

WEEK_9_DAY_3_MORNING

Dimensionality Reduction

- Importance of Dimension Reduction in machine learning

Common methods to perform Dimension Reduction - T

Dimensionality Reduction using PCA in python

Dimensionality Reduction

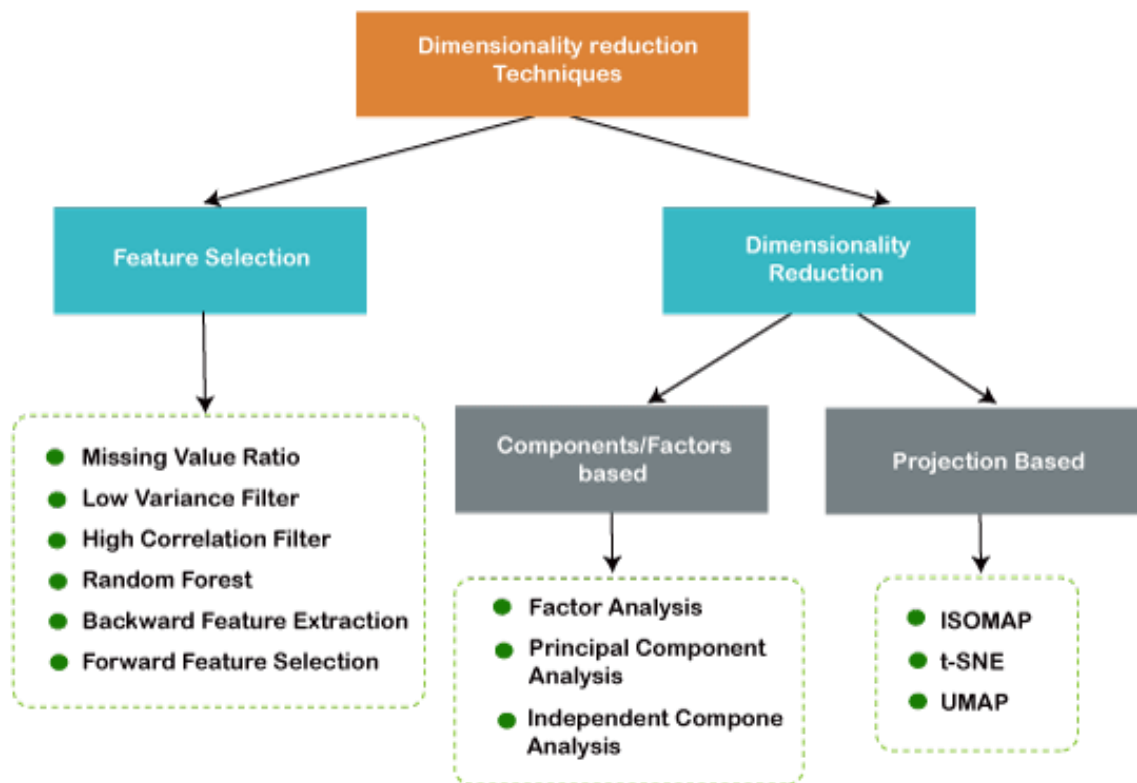
What is Dimensionality Reduction?

The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

A dataset contains a huge number of input features in various cases, which makes the predictive modeling task more complicated. Because it is very difficult to visualize or make predictions for the training dataset with a high number of features, for such cases, dimensionality reduction techniques are required to use.

Dimensionality reduction technique can be defined as, *"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information."*

It is commonly used in the fields that deal with high-dimensional data, such as **speech recognition, signal processing, bioinformatics, etc.** It can also be used for **data visualization, noise reduction, cluster analysis, etc.**



Benefits of applying Dimensionality Reduction

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- It removes the redundant features (if present) by taking care of multicollinearity.

Disadvantages of dimensionality Reduction

- Some data may be lost due to dimensionality reduction.
- In the PCA dimensionality reduction technique, sometimes the principal components required to consider are unknown.

Approaches of Dimension Reduction

There are two ways to apply the dimension reduction technique, which are given below:

Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy.

Three methods are used for the feature selection:

1. Filters Methods

In this method, the dataset is filtered, and a subset that contains only the relevant features is taken. Some common techniques of filters method are:

- **Correlation**
- **Chi-Square Test**
- **ANOVA**
- **Information Gain, etc.**

2. Wrappers Methods

The wrapper method has the same goal as the filter method, but it takes a machine learning model for its evaluation. In this method, some features are fed to the ML model, and evaluate the performance. The performance decides whether to add those features or remove to increase the accuracy of the model. This method is more accurate than the filtering method but complex to work. Some common techniques of wrapper methods are:

- Forward Selection
- Backward Selection
- Bi-directional Elimination

3. Embedded Methods: Embedded methods check the different training iterations of the machine learning model and evaluate the importance of each feature. Some common techniques of Embedded methods are:

- **LASSO**
- **Elastic Net**
- **Ridge Regression, etc.**

Feature Extraction:

Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

Some common feature extraction techniques are:

- a. Principal Component Analysis
- b. Linear Discriminant Analysis
- c. Kernel PCA
- d. Quadratic Discriminant Analysis

Common techniques of Dimensionality Reduction

- a. **Principal Component Analysis**
- b. **Backward Elimination**
- c. **Forward Selection**
- d. **Score comparison**
- e. **Missing Value Ratio**
- f. **Low Variance Filter**
- g. **High Correlation Filter**
- h. **Random Forest**
- i. **Factor Analysis**
- j. **Auto-Encoder**

Dimensionality Reduction using PCA in python

Principal Component Analysis (PCA)

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in [machine learning](#). It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are *image processing, movie recommendation system, optimizing the power allocation in various communication channels*. It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The PCA algorithm is based on some mathematical concepts such as:

- **Variance and Covariance**
- **Eigenvalues and Eigen factors**

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the **number of columns present in the dataset**.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M , and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v .
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Principal Components in PCA

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

- The principal component must be the **linear combination of the original features**.
- These components are orthogonal, i.e., the **correlation between a pair of variables is zero**.
- 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.

Steps for PCA algorithm

1. Getting the dataset

Firstly, we need to take the input dataset and divide it into two subparts X and Y , where X is the training set, and Y is the validation set.

2. Representing data into a structure

Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X . Here each row

corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.

3. **Standardizing the data**

In this step, 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 .

4. **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 .

5. **Calculating the Eigen Values and Eigen Vectors**

Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z . 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.

6. **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^* .

7. **Calculating the new features Or Principal Components**

Here we will calculate the new features. To do this, 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.

8. **Remove less or unimportant features from the new dataset.**

The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.