**A PYTHON PROGRAM TO IMPLEMENT DIMENSIONALITY**

**REDUCTION USING PCA**

**Ex.No.: 10**

**Date of Experiment: 8/10/2024**

**AIM:-**

To implement Dimensionality Reduction using PCA in a python program.

# ALGORITHM:-

Step1: Import all the necessary libraries and modules(pandas as pd, StandardScalar from sklearn.preprocessing, PCA from sklearn.decomposition and seaborn as sns).

Step2: Import the “iris dataset” from the sklearn library.

Step3: Load the dataset and convert it into a pandas dataframe.

Step4: Standardize the features using the “StandardScalar()” function and create an object of that and display it.

Step5: Use the “sns.heatmap()” function and find out the correlation between the features prior to applying PCA and display it visually.

Step6: Consider 3 principal components, fit them using the “pca.fit()” function, transform using the “pca.transform()” function and convert into a pandas dataframe.

Step7: Once again check the correlation between the features after PCA has been applied on the dataset.

Step8: Visually display the correlation after applying PCA using the “sns.heatmap()” function. **IMPLEMENTATION:-**

from sklearn import datasets

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

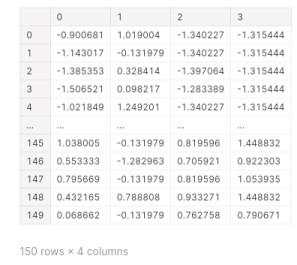
import seaborn as sns

iris = datasets.load\_iris()

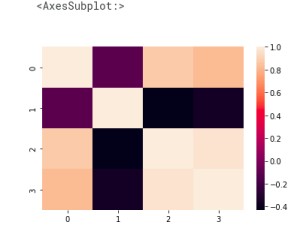
df = pd.DataFrame(iris['data'], columns = iris['feature\_names']) df.head()



scalar = StandardScaler() scaled\_data = pd.DataFrame(scalar.fit\_transform(df)) #scaling the data scaled\_data



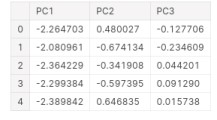
sns.heatmap(scaled\_data.corr())



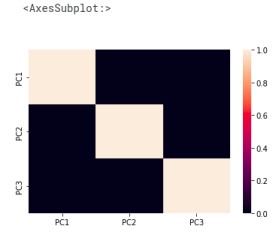
pca = PCA(n\_components = 3)

pca.fit(scaled\_data)

data\_pca = pca.transform(scaled\_data) data\_pca = pd.DataFrame(data\_pca,columns=['PC1','PC2','PC3']) data\_pca.head()



sns.heatmap(data\_pca.corr())



# RESULT:-

Thus Dimensionality Reduction has been implemented using PCA in a python program successfully and the results have been analyzed.