Practical 1

Problem Statement:

To use PCA Algorithm for dimensionality reduction on wine dataset and visualize data after applying PCA.

Dataset: wine Dataset.

dataset Link: https://media.geeksforgeeks.org/wp-content/uploads/Wine.csv

Objective:

1. Understand working of PCA algorithm
2. Visualize Data after applying PCA algorithm.

Outcome:

PCA Algorithm for dimensionality reduction should be understood.

Theory:

PCA is an unsupervised linear dimensionality reduction algorithm to find a more meaningful basis or coordinate system for our data and works based on covariance matrix to find the strongest features of your samples.

PCA is used for -

1. Dimensionality Reduction
2. Increasing Performance
3. Visualizing Higher Dimensional Data
4. Obscure Data
5. Create Independent Features

Requirements:

1) Get the Dataset

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset.

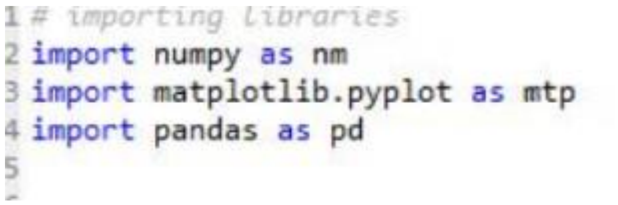
Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV file. However, sometimes, we may also need to use an HTML or xlsx file.

.CSV File

CSV stands for "Comma-Separated Values" files; it is a file format which allows us to save the tabular data, such as spreadsheets. It is useful for huge datasets and can use these datasets in programs.
For real-world problems, we can download datasets online from various sources such ashttps://www.kaggle.com. We can also create our dataset by gathering data using various API with Python and put that data into a .csv file.

2) Importing Libraries

In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:
Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:
import numpy as nm
Here we have used nm, which is a short name for Numpy, and it will be used in the whole program.
Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. It will be imported as below:
import matplotlib.pyplot as mpt
Here we have used mpt as a short name for this library.
Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. It will be imported as below:
Here, we have used pd as a short name for this library. Consider the below image



```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

3) Importing the Datasets

Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory and import wine dataset.

```
df = pd.read_csv('winequality.csv')
```

4) Scaling of data

Feature scaling is a data preprocessing technique used to transform the values of features or variables in a dataset to a similar scale. The purpose is to ensure that all features contribute equally to the model and to avoid the domination of features with larger values.

```
scaler = MinMaxScaler()
```

5) Applying Principal Component Analysis

PCA is an unsupervised linear dimensionality reduction algorithm to find a more meaningful basis or coordinate system for our data and works based on covariance matrix to find the strongest features of your samples. We call PCA function on dataset.

```
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=3)
```

6) Visualize dataset using matplotlib library

After a PCA, the observations are expressed in principal component scores. Therefore, it is important to visualize the spread of the data along the new axes (principal components) to interpret the relations in the dataset.

```
plt.figure(figsize=(8,6))
plt.scatter(x_pca[:,0],x_pca[:,1])#first two columns
plt.xlabel('first principle component')
plt.ylabel('second principle component')
```

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.

Conclusion:

Principal component analysis is a technique to summarize data, and is highly flexible depending on your use case. It can be valuable in both displaying and analysing a large number of possibly dependent variables. Techniques of performing principal component analysis range from arbitrarily selecting principal components, to automatically finding them until a variance is reached.

Oral Questions:

1. What all libraries are used for performing PCA?
2. How the data is imported in python?
3. Explain working of PCA?
4. Explain how PCA is visualized?