Machine Learning Lab – 20ISL68A

Program 1 - Implement and demonstrate the Principal Component Analysis for dimensionality reduction. Read the training data set from a .CSV file.

```
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

iris=load_iris()

numsamples,numfeatures=iris.data.shape
print(numsamples)
print(numfeatures)
print(iris.target_names)

x=iris.data
pca=PCA(n_components=2).fit(x)
z=pca.transform(x)
z

print(pca.components_)
print(pca.explained_variance_ratio_)
print(sum(pca.explained_variance_ratio_))

y=iris.target
plt.scatter(z[:,0],z[:,1],c=y)
```

Description

- It is a linear dimension reduction algorithm.
- It is a projection based method that transforms the data by projecting it onto a set of orthogonal (perpendicular) axes.

Execution

Step 1: Importing Libraries

```
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

Step 2: Loading the Iris Dataset

```
iris=load_iris()
```

Step 3: Understanding the dataset

```
numsamples,numfeatures=iris.data.shape
print(numsamples)
print(numfeatures)
print(iris.target_names)
```

```
150
4
['setosa' 'versicolor' 'virginica']
```

Step 4: Applying PCA

```
x=iris.data
pca=PCA(n_components=2).fit(x)
z=pca.transform(x)
z
```

```
Out[26]: array([[-2.68412563,  0.31939725],
                [-2.71414169, -0.17700123],
                [-2.88899057, -0.14494943],
                [-2.74534286, -0.31829898],
                [-2.72871654,  0.32675451],
                [-2.28085963,  0.74133045],
                [-2.82053775, -0.08946138],
                [-2.62614497,  0.16338496],
                [-2.88638273, -0.57831175],
                [-2.6727558 , -0.11377425],
                [-2.50694709,  0.6450689 ],
                [-2.61275523,  0.01472994],
                [-2.78610927, -0.235112  ],
                [-3.22380374, -0.51139459],
                [-2.64475039,  1.17876464],
                [-2.38603903,  1.33806233],
                [-2.62352788,  0.81067951],
                [-2.64829671,  0.31184914],
```

Step 5: PCA Components

```
print(pca.components_)
print(pca.explained_variance_ratio_)
print(sum(pca.explained_variance_ratio_))
```

```
[[ 0.36138659 -0.08452251  0.85667061  0.3582892 ]
 [ 0.65658877  0.73016143 -0.17337266 -0.07548102]]
[0.92461872 0.05306648]
0.9776852063187951
```

Step 6: Visualization

```
y=iris.target
plt.scatter(z[:,0],z[:,1],c=y)
```