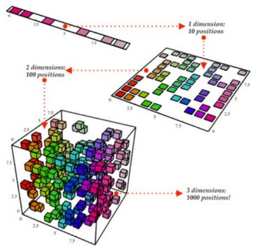
Principal Component Analysis

Tushar B. Kute,
http://tusharkute.com



Dimensionality Reduction

- Dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables.
- It can be divided into feature selection and feature extraction.
  - Feature selection approaches try to find a subset of the original variables (also called features or attributes).
  - Feature projection or Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions.

Large Dimensions

- Large number of features in the dataset is one of the factors that affect both the training time as well as accuracy of machine learning models. You have different options to deal with huge number of features in a dataset.
  - Try to train the models on original number of features, which take days or weeks if the number of features is too high.
  - Reduce the number of variables by merging correlated variables.
  - Extract the most important features from the dataset that are responsible for maximum variance in the output. Different statistical techniques are used for this purpose e.g. linear discriminant analysis, factor analysis, and principal component analysis.

Principal Component Analysis

- Principal component analysis, or PCA, is a statistical technique to convert high dimensional data to low dimensional data by selecting the most important features that capture maximum information about the dataset.
- The features are selected on the basis of variance that they cause in the output.
- The feature that causes highest variance is the first principal component. The feature that is responsible for second highest variance is considered the second principal component, and so on.
- It is important to mention that principal components do not have any correlation with each other.

Steps in PCA

- Standardization
- Covariance Matrix Computation
- Computer Eigen vector and eigen values
- Feature vector
- Recast the Data Along the Principal Components Axes

Standardization

- The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
- More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables.
- That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges

Standardization

- Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

- Once the standardization is done, all the variables will be transformed to the same scale.

Covariance Matrix Computation

- The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them.
- Because sometimes, variables are highly correlated in such a way that they contain redundant information.
- So, in order to identify these correlations, we compute the covariance matrix.

Covariance Matrix Computation

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Covariance Matrix Computation

- What do the covariances that we have as entries of the matrix tell us about the correlations between the variables?
- It’s actually the sign of the covariance that matters :
  - if positive then : the two variables increase or decrease together (correlated)
  - if negative then : One increases when the other decreases (Inversely correlated)
- Now, that we know that the covariance matrix is not more than a table that summaries the correlations between all the possible pairs of variables

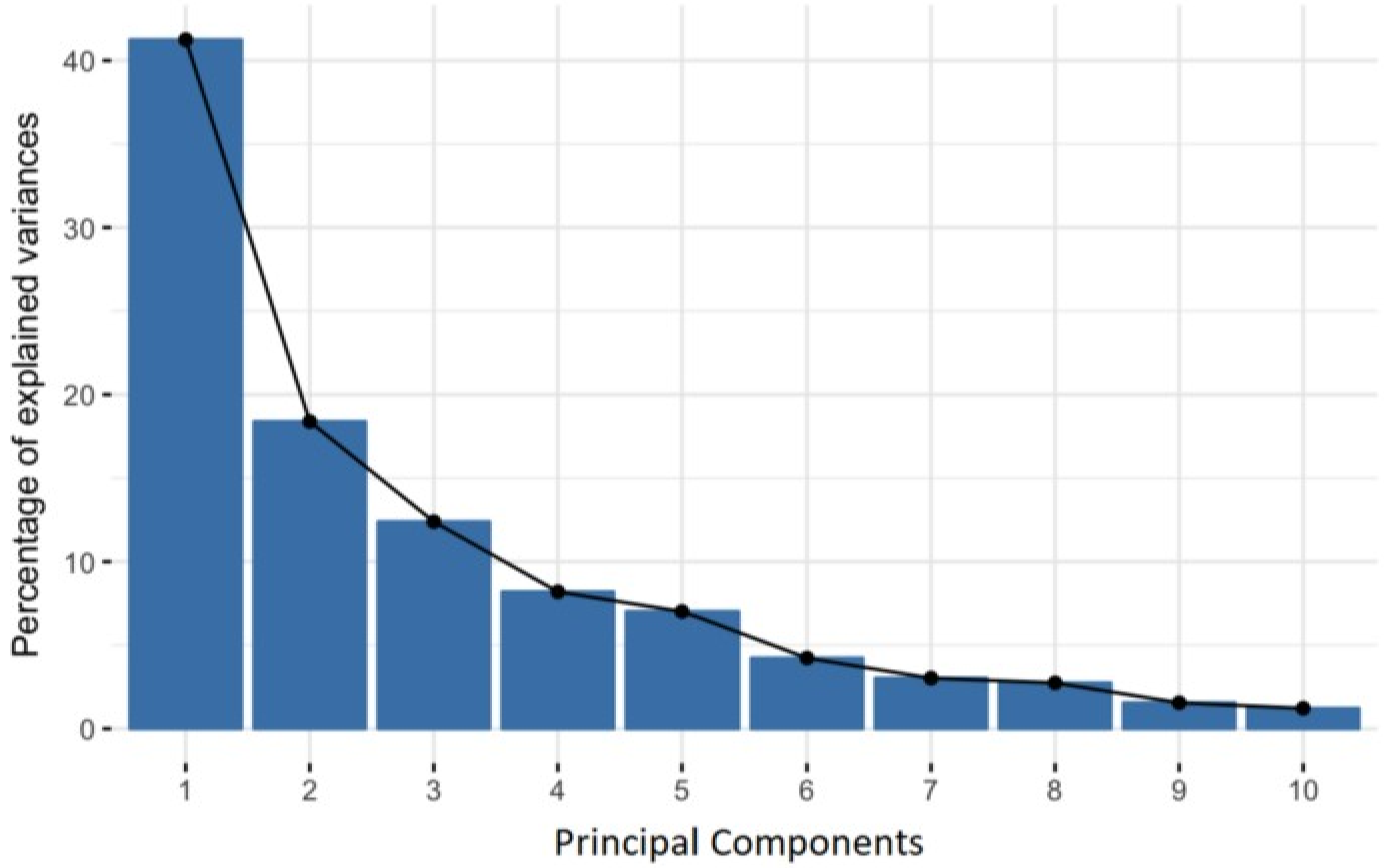
Eigenvector and eigenvalues

- Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.
- Before getting to the explanation of these concepts, let’s first understand what do we mean by principal components.

Eigenvector and eigenvalues

- Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables.
- These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components.
- So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.

Eigenvector and eigenvalues



Principal Components

- Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.
- An important thing to realize here is that, the principal components are less interpretable and don’t have any real meaning since they are constructed as linear combinations of the initial variables.
- Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data.

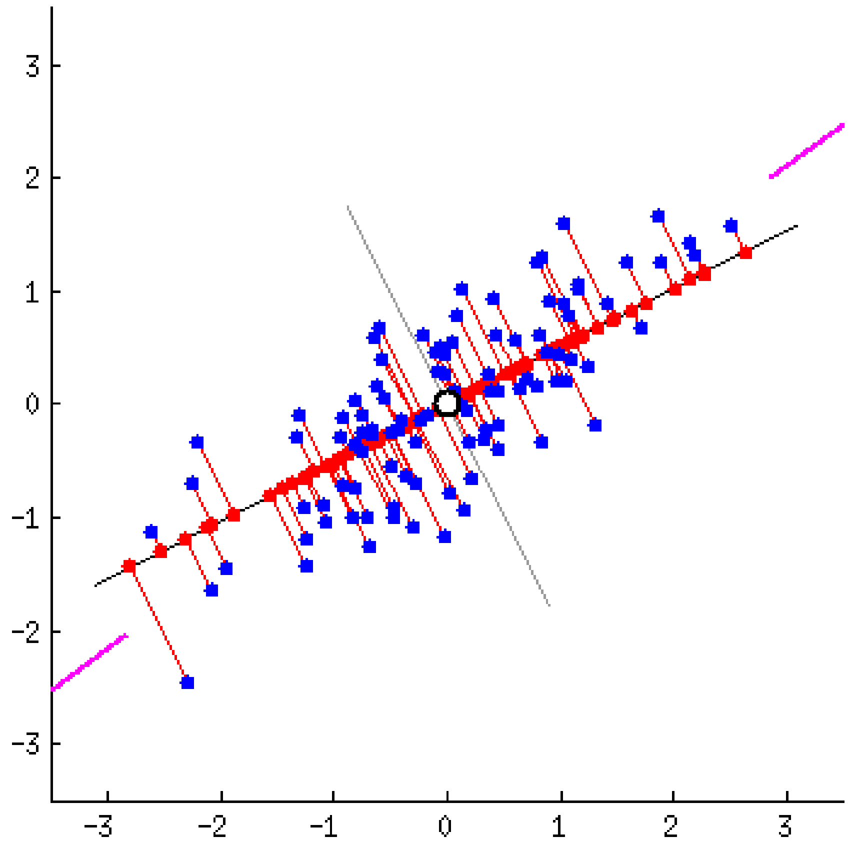
Principal Components

- As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set.

Principal Components

- For example, let’s assume that the scatter plot of our data set is as shown below, can we guess the first principal component ?
- Yes, it’s approximately the line that matches the purple marks because it goes through the origin and it’s the line in which the projection of the points (red dots) is the most spread out.
- Or mathematically speaking, it’s the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).

Principal Components



Example

- Let’s suppose that our data set is 2-dimensional with 2 variables x,y and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

$$v1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix} \qquad \lambda_1 = 1.284028$$

$$v2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix} \qquad \lambda_2 = 0.04908323$$

- If we rank the eigenvalues in descending order, we get λ1>λ2, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) isv2.

Feature Vector

- As we saw in the previous step, computing the eigenvectors and ordering them by their eigenvalues in descending order, allow us to find the principal components in order of significance.
- In this step, what we do is, to choose whether to keep all these components or discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call Feature vector.
- So, the feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep.
- This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.

Example:

- Continuing with the example from the previous step, we can either form a feature vector with both of the eigenvectors v1 and v2:

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

- Or discard the eigenvector v2, which is the one of lesser significance, and form a feature vector with v1 only:

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

- Discarding the eigenvector v2 will reduce dimensionality by 1, and will consequently cause a loss of information in the final data set.

Last step

- The aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis).
- This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

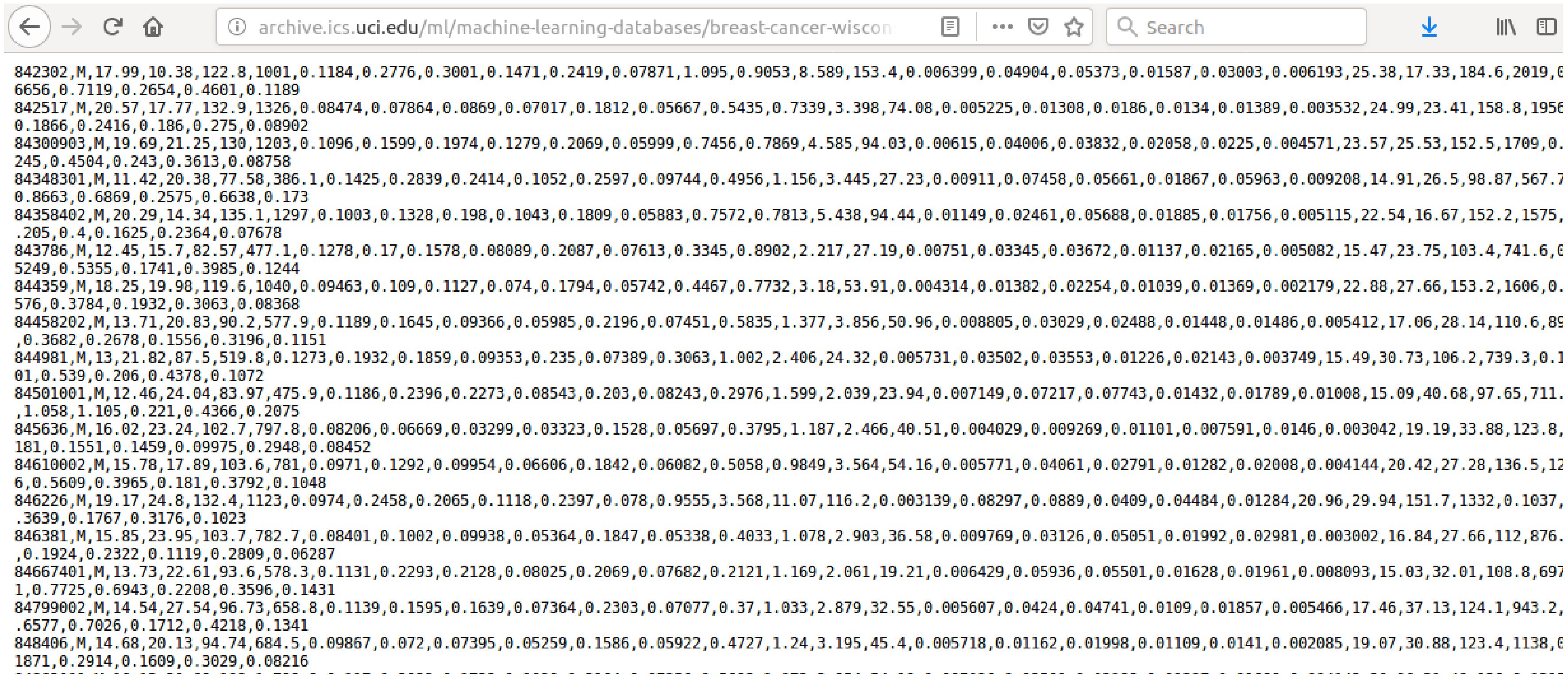
Advantages of PCA

- The training time of the algorithms reduces significantly with less number of features.
- It is not always possible to analyze data in high dimensions. For instance if there are 100 features in a dataset. Total number of scatter plots required to visualize the data would be 100(100-1)2 = 4950. Practically it is not possible to analyze data this way.

Normalization of features

- It is imperative to mention that a feature set must be normalized before applying PCA. For instance if a feature set has data expressed in units of Kilograms, Light years, or Millions, the variance scale is huge in the training set. If PCA is applied on such a feature set, the resultant loadings for features with high variance will also be large. Hence, principal components will be biased towards features with high variance, leading to false results.
- Finally, the last point to remember before we start coding is that PCA is a statistical technique and can only be applied to numeric data. Therefore, categorical features are required to be converted into numerical features before PCA can be applied.

Example:


archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wiscon
Search

```
842302,M,17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.05373,0.01587,0.03003,0.006193,25.38,17.33,184.6,2019,
6656,0.7119,0.2654,0.4601,0.1189
842517,M,20.57,17.77,132.9,1326,0.08474,0.07864,0.0869,0.07017,0.1812,0.05667,0.5435,0.7339,3.398,74.08,0.005225,0.01308,0.0186,0.0134,0.01389,0.003532,24.99,23.41,158.8,195
0.1866,0.2416,0.186,0.275,0.08902
84300903,M,19.69,21.25,130,1203,0.1096,0.1599,0.1974,0.1279,0.2069,0.05999,0.7456,0.7869,4.585,94.03,0.00615,0.04006,0.03832,0.02058,0.0225,0.004571,23.57,25.53,152.5,1709,0.
245,0.4504,0.243,0.3613,0.08758
84348301,M,11.42,20.38,77.58,386.1,0.1425,0.2839,0.2414,0.1052,0.2597,0.09744,0.4956,1.156,3.445,27.23,0.00911,0.07458,0.05661,0.01867,0.05963,0.009208,14.91,26.5,98.87,567.
0.8663,0.6869,0.2575,0.6638,0.173
84358402,M,20.29,14.34,135.1,1297,0.1003,0.1328,0.198,0.1043,0.1809,0.05883,0.7572,0.7813,5.438,94.44,0.01149,0.02461,0.05688,0.01885,0.01756,0.005115,22.54,16.67,152.2,1575,
.205,0.4,0.1625,0.2364,0.07678
843786,M,12.45,15.7,82.57,477.1,0.1278,0.17,0.1578,0.08089,0.2087,0.07613,0.3345,0.8902,2.217,27.19,0.00751,0.03345,0.03672,0.01137,0.02165,0.005082,15.47,23.75,103.4,741.6,
5249,0.5355,0.1741,0.3985,0.1244
844359,M,18.25,19.98,119.6,1040,0.09463,0.109,0.1127,0.074,0.1794,0.05742,0.4467,0.7732,3.18,53.91,0.004314,0.01382,0.02254,0.01039,0.01369,0.002179,22.88,27.66,153.2,1606,0.
576,0.3784,0.1932,0.3063,0.08368
84458202,M,13.71,20.83,90.2,577.9,0.1189,0.1645,0.09366,0.05985,0.2196,0.07451,0.5835,1.377,3.856,50.96,0.008805,0.03029,0.02488,0.01448,0.01486,0.005412,17.06,28.14,110.6,8
,0.3682,0.2678,0.1556,0.3196,0.1151
844981,M,13,21.82,87.5,519.8,0.1273,0.1932,0.1859,0.09353,0.235,0.07389,0.3063,1.002,2.406,24.32,0.005731,0.03502,0.03553,0.01226,0.02143,0.003749,15.49,30.73,106.2,739.3,0.
01,0.539,0.206,0.4378,0.1072
84501001,M,12.46,24.04,83.97,475.9,0.1186,0.2396,0.2273,0.08543,0.203,0.08243,0.2976,1.599,2.039,23.94,0.007149,0.07217,0.07743,0.01432,0.01789,0.01008,15.09,40.68,97.65,711.
,1.058,1.105,0.221,0.4366,0.2075
845636,M,16.02,23.24,102.7,797.8,0.08206,0.06669,0.03299,0.03323,0.1528,0.05697,0.3795,1.187,2.466,40.51,0.004029,0.009269,0.01101,0.007591,0.0146,0.003042,19.19,33.88,123.8,
181,0.1551,0.1459,0.09975,0.2948,0.08452
84610002,M,15.78,17.89,103.6,781,0.0971,0.1292,0.09954,0.06606,0.1842,0.06082,0.5058,0.9849,3.564,54.16,0.005771,0.04061,0.02791,0.01282,0.02008,0.004144,20.42,27.28,136.5,12
6,0.5609,0.3965,0.181,0.3792,0.1048
846226,M,19.17,24.8,132.4,1123,0.0974,0.2458,0.2065,0.1118,0.2397,0.078,0.9555,3.568,11.07,116.2,0.003139,0.08297,0.0889,0.0409,0.04484,0.01284,20.96,29.94,151.7,1332,0.1037,
.3639,0.1767,0.3176,0.1023
846381,M,15.85,23.95,103.7,782.7,0.08401,0.1002,0.09938,0.05364,0.1847,0.05338,0.4033,1.078,2.903,36.58,0.009769,0.03126,0.05051,0.01992,0.02981,0.003002,16.84,27.66,112,876.
,0.1924,0.2322,0.1119,0.2809,0.06287
84667401,M,13.73,22.61,93.6,578.3,0.1131,0.2293,0.2128,0.08025,0.2069,0.07682,0.2121,1.169,2.061,19.21,0.006429,0.05936,0.05501,0.01628,0.01961,0.008093,15.03,32.01,108.8,697
1,0.7725,0.6943,0.2208,0.3596,0.1431
84799002,M,14.54,27.54,96.73,658.8,0.1139,0.1595,0.1639,0.07364,0.2303,0.07077,0.37,1.033,2.879,32.55,0.005607,0.0424,0.04741,0.0109,0.01857,0.005466,17.46,37.13,124.1,943.2,
.6577,0.7026,0.1712,0.4218,0.1341
848406,M,14.68,20.13,94.74,684.5,0.09867,0.072,0.07395,0.05259,0.1586,0.05922,0.4727,1.24,3.195,45.4,0.005718,0.01162,0.01998,0.01109,0.0141,0.002085,19.07,30.88,123.4,1138,
1871,0.2914,0.1609,0.3029,0.08216
```

Reading the dataset

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

dataset = pd.read_csv('wisc.csv')
print dataset.head()

# Input features
X = dataset.drop(['id','diagnosis'], 1)

# Output feature
y = dataset['diagnosis']
```

Normalize

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=0)

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

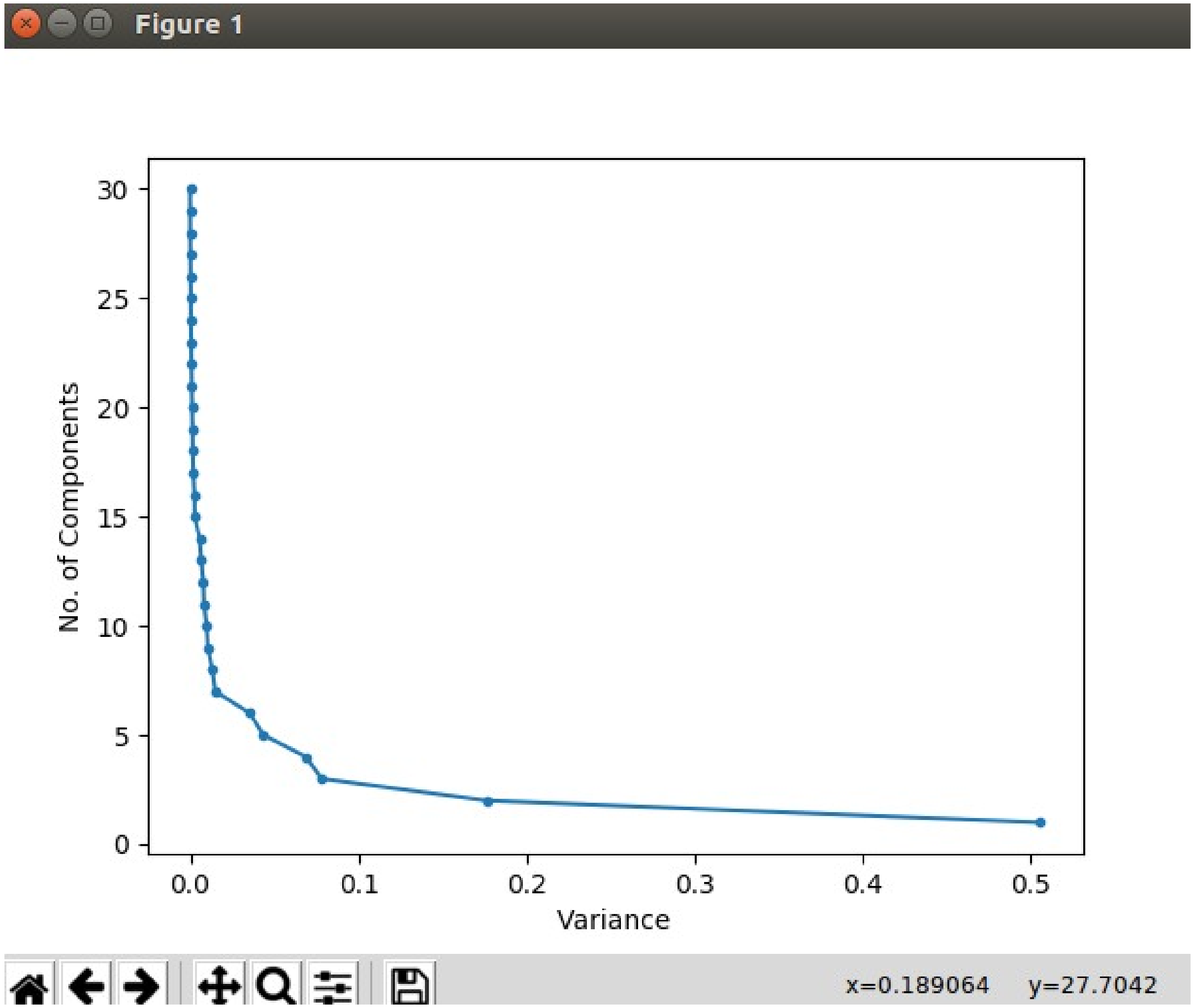
Apply PCA

```
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier

# Create any calssifier
classifier = RandomForestClassifier(n_estimators=10,
                                    random_state=0)

# Object of PCA
pca = PCA()

X_train0 = pca.fit_transform(X_train)
X_test0 = pca.transform(X_test)
classifier.fit(X_train0, y_train)
y_pred = classifier.predict(X_test0)
```

Calculate variance

```
from sklearn.metrics import confusion_matrix,
                            accuracy_score
print 'Confusion Matrix (default):\n',
                confusion_matrix(y_test, y_pred)
print 'Accuracy (default):',
                accuracy_score(y_test, y_pred) * 100
variance = pca.explained_variance_ratio_
print variance

# Plot variance vs. no. of components
plt.plot(variance, range(1,31), '.-')
plt.xlabel('Variance')
plt.ylabel('No. of Components')
plt.show()
```

Variance plot



Variance Ratio

- The PCA class contains explained_variance_ratio_ which returns the variance caused by each of the principal components.

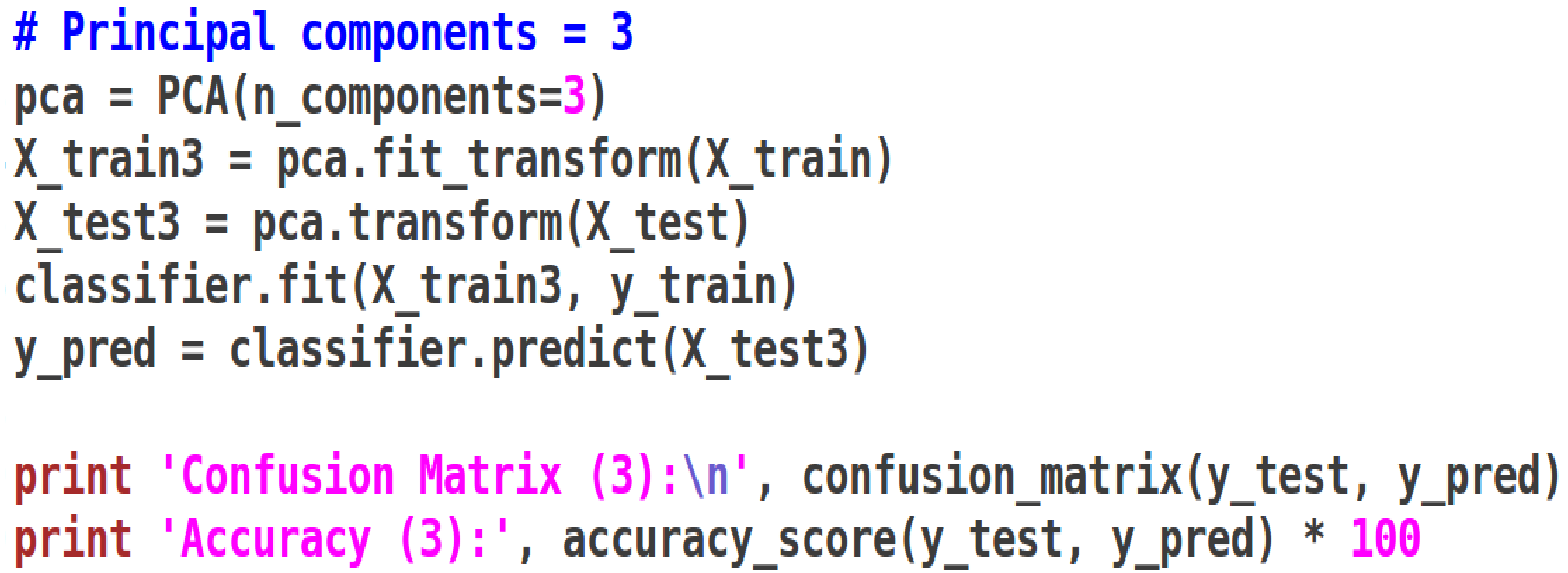
Principal Components = 1

```
# Principal components = 1
pca = PCA(n_components=1)
X_train1 = pca.fit_transform(X_train)
X_test1 = pca.transform(X_test)

classifier.fit(X_train1, y_train)
y_pred = classifier.predict(X_test1)

print 'Confusion Matrix (1):\n', confusion_matrix(y_test, y_pred)
print 'Accuracy (1):', accuracy_score(y_test, y_pred) * 100
```

Principal Components = 2

```
# Principal components = 2
pca = PCA(n_components=2)
X_train2 = pca.fit_transform(X_train)
X_test2 = pca.transform(X_test)

classifier.fit(X_train2, y_train)
y_pred = classifier.predict(X_test2)

print 'Confusion Matrix (2):\n', confusion_matrix(y_test, y_pred)
print 'Accuracy (2):', accuracy_score(y_test, y_pred) * 100
```

Principal Components = 3



```
# Principal components = 3
pca = PCA(n_components=3)
X_train3 = pca.fit_transform(X_train)
X_test3 = pca.transform(X_test)
classifier.fit(X_train3, y_train)
y_pred = classifier.predict(X_test3)

print 'Confusion Matrix (3):\n', confusion_matrix(y_test, y_pred)
print 'Accuracy (3):', accuracy_score(y_test, y_pred) * 100
```

Linear Discriminant Analysis

- Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition, and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events.
- The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

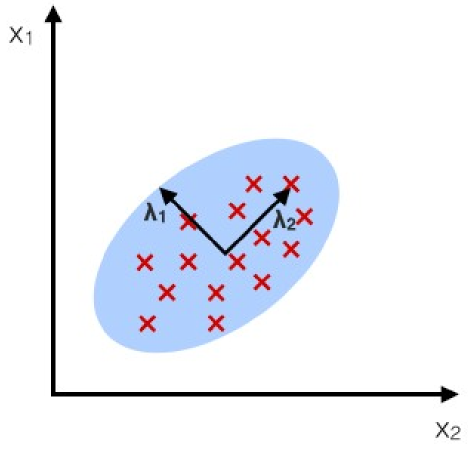
How LDA is performed ?

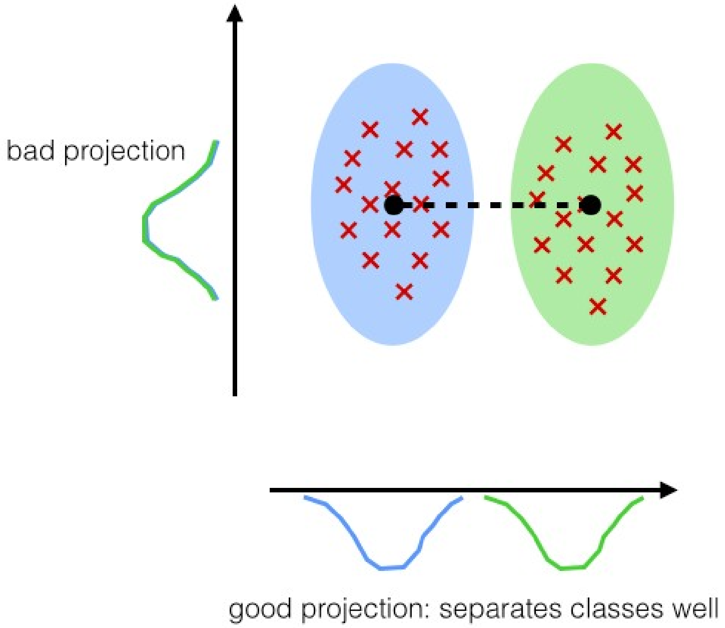
- Compute d-dimensional mean vectors for different classes from the dataset, where d is the dimension of feature space.
- Compute in-between class and with-in class scatter matrices.
- Compute eigen vectors and corresponding eigen values for the scatter matrices.
- Choose k eigen vectors corresponding to top k eigen values to form a transformation matrix of dimension d x k.
- Transform the d-dimensional feature space X to k-dimensional feature space X_lda via the transformation matrix.

PCA vs. LDA

PCA:
component axes that maximize the variance

LDA:
maximizing the component axes for class-separation





Reference: sebastianraschka.com

Useful resources

- https://stackabuse.com
- http://archive.ics.uci.edu/ml/index.php
- https://scikit-learn.org
- https://en.wikipedia.org
- www.towardsdatascience.com
- www.analyticsvidhya.com
- www.kaggle.com
- www.github.com

Thank you

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

c/MITUSkillologies

Web Resources
http://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com