PCA in Machine Learning

In the dynamic landscape of machine learning, the abundance of data presents both opportunities and challenges. Dealing with high-dimensional data can be both a blessing and a curse. While data richness enhances model performance, it often comes at the cost of increased dimensionality and computational complexity. Principal Component Analysis (PCA) emerges as a pivotal technique to navigate this complexity by extracting essential features and reducing the dimensionality of the data. In this article, we will delve into the application of PCA in machine learning and demonstrate its efficacy with a practical example.

What is PCA?

Principal Component Analysis is a technique that helps to find out the most common dimensions of the dataset and makes result analysis simpler. In the available dataset not all these datasets dimension is critical, some may be the primary key datasets, whereas others are not. So, PCA Method of factor analysis gives a calculative way of eliminating a few extra less important variables, thereby maintaining the transparency of all information.

PCA is also called dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while retaining most of its variance. With reduced data and dimensions, it is easily to explore and visualize the algorithms without wasting your valuable time.

Therefore, PCA statistically analyze all dimensions and reduce them as much as possible while preserving the exact information.

Principal Components in PCA:

1. The principal component must be the linear combination of the original features.
2. These components are orthogonal, i.e., the correlation between a pair of variables is zero.
3. The importance of each component decreases when going to 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

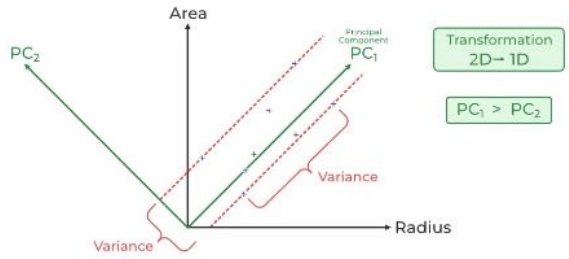

figure:**PCA**

Applications of PCA:

The application of PCA in machine learning spans across various domains, from image recognition to financial forecasting. One of the primary applications of PCA in machine learning is preprocessing data before feeding it into a model. By reducing the dimensionality of the feature space, PCA alleviates the curse of dimensionality, thereby improving model performance and generalization. Additionally, PCA facilitates visualization and interpretation of high-dimensional data, aiding in exploratory data analysis and model understanding. It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc. In Machine Leaning, PCA is used to visualize multidimensional data. In healthcare data. It can explore the factors assumed to be very important in increasing the risk of any chronic disease. It helps to resize an image.It is used to analyze stock data and forecasting data. It can analyze the patterns in high-dimensional data sets.

Terminologies of PCA:

1. **Variance-** Variance calculates the variation of data distributed across dimensionality of graph.
2. **Covariance-** It calculates dependencies and relationship between features or datasets.
3. **Standardizing data-** It is a process of scaling our dataset within a specific range for unbiased output.
4. **Covariance matrix-** This matrix is used to calculate interdependencies between the features or datasets and helps in reduce the datasets to improve the performance.
5. **Eigenvectors-** Eigenvectors' purpose is to find out the largest variance that exists in the dataset to calculate Principal Component. eigenvector is expanding or contracting X-Y (2D) graph without altering the direction.
6. **Eigenvalues-** Eigenvalue means the magnitude of the Eigenvector. Eigenvalue indicates variance in a particular direction.
7. **Dimensionality Reduction-** Under this we Transpose the original data and multiply it by transposing of the derived feature vector. It Reduces the features/datasets without losing information.

Step-By-Step Explanation of PCA:

Step 1: Standardization

we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance.
If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.

Step 2: Calculating the Covariance of Z

To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.

Step 3: Calculating the Eigen Values and Eigen Vectors

Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.

Step 4: Sorting the Eigen Vectors

In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.

Step 5: Calculating Principal Components

we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.

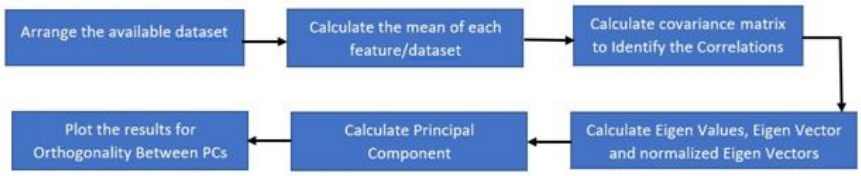

figure:Steps of PCA

Real-Life example of PCA in Python using scikit-learn library:

To harness the power of PCA, we first standardize the features to ensure they have zero mean and unit variance. Next, we apply PCA to the standardized dataset to extract the principal components. By retaining a sufficient amount of variance – say, 95% – we reduce the dimensionality of the dataset while preserving its essential structure.

Import Libraries-

Import necessary libraries including NumPy, Matplotlib, PCA from scikit-learn, and the dataset.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
```

figure:Important libraries imported

Load and Standardize Data-

Load the dataset and standardize the features by subtracting the mean and dividing by the standard deviation.

Create PCA Instance-

Create an instance of the PCA class and fit it to the standardized data.

Plot Explained Variance Ratio-

Plot the cumulative explained variance to help decide the number of principal components to retain.

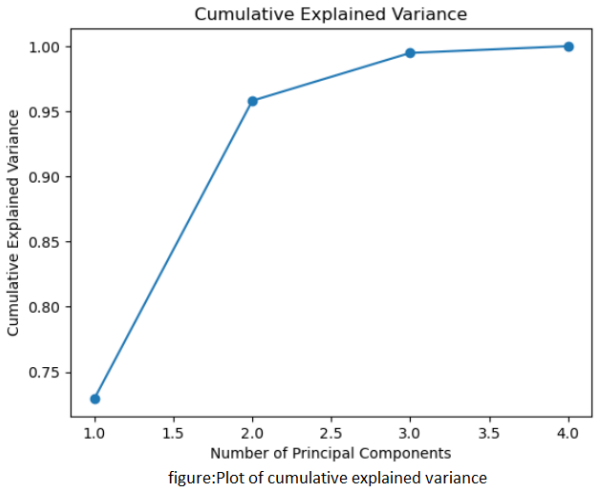

figure:Plot of cumulative explained variance

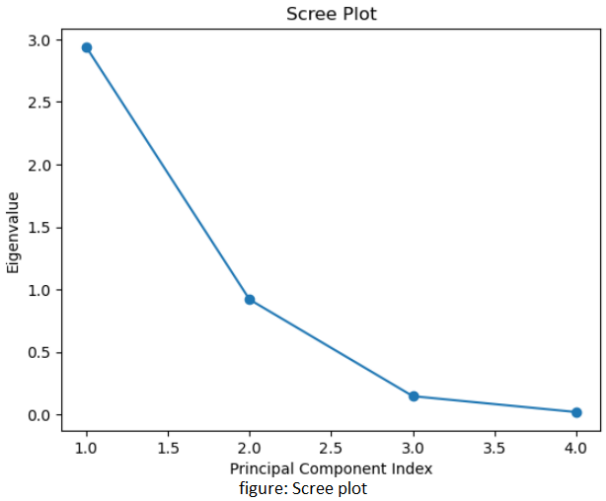

figure: Scree plot

Choose Number of Components-

Based on the plot or a desired threshold, choose the number of principal components.Apply PCA with the selected number of components.

Visualize Transformed Data-

Scatter plot the transformed data using the selected principal components. In this example, we use two components for visualization.

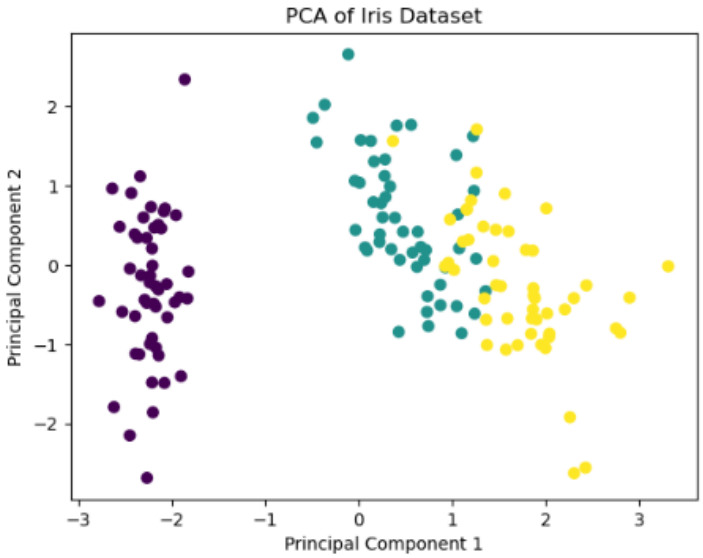

figure: Scatter plot for PCA

Advantages of PCA:

- Dimensionality Reduction: PCA simplifies data by reducing the number of variables, making analysis easier.
- Feature Selection: It helps in selecting the most important variables in large datasets, aiding machine learning tasks.
- Data Visualization: PCA helps visualize high-dimensional data in 2 or 3 dimensions, making it easier to understand.
- Multicollinearity: PCA addresses multicollinearity by creating new, uncorrelated variables for regression analysis.
- Noise Reduction: It filters out noise by removing principal components with low variance, improving signal-to-noise ratio.
- Data Compression: PCA compresses data by representing it with fewer principal components, reducing storage and processing needs.
- Outlier Detection: It identifies outliers as data points far from others in the principal component space, aiding in anomaly detection.

Disdvantages of PCA:

- Interpretation Difficulty: PCA creates combinations of original variables that can be hard to interpret directly, making it challenging to explain results.
- Data Scaling Importance: PCA's effectiveness relies on properly scaled data; if data isn't scaled correctly, PCA may not yield accurate results.
- Information Loss: PCA reduces variables, but this can lead to information loss. How much information is lost depends on how many principal components are retained.
- Non-linear Relationships: PCA assumes linear relationships between variables; if relationships are non-linear, PCA might not be suitable.
- Computational Complexity: PCA can be slow for large datasets, especially with many variables, due to its computational demands.
- Risk of Overfitting: Overfitting can occur if too many principal components are used or if the model is trained on a small dataset, leading to poor performance on new data.

Principal Component Analysis (PCA) offers a powerful framework for dimensionality reduction and visualization, as exemplified by its application to the Iris dataset. By distilling complex datasets into their essential components, PCA enables us to uncover hidden patterns, facilitate visualization, and enhance machine learning algorithms' performance. As we continue to explore the frontiers of data analysis and machine learning, PCA remains a valuable tool for unlocking insights and driving innovation.