Machine Learning (ML)

Name : Brekisha A. Patel
Sapid : 60004210126
Branch : Computer Engineering
Div : C2 , Batch : 1

Experiment no. 7

Aim :- To implement PCA.

Theory :-

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. It increases interpretability yet, at the same time, it minimizes information loss. It helps to find the most significant features in a dataset and makes the data easy for plotting in 2D and 3D. PCA helps in finding a sequence of linear combinations of variables.
The Principal Components are a straight line that captures most of the variance of the data. They have a direction and magnitude. Principal components are orthogonal projections (perpendicular) of data onto lower-dimensional space.
The term "dimensionality" describes the quantity of features or variables used in the research. It can be difficult to visualize and interpret the relationships between variables when dealing with high dimensional data, such as datasets with numerous variables. While reducing the number of variables in the dataset, dimensionality reduction methods like PCA are used to preserve the most crucial data. The original variables are converted to into a new set of variables called principal components, which are linear combinations of the original variables, by PCA in order to accomplish this. The dataset's reduced dimensionality depends on how many principal components are used in the study. The objective of PCA is to select fewer principal components that account for the data's most important variation. PCA can help to streamline data analysis,

PTO

enhance visualization, and make it simpler to spot trends and relationships between factors by reducing the dimensionality of the dataset.

A statistical measure known as correlation expresses the direction & strength of the linear connection between two variables. The Covariance matrix, a square matrix that displays the pairwise correlations between all pairs of variables in the dataset, is calculated in the setting of PCA using correlation. The Covariance matrix's diagonal elements stand for each variable's variance, while the off-diagonal elements indicate the covariances between different pairs of variables. The strength and direction of the linear connection between two variables can be determined using the correlation coefficient, a standardized measure of correlation with a range of -1 to 1. A correlation coefficient of 0 denotes no linear connection between the two variables, while correlation coefficients of 1 and -1 denote the perfect positive and negative correlations, respectively. The principal components in PCA are linear combinations of the initial variables that maximize the variance explained by the data. Principal components are calculated using the correlation matrix. The main components of the data are calculated using the eigenvectors. The ways in which the data varies most are represented by the eigenvectors of the data's covariance matrix. The new coordinate system in which the data is represented is then defined using these coordinates.

eg :-

	a	b
1	4	6
2	8	2
3	13	3
4	7	15
	8	6.5

$$S = \begin{bmatrix} \text{var}(x) & \text{cov}(x,y) \\ \text{cov}(x,y) & \text{var}(y) \end{bmatrix} \quad \ldots (\because \text{covariance matrix})$$

formula :- $\text{cov}(a,a) = \frac{1}{N-1} \sum_{i=1}^{4} (a_i - \bar{a})(a_i - \bar{a})$

$\bar{a} = 8$, $\bar{b} = 6.5$

$\text{Mean}(a) = \bar{a} = \frac{32}{4} = 8$

$\text{Mean}(b) = \bar{b} = \frac{26}{4} = 6.5$

$$\text{cov}(a,b) = \frac{1}{3}[(4-8)(6-6.5) + (8-8)(2-6.5) + (13-8)(3-6.5) + (7-8)(15-6.5)]$$

$\text{cov}(a,b) = -8$ //

$$\text{cov}(a,a) = \frac{1}{3}[(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2]$$

$\text{cov}(a,a) = 14$ //

$$\text{cov}(b,b) = \frac{1}{3}[(6-6.5)^2 + (2-6.5)^2 + (3-6.5)^2 + (15-6.5)^2]$$

$\text{cov}(b,b) = 35$ //

$$S = \begin{bmatrix} 14 & -8 \\ -8 & 35 \end{bmatrix} \longrightarrow \text{covariance matrix}$$

$|S - \lambda I| = 0$

$$\left|\begin{matrix} 14 & -8 \\ -8 & 35 \end{matrix}\right| - \lambda \left|\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\right| = 0$$

$$\left|\begin{matrix} 14-\lambda & -8 \\ -8 & 35-\lambda \end{matrix}\right| = 0$$

$(14-\lambda)(35-\lambda) - 64 = 0$

$490 - 49\lambda + \lambda^2 - 64 = 0$

$\lambda^2 - 49\lambda + 426 = 0$

$\lambda_1 = 37.7, \lambda_2 = 11.3 \longrightarrow$ (eigen values)

Thus, eigen values are : $\lambda_1 = 37.7, \lambda_2 = 11.3$

To find eigen vector :

$(S - \lambda I)(u_1) = 0$

Let $u_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

$$\begin{bmatrix} 14-\lambda_1 & -8 \\ -8 & 35-\lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$14 - \lambda_1 u_1 - 8u_2 = 0$

$(14-\lambda_1)u_1 = 8u_2$

$$\frac{u_1}{8} = \frac{u_2}{14-\lambda_1} = A$$

$(\because$ Let $A = 1)$

$$u_1 = \begin{bmatrix} 8 \\ -23.7 \end{bmatrix} \Rightarrow n u_1 = \begin{bmatrix} 0.32 \\ -0.95 \end{bmatrix} \quad \ldots \left(\text{by } \frac{8}{\sqrt{8^2 + (23-7)^2}}\right)$$

$$\left(\begin{bmatrix} 14 & -8 \\ -8 & 35 \end{bmatrix} - \begin{bmatrix} 11.3 & 0 \\ 0 & 11.3 \end{bmatrix} \right) \begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix} = 0$$

$u_2 = \begin{bmatrix} 8 \\ 2.7 \end{bmatrix} \rightarrow nu_2 = \begin{bmatrix} 0.95 \\ 0.32 \end{bmatrix}$

Normalized eigen vectors are :-

$nu_1 = \begin{bmatrix} 0.32 \\ -0.95 \end{bmatrix}$ & $nu_2 = \begin{bmatrix} 0.92 \\ 0.32 \end{bmatrix}$

$\left(\frac{\lambda_1}{\lambda_1+\lambda_2}, \frac{\lambda_2}{\lambda_1+\lambda_2}\right) = \left(\frac{37.7}{49}, \frac{11.3}{49}\right) = (0.77, 0.23)$

$nu_1^T = \begin{bmatrix} a-\bar{a} \\ b-\bar{b} \end{bmatrix}$

$nu_1 = \begin{bmatrix} 0.32 \\ -0.95 \end{bmatrix}$ $nu_2 = \begin{bmatrix} 0.95 \\ 0.32 \end{bmatrix}$

$nu_1^T = [0.32 \quad -0.95]$

$(\because a=4, b=6)$

$P_{11} = [0.32 \quad -0.95] \begin{bmatrix} 4 & -8 \\ 6 & -6.5 \end{bmatrix} = -0.805 \approx 0.81$

$(\because a=8, b=2)$

$P_{12} = [0.32 \quad -0.95] \begin{bmatrix} 8 & -8 \\ 2 & -6.5 \end{bmatrix} = 4.27$

$(\because a=13, b=3)$

$P_{13} = [0.32 \quad -0.95] \begin{bmatrix} 13 & -8 \\ 3 & -6.5 \end{bmatrix} = 4.921 \approx 4.91$

$(\because a=7, b=15)$

$P_{14} = [0.32 \quad -0.95] \begin{bmatrix} 7 & -8 \\ 15 & -6.5 \end{bmatrix} = -8.37$

Conclusion :

In conclusion, Principal Component Analysis (PCA) is a powerful technique in machine learning for reducing the dimensionality of data while preserving its most important information. By finding the most significant features and representing the data in a lower-dimensional space, PCA facilitates easier visualization and interpretation of relationships between variables. In this experiment, we implemented PCA and observed several key components : the centered data, covariance matrix, eigen values, eigenvectors, and row values. Through these observations, we gained insights into how PCA works and its applications in simplifying data analysis, enhancing visualization, and identifying trends & relationships within datasets. Overall, PCA proves to be a valuable tool for data processing and analysis in various domains.

NAME:-PREKSHA ASHOK PATEL
BRANCH:-COMPUTER ENGINEERING
SAPID:-60004210126
DIV:-C2 ; BATCH:-1

MACHINE LEARNING

EXPERIMENT NO-7

CODE AND OUTPUT :-
USING LIBRARY:-
DATASET 1]

```
import numpy as np
from sklearn.decomposition import PCA

dataset = np.array([[4,6], [8,2], [13,3],[7,15]])

mean_vector = np.mean(dataset, axis=0)
centered_data = dataset - mean_vector

covariance_matrix = np.cov(centered_data.T)

eigen_values, eigen_vectors = np.linalg.eig(covariance_matrix)

sorted_indices = np.argsort(eigen_values)[::-1]
sorted_eigen_values = eigen_values[sorted_indices]
sorted_eigen_vectors = eigen_vectors[:, sorted_indices]

num_components = 2
selected_eigen_vectors = sorted_eigen_vectors[:, :num_components]

new_values = np.dot(centered_data, selected_eigen_vectors)

# Print results
print("Centered Data:")
print(centered_data)
print("\nCovariance Matrix:")
print(covariance_matrix)
print("\nEigen Values:")
print(sorted_eigen_values)
print("\nEigen Vectors:")
print(sorted_eigen_vectors)
print("\nNew Values:")
print(new_values)
```

```
Centered Data:
[[-4.  -0.5]
 [ 0.  -4.5]
 [ 5.  -3.5]
 [-1.   8.5]]

Covariance Matrix:
[[14. -8.]
 [-8. 35.]]

Eigen Values:
[37.70037878 11.29962122]

Eigen Vectors:
[[ 0.31981892 -0.94747869]
 [-0.94747869 -0.31981892]]

New Values:
[[-0.80553633  3.9498242 ]
 [ 4.26365409  1.43918513]
 [ 4.91526999 -3.61802722]
 [-8.37338775 -1.77098211]]
```

DATASET-2]

```
[6]  import numpy as np
     from sklearn.decomposition import PCA

     dataset = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

     mean_vector = np.mean(dataset, axis=0)
     centered_data = dataset - mean_vector

     covariance_matrix = np.cov(centered_data.T)

     eigen_values, eigen_vectors = np.linalg.eig(covariance_matrix)

     sorted_indices = np.argsort(eigen_values)[::-1]
     sorted_eigen_values = eigen_values[sorted_indices]
     sorted_eigen_vectors = eigen_vectors[:, sorted_indices]

     num_components = 2
     selected_eigen_vectors = sorted_eigen_vectors[:, :num_components]

     new_values = np.dot(centered_data, selected_eigen_vectors)

     print("Centered Data:")
     print(centered_data)
     print("\nCovariance Matrix:")
     print(covariance_matrix)
     print("\nEigen Values:")
     print(sorted_eigen_values)
     print("\nEigen Vectors:")
     print(sorted_eigen_vectors)
     print("\nNew Values:")
     print(new_values)
```

```
Centered Data:
[[-3. -3. -3.]
 [ 0.  0.  0.]
 [ 3.  3.  3.]]

Covariance Matrix:
[[9. 9. 9.]
 [9. 9. 9.]
 [9. 9. 9.]]

Eigen Values:
[27.  0.  0.]

Eigen Vectors:
[[ 0.57735027  0.          -0.81649658]
 [ 0.57735027 -0.70710678  0.40824829]
 [ 0.57735027  0.70710678  0.40824829]]

New Values:
[[-5.19615242e+00  4.44089210e-16]
 [ 0.00000000e+00  0.00000000e+00]
 [ 5.19615242e+00 -4.44089210e-16]]
```

WITHOUT USING LIBRARY :-
DATASET-1]

```python
import numpy as np

dataset = np.array([[4, 6], [8, 2], [13, 3], [7, 15]])

mean_vector = np.mean(dataset, axis=0)
centered_data = dataset - mean_vector

covariance_matrix = np.dot(centered_data.T, centered_data) / (len(dataset) - 1)

eigen_values, eigen_vectors = np.linalg.eig(covariance_matrix)

sorted_indices = np.argsort(eigen_values)[::-1]
sorted_eigen_values = eigen_values[sorted_indices]
sorted_eigen_vectors = eigen_vectors[:, sorted_indices]

num_components = 2
selected_eigen_vectors = sorted_eigen_vectors[:, :num_components]

new_values = np.dot(centered_data, selected_eigen_vectors)

print("Centered Data:")
print(centered_data)
print("\nCovariance Matrix:")
print(covariance_matrix)
print("\nEigen Values:")
print(sorted_eigen_values)
print("\nEigen Vectors:")
print(sorted_eigen_vectors)
print("\nNew Values:")
print(new_values)
```

```
Centered Data:
[[-4.   -0.5]
 [ 0.   -4.5]
 [ 5.   -3.5]
 [-1.    8.5]]

Covariance Matrix:
[[14. -8.]
 [-8. 35.]]

Eigen Values:
[37.70037878 11.29962122]

Eigen Vectors:
[[ 0.31981892 -0.94747869]
 [-0.94747869 -0.31981892]]

New Values:
[[-0.80553633  3.9498242 ]
 [ 4.26365409  1.43918513]
 [ 4.91526999 -3.61802722]
 [-8.37338775 -1.77098211]]
```

DATASET-2]

```
[9]  import numpy as np

     dataset = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

     mean_vector = np.mean(dataset, axis=0)
     centered_data = dataset - mean_vector

     covariance_matrix = np.dot(centered_data.T, centered_data) / (len(dataset) - 1)

     eigen_values, eigen_vectors = np.linalg.eig(covariance_matrix)

     sorted_indices = np.argsort(eigen_values)[::-1]
     sorted_eigen_values = eigen_values[sorted_indices]
     sorted_eigen_vectors = eigen_vectors[:, sorted_indices]

     num_components = 2
     selected_eigen_vectors = sorted_eigen_vectors[:, :num_components]

     new_values = np.dot(centered_data, selected_eigen_vectors)

     # Print results
     print("Centered Data:")
     print(centered_data)
     print("\nCovariance Matrix:")
     print(covariance_matrix)
     print("\nEigen Values:")
     print(sorted_eigen_values)
     print("\nEigen Vectors:")
     print(sorted_eigen_vectors)
     print("\nNew Values:")
     print(new_values)
```

```
Centered Data:
[[-3. -3. -3.]
 [ 0.  0.  0.]
 [ 3.  3.  3.]]

Covariance Matrix:
[[9. 9. 9.]
 [9. 9. 9.]
 [9. 9. 9.]]

Eigen Values:
[27.  0.  0.]

Eigen Vectors:
[[ 0.57735027  0.          -0.81649658]
 [ 0.57735027 -0.70710678  0.40824829]
 [ 0.57735027  0.70710678  0.40824829]]

New Values:
[[-5.19615242e+00  4.44089210e-16]
 [ 0.00000000e+00  0.00000000e+00]
 [ 5.19615242e+00 -4.44089210e-16]]
```