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()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

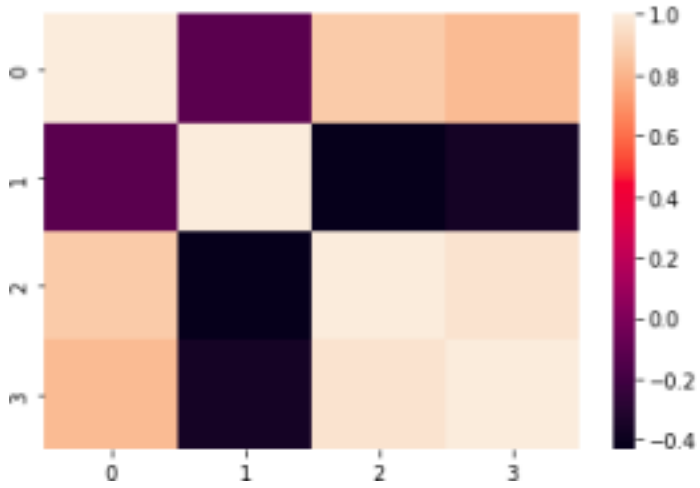
```
scalar = StandardScaler()
scaled_data = pd.DataFrame(scalar.fit_transform(df)) #scaling the data
scaled_data
```

	0	1	2	3
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444
3	-1.506521	0.098217	-1.283389	-1.315444
4	-1.021849	1.249201	-1.340227	-1.315444
...	...	...	...	...
145	1.038005	-0.131979	0.819596	1.448832
146	0.553333	-1.282963	0.705921	0.922303
147	0.795669	-0.131979	0.819596	1.053935
148	0.432165	0.788808	0.933271	1.448832
149	0.068662	-0.131979	0.762758	0.790671

150 rows × 4 columns

```
sns.heatmap(scaled_data.corr())
```

```
<AxesSubplot:>
```

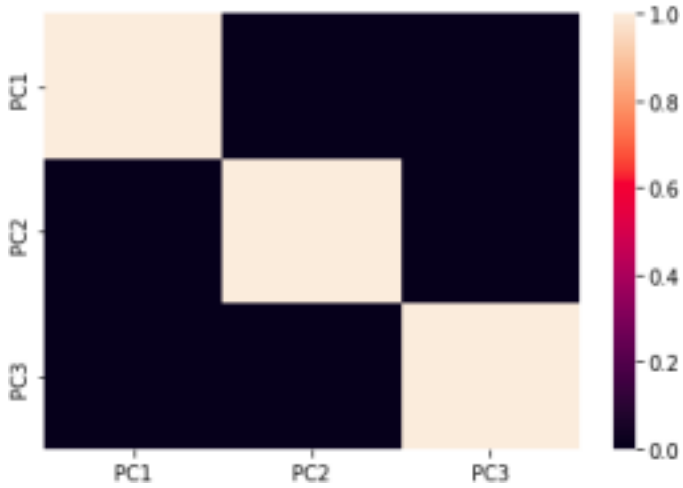



```
pca = PCA(n_components = 3)
pca.fit(scaled_data)
data_pca = pca.transform(scaled_data)
data_pca = pd.DataFrame(data_pca,columns=['PC1','PC2','PC3'])
data_pca.head()
```

	PC1	PC2	PC3
0	-2.264703	0.480027	-0.127706
1	-2.080961	-0.674134	-0.234609
2	-2.364229	-0.341908	0.044201
3	-2.299384	-0.597395	0.091290
4	-2.389842	0.646835	0.015738

```
sns.heatmap(data_pca.corr())
```

```
<AxesSubplot:>
```



RESULT:-

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