#Principal component analysis (PCA)

#Linear dimensionality reduction using Singular Value Decomposition of the data and keeping #only the most significant singular vectors to project the data to a lower #dimensional space.

#This implementation uses the scipy.linalg implementation of the singular value decomposition. It #only works for dense arrays and is not scalable to large dimensional data.

>>> import numpy as np

>>> from sklearn.decomposition import PCA

>>> X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])

>>> pca = PCA(n\_components=2)

>>> pca.fit(X)

PCA(copy=True, n\_components=2, whiten=False)

>>> print(pca.explained\_variance\_ratio\_)

[ 0.99244... 0.00755...]

#Implement PCA to GLASS Data set

**#explained\_variance\_ratio\_** : array, [n\_components]

#Percentage of variance explained by each of the selected components.

#If n\_componentsis not set then all components are stored and the sum of explained variances is equal to 1.0

glass=pd.read\_csv(‘glass.csv’)

>>> pca = PCA(n\_components=2)

>>> pca.fit(glass)

Test=pca.[transform](http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA.transform)(glass)

print(pca.explained\_variance\_ratio\_)

glass\_reduce= PCA(n\_components=3).fit\_transform(glass)