**AI PHASE-1 PROJECT: HOUSE PRICE PREDICTION**

**PROBLEM DEFENITION AND THINKING:**

**Problem Definition:** The problem is to predict houses prices using machine learning techniques. The objective is to develop a model that accurately predicts the prices of houses based on set of features such as location, square footage, number of bedrooms and bathrooms, and other relevant factors. This project involves data preprocessing, feature engineering, model selection, training and evaluation.

**Design Thinking:**

**Data Source: Choose a dataset containing information about houses, including features like location, square footage, bedrooms, bathrooms, and price.**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

%matplotlib inline

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

#understanding the data

data.info()

**Data Preprocessing: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.**

data.head()

data.describe()

**Feature Selection: Select the most relevant features for predicting house prices.**

USA\_Housing=pd.read\_csv('/content/USA\_Housing.csv')

sns.pairplot(USA\_Housing)

sns.distplot(USA\_Housing['Price'],hist\_kws=dict(edgecolor="black", linewidth=1),color='Blue')

#Displaying correlation among all the columns

USA\_Housing.corr()

sns.heatmap(USA\_Housing.corr(), annot = True)

**Model Selection: Choose a suitable regression algorithm (e.g., Linear Regression) for predicting house prices.**

#getting all column names

USA\_Housing.columns

# Columns as Features

X = USA\_Housing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',

       'Avg. Area Number of Bedrooms', 'Area Population']]

# Price is my Target Variable, what we trying to predict

y = USA\_Housing['Price']

**Model Training: Train the selected model using the preprocessed data.**

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=101)

#importing the Linear Regression Algorithm

from sklearn.linear\_model import LinearRegression

#creating LinearRegression Object

lm = LinearRegression()

#Training the Data Model

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

**Evaluation: Evaluate the model's performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.**

#Displaying the Intercept

print(lm.intercept\_)

coeff\_data = pd.DataFrame(lm.coef\_, X.columns, columns=['Coefficient'])

coeff\_data

#predictions

predictions = lm.predict(X\_test)

plt.scatter(y\_test, predictions, edgecolor='black')

sns.distplot((y\_test - predictions), bins = 50, hist\_kws=dict(edgecolor="black", linewidth=1),color='Blue')

from sklearn import metrics

print('MAE:', metrics.mean\_absolute\_error(y\_test, predictions))

print('MSE:', metrics.mean\_squared\_error(y\_test, predictions))

print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, predictions)))