UNIT 3

DATA PRE-PROCESSING & Dimensionality Reduction

METHOD 2: LDA (Linear Discriminant Analysis)

NEED FOR
LDA (Linear Discriminant Analysis)

Problem Statement

There are two groups -

- one who owns expensive cars
- other who owns affordable cars.

Find the characteristic / feature which can be used to predict the possibility of owning an expensive car. Apply LDA by using Fisher's Discriminant.

Most probable feature **Income**

Technique in Brief

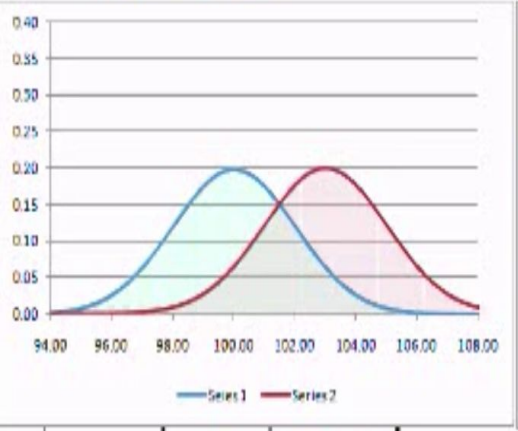
- **Group 1 :** Contains the dataset of people who owns the expensive car
- **Group 2 :** Contains the dataset of people who does not own the expensive car

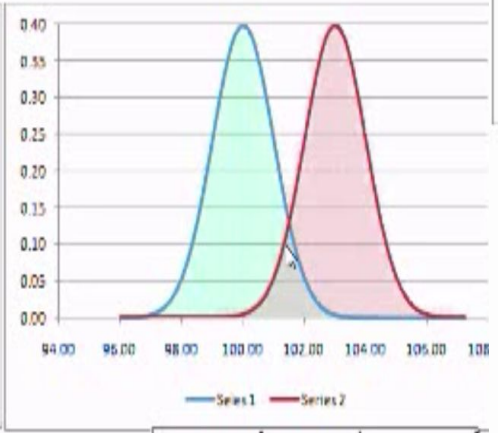
- Now we can calculate mean and variance of each of the independent variables for group 1 and group 2.
- Let's calculate Fisher's linear discriminant ratio for each variable using formula

$$\frac{(\text{Mean 1} - \text{Mean 2})^2}{\text{Var 1} + \text{Var 2}}$$

Bigger the value of ratio, greater the power of variable

- Now select, those variable which are having bigger values for this ratio
- What is the rational behind this ratio, let's see graphically

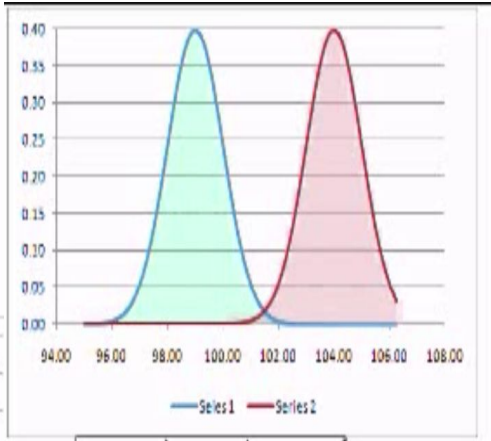




	series	series 2
Mean	100	103
std dev	2	2

NOTE:
Std dev = Root Mean Square of Variance

	Series	Series 2
Mean	100	103
std dev	1	1



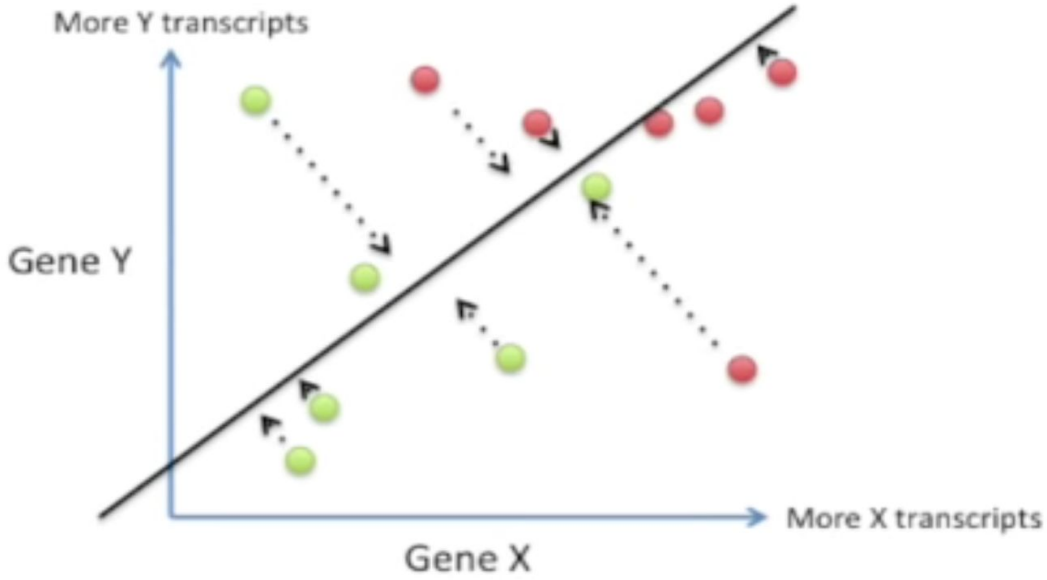
	Series	Series 2
Mean	99	104
std dev	1	1

Fischer's Ratio Significance

- So lesser the sum of variance: more clear is the distnction
- Greater the difference of the means of the population: clearer the distinction between two sets

A bit more insight into the TECHNIQUE OF LDA (Linear Discriminant Analysis)

Reducing a 2-D graph to a 1-D graph with LDA



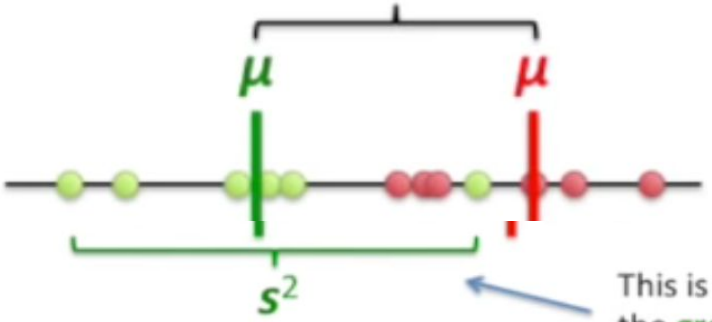
LDA uses both genes to create a new axis...

...and projects the data onto this new axis in a way to maximize the separation of the two categories.

How LDA creates a new axis…

The new axis is created according to two criteria (considered simultaneously):

1) Maximize the distance between means.



Let's call $(\mu - \mu)$ ***d*** for ***d***istance.

This is the scatter around the **green** dots.

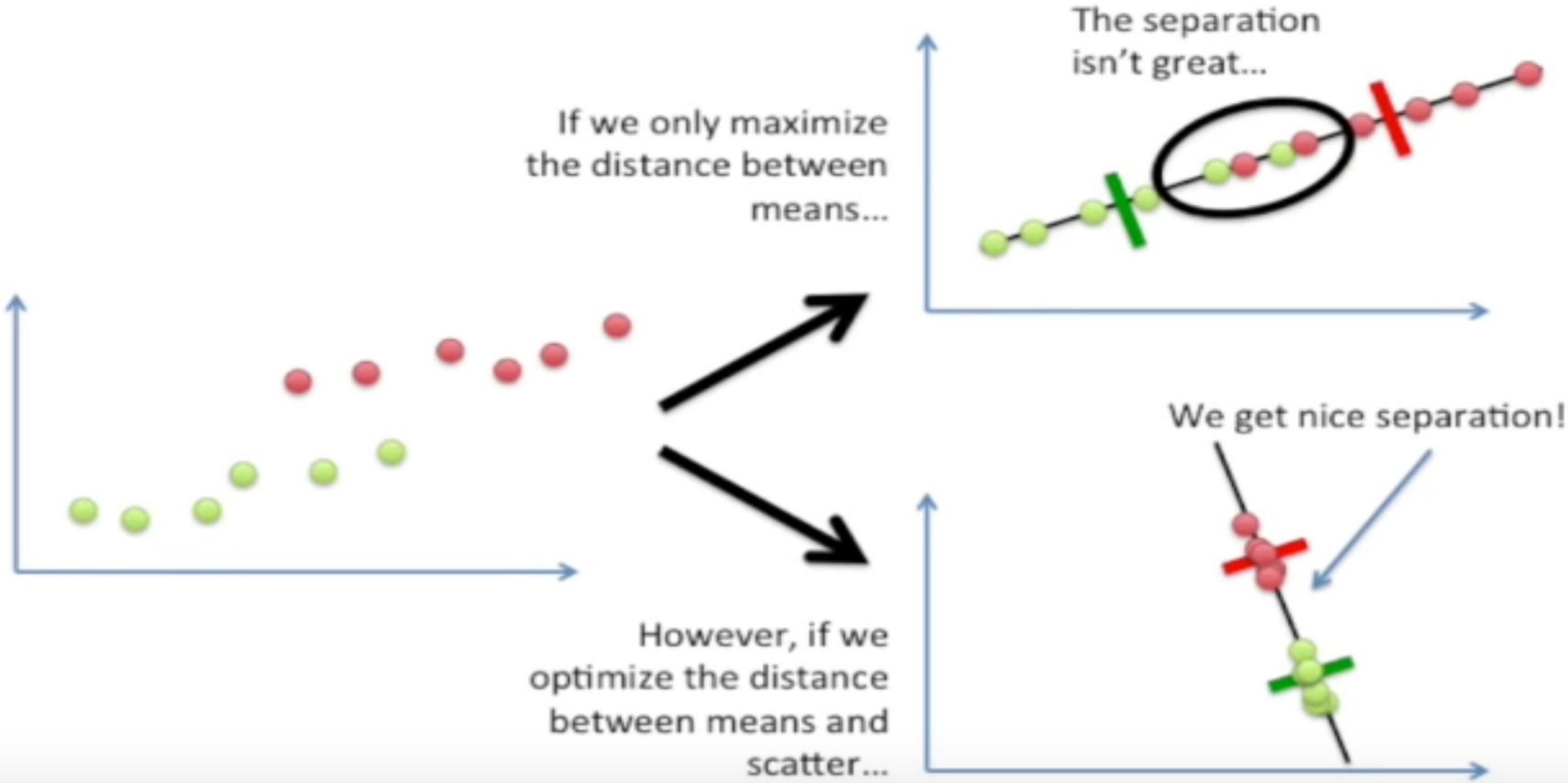
$$\frac{(\mu - \mu)^2}{s^2 + s^2} \rightarrow \frac{\text{Ideally large}}{\text{Ideally small}}$$

2) Minimize the variation (which LDA calls "scatter" and is represented by s^2) within each category.

$$\frac{d^2}{s^2 + s^2}$$

This is called the **"Fischer Ratio"**

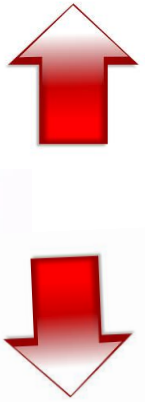
An example showing why both distance and scatter are important.



TECHNIQUE OF
LDA (Linear Discriminant Analysis)

Create an axis that maximizes the distance between the means for the two categories while minimizing the scatter.

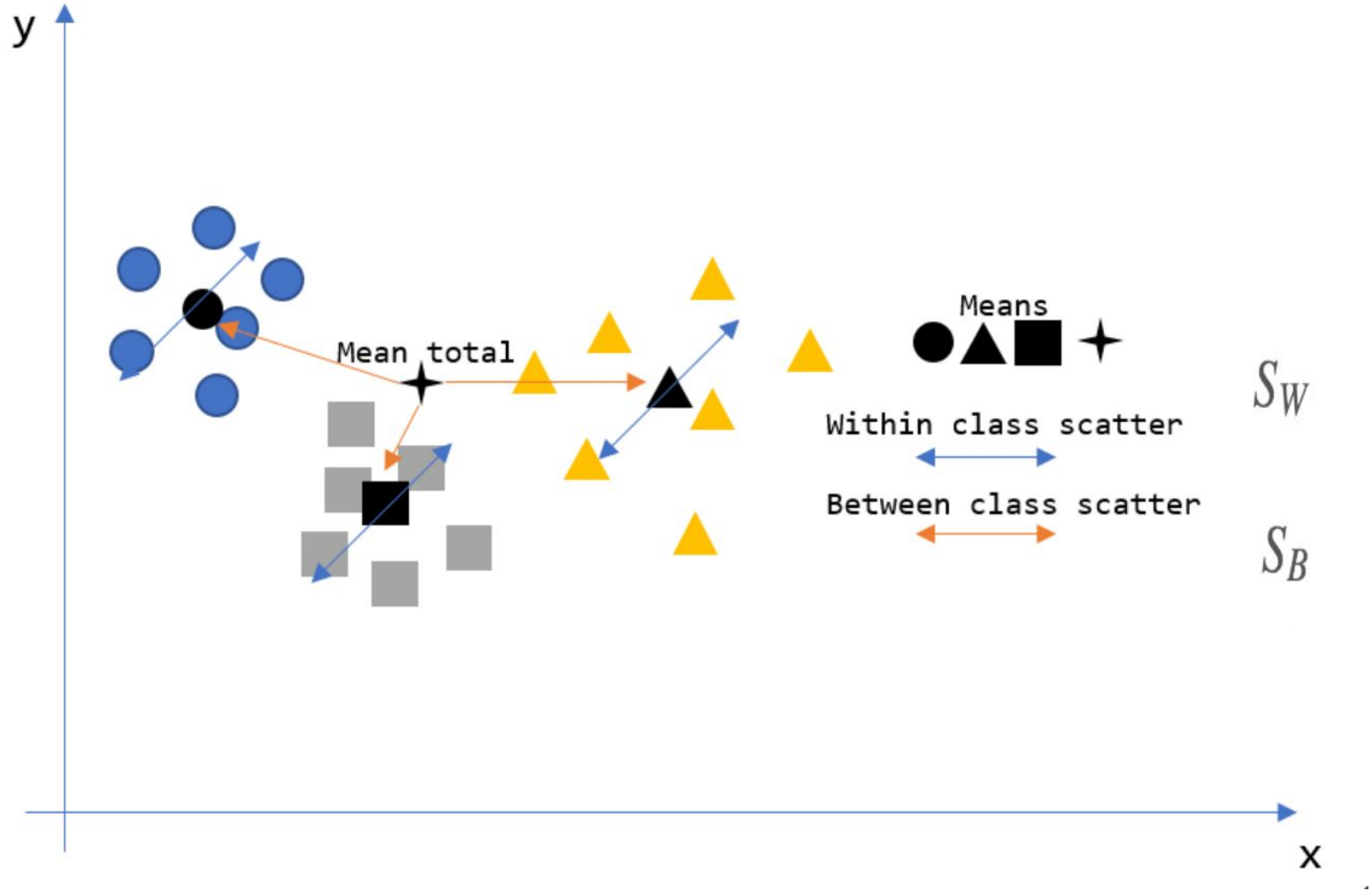
$$\frac{(\text{Mean 1} - \text{Mean 2})^2}{\text{Var 1} + \text{Var 2}}$$



ALGORITHM
LDA (Linear Discriminant Analysis)

LDA PROCESS SUMMARIZED IN 6 STEPS BY Raschka, S. (2015) p.139-140

1. Standardize the dataset (zero mean, standard deviation of 1)
2. Compute the total mean vector μ as well as the mean vectors per class μ_c
3. Compute the scatter withing and scatter between matrices S_B and S_W
4. Compute the eigenvalues and eigenvectors of $S_W^{-1} S_B$ to find the w which maximizes $\frac{w^T S_B w}{w^T S_W w}$
5. Select the Eigenvectors of the corresponding k largest Eigenvalues to create a dxk dimensional transformation matrix w where the Eigenvectors are the columns of this matrix
6. Use w to transform the original nxd dimensional dataset x into a lower, nxk dimensional dataset y


y
Mean total
Means
Within class scatter
S_W
Between class scatter
S_B
x

LDA : How Dataset (e.g.IRIS)and Feature Vector are related

	sepal length in cm	sepal width in cm	petal length in cm	petal width in cm	class label
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

Tip: LDA works only with numerical values.Textual categorical attribute should be converted into numerical values.

$$\boldsymbol{X} = \underbrace{\begin{bmatrix} x_{1_{\text{sepal length}}} & x_{1_{\text{sepal width}}} & x_{1_{\text{petal length}}} & x_{1_{\text{petal width}}} \\ x_{2_{\text{sepal length}}} & x_{2_{\text{sepal width}}} & x_{2_{\text{petal length}}} & x_{2_{\text{petal width}}} \\ \cdots & & & \\ x_{150_{\text{sepal length}}} & x_{150_{\text{sepal width}}} & x_{150_{\text{petal length}}} & x_{150_{\text{petal width}}} \end{bmatrix}}_{\text{Feature Vector}}, \quad \boldsymbol{y} = \begin{bmatrix} \omega_{\text{setosa}} \\ \omega_{\text{setosa}} \\ \cdots \\ \omega_{\text{virginica}} \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 1 \\ \cdots \\ 3 \end{bmatrix}$$

LDA : Dataset (e.g.IRIS) Mean Calculation and Resultant Vector

Computing the d-dimensional mean vectors

In this first step, we will start off with a simple computation of the mean vectors $\boldsymbol{m}_i$, $(i = 1, 2, 3)$ of the 3 different flower classes:

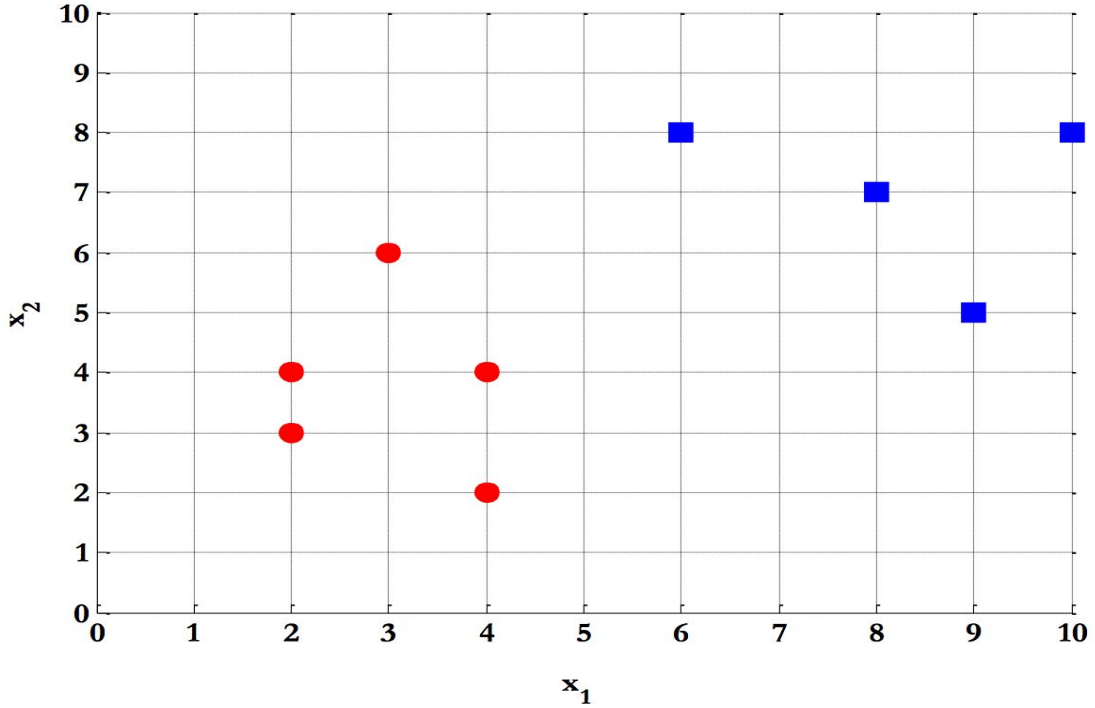
$$\boldsymbol{m}_i = \begin{bmatrix} \mu_{\omega_i}(\text{sepal length}) \\ \mu_{\omega_i}(\text{sepal width}) \\ \mu_{\omega_i}(\text{petal length}) \\ \mu_{\omega_i}(\text{petal width}) \end{bmatrix}, \quad \text{with} \quad i = 1, 2, 3$$

Dataset is n x d. In IRIS it is 150 x 4.
Mean Vector is d x 1. Here it is 4 x 1.

MATH BEHIND
And
Numerical solving using
LDA (Linear Discriminant Analysis)

Compute the Linear Discriminant projection for the following two-dimensional dataset.

- Samples for class ω_1 : $\mathbf{X_1}=(x_1,x_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$
- Sample for class ω_2 : $\mathbf{X_2}=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$



Feature Matrix:
n x d
n is # of samples
d is # of features
d is dimension of feature matrix

```
% samples for class 1
X1 = [4,2;
      2,4;
      2,3;
      3,6;
      4,4];


% samples for class 2
X2 = [9,10;
      6,8;
      9,5;
      8,7;
      10,8];
```

n x d here is **4 x 2**
d is 2 i.e. 2 dimensional feature matrix

Step 1: Computing the d-dimensional mean vectors

$\boldsymbol{m}_i$ is the mean vector

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$$

The Math

The Numerical

The classes mean are :

$$\mu_1 = \frac{1}{N_1} \sum_{x \in \omega_1} x = \frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix} + \begin{pmatrix}2\\4\end{pmatrix} + \begin{pmatrix}2\\3\end{pmatrix} + \begin{pmatrix}3\\6\end{pmatrix} + \begin{pmatrix}4\\4\end{pmatrix}\right] = \begin{pmatrix}3\\3.8\end{pmatrix}$$

$$\mu_2 = \frac{1}{N_2} \sum_{x \in \omega_2} x = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix} + \begin{pmatrix}6\\8\end{pmatrix} + \begin{pmatrix}9\\5\end{pmatrix} + \begin{pmatrix}8\\7\end{pmatrix} + \begin{pmatrix}10\\8\end{pmatrix}\right] = \begin{pmatrix}8.4\\7.6\end{pmatrix}$$

Step 2: Computing the Scatter Matrices

Now, we will compute the two *4x4*-dimensional matrices: The within-class and the between-class scatter matrix.

2.1 Within-class scatter matrix S_W

The **within-class scatter** matrix S_W is computed by the following equation:

$$S_W = \sum_{i=1}^{c} S_i$$

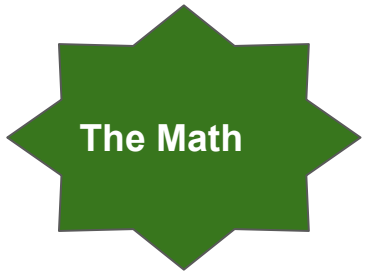
where

$$S_i = \sum_{\boldsymbol{x} \in D_i}^{n} (\boldsymbol{x} - \boldsymbol{m}_i)\ (\boldsymbol{x} - \boldsymbol{m}_i)^T$$

(scatter matrix for every class)

and $\boldsymbol{m}_i$ is the mean vector

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$$



Step 2: Computing the Scatter Matrices

Covariance matrix of the first class: For Feature X1

$$S_1 = \sum_{x \in \omega_1} (x - \mu_1)(x - \mu_1)^T = \left[\begin{pmatrix} 4 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 2 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2$$

$$+ \left[\begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 3 \\ 6 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix}$$

The Numerical

Step 2: Computing the Scatter Matrices

Covariance matrix of the second class: For Feature X2

$$S_2 = \sum_{x \in \omega_2} (x - \mu_2)(x - \mu_2)^T = \left[\begin{pmatrix} 9 \\ 10 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 6 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$+ \left[\begin{pmatrix} 9 \\ 5 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 8 \\ 7 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 10 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

The Numerical

The **within-class scatter** matrix S_W is computed by the following equation:

Within-class scatter matrix:

The Math

$$S_W = \sum_{i=1}^{c} S_i$$

$$S_w = S_1 + S_2 = \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix} + \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

$$= \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}$$

The Numerical

2.2 Between-class scatter matrix S_B

The **between-class scatter** matrix S_B is computed by the following equation:

$$S_B = \sum_{i=1}^{c} N_i(\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

where

$\boldsymbol{m}$ is the overall mean, and $\boldsymbol{m}_i$ and N_i are the sample mean and sizes of the respective classes.

The Math

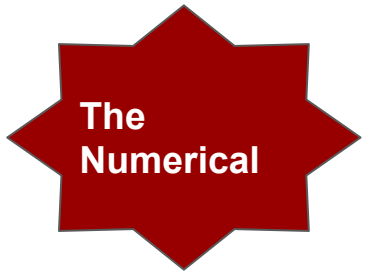
Between-class scatter matrix:

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

$$= \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]\left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^T$$

$$= \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}\begin{pmatrix} -5.4 & -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix}$$



Step 3: Solving the generalized eigenvalue problem for the matrix $S_W^{-1} S_B$

Next, we will solve the generalized eigenvalue problem for the matrix $S_W^{-1} S_B$ to obtain the linear discriminants.

Checking the eigenvector-eigenvalue calculation

A quick check that the eigenvector-eigenvalue calculation is correct and satisfy the equation:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$

where

$\boldsymbol{A} = S_W^{-1} S_B$

$\boldsymbol{v} =$ Eigenvector

$\lambda =$ Eigenvalue

The LDA projection is then obtained as the solution of the generalized eigen value problem

$$S_W^{-1} S_B w = \lambda w$$

$$\Rightarrow \left| S_W^{-1} S_B - \lambda I \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{pmatrix} \right|$$

$$= (9.2213 - \lambda)(2.9794 - \lambda) - 6.489 \times 4.2339 = 0$$

$$\Rightarrow \lambda^2 - 12.2007\lambda = 0 \Rightarrow \lambda(\lambda - 12.2007) = 0$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$

The Numerical

Computing the Eigen Vectors (w_1 and w_2) of the Eigen Values obtained (λ_1 and λ_2)

Hence

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_1 = \underbrace{0}_{\lambda_1} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

and

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_2 = \underbrace{12.2007}_{\lambda_2} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

Thus;

$$w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix} \quad and \quad w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*$$

The optimal projection is the one that given maximum $\lambda = J(w)$

The Numerical

Step 4: Selecting linear discriminants for the new feature subspace

The Math

4.1. Sorting the eigenvectors by decreasing eigenvalues

4.2. Choosing *k* eigenvectors with the largest eigenvalues

After sorting the eigenpairs by decreasing eigenvalues, it is now time to construct our $k \times d$-dimensional eigenvector matrix $\boldsymbol{W}$

The Numerical

Eigen Vectors	Eigenvalues
$\begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix}$ ✓	$\lambda_2 = 12.2007$
$= \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix}$	$\lambda_1 = 0$

Step 5: Transforming the samples onto the new subspace

In the last step, we use the $n \times d$-dimensional matrix $\boldsymbol{W}$ that we just computed to transform our samples onto the new subspace via the equation

$$\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}.$$

(where $\boldsymbol{X}$ is a $n \times d$-dimensional matrix representing the n samples, and $\boldsymbol{Y}$ are the transformed $n \times k$-dimensional samples in the new subspace).

X_1

$$\begin{matrix}4,2\\2,4\\2,3\\3,6\\4,4\end{matrix} \times \begin{pmatrix}0.9088\\0.4173\end{pmatrix} = \begin{pmatrix}4 \times 0.9 + 2 \times 0.4\\2 \times 0.9 + 4 \times 0.4\\2 \times 0.9 + 3 \times 0.4\\3 \times 0.9 + 6 \times 0.4\\4 \times 0.9 + 4 \times 0.4\end{pmatrix} = \begin{pmatrix}4.4\\3.4\\3\\5.1\\5.2\end{pmatrix}$$

The Numerical

$$X \times W = Y$$

Eigenvalues

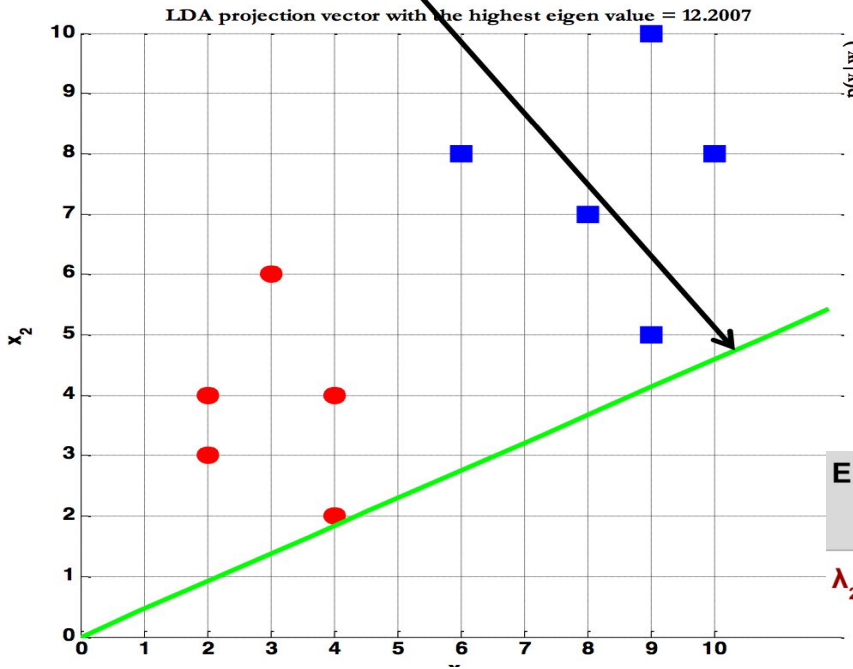
$\lambda_2 = 12.2007$

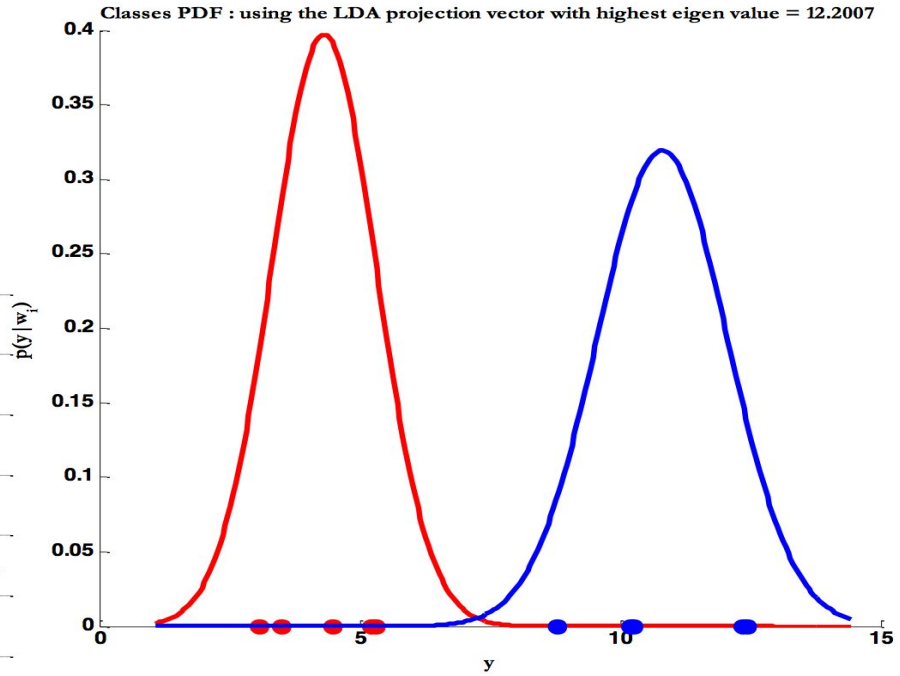
$$\begin{matrix}9,10\\6,8\\9,5\\8,7\\10,8\end{matrix} \times \begin{pmatrix}0.9088\\0.4173\end{pmatrix} = \begin{pmatrix}9 \times 0.9 + 10 \times 0.4\\6 \times 0.9 + 8 \times 0.4\\9 \times 0.9 + 5 \times 0.4\\8 \times 0.9 + 7 \times 0.4\\10 \times 0.9 + 8 \times 0.4\end{pmatrix} = \begin{pmatrix}12.1\\8.6\\10.1\\10\\12.2\end{pmatrix}$$

X_2

LDA - Projection

The projection vector corresponding to the **highest** eigen value





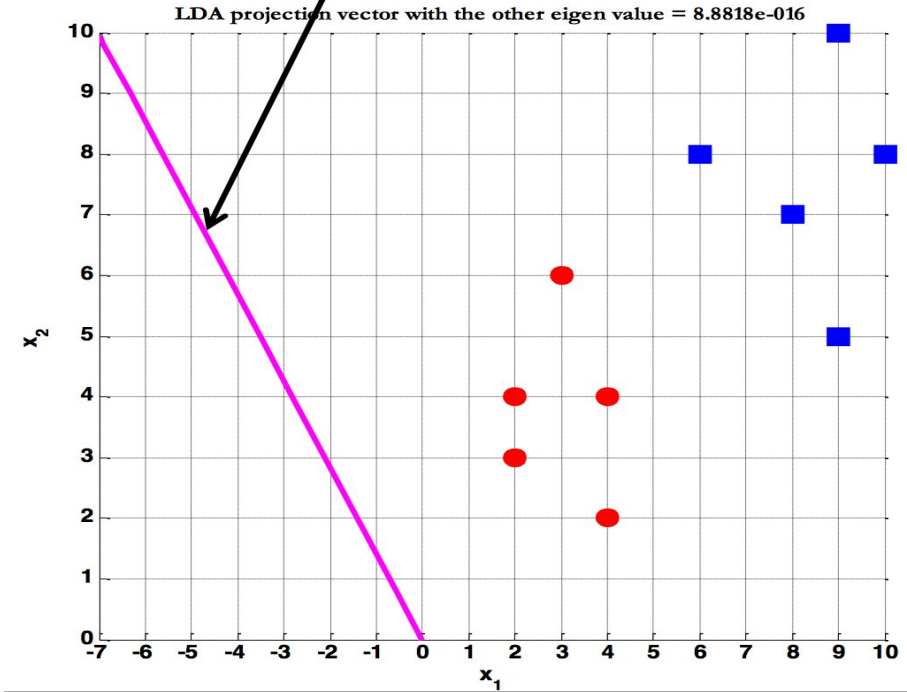
Eigenvalues

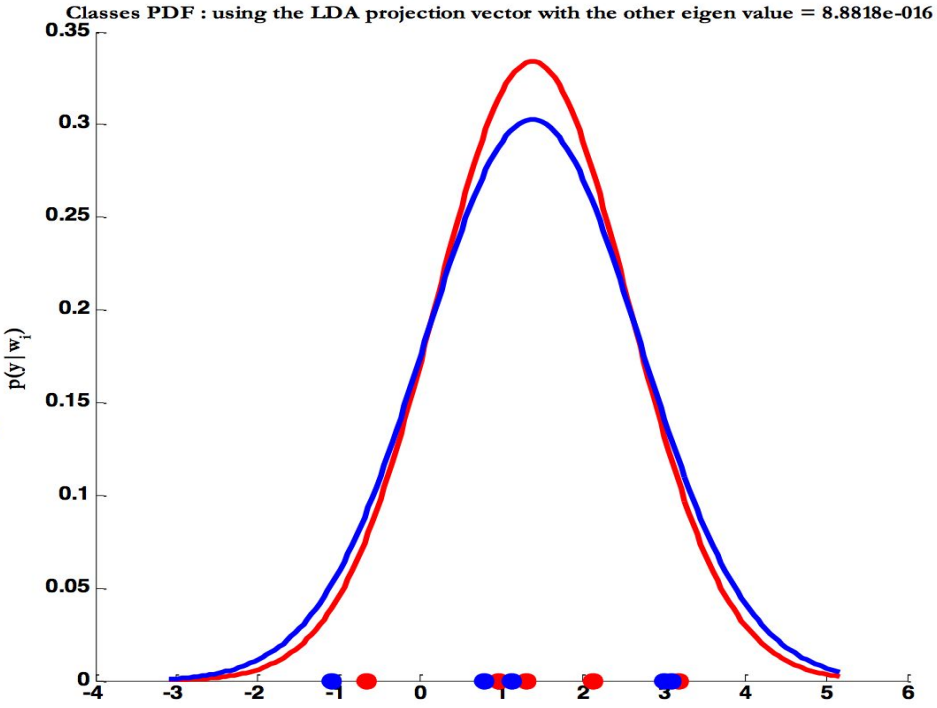
$\lambda_2 = 12.2007$

Using this vector leads to **good separability** between the two classes

LDA - Projection

The projection vector corresponding to the **smallest** eigen value



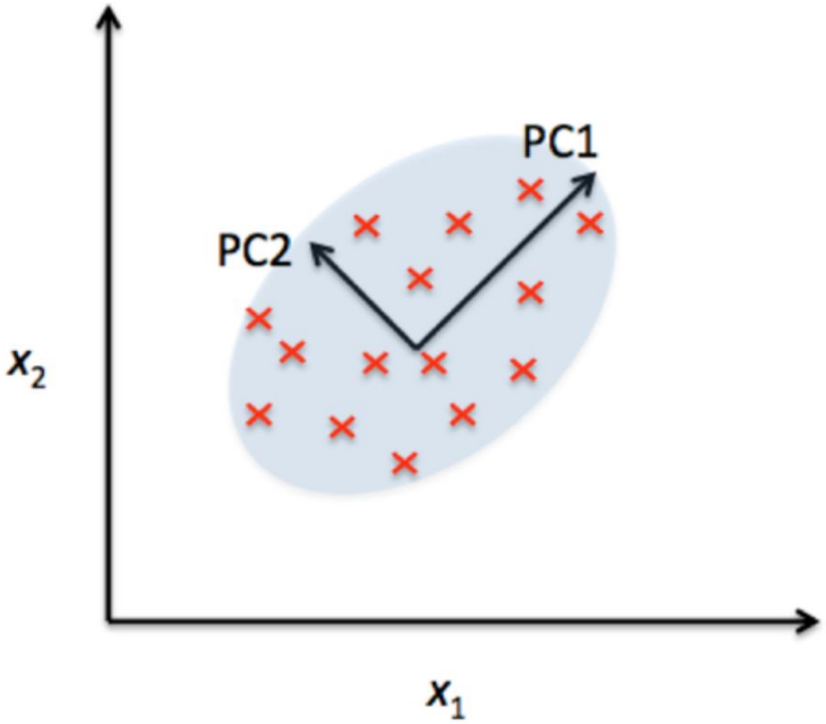


Eigenvalues

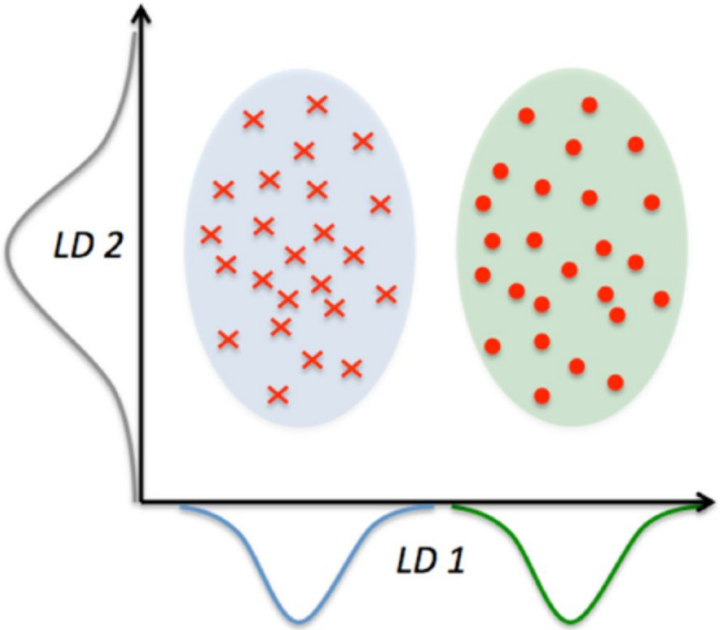
$\lambda_1 = 0$

Using this vector leads to **bad separability** between the two classes

Difference between
PCA (Principal Component Analysis)
&
LDA (Linear Discriminant Analysis)

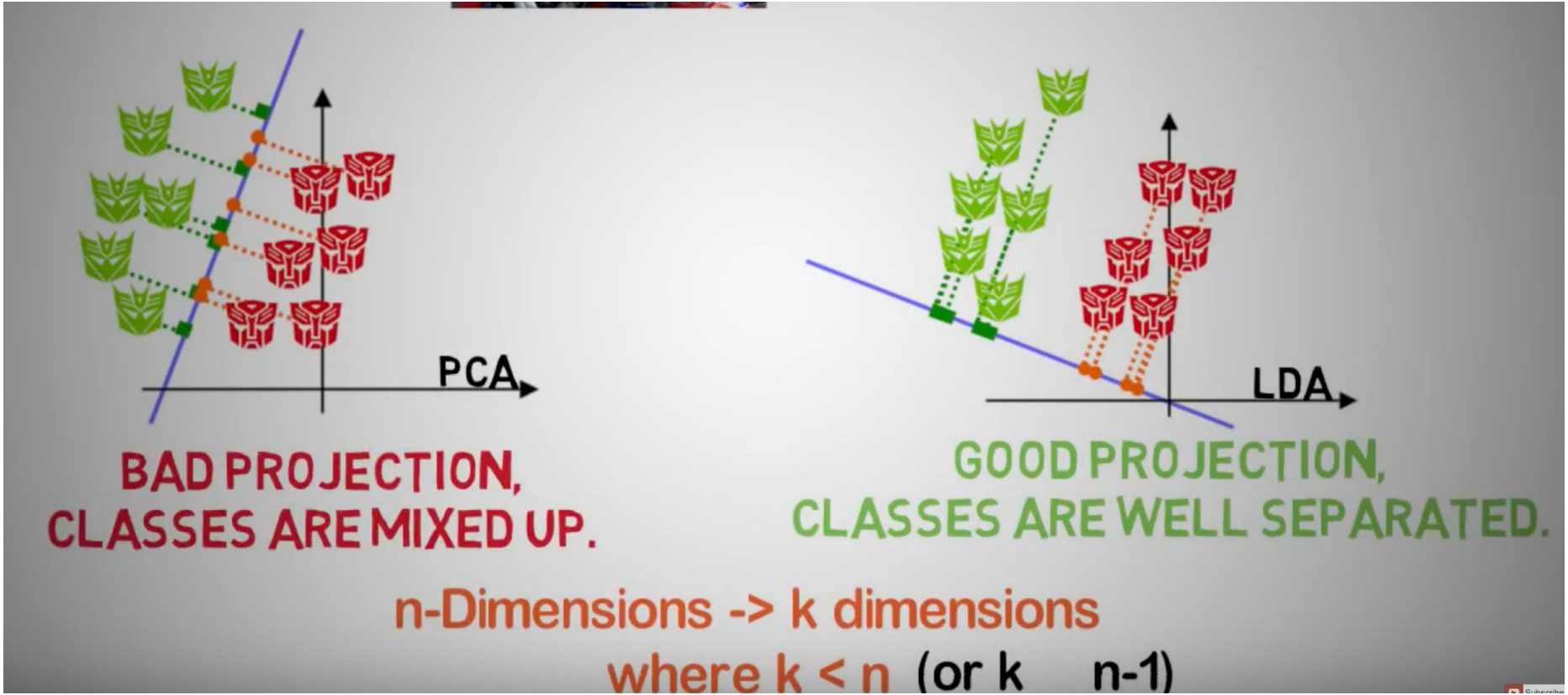


PCA as a technique that finds the directions of maximal variance



LDA attempts to find a feature subspace that maximizes class separability

PCA	LDA
• PCA applied to data tries to identify directions in the feature space that account to the most variance between the data	• LDA tries to identify characteristics that account to the most variance between classes. Its underlying principle is the 'Fischer Ratio'.
• PCA is "un-supervised", in the sense it ignores class labels.	• LDA is supervised and takes into account the class labels.
• Training set is small, PCA out-performs LDA. • Loss function is MSE (Mean Square Error)	• When the number of training samples is large, and is representative of each class, LDA is better. • Loss Function is Fischer's Criteria ($\frac{w^T S_B w}{w^T S_W w}$
• No distinction between inter-class and inter-class variability	• Finds a feature sub-space that maximises the ratio of inter-class and inter-class variability.
• PCA can be used even when the number of samples is less than the number of dimensions(features).	• LDA requires the number of samples is to be at least equal (ideally greater) to number of dimensions. Else it will give incorrect results.
• PCA is generally not used for classification as it ignores class labels and is un-supervised.	• LDA can be used as a classification algorithm also, apart from dimensionality reduction. In fact, it is used for multi-class classification in the MLDA form.

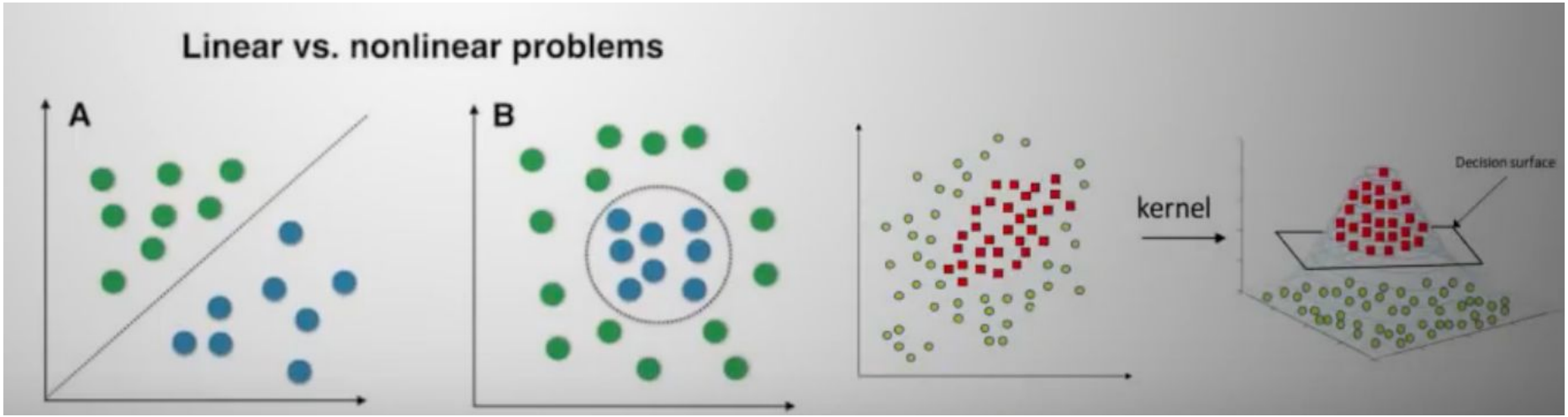


Tip: LDA uses the class labels provided in the data set. So it is well-suited for multi-class classification. This is the variant of LDA called MLDA - Multiple Linear Discriminant Analysis.

Limitations of LDA

1. This is applicable only for numeric variables
2. It can easily give variable more suitable for classification kind of work
3. It is computationally less demanding than info value method because you are not creating groups for a numeric variable
4. If the dataset is labelled in imbalanced way,or if some features are missing it can provide misleading result because mean and variance might be getting calculated just based on very few records

Limitations of LDA



LDA cannot be used if the data is not Linearly Separable. In such cases, Kernel Tricks need to be used to transform the data to a three dimensional space and them seperate it out using a hyperplane.

Tips: Computer Vision Projects prefer LDA over PCA. LDA works well in different lighting conditions and different facial expressions. Refer Paper by A.M.Martiniz(2001)