**Department of Computer Science and Engineering** **Data Science**

**Academic Year:** 2024-2025 **Name of Student:** Diya Thakkar

**Semester:** VI **Student ID:** 22107040

**Class / Branch:** TE CSE (DS) **Date of Performance:**27-1-25 **Subject:** ML Lab **Date of Submission:**27-1-25

**Name of Instructor:** Prof.Ujwala Pagare

**Experiment No. 2**

**Aim:**- To implement Linear Regression using python.

**Program:-**

**import pandas as pd**

**import** numpy **as** np

**import** seaborn **as** sns

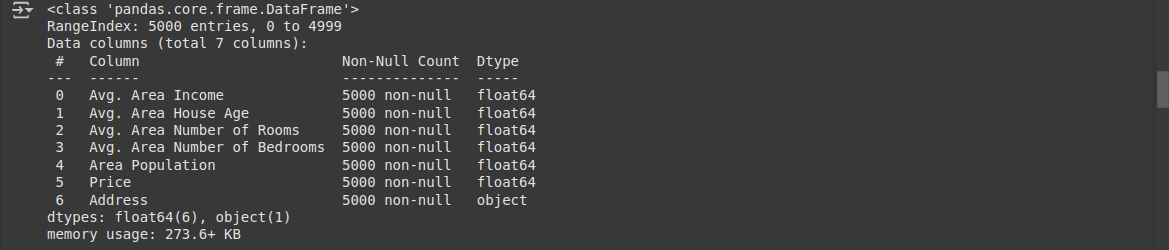
**import** matplotlib.pyplot **as** plt

**from** sklearn.metrics **import** mean\_squared\_error

HouseDF **=** pd.read\_csv('USA\_Housing.csv')

HouseDF.head()

**Output:**

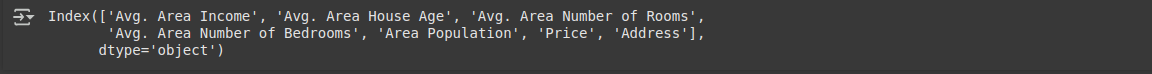
HouseDF.info()

HouseDF.describe()

**Output:**

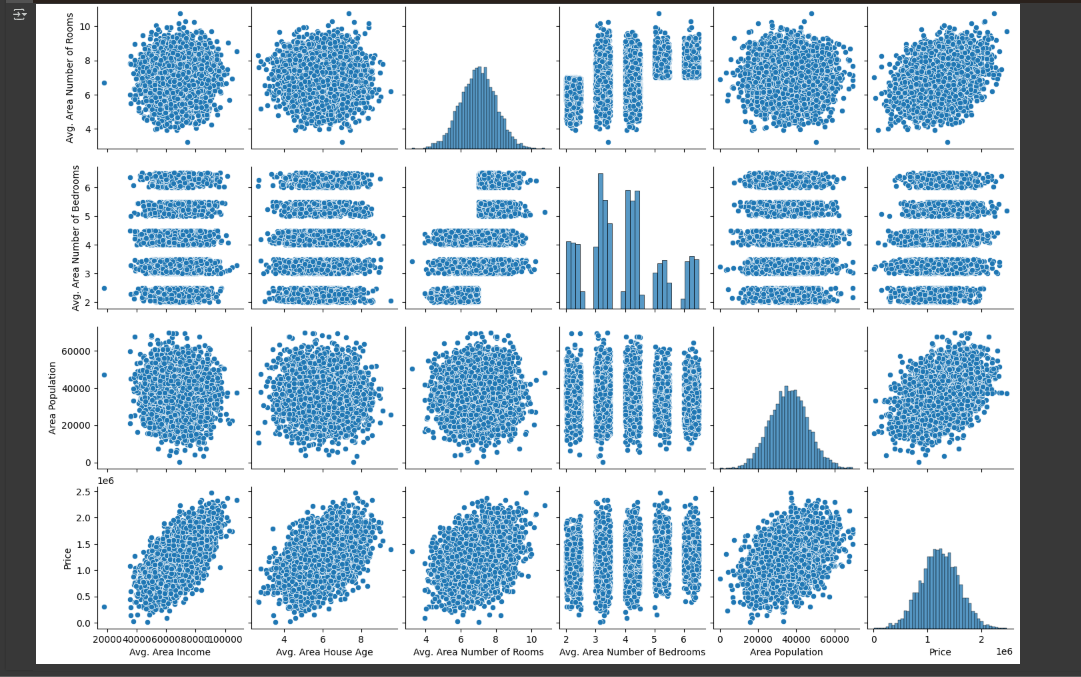
HouseDF.columns

Output:

****

sns.pairplot(HouseDF)

**Output:**

****

*#sns.heatmap(HouseDF.corr(), annot=True)*

*# Select only numerical features for correlation calculation*

numerical\_features **=** HouseDF.select\_dtypes(include**=**np.number)

*# Calculate and plot the correlation matrix*

sns.heatmap(numerical\_features.corr(), annot**=True**)

**Output:**



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

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

y **=** HouseDF['Price']

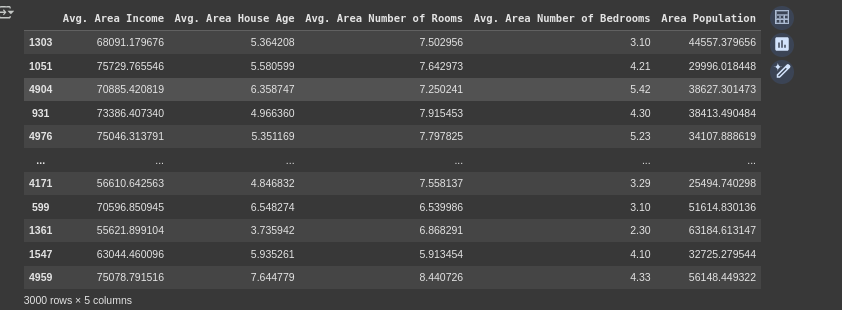
**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size**=0.4**,

random\_state**=101**)

X\_train

Output:



**from** sklearn.linear\_model **import** LinearRegression

lm **=** LinearRegression()

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

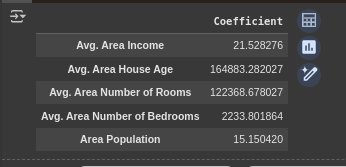
**Output:**

****

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

coeff\_df

**Output:**

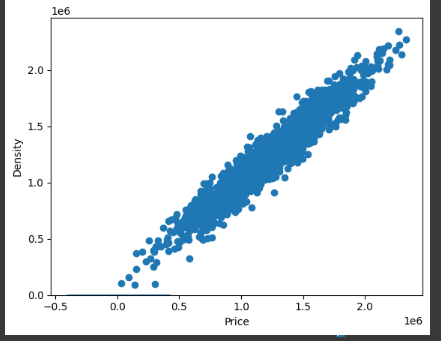
****

predictions **=** lm.predict(X\_test)

plt.scatter(y\_test,predictions)

sns.distplot((y\_test**-**predictions),bins**=50**);

Output:



*#Regression Evaluation Metrics*

*#Mean Squared Error and R-squared for model evaluation*

mse **=** mean\_squared\_error(y\_test, predictions)

mse

**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)))

*# Instead of overwriting y\_test, create a new variable*

*# for the numpy array to avoid confusion and potential errors.*

y\_test\_numeric **=** np.array(HouseDF.loc[X\_test.index, 'Price'], dtype**=**np.float64)

*# Now use y\_test\_numeric for the MSE calculation*

mse\_sum **=** **0**

**for** i **in** **range**(**len**(y\_test\_numeric)):

prediction **=** lm.predict(X\_test.iloc[[i]])

mse\_sum **+=** (y\_test\_numeric[i] **-** prediction)**\*\*2**

mse **=** mse\_sum **/** **len**(y\_test\_numeric)

**print**(mse)

**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)))

