Experiment – 2

Student Name:Simranpreet Kaur **UID: 24MAI10051**

Branch: CSE-AIML **Section/Group: 24MAI-1**

Semester: 2 **Date of Performance:**

Subject: Machine Learning Lab **Subject Code: 24CSH-667**

Aim: Write a program in Python to implement Multiple Regression Algorithm.

Software Requirements:

- Windows 11
- Python IDE
- Jupyter Notebook

Theory: Multiple linear regression is a statistical method used to predict the value of a dependent variable based on the values of two or more independent variables. It extends the concept of simple linear regression, which considers only one independent variable, to incorporate multiple predictors.

The equation for multiple linear regression can be expressed as:

$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + ... + \beta_pX_p + \varepsilon$

Where:

Y:The dependent variable (the variable we're trying to predict)

β_0:The intercept (the value of Y when all independent variables are zero)

$\beta_1...\beta_p$:The coefficients (slopes) for each independent variable ($X_1...X_p$)

$X_1...X_p$:The independent variables (the predictors)

ε:The error term (the difference between the predicted value of Y and the actual value)

The equation shows that the predicted value of Y is a combination of:

1. A constant term (β_0):This represents the starting point or baseline value of Y.
2. The weighted sum of the independent variables: Each independent variable (X) is multiplied by its corresponding coefficient (β), and these products are then summed. The coefficients represent the influence of each independent variable on Y.
3. An error term (ε): This accounts for the fact that the model may not perfectly predict the actual value of Y.

Key Points:

1. Multiple predictors:Multiple linear regression allows us to consider the combined influence of multiple factors on the dependent variable.
2. Flexibility:It can handle various types of relationships between the variables, as long as they are approximately linear.
3. Wide applications:It's used in various fields, including economics, finance, social sciences, and engineering, for tasks like forecasting, risk assessment, and decision-making.

By estimating the coefficients (β_0, β_1, β_2, ...) from the available data, we can use the multiple linear regression equation to predict the value of the dependent variable for new observations.

Source Code:

Dataset: https://drive.google.com/file/d/1sQmsFM4G_YilGZtuwl1LT3sVBxsZfR5G/view

```
import pandas as pd

import numpy as np

import warnings

from sklearn import linear_model

import seaborn as sns

import matplotlib.pyplot as plt
```

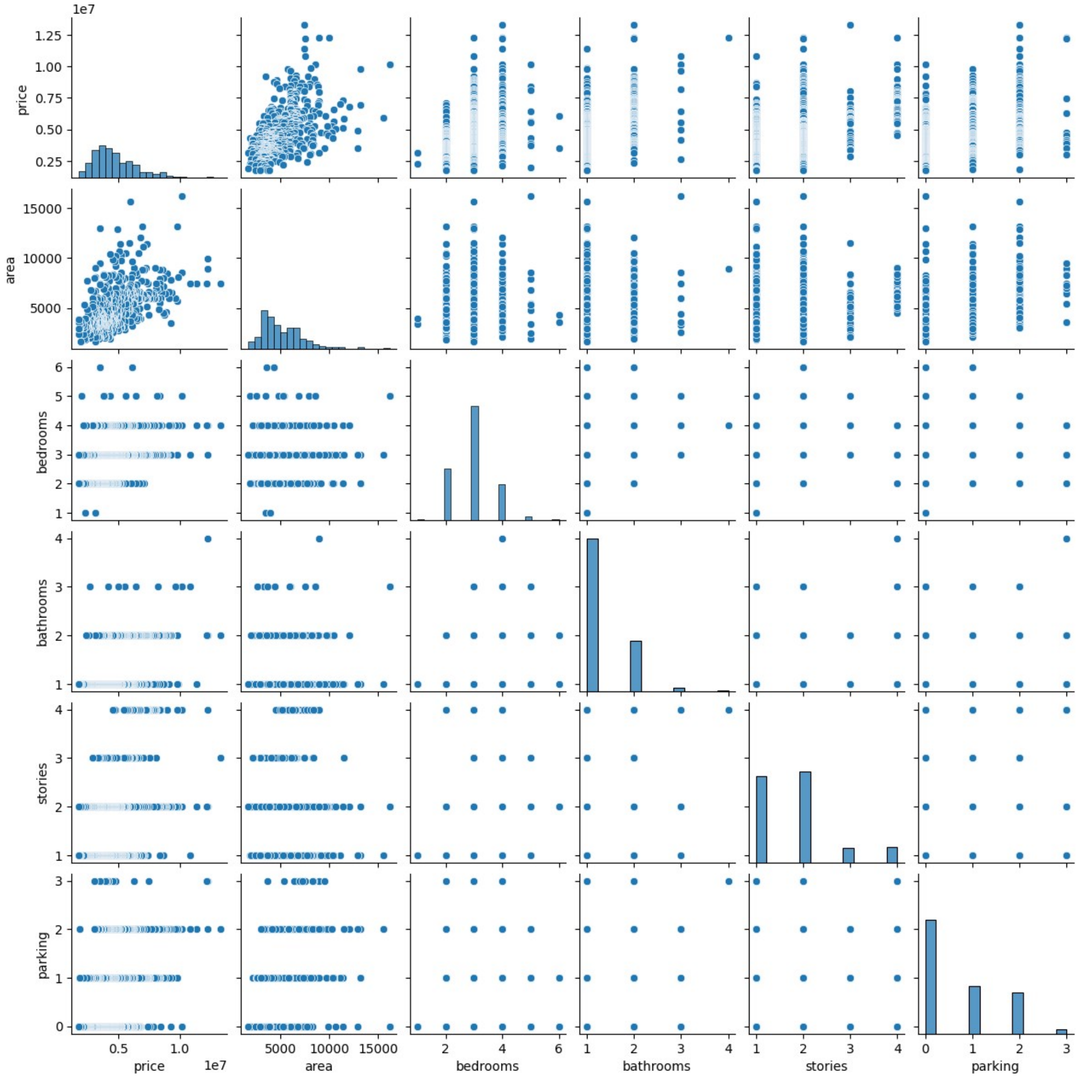
```
from mpl_toolkits.mplot3d import Axes3D

%matplotlib inline


# Load the dataset

data = pd.read_csv("Housing.csv")


# Visualizing the relationships between features using pair plots

sns.pairplot(data=data, height=2)


# Visualizing multicollinearity between independent features using a heatmap

corr = data[['area', 'bedrooms', 'stories']].corr()


print('Pearson correlation coefficient matrix for each independent variable: \n', corr)


# Masking the diagonal cells

masking = np.zeros_like(corr, dtype=bool)

np.fill_diagonal(masking, val=True)


# Initializing a matplotlib figure

figure, axis = plt.subplots(figsize=(4, 3))


# Generating a custom colormap

c_map = sns.diverging_palette(223, 14, as_cmap=True, sep=100)
```

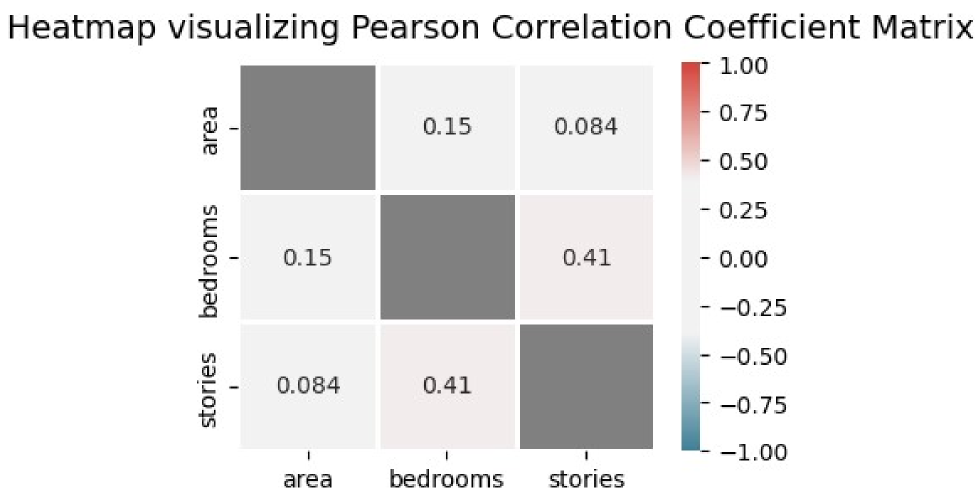
```
c_map.set_bad('grey')

# Displaying the heatmap with the masking and the correct aspect ratio
sns.heatmap(corr, mask=masking, cmap=c_map, vmin=-1, vmax=1, center=0,
linewidths=1, annot=True)

figure.suptitle('Heatmap visualizing Pearson Correlation Coefficient Matrix', fontsize=14)
axis.tick_params(axis='both', which='major', labelsize=10)

# Building the Multiple Linear Regression Model
# Setting the independent and dependent features
X = data.drop("price", axis=1).values
y = data["price"].values

# One-hot encode categorical variables
X = pd.get_dummies(data.drop("price", axis=1), drop_first=True).values

# Initializing the model class from the sklearn package and fitting our data into it
reg = linear_model.LinearRegression()
reg.fit(X, y)

# Printing the intercept and the coefficients of the regression equation
print('Intercept: ', reg.intercept_)
print('Coefficients array: ', reg.coef_)
```

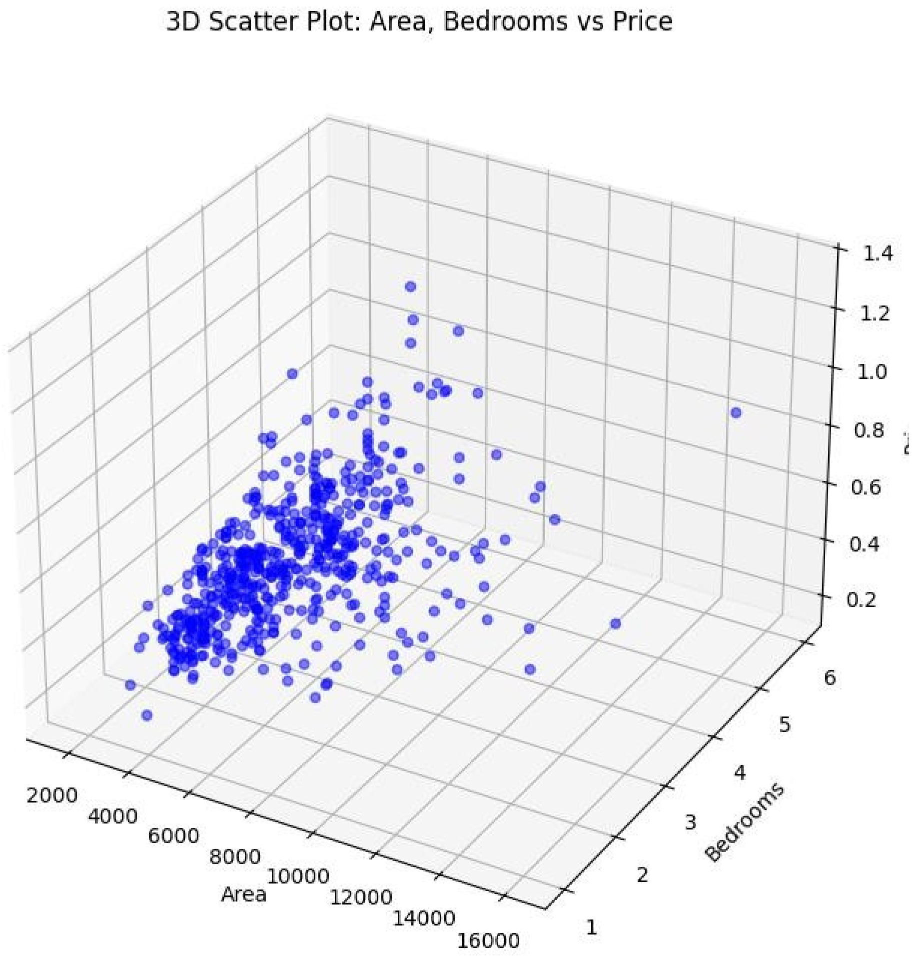
```
# Plotting a 3-D plot for visualizing the Multiple Linear Regression Model

fig = plt.figure(figsize=(10, 8))

ax = fig.add_subplot(111, projection='3d')

ax.scatter(data["area"], data["bedrooms"], data["price"], c='blue', marker='o', alpha=0.5)

ax.set_title("3D Scatter Plot: Area, Bedrooms vs Price")

ax.set_xlabel("Area")

ax.set_ylabel("Bedrooms")

ax.set_zlabel("Price") plt.show()
```

Output:

```
Pearson correlation coefficient matrix for each independent variable:
              area  bedrooms   stories
area      1.000000  0.151858  0.083996
bedrooms  0.151858  1.000000  0.408564
stories   0.083996  0.408564  1.000000
Intercept:  42771.693918105215
Coefficients array:  [ 2.44139386e+02  1.14787560e+05  9.87668107e+05  4.50848003e+05
  2.77107101e+05  4.21272589e+05  3.00525860e+05  3.50106904e+05
  8.55447145e+05  8.64958311e+05  6.51543800e+05 -4.63446200e+04
 -4.11234386e+05]
```


1e7
price
area
bedrooms
bathrooms
stories
parking
1e7


Heatmap visualizing Pearson Correlation Coefficient Matrix
area
bedrooms
stories
0.15
0.084
0.15
0.41
0.084
0.41
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



Learning Outcomes:

1. Understood multiple linear regression
2. Learned data processing process for multiple variables
3. Understood the multiple regression equation
4. Used python libraries for multiple regression
5. Learned about fitting the model and model evaluation