**AI Project Report**

**Group Members :**

**Abdur Rahman 21MDBCS153**

**Wasif Ullah 21MDBCS105**

**Muhammad Mubashir 21MDBCS164**

### Report: House Price Prediction using Linear Regression

#### Project Title: House Price Prediction using Linear Regression

#### Objective:

To develop a Python application that predicts house prices based on various features such as area, number of bedrooms, age of the house, etc., using a predefined dataset and the linear regression algorithm.

#### Introduction:

Predicting house prices is a common problem in real estate and finance. The ability to accurately estimate house prices can provide significant benefits to buyers, sellers, and real estate professionals. In this project, we use a linear regression model to predict house prices based on the California Housing Prices dataset.

#### Dataset:

We use the California Housing Prices dataset available in the scikit-learn library. This dataset contains information on various features such as the average number of rooms, average number of bedrooms, population, median income, and more, across different districts in California.

#### Tools and Technologies:

* **Programming Language**: Python
* **Libraries**:
  + pandas for data manipulation
  + numpy for numerical operations
  + scikit-learn for machine learning
  + matplotlib for data visualization

#### Methodology:

**Data Loading**: The dataset is loaded using the fetch\_california\_housing function from scikit-learn.

from sklearn.datasets import fetch\_california\_housing

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

df = housing.frame

**Data Exploration**: We explore the dataset by displaying the first few rows and basic statistics. Visualization is done using matplotlib.

import pandas as pdimport matplotlib.pyplot as plt

print(df.head())print(df.describe())

pd.plotting.scatter\_matrix(df, figsize=(12, 12))

plt.show()

**Data Splitting**: The dataset is split into training and testing sets using train\_test\_split.

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

X = df.drop(columns=["MedHouseVal"])

y = df["MedHouseVal"]

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

**Model Training**: We train a linear regression model using the training set.

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

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

**Model Evaluation**: The model's performance is evaluated using Mean Squared Error (MSE). Predictions are made on the test set and compared with the true values.

from sklearn.metrics import mean\_squared\_error

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

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

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

plt.xlabel("True Values")

plt.ylabel("Predictions")

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

plt.show()

#### Results:

The model's Mean Squared Error (MSE) is calculated to assess the accuracy of the predictions. A scatter plot is used to visualize the relationship between the true and predicted house prices. The closer the points are to the diagonal line, the better the model's predictions.





