Assignment No. 3

Problem Statement: a. Find the correlation matrix on the iris dataset. b. Plot the correlation plot on the dataset and visualize giving an overview of relationships among data on iris dataset.

Objective:

To analyze the relationships among the features of the iris dataset through a correlation matrix and visual representation helps in understanding how different variables are related to each other. By identifying the strength and direction of correlations, we can uncover patterns that may not be immediately apparent from the raw data. This analysis is essential for feature selection in machine learning, guiding us in choosing the most relevant features for model training.

Prerequisite :

1. Understanding of libraries like Pandas, NumPy, Matplotlib, and Seaborn.
2. Experience with data visualization techniques using libraries like Matplotlib or Seaborn (optional).
3. Text editor and basic knowledge of file handling in Python.

Theory :

Covariance and **correlation** are two terms that are opposed and are both used in statistics and regression analysis.

Covariance:

- Covariance is a measure of the relationship between two random variables. It measures how two random random variables vary together.
- Covariance is a quantitative measure of the degree to which the deviation of one variable (X) from its mean is related to the deviation of another variable (Y) from its mean.
- Covariance measures the joint variability of two random variables.
- Covariance can be negative or positive (or zero). A positive value of Covariance means that two random variables tend to vary in the same direction
- Negative value means that they vary in opposite directions, and a 0 means that they don’t vary together.
- Covariance values can vary from $-\infty$ to $+\infty$.

- **Formula:**

$$Cov_{x,y} = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

- **Types of Covariance:**
- Positive and negative covariance.
- Positive covariance means both the variables (X, Y) move in the same direction (i.e. show similar behavior).
- Negative covariance means both the variables (X, Y) move in the opposite direction.

Covariance Matrix

The covariance matrix is also known as the variance-covariance matrix, as the diagonal values of the covariance matrix show variances and the other values are the covariances. To determine the covariance matrix, the formulas for variance and covariance are required. Depending upon the type of data available, the variance and covariance can be found for both sample data and population data.

Population Variance: $\text{var}(x) = \frac{\sum_1^n (x_i - \mu)^2}{n}$

Population Covariance: $\text{cov}(x, y) = \frac{\sum_1^n (x_i - \mu_x)(y_i - \mu_y)}{n}$

Sample Variance: $\text{var}(x) = \frac{\sum_1^n (x_i - \bar{x})^2}{n-1}$

Sample Covariance: $\text{cov}(x, y) = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n-1}$

μ = mean of population data.

$\bar{x}$ = mean of sample data.

n = number of observations in the dataset.

xi = observations in dataset x.

Correlation:

- Correlation tells us both the strength and the direction of this relationship. Correlation helps us to determine whether or not, and how strongly, changes in various variables relate to each other.
- The most important part of Correlation is that it is bounded between -1 and 1. -1 and 1 are known as perfect Correlation.
- **Types of Correlation:**
- **Negative** and **Positive** correlation.
    - **Positive correlation:**
      The correlation is said to be positive when the variables move together in the same direction. Ex. When the income rises, consumption also rises.
    - **Negative correlation:**
      The correlation is negative when they move in opposite directions.
      Ex. When the price of apples falls its demand increases.

- If Correlation is 0 means no relationship between two variables.

TECHNIQUES FOR MEASURING CORRELATION:

Scatter Diagram: A scatter diagram is a useful technique for visually examining the form of relationship, without calculating any numerical value.

Karl Pearson's Coefficient of Correlation: This is also known as product moment correlation coefficient or simple correlation coefficient. It gives a precise numerical value of the degree of linear relationship between two variables X and Y. It is important to note that Karl Pearson's coefficient of correlation should be used only when there is a **linear relation between the variables.**

formula:

$$r = \frac{\Sigma(X-\overline{X})(Y-\overline{Y})}{\sqrt{\Sigma(X-\overline{X})^2}\sqrt{\Sigma(Y-\overline{Y})^2}}$$

Correlation Matrix:

- We use correlation coefficients to determine the relationship between two variables. if we want to evaluate the correlation among multiple pairs of variables. Then we use a correlation matrix.
- A correlation matrix is essentially a table depicting the correlation coefficients for various variables.

Algorithm (if any to achieve the objective)

1. I**mport Libraries:**
    - Import necessary libraries: pandas, seaborn, matplotlib.pyplot.
2. **Load the Iris Dataset:**
    - Use a suitable method to load the iris dataset into a DataFrame.
3. **Calculate the Correlation Matrix:**
    - Compute the correlation matrix.
4. **Display the Correlation Matrix:**
    - Print the correlation matrix to review the correlation coefficients.
5. **Plot the Correlation Matrix:**
    - Create a heatmap to visualize the correlation matrix:
        a. Use sns.heatmap() to plot the correlation matrix.
        b. Customize the heatmap with options for annotations, color map, and square formatting.

Code & Output

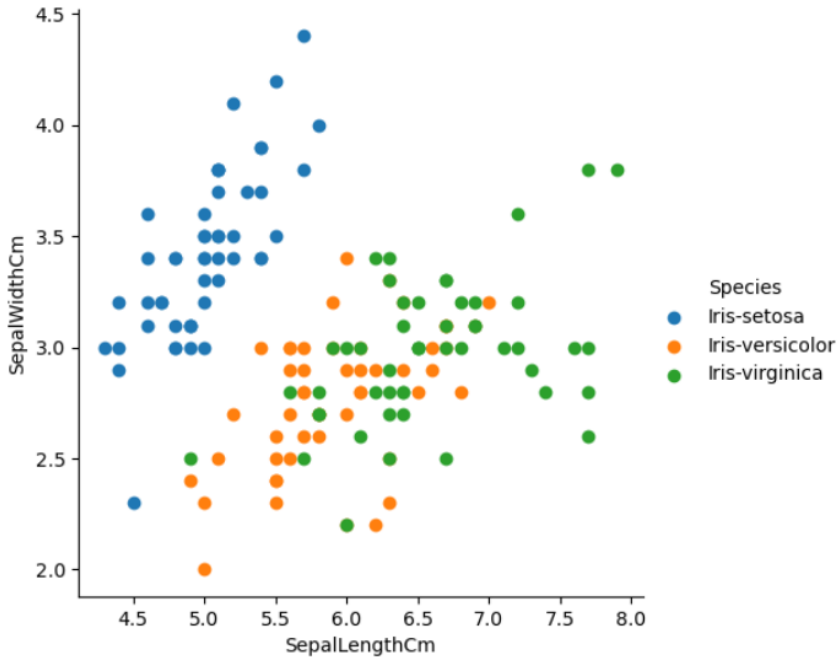
```
[11]: import pandas as pd
      dt = pd.read_csv("C:/Users/dnyan/FODS Assignments/Datasets/iris.csv")

[15]: import seaborn as sns
      import matplotlib.pyplot as plt
      from math import sqrt

[17]: dt.head()
```

[17]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

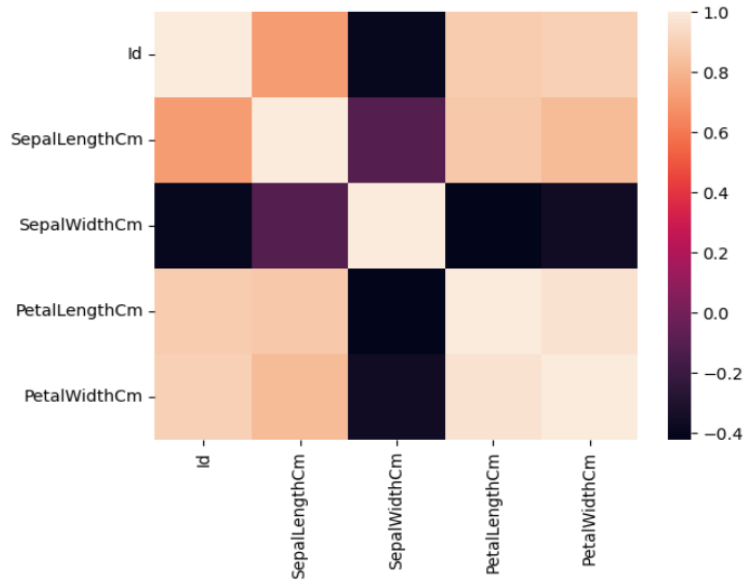
```
[32]: #classes separation
      import seaborn as sns
      import matplotlib.pyplot as plt

      g = sns.FacetGrid(dt, hue="Species", height=5)
      g.map(plt.scatter, "SepalLengthCm", "SepalWidthCm")
      g.add_legend()
```

[32]: <seaborn.axisgrid.FacetGrid at 0x26a28fda5d0>



[34]:
```
numeric_df = dt.select_dtypes(include = [float, int])
```

[40]:
```
# import correlation matrix to see parametrs which best correlate each other
import seaborn as sns
corr = numeric_df.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values)
plt.show()
```



[7]:
```
# function for calculating mean
import pandas as pd
import numpy as np
def mean_impl(dt):
    data=np.asarray(dt).flatten()
    s=dt.sum()/len(dt)
    return s
```

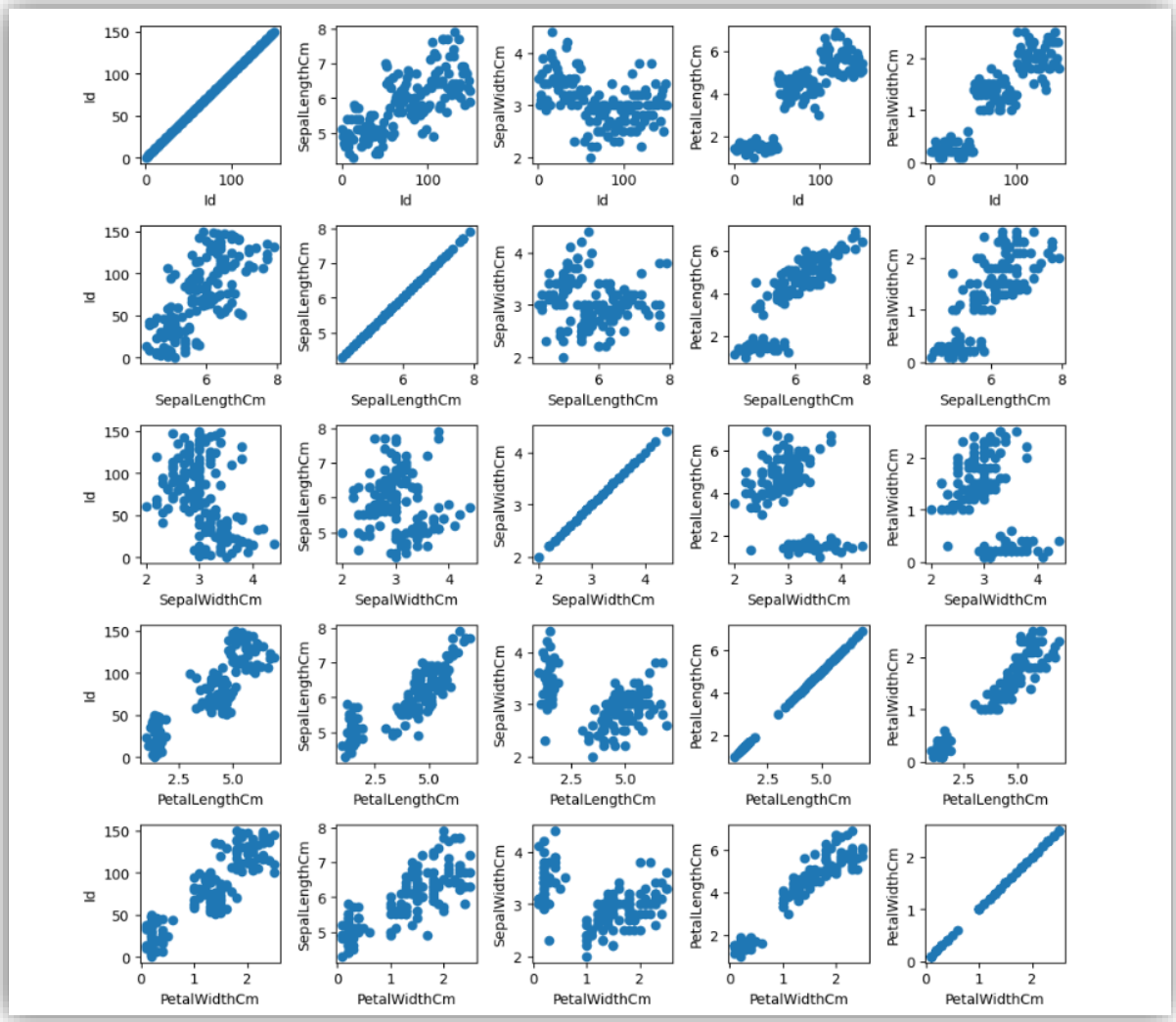
[8]:
```
# Covariance calculation
def covariance_implmentation(dt):
    size = dt.shape[1]  # Gives the number of columns

    # Computes the deviation of the values in the first column from the mean of that column
    a = np.asarray(dt.iloc[:, 0]).flatten() - mean_impl(dt.iloc[:, 0])
    if size == 2:
        b = np.asarray(dt.iloc[:, 1]).flatten() - mean_impl(dt.iloc[:, 1])
    else:
        b = a  # Default case for larger dimensions

    # Computes the sum of the element-wise product of a and b
    p = (a * b).sum()
    return p / (dt.shape[0] - 1)  # Return covariance

# Load the Iris dataset
dt = pd.read_csv("C:/Users/dnyan/FODS Assignments/Datasets/iris.csv")

# Compute the covariance between the first two columns
cov_value = covariance_implmentation(dt.iloc[:, 0:2])
print(cov_value)
```

```
25.782885906040267
```

[13]:
```
import numpy as np

def Covariance_matrix(dt):
    # Filter for numeric columns only
    data_numeric = dt.select_dtypes(include=[np.number])

    result=[]

    # Iterate over each pair of columns in the dataset
    for i in range(data_numeric.shape[1]):
        for j in range(data_numeric.shape[1]):

            # Compute the covariance between the ith and jth columns using covariance_implmentation
            result.append(covariance_implmentation(data_numeric.iloc[:,[i,j]]))
    print(np.asarray(result).reshape(data_numeric.shape[1],data_numeric.shape[1]))
Covariance_matrix(dt)
```

```
[[ 1.88750000e+03  2.57828859e+01 -7.49228188e+00  6.76677852e+01
   2.98322148e+01]
 [ 2.57828859e+01  6.85693512e-01 -3.92684564e-02  1.27368233e+00
   5.16903803e-01]
 [-7.49228188e+00 -3.92684564e-02  1.88004027e-01 -3.21712752e-01
  -1.17981208e-01]
 [ 6.76677852e+01  1.27368233e+00 -3.21712752e-01  3.11317942e+00
   1.29638747e+00]
 [ 2.98322148e+01  5.16903803e-01 -1.17981208e-01  1.29638747e+00
   5.82414318e-01]]
```

[18]:
```
import matplotlib.pyplot as plt
%matplotlib inline

# generate a grid of scatter plots showing pairwise comparisons
# between different columns of a dataset
def scatterplot_impl(dt):

    # Plot position index
    p=1

    # Set figure size for the plot grid
    plt.figure(figsize=(10, 10))

    for i in range(dt.shape[1]):
        for j in range(dt.shape[1]):
            plt.subplot(dt.shape[1],dt.shape[1],p)    # Create a subplot in the grid
            plt.scatter(dt.iloc[:,i], dt.iloc[:,j])   # Create a scatter plot for column pair (i, j)
            plt.xlabel(dt.columns[i])    # Label the x-axis with the name of the ith column
            plt.ylabel(dt.columns[j])    # Label the y-axis with the name of the jth column

            # Move to the next plot position
            p+=1

    plt.tight_layout()   # Adjust layout so plots do not overlap
    plt.show()           # Display the plot grid
scatterplot_impl(dt.select_dtypes(include=[np.number]))
```



[41]:
```
# Select only numeric columns for correlation
dt = pd.read_csv("C:/Users/dnyan/FODS Assignments/Datasets/iris.csv")

numeric_columns = dt.select_dtypes(include='number')

# Calculate the correlation matrix
correlation_matrix = numeric_columns.corr()

# Display the correlation matrix
print("Correlation Matrix:")
correlation_matrix
```

```
Correlation Matrix:
```

[41]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Id	1.000000	0.716676	-0.397729	0.882747	0.899759
SepalLengthCm	0.716676	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.397729	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.882747	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.899759	0.817954	-0.356544	0.962757	1.000000

References :

1. **https://colab.research.google.com/drive/1vLl9UjTLVbiJmz29tQ6uYiQGX77StDjc**
2. **https://colab.research.google.com/drive/1VxeVlhoS2VCYHayqizbk09hG9jfFxldn**
3. **https://colab.research.google.com/drive/1HHBJ4ovZ-NaYn1l3xF1katqPT6JMOinx**
4. **https://www.kaggle.com/code/dronio/iris-plots-correlation-matrix**
5. **https://www.kaggle.com/code/ajay101tiwari/covariance-and-correlation-python-implementation**
6. **https://www.cuemath.com/algebra/covariance-matrix/**
7.

Conclusion :

In this assignment, we analyzed the correlation matrix of the iris dataset provides insights into the relationships between different features. Visualizing these correlations through a heatmap enhances understanding and can guide further analysis or modeling efforts. This foundational analysis is crucial for more advanced statistical methods and machine learning applications.