Aayush Shah
19BCE245
1 May 2021

Practical 10

Correlation Coefficient

- **Definition :** Write a program to implement (i) Pearson correlation and display heatmap 4 (ii) chi-square test

- **Code :**

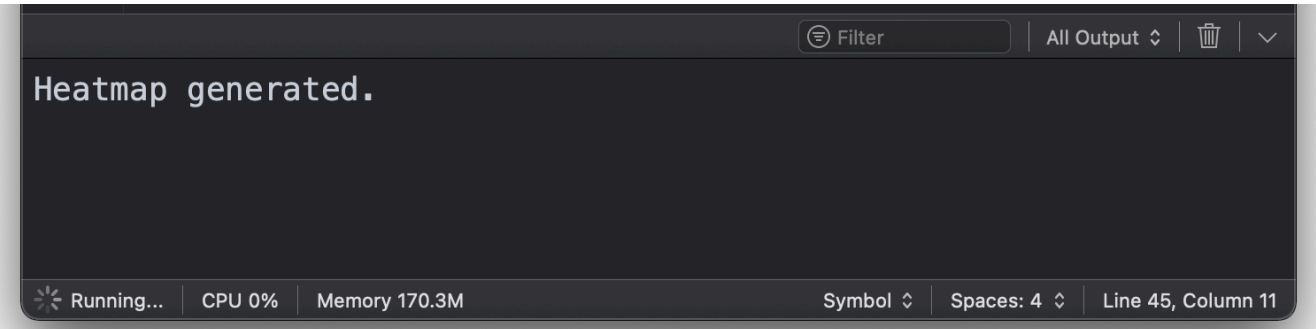
```
1. from sklearn import datasets
2. import numpy as np
3. import seaborn as sb
4. import matplotlib.pyplot as plt
5.
6. # Loading the boston_dataset in x and y
7. x , y = datasets.load_boston(return_X_y=True)
8.
9. # Creating the List
10.data_set = np.array(x[0:506,:])
11.
12.# Reshaping the 1-d array in 2-d array of 13 rows
13.data_set = data_set.reshape(13,506)
14.
15.x_mean = [ np.array(item).mean() for item in data_set ]  # Calulating the mean of all the 13 arrays and storing in the list
16.
17.data_val = []
18.for i in range(len(x_mean)):
19.    temp = []
20.    for j in data_set:
21.        num = 0
22.        den = 0
23.        for k in range(len(j)):
```

```
            num += (data_set[i][k]-x_mean[i])*(j[k]-
  np.array(j).mean())
            den += (((data_set[i][k]-x_mean[i])**2)*((j[k]-
  np.array(j).mean())**2))**0.5
        temp.append(num/den)
    data_val.append(temp)

"""
In the above nested loop, I am calculating the the pearson
  coefficient for all the 13 arrays I have created before.
for all the 13 arrays.
Let i , j represent array[i] and array[j]
and matrix[i][j] represents the pearson coefficient for
  array i and array j
theefore, the pearson coefficient for matrix[i][j] =
  matrix[j][i]
and for j = i, the pearson coefficient for matrix[i][i] = 1
"""

# Plotting the heatmap of that matrix
heat_map = sb.heatmap(data_val,annot=True,cmap='Oranges')
plt.title("Heat Map")
plt.xlabel("X - Axis")
plt.ylabel("Y - Axis")
plt.show()
```

- **Sample I/O :**



- **Generated Graph :**