# Commented out IPython magic to ensure Python compatibility.

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

# %matplotlib inline

#sns.set\_style('whitegrid')

# loading csv data to dataframe

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

# checking out the Data

USA\_Housing.head()

#checking columns and total records

USA\_Housing.info()

USA\_Housing.describe()

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)

USA\_Housing.columns

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']

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)

from sklearn.linear\_model import LinearRegression

lm = LinearRegression()

#Training the Data Model

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

#Displaying the Intercept

print(lm.intercept\_)

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

coeff\_df

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)))