Dimension Reduction Techniques

Agenda

- Dimension Reduction
- Principal Component Analysis (PCA)
    - Procedure
    - Terminologies
    - Selecting Principal Components
- Case Study

Agenda

- Linear Discriminant Analysis (LDA)
  - Procedure
  - Terminologies

Dimension Reduction

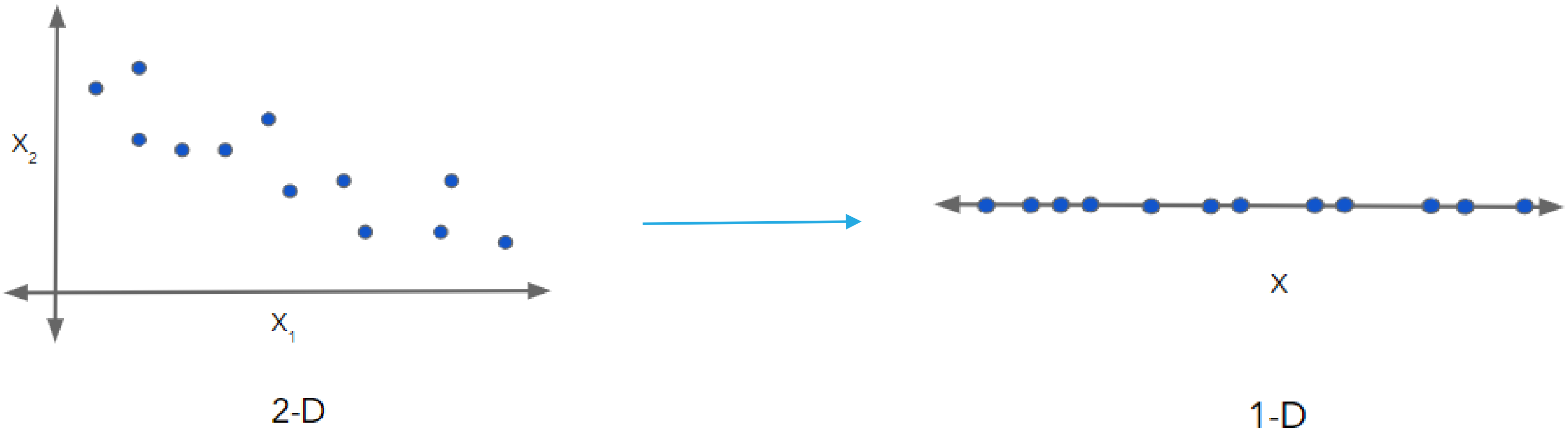
Dimension reduction

- The real-world dataset may contain a large number of features under study
- A dataset with a large number of features needs more time for model training
- As the number of variables increase, the data becomes more sparse. Overfitting can occur when a model is built on such data
- To avoid the sparsity of the data, we require more observations (i.e. rows) which may not be easily available in most cases
- The distance between the different points starts converging to a single value with an increase in dimensions. This is known as ‘Distance Concentration’

Dimension reduction

- To avoid such issues, one can reduce the dimension of the dataset
- The dimension reduction techniques remove the redundant variables/ noise in the original data, which reduces the training time
- Reducing the dataset to 2 or 3 dimensions helps in visualization of the data
- Various dimension reduction techniques:
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - Factor Analysis

Dimension reduction



Approaches for dimension reduction

- Two different approaches can be used for dimension reduction: Projection, Manifold learning
- In the projection approach, the original dataset is projected onto the lower-dimensional plane
- PCA uses the projection approach for dimension reduction
- This method is not effective if the dataset has different layers in the higher dimensions
- In manifold learning, a manifold is created on which the dataset lies

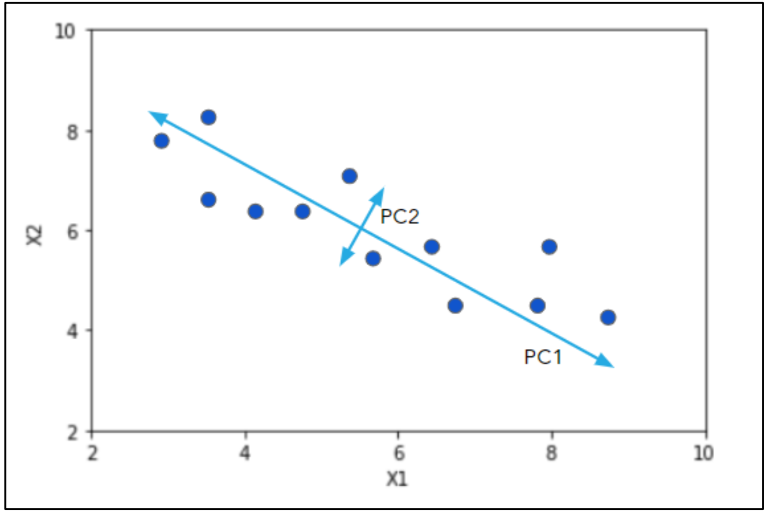
Principal Component Analysis (PCA)

PCA

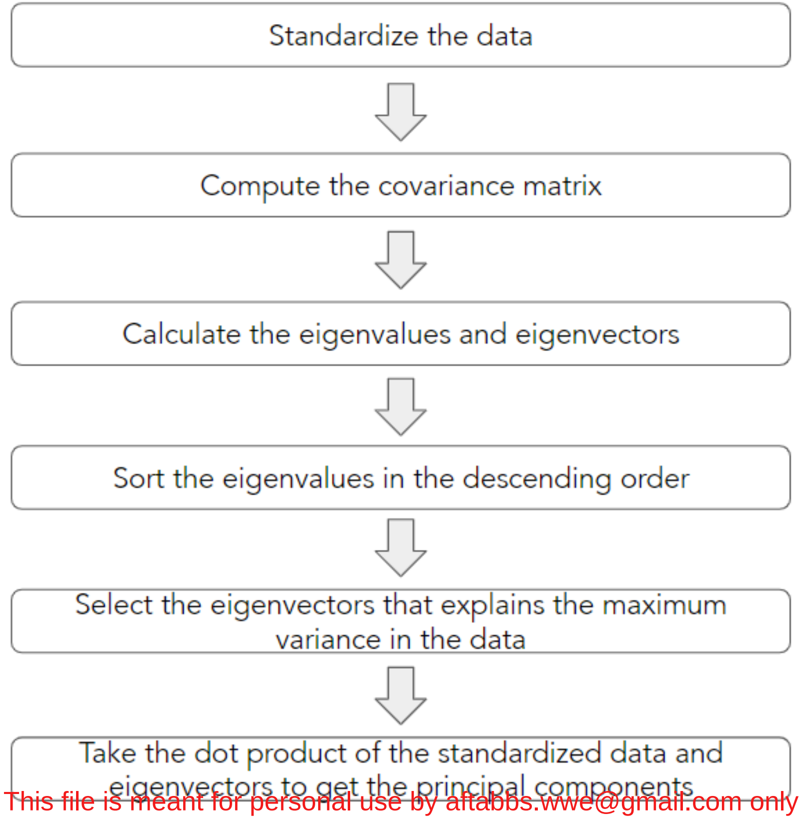
- It is one of the dimensionality reduction techniques that is used to reduce the dimensions of the large datasets
- It transforms the large set of features into a small set such that it will contain the maximum information in the original data
- The number of components is less than or equal to the number of independent variables
- PCA projects the original dataset on the lower dimensional plane
- It transforms the original data to a set of uncorrelated principal components

PCA

- The first principal component (PC1) exhibits the direction of maximum variance in the data
- It is used to remove the redundancy in the data
- PCA reduces the multicollinearity (if present) in the original data
- Principal components are always orthogonal to each other



PCA - procedure



Standardize the data

Compute the covariance matrix

Calculate the eigenvalues and eigenvectors

Sort the eigenvalues in the descending order

Select the eigenvectors that explains the maximum variance in the data

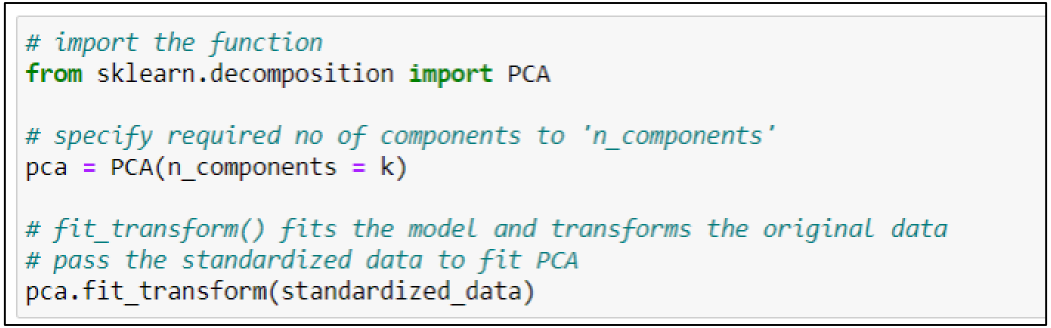
Take the dot product of the standardized data and eigenvectors to get the principal components

Application

- PCA is mainly used in image compression, facial recognition models
- It is also used in the exploratory analysis to reduce the dimension of data before applying machine learning methods
- Used in the field of psychology, finance to identify the patterns high dimensional data

Python code

In python, we use the following code to perform PCA:



```python
# import the function
from sklearn.decomposition import PCA

# specify required no of components to 'n_components'
pca = PCA(n_components = k)

# fit_transform() fits the model and transforms the original data
# pass the standardized data to fit PCA
pca.fit_transform(standardized_data)
```

Terminologies

Covariance

- The covariance measures how co-dependent two variables are
- Positive covariance value means that the two variables are directly proportional to each other
- Negative covariance value means that the two variables are inversely proportional to each other
- It is similar to variance, but the variance illustrates the variation of the single variable and covariance explains how two variables vary together

Covariance

The covariance between two variables X and Y is given by

$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{n-1}$$

Xi = values taken by variable X ∈ [1, n] to i ∈ [1, n]

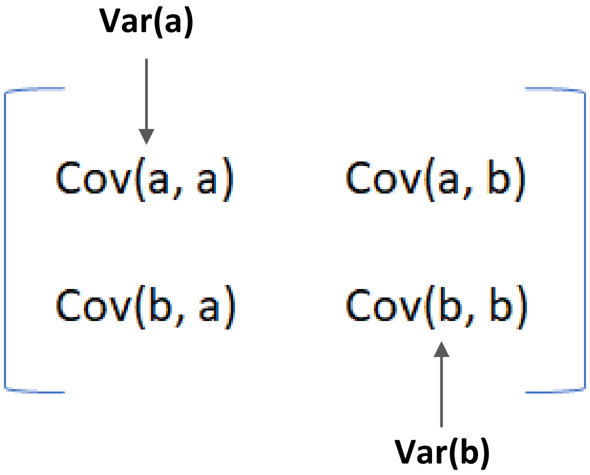
Yi = values taken by variable Y ∈ [1, n] to i ∈ [1, n]

$\overline{X}$ = mean of Xi

$\overline{Y}$ = mean of Yi

Covariance matrix

- The covariance matrix explains the covariance between the pair of variables
- The diagonal entries represent the variance of the variable, as it is the covariance of the variable with itself
- The diagonal matrix is always symmetric
- The off-diagonal entries are covariance between the variables that represent the distortions (redundancy) in the data



Var(a)

$$\begin{bmatrix} \mathrm{Cov}(a, a) & \mathrm{Cov}(a, b) \\ \mathrm{Cov}(b, a) & \mathrm{Cov}(b, b) \end{bmatrix}$$

Var(b)

Eigenvalue

- For any nxn matrix A, we can find n eigenvalues that satisfy the characteristic equation
- A characteristic equation is defined as: |A - λI| = 0 i.e. det(A - λI) = 0

  where **I** is the identity matrix
- The characteristic polynomial for matrix A given as |A - λI|
- The scalar value λ is known as the eigenvalue of the matrix A
- Eigenvalues can be real/ complex in nature

Eigenvalues

- In PCA, the eigenvalue represents the total variance explained by the principal component
- The percentage of variation explained by the ith component is given as

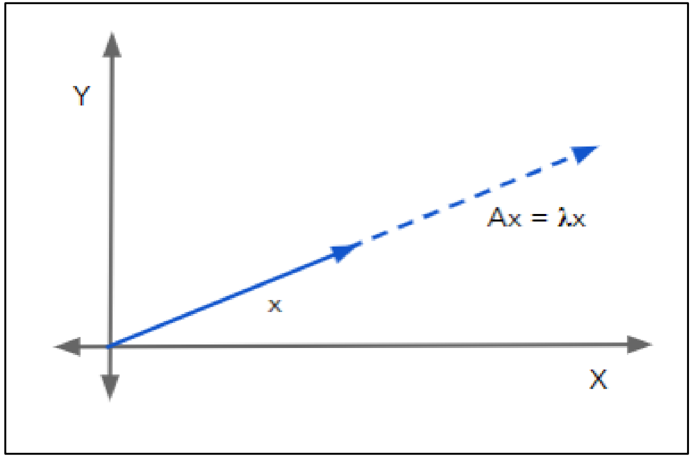
$$\left(\frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i}\right) * 100$$

where λ_i is the ith eigenvalue.

- The eigenvalues are considered in selecting the number of principal components

Eigenvector

- For each eigenvalue λ of a matrix A, there exist a non-zero vector x, which satisfy the equation: (A - λI)x = 0 i.e. Ax = λx
- The vector x is known as the eigenvector corresponding to the eigenvalue λ
- Eigenvectors of the distinct eigenvalues are always linearly independent
- The eigenvector is a vector that does not changes its direction, after transformation by matrix A



Loadings

- Mathematically, a principal component is the linear combination of the scaled (with mean = 0, standard deviation = 1) independent variables.
- i.e. PC1 = $w_{11}X_1 + w_{21}X_2 + \dots + w_{n1}X_n$.
- The coefficients of the principal components are also known as ‘loadings’
- w_1 is the loading vector consisting of elements (w_{11}, w_{21}, …, w_{n1}) of the first principal component
- The sum of the squares of the loadings for a principal component is 1

do it YOURSELF

Calculate the eigenvalues and eigenvectors of the given matrix.

$$A = \begin{bmatrix} 2 & -3 \\ 1 & 6 \end{bmatrix}$$

Selecting Principal Components

Selecting principal components

- The principal components are obtained by taking the dot product of the scaled data and the eigenvectors

- We use the eigenvalues of the covariance matrix to select the optimal number of principal components

- In order to reduce the data dimension, we consider first few principal components that explains most of the variation in the data

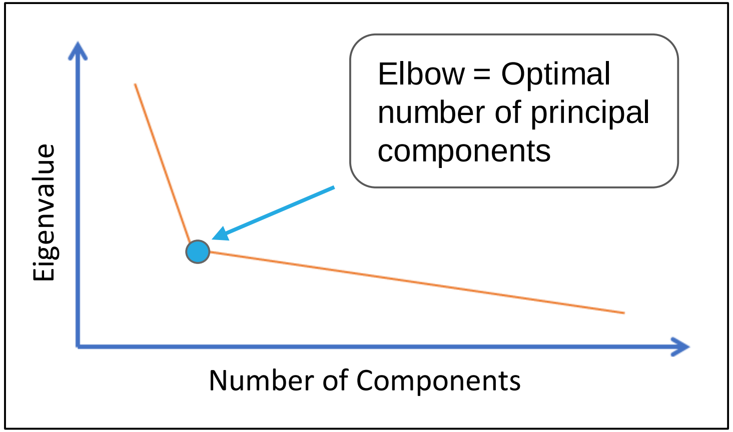
- Different criteria to select the principal components:
    - Kaiser Criterion
    - Scree Plot

Kaiser criterion

- It is an easy criterion to choose the principal components
- It selects the number of principal components for which the eigenvalue is greater than 1
- Higher eigenvalues explain most of the variance in the data

Scree plot

- This method plots the eigenvalues (Y-axis) against the number of principal components (X-axis)
- The elbow point in the scree plot corresponds to the optimal number of components
- After the elbow point, the components do not contribute much to the variance in the data
- This method fails if there is no explicit elbow point in the scree plot



Percentage of total variance

- One can decide the number of principal components based on the percentage of variance explained by the variables

- In most of the cases, the components explaining 70-80% of the variance can be considered as the principal components

- On the other hand, in some examples, the first few components explain only 50-60% of the total variance

Summary

- PCA is an unsupervised dimension reduction technique that uses the projection method to reduce the data dimension
- It finds the principal components that best represents the data in much lower dimensions
- The first principal component explains most of the variation in the data, the second principal component explains 2nd most variation in the data and so on
- The principal components are always orthogonal to one another
- Reducing the dimension of the data to 2 or 3 dimensions aids in data visualization

Case Study

Business example

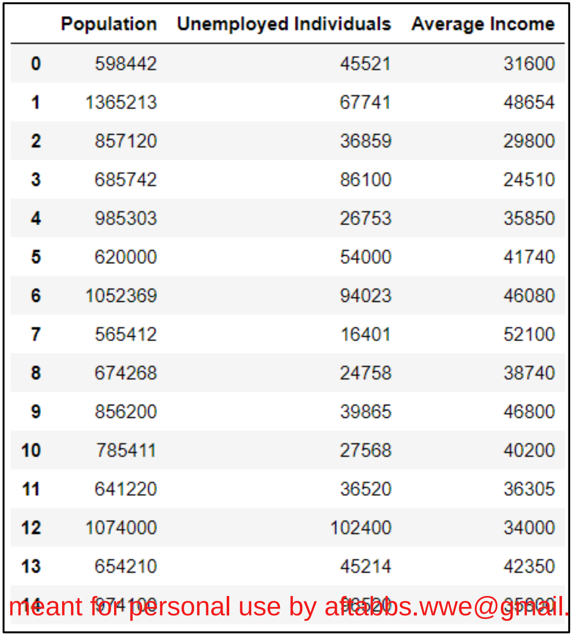
The Department of Social Welfare in Canada has collected data on the various factors that influence the crime rate (per 1,00,000 individuals) in different cities.

Three different factors are considered for the study: Total population, Unemployed individuals (between age 18-65) and Average annual income of the individual.

Let us reduce the 3-D data to 2-D to make the data more interpretable. In order to do so, we try to obtain 2 principal components which preserve the maximum information in the original data.

Data

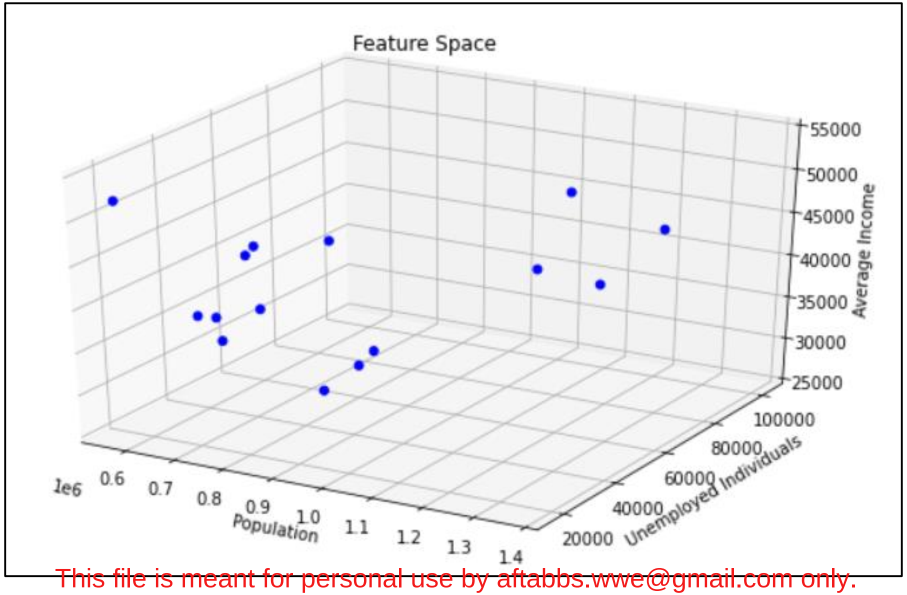
Consider the independent features affecting the crime rate in Canada.



	Population	Unemployed Individuals	Average Income
0	598442	45521	31600
1	1365213	67741	48654
2	857120	36859	29800
3	685742	86100	24510
4	985303	26753	35850
5	620000	54000	41740
6	1052369	94023	46080
7	565412	16401	52100
8	674268	24758	38740
9	856200	39865	46800
10	785411	27568	40200
11	641220	36520	36305
12	1074000	102400	34000
13	654210	45214	42350
14	974100	98520	35800

Data

The original feature space in 3-D.



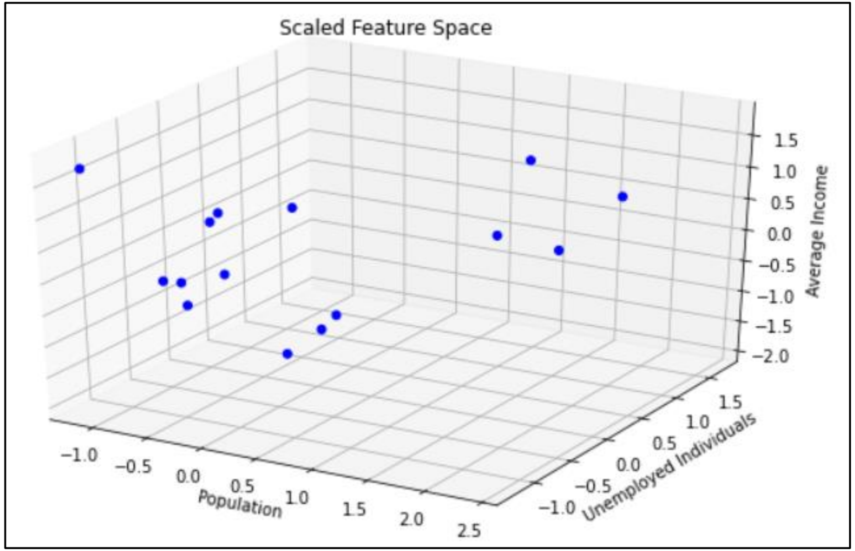
Step 1: standardize the data

$$x_{new} = \frac{x - \mu}{\sigma}$$

Where, μ: Mean of the variable

σ: Standard deviation of the variable

x: Original data points

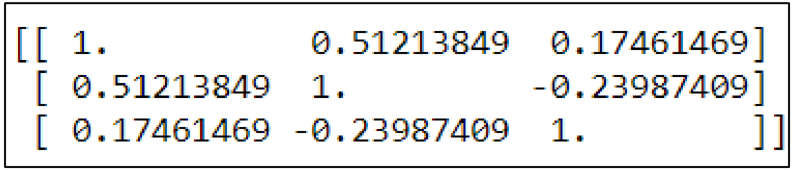


Step 2: covariance matrix

- The diagonal entries of the matrix represents the variance of each variable

- Here, for the standardized data, the variance is 1

- The off-diagonal values exhibits the relation between pair of variables



```
[[ 1.          0.51213849  0.17461469]
 [ 0.51213849  1.         -0.23987409]
 [ 0.17461469 -0.23987409  1.        ]]
```

Step 3: eigenvalues and eigenvectors

Let A be a covariance matrix and λ be the eigenvalue of A.

Eigenvalues are the roots of the equation:

$$det(A - \lambda I) = 0$$

$$det\left(\begin{bmatrix} 1 & 0.51213849 & 0.17461469 \\ 0.51213849 & 1 & -0.23987409 \\ 0.17461469 & -0.23987409 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}\right) = 0$$

Step 3: eigenvalues and eigenvectors

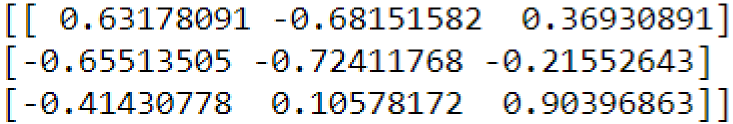
After solving the equation, we get

$\lambda_1 = 0.35442161,\ \lambda_2 = 1.51704963,\ \lambda_3 = 1.12852876$

Now find the eigenvectors by solving the following equation:

$$(A - \lambda I)x = 0$$

After solving the equation, we get the eigenvectors as:



```
[[ 0.63178091 -0.68151582  0.36930891]
 [-0.65513505 -0.72411768 -0.21552643]
 [-0.41430778  0.10578172  0.90396863]]
```

Step 4: sort the eigenvalues

Sort the eigenvalues in the descending order.

λ_2 = 1.51704963,

λ_3 = 1.12852876,

λ_1 = 0.35442161

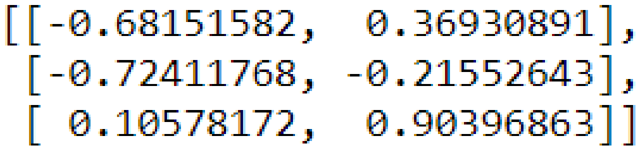
Since there are only 3 eigenvalues, we use the Kaiser criterion to decide the number of principal components.

Here, λ_2 and λ_3 are greater than 1. Thus we consider the eigenvectors corresponding to λ_2 and λ_3 as the loadings for principal components.

Step 5: select the eigenvectors

- Consider the eigenvectors corresponding to the eigenvalues λ_2 and λ_3

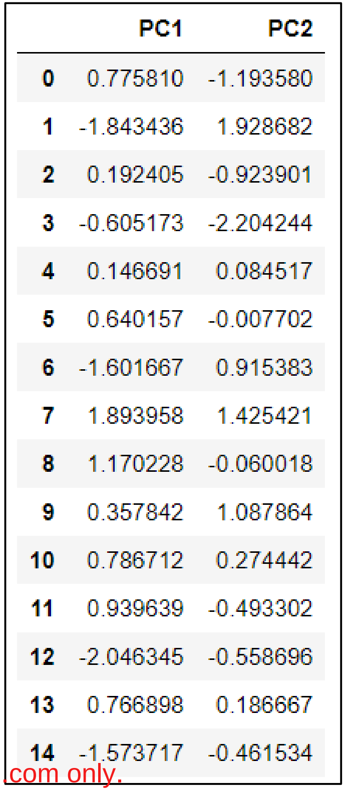
Coefficients of the first two principal components:



```
[[-0.68151582,  0.36930891],
 [-0.72411768, -0.21552643],
 [ 0.10578172,  0.90396863]]
```

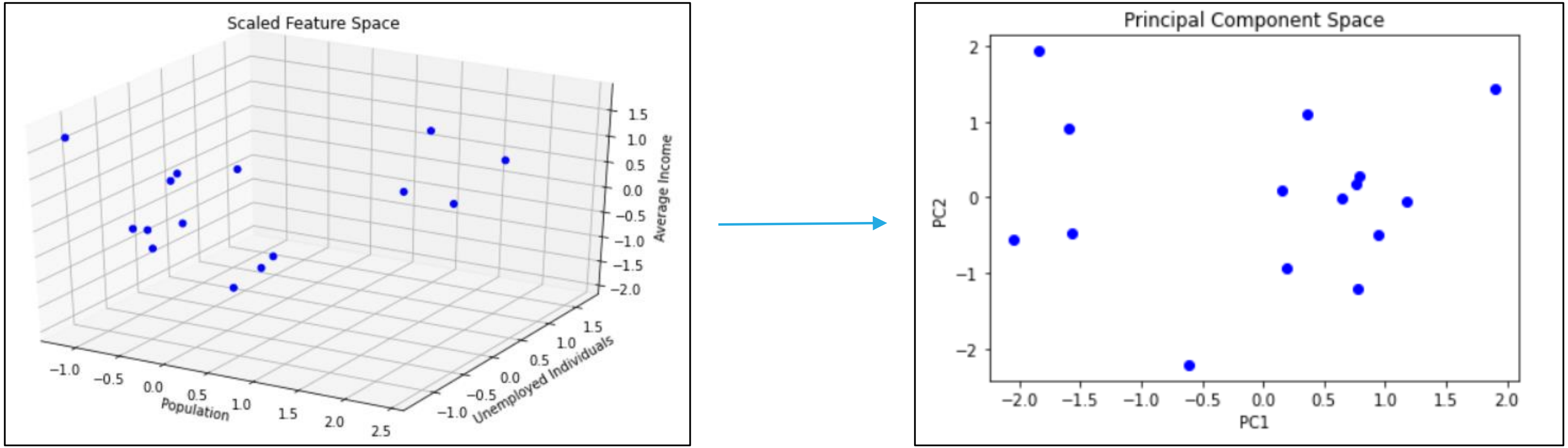
Step 6: select the components

Transform the original data by taking the dot product of the original data with the eigenvectors to get the principal components.



	PC1	PC2
0	0.775810	-1.193580
1	-1.843436	1.928682
2	0.192405	-0.923901
3	-0.605173	-2.204244
4	0.146691	0.084517
5	0.640157	-0.007702
6	-1.601667	0.915383
7	1.893958	1.425421
8	1.170228	-0.060018
9	0.357842	1.087864
10	0.786712	0.274442
11	0.939639	-0.493302
12	-2.046345	-0.558696
13	0.766898	0.186667
14	-1.573717	-0.461534

Summary


Scaled Feature Space
Population
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Unemployed Individuals
−1.0
−0.5
0.0
0.5
1.0
1.5
Average Income
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Principal Component Space
PC2
2
1
0
−1
−2
PC1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

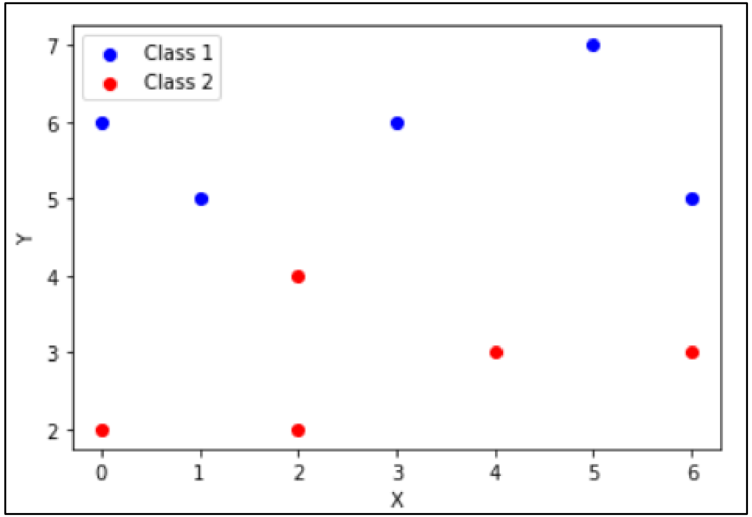
Linear Discriminant Analysis (LDA)

LDA

- LDA was formulated by Ronald A. Fisher in 1936
- It is a linear transformation technique and is most widely used for dimensionality reduction
- It is used as a pre-processing stage for pattern-classification
- The purpose of LDA is to lower the dimension space with a good separability between the classes
- It is known to have a practical use as a binary as well as multiclass classifier

LDA

- Consider the observations are divided into ‘c’ known classes/ categories with ‘n’ independent features
- In LDA, we project the original data on the k (k < c) vectors that maximizes the separation between the classes
- For the given figure, we project the data on the 1-D line that maximizes the separation between class 1 and class 2

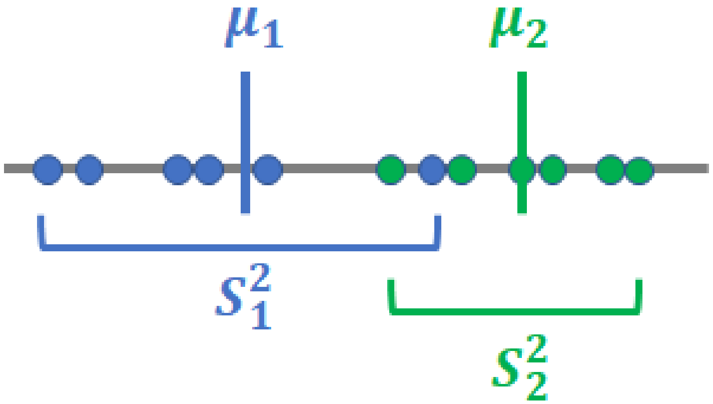


LDA

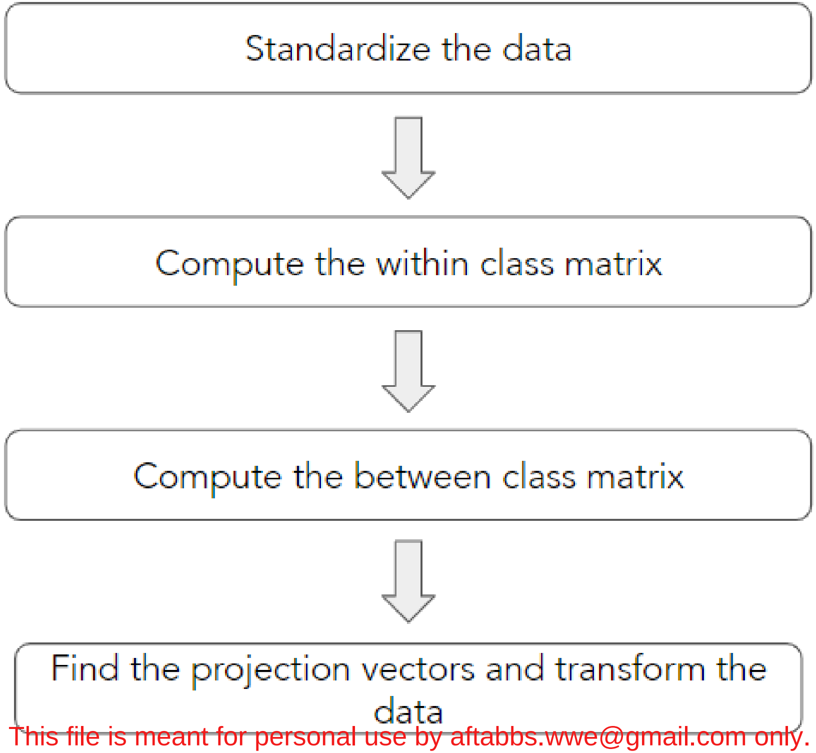
Goal of LDA is to:

- Maximize the distance between the means (i.e. between μ_1 and μ_2) of classes
- Minimize the variance within each class

In the given figure, μ_1 and μ_2 are the means, and S_1^2 and S_2^2 are the variances of the two classes.



LDA - procedure



Standardize the data

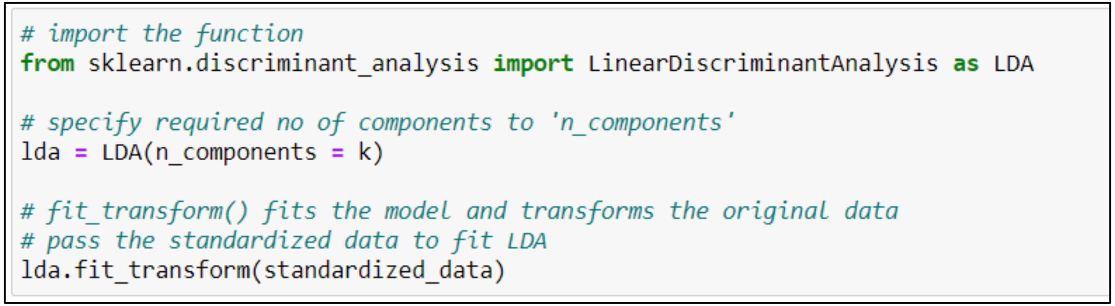
Compute the within class matrix

Compute the between class matrix

Find the projection vectors and transform the data

Python code

In python, we use the following code to perform LDA:



```
# import the function
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

# specify required no of components to 'n_components'
lda = LDA(n_components = k)

# fit_transform() fits the model and transforms the original data
# pass the standardized data to fit LDA
lda.fit_transform(standardized_data)
```

Terminologies

Within class matrix (S_W)

- It captures how precisely the data is scattered within the class
- Consider the data divided into two classes C_1 and C_2, the within class matrix is given by the summation of the covariance matrix (S_1) of the class C_1 and the covariance matrix (S_2) of the class C_2

$$S_w = S_1 + S_2$$

- We want to minimize S_W. i.e. the distance between the elements of the class should be minimum

Between class matrix (S_B)

- It represents how precisely the data is scattered across the classes

- Suppose the means of classes C_1 and C_2 are μ_1 and μ_2 respectively. The formula to find S_B is given as:

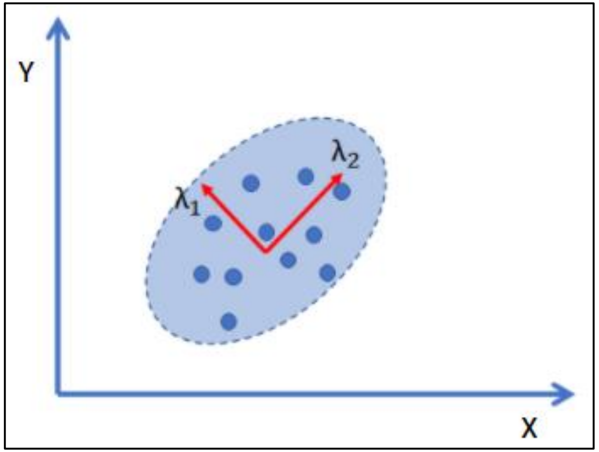
$$S_B = (\mu_1 - \mu_2).\,(\mu_1 - \mu_2)^T$$

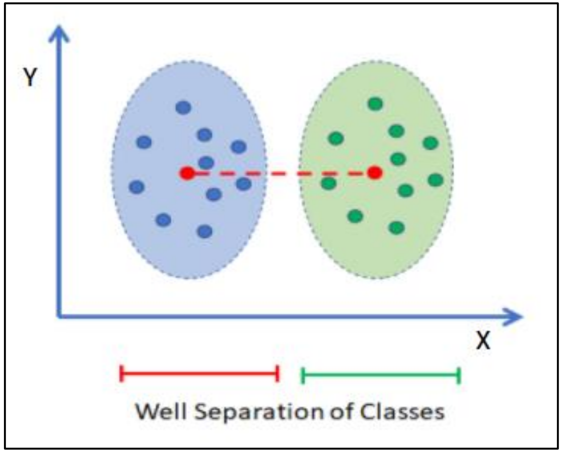
- We want to maximize S_B. i.e. the distance between the two classes should be higher

Projection vector

- The data samples are to be projected on the projection vectors with lower dimension
- These vectors are the eigenvectors of the matrix $S_w^{-1} . S_B$
- The eigenvector associated with the highest eigenvalue represents the direction of highest separability between the classes
- The projection vectors are the coefficients of the linear discriminants that exhibit the direction of maximum separability of classes
- Like PCA, transform the original data by taking a dot product of original data with projection vectors

PCA and LDA





In PCA the components maximize the variance

In LDA the component axes maximizes the classes

Summary

PCA

- It is an unsupervised machine learning technique
- Finds the components that maximize the variance in the data
- It obtains the components that 'best represents' the dataset into lower dimensions
- It identifies the direction of the maximum variation in the data

LDA

- It is a supervised machine learning technique
- Calculates the linear discriminants to obtain the maximum separation between the classes of the dependent variable
- It obtains the discriminants that 'best discriminates' between the classes
- It seeks to maximize the ratio of between-class and within-class variation

Thank You