefficients (for linear regression), or any other insights to explain how it makes predictions.

oefficients = model.coef\_

print("Model Coefficients:", coefficients)

**Step 7: Visualization and Analysis** Create visualizations to help understand the data and model predictions. Visualizations may include scatter plots, regression plots, and residual plots.

import matplotlib.pyplot as plt

plt.scatter(y\_test, y\_pred)

plt.xlabel("Actual Prices")

plt.ylabel("Predicted Prices")

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

plt.show()