Introduction to Linear Discriminant Analysis

Jaclyn Kokx

1. Who am I?
2. What is LDA?
3. Why use LDA?
4. How do I use LDA?

https://memegenerator.net/instance/55587306/winter-is-coming-brace-yourself-powerpoint-slides-are-coming

Who Am I?

Jackie Kokx

MS Mechanical Engineering

BS Materials Science & Engineering

Biotech Industry

Algebra Instructor

Self-Taught Data Enthusiast

Ultimate Frisbee, Running, Mom

www.jaclynkokx.com

What is LDA?

A linear combination of features

A supervised dimensionality reduction technique to be used with continuous independent variables and a categorical dependent variables

Linear Discriminant Analysis

separates two or more classes

Because it works with numbers and sounds science-y

More LDA

Ronald Fisher

Fisher's Linear Discriminant (2-Class Method) - 1936

C R Rao

Multiple Discriminant Analysis - 1948

Also used

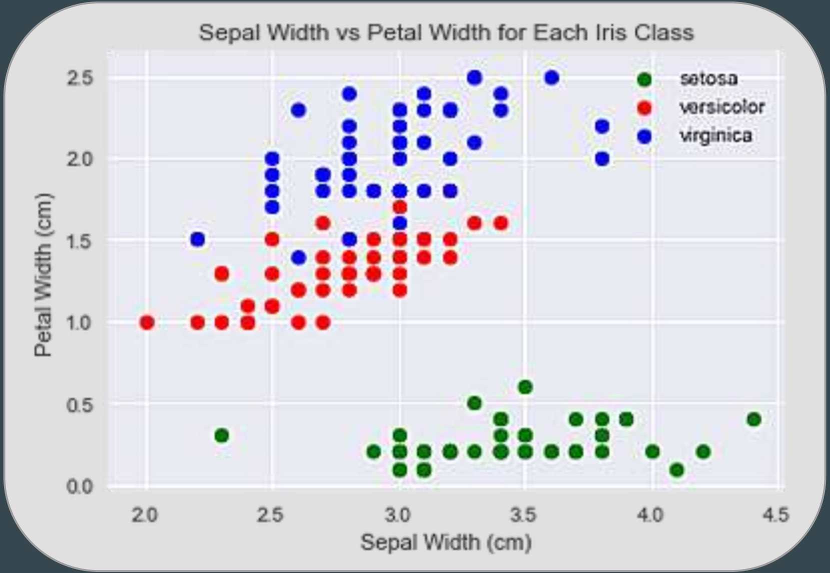
As a data classification technique

http://www.swlearning.com/quant/kohler/stat/biographical_sketches/Fisher_3.jpeg
http://www.buffalo.edu/ubreporter/archive/2011_07_21/rao_guy_medal.html

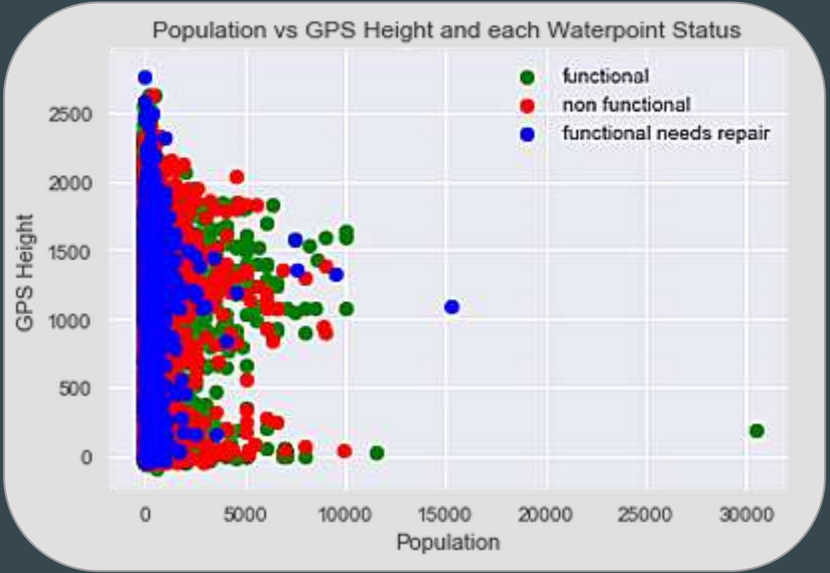
Why Would I Want to Use LDA?

Because real life data is ugly.

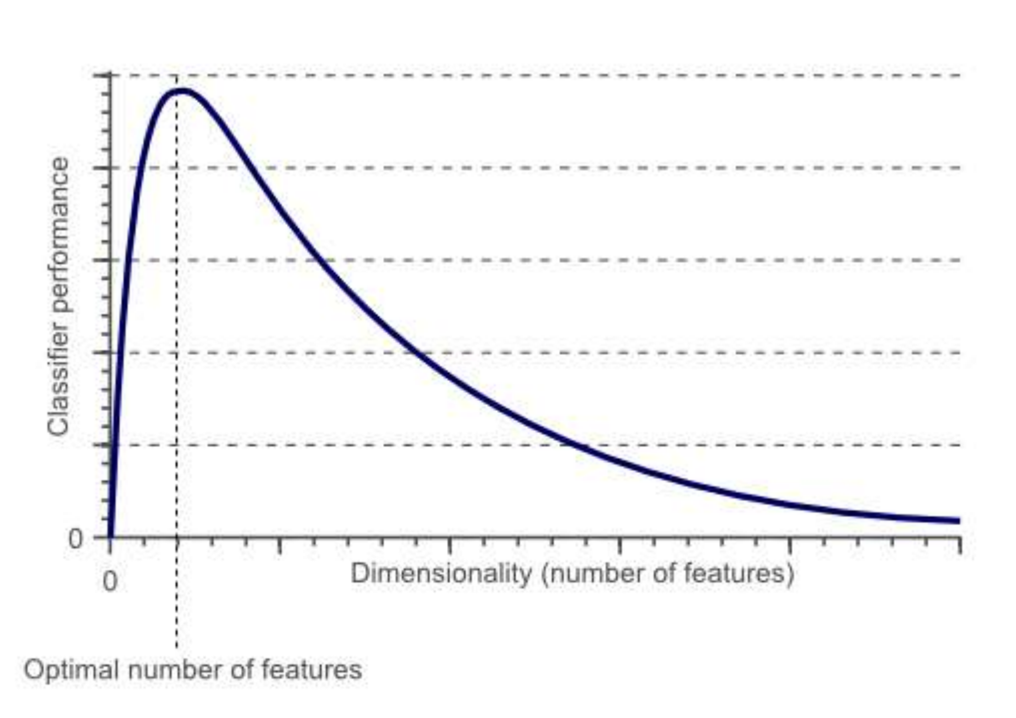
Iris Dataset



Your (My) Dataset



The curse of dimensionality



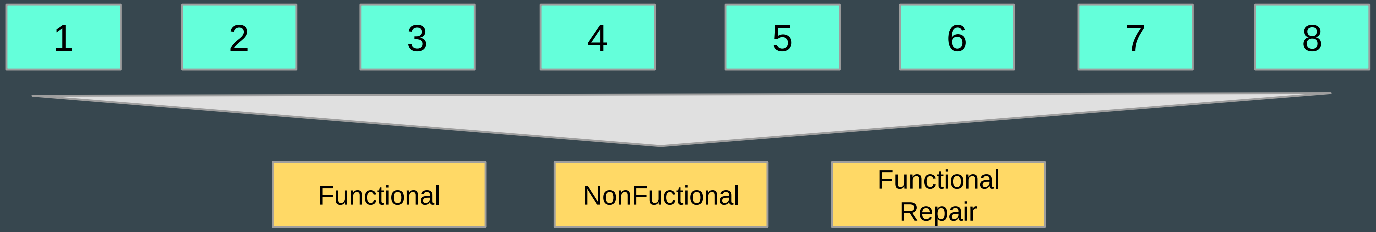
Reducing the number of your features might improve your classifier

http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

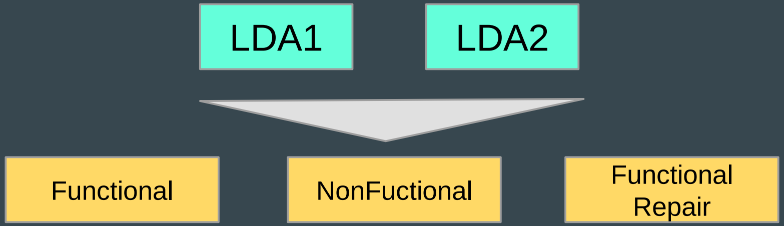
How much can I reduce my number of features?

<u>One less than the number of labeled classes you have</u>

My waterpoint dataset



My waterpoint dataset after LDA



3 Classes - 1 = 2 features (dimensions)

How Do I Use LDA?

Due credit to Sebastian Raschka http://sebastianraschka.com/Articles/2014_python_lda.html

sebastianraschka

ABOUT BOOKS DEEP LEARNING ELSEWHERE PROJECTS RESOURCES BLOG

MACHINE LEARNING FAQ

[RSS]

Linear Discriminant Analysis

– Bit by Bit

Aug 3, 2014
by Sebastian Raschka

Sections

Linear discriminant analysis: A detailed tutorial

Alaa Tharwat [a,b,*,**], Tarek Gaber [c,*], Abdelhameed Ibrahim [d,*] and Aboul Ella Hassanien [e,*]

[a] *Department of Computer Science and Engineering, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany*

[b] *Faculty of Engineering, Suez Canal University, Egypt*

E-mail: engalaatharwat@hotmail.com

[c] *Faculty of Computers and Informatics, Suez Canal University, Egypt*

E-mail: tmgaber@gmail.com

[d] *Faculty of Engineering, Mansoura University, Egypt*

E-mail: afai79@yahoo.com

[e] *Faculty of Computers and Information, Cairo University, Egypt*

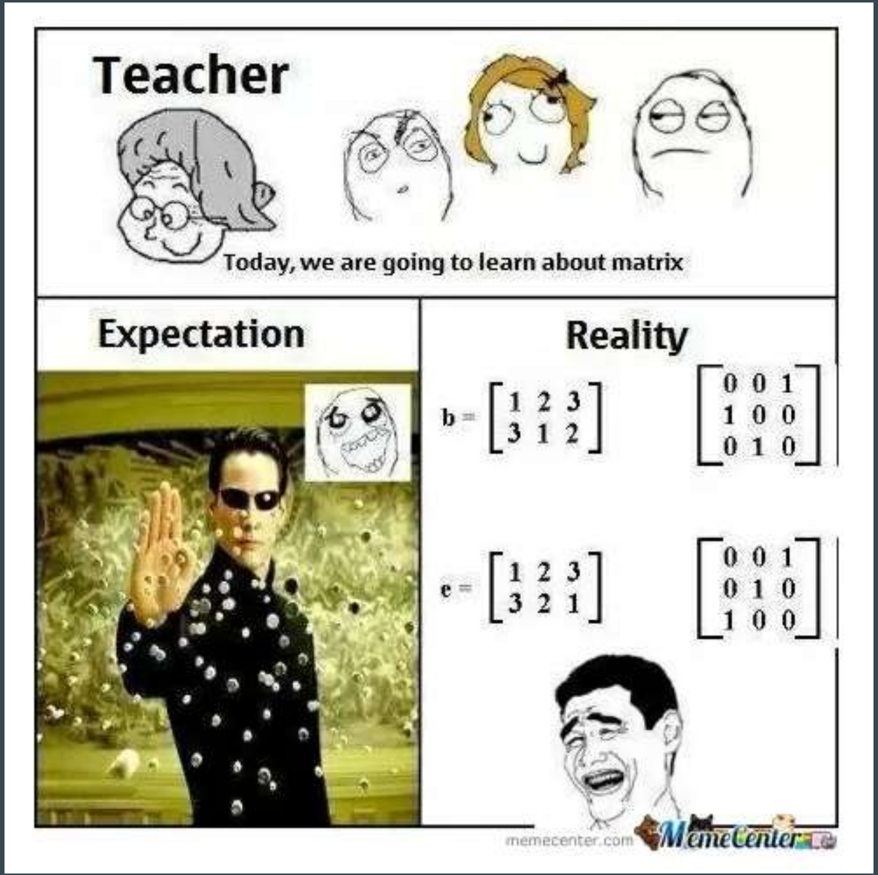
E-mail: aboitcairo@gmail.com

Abstract. Linear Discriminant Analysis (LDA) is a very common technique for dimensionality reduction problems as a pre-processing step for machine learning and pattern classification applications. At the same time, it is usually used as a black box, but (sometimes) not well understood. The aim of this paper is to build a solid intuition for what is LDA, and how LDA works, thus enabling readers of all levels be able to get a better understanding of the LDA and to know how to apply this technique in different applications. The paper first gave the basic definitions and steps of how LDA technique works supported with visual explanations of these steps. Moreover, the two methods of computing the LDA space, i.e. *class-dependent* and *class-independent* methods, were explained in details. Then, in a step-by-step approach, two numerical examples are demonstrated to show how the LDA space can be calculated in case of the class-dependent and class-independent methods. Furthermore, two of the most common LDA problems (i.e. *Small Sample Size (SSS)* and non-linearity problems) were highlighted and illustrated, and state-of-the-art solutions to these problems were investigated and explained. Finally, a number of experiments was conducted with

https://www.researchgate.net/publication/316994943_Linear_discriminant_analysis_A_detailed_tutorial

What you need to start:

- Labeled data
- Ordinal features (the number kind, not the category kind)
- Some idea about matrix algebra
- (Python)



https://www.memecenter.com/fun/330902/matrix--expectation-vs-reality

The Big Picture

Create a subspace that separates our classes really well, then we'll project our data onto that subspace.

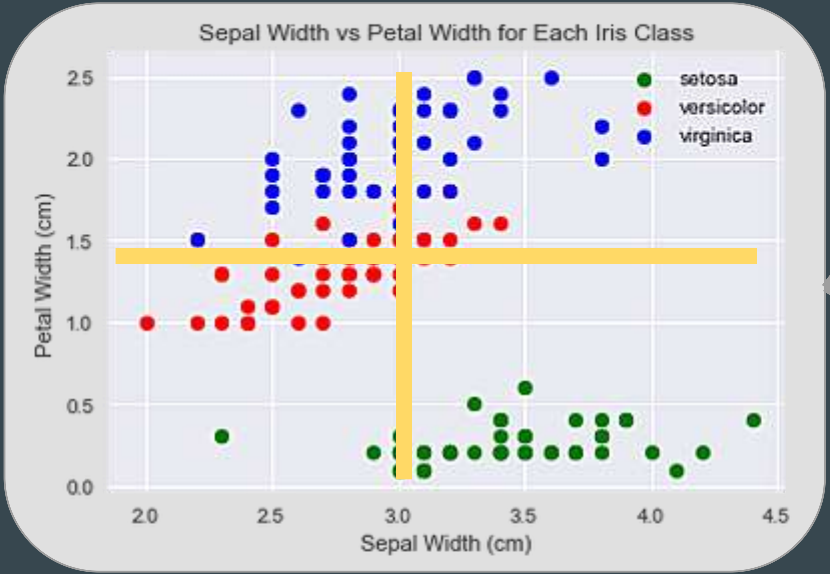
Linear Discriminant: $$y = W^T * x$$

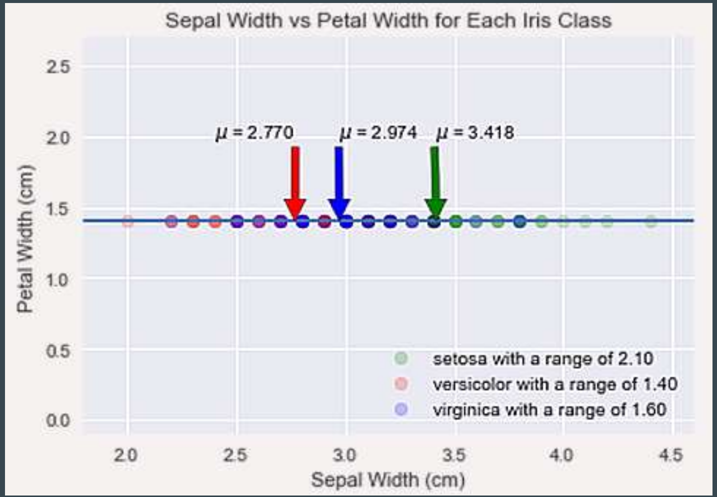
To find that optimal subspace we're going to:

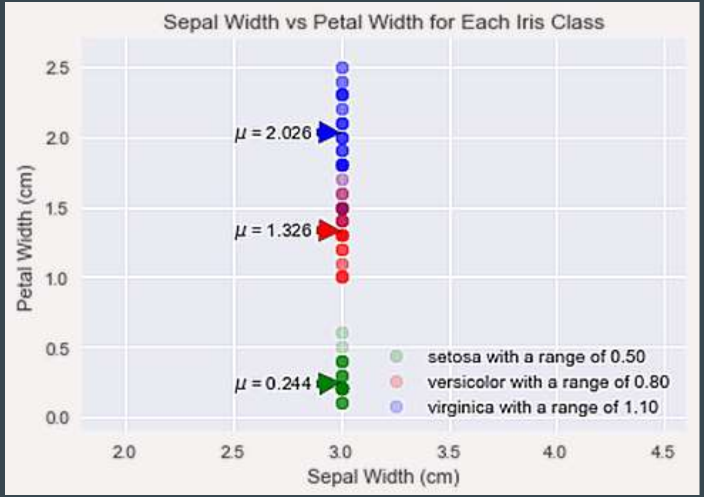
- Minimize within-class variance (we want tight groups)
- Maximize the between-class distance (we want our groups as far apart as possible)

Let’s visualize it

Iris Dataset







Another View

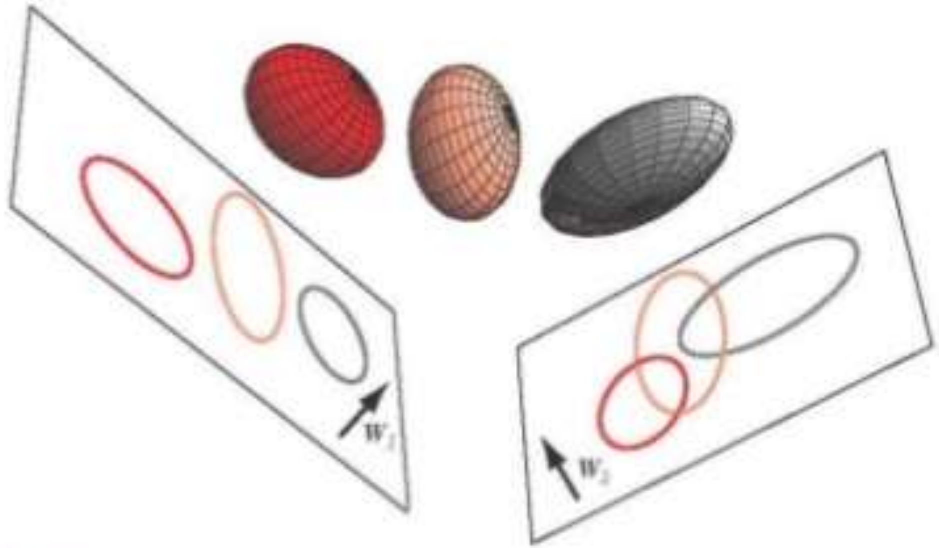


FIGURE 3.6. Three three-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors $\mathbf{W}_1$ and $\mathbf{W}_2$. Informally, multiple discriminant methods seek the optimum such subspace, that is, the one with the greatest separation of the projected distributions for a given total within-scatter matrix, here as associated with $\mathbf{W}_1$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*.

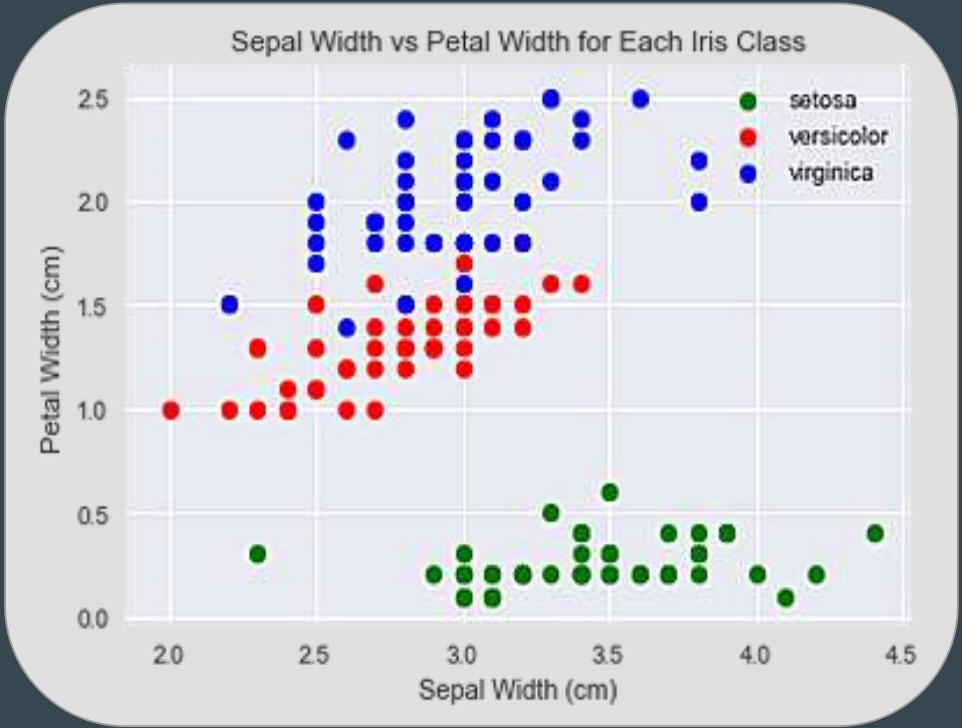
Let’s start the calculations

5 Steps to LDA

1) Means
2) Scatter Matrices
3) Finding Linear Discriminants
4) Subspace
5) Project Data

$$y = W^T * x$$

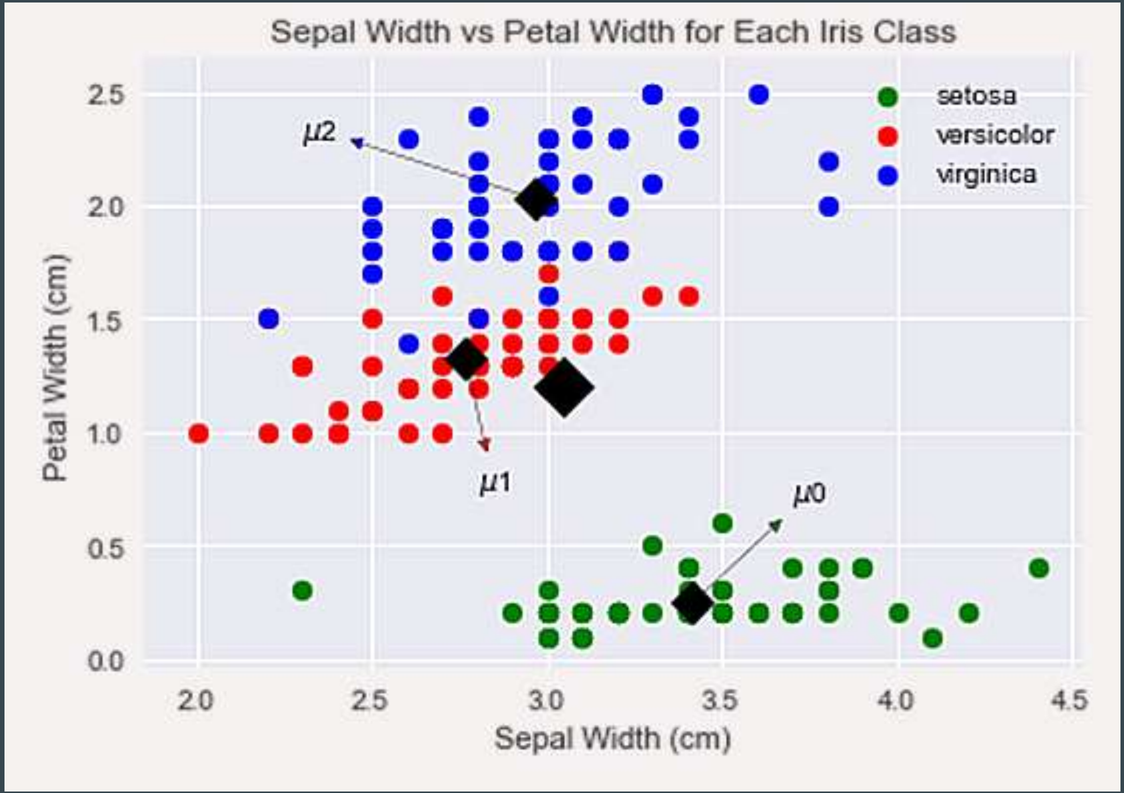
Iris Dataset



Step 1: Means

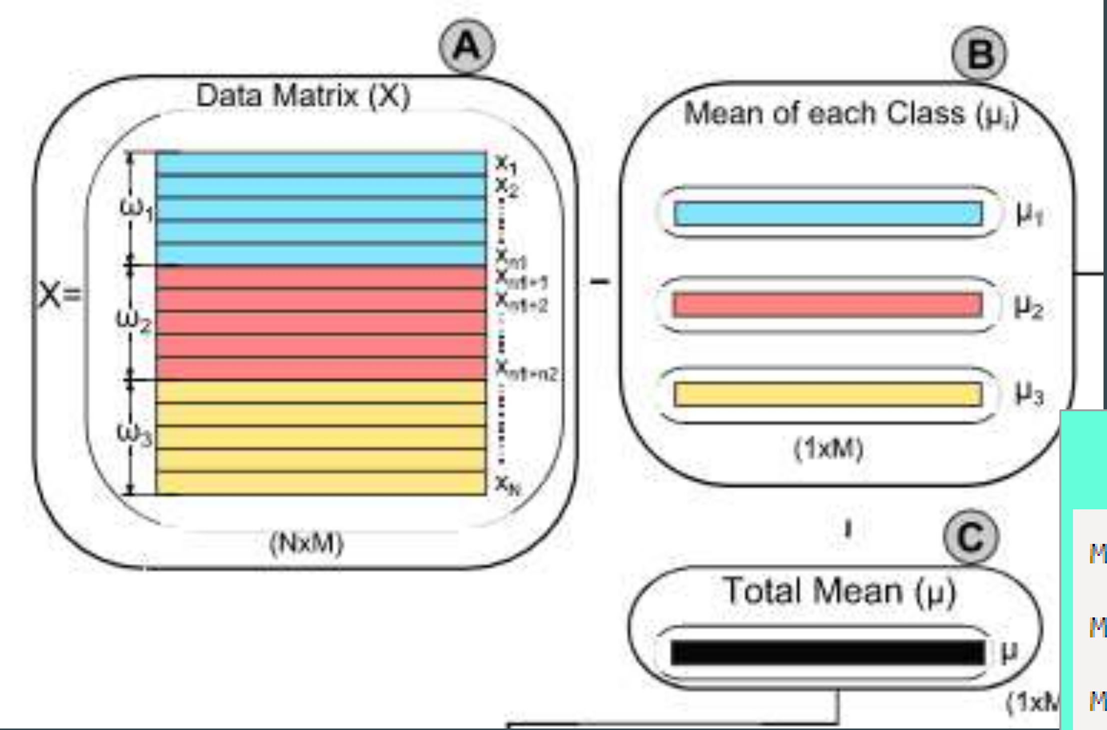
1. Find each class mean

1. Find the overall mean (central point)



Means

Sepal Width	Petal Width
3.504	0.248
3.408	0.213
2.975	0.327
3.225	0.289
2.648	0.222
3.346	0.203
3.198	0.255
2.899	0.301
3.045	0.297



	Sepal Width	Petal Width	
Mean Vector class 0: [	3.418	0.244]	μ0
Mean Vector class 1: [	2.77	1.326]	μ1
Mean Vector class 2: [	2.974	2.026]	μ2

```
The central point (overall mean) is: (3.054, 1.199)
```

5 Steps to LDA

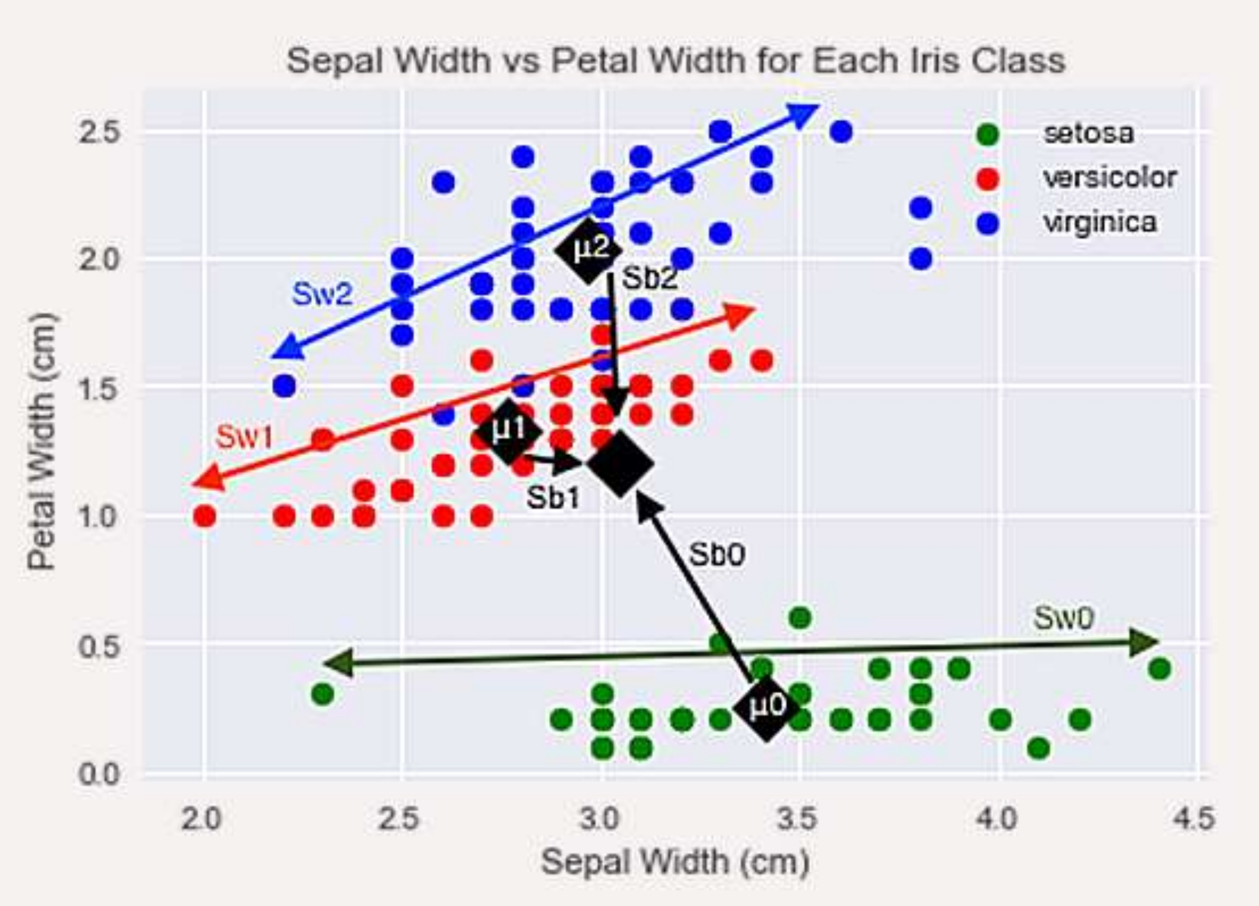
1) Means
2) Scatter Matrices
3) Finding Linear Discriminants
4) Subspace
5) Project Data

Iris Dataset

Step 2 : Scatter Matrices

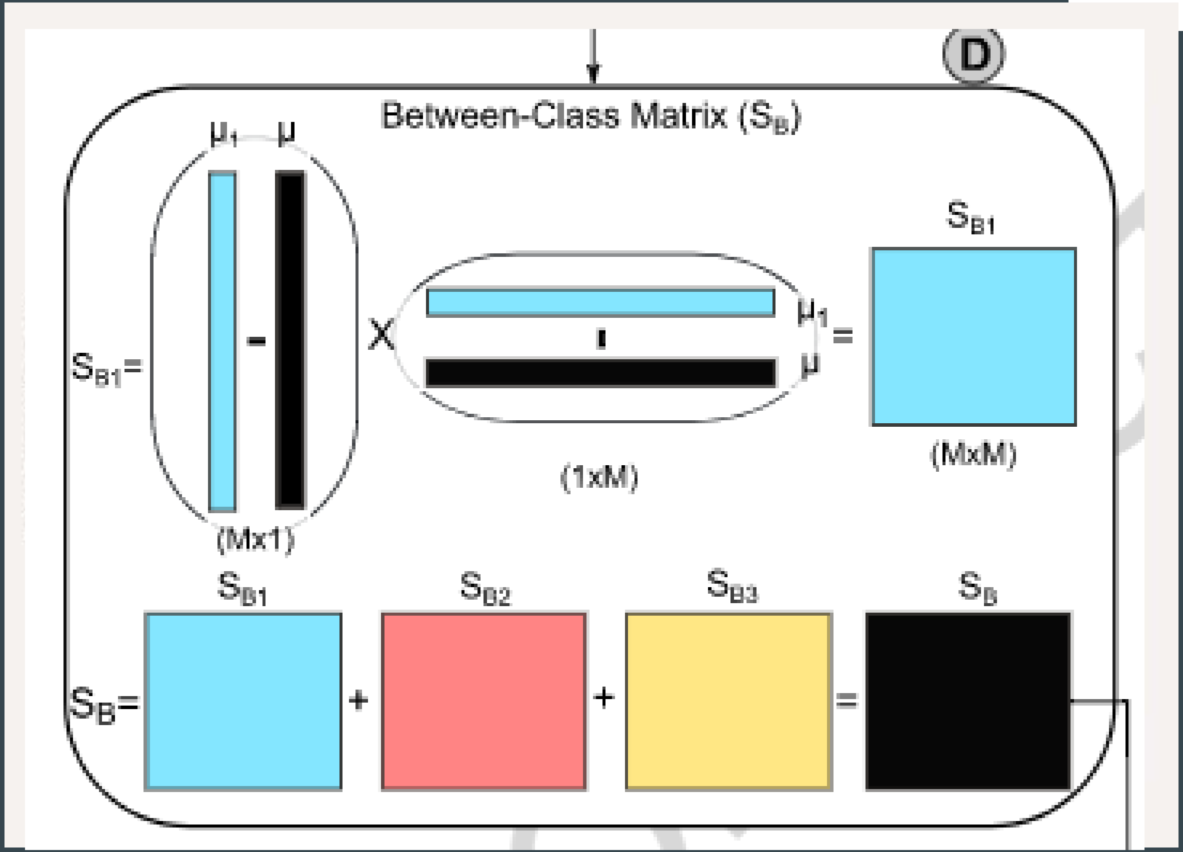
2a) Between-Class Scatter Matrix (Sb)

2b) Within-Class Scatter Matrix (Sw)



2a) Between-Class Scatter Ma

$$S_B = \sum_{i=1}^{c} N_i(\ \mu_i - CP\ )(\ \mu_i - CP\ )^T$$



N_i = Sample Size of Class

Squared Difference

M = # of Features

Between-Class Scatter Matrix

```
# Between-Class Scatter Matrix

# The mean sepal width and mean petal width
# AKA the Central Point
overall_mean = np.mean(X, axis = 0)

# empty scatter matrix
S_B = np.zeros((2,2))

for i, mean_vec in enumerate(mean_vectors):
    n = X[y == i, :].shape[0]
    mean_vec = mean_vec.reshape(2,1)
    overall_mean = overall_mean.reshape(2,1)

    # add up the differences of the class means and the overall mean
    S_B += n * (mean_vec - overall_mean).dot((mean_vec - overall_mean).T)

print('Between-Class Scatter Matrix:\n', S_B)
```

```
Between-Class Scatter Matrix:
 [[ 10.9776 -22.4924]
 [-22.4924  80.6041]]
```

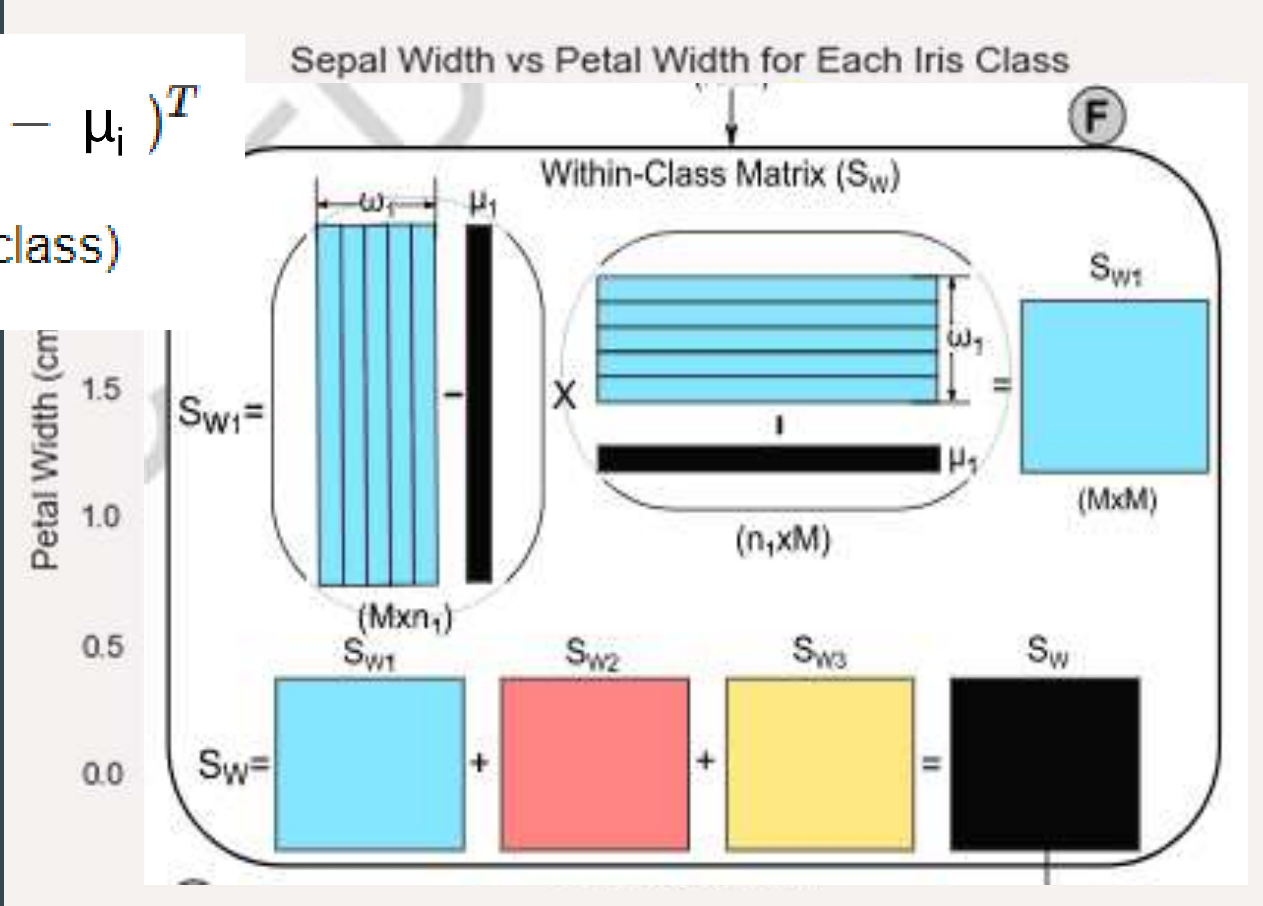
$$S_B = \sum_{i=1}^{c} N_i (\ \mu_i - CP\ )(\ \mu_i - CP\ )^T$$

2b) Within-Class Scatter Matrix

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

(scatter matrix for every class)

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



Within-Class Scatter Matrix

```
# Within-Class Scatter Matrix
# Calculating the covariance

S_W = np.zeros((2,2))
for cl, mv in zip(range(0,3), mean_vectors):
    # empty class covariance matrix
    class_sc_mat = np.zeros((2,2))

    for row in X[y == cl]:
        row, mv = row.reshape(2,1), mv.reshape(2,1)
        # calculates the covariance of the features for within a class
        class_sc_mat += (row - mv).dot((row - mv).T)

    # add up the 3 covariance matrices
    S_W += class_sc_mat

print('Within-Class Scatter matrix:\n', S_W)



Within-Class Scatter matrix:
 [[ 17.035    4.9132]
 [  4.9132   6.1756]]
```

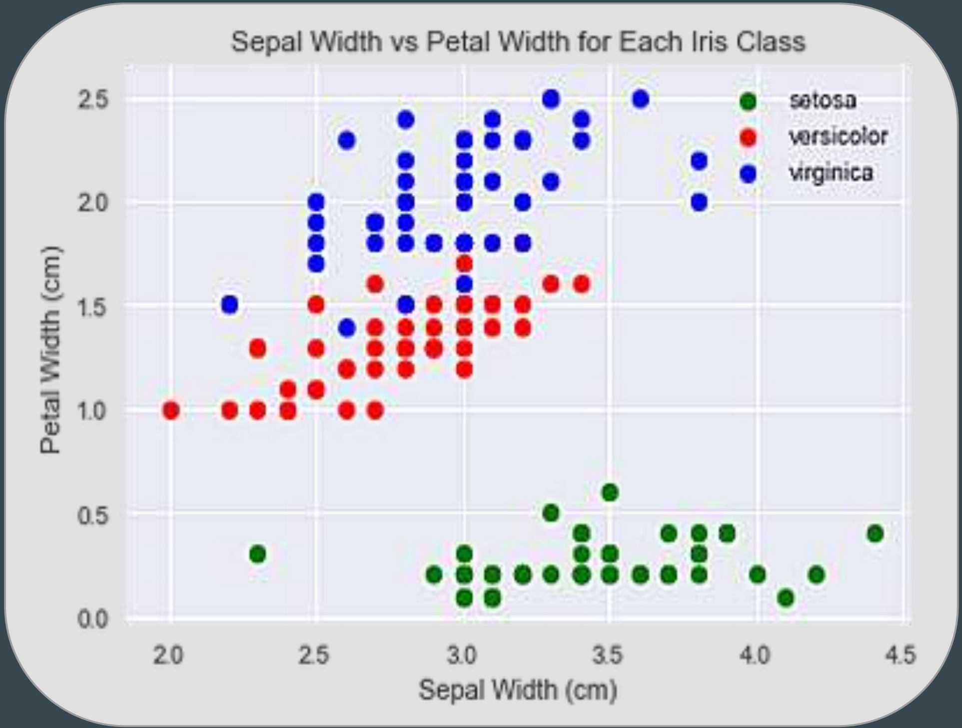
$$S_i = \sum_{\boldsymbol{x} \in D_i}^{n} (\boldsymbol{x} - \mu_i)(\boldsymbol{x} - \mu_i)^T$$

(scatter matrix for every class)

5 Steps to LDA

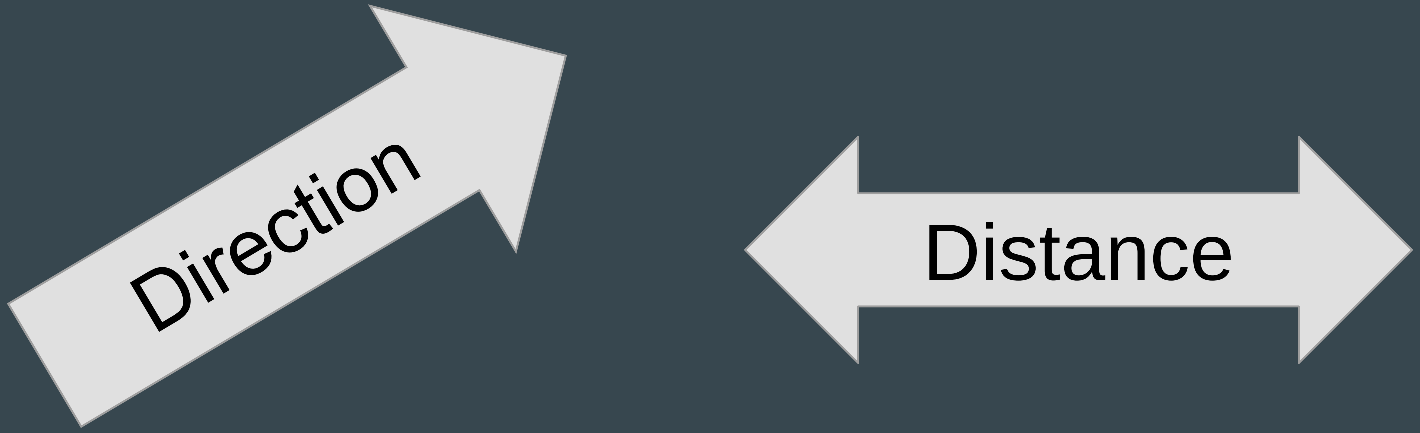
1) Means
2) Scatter Matrices
3) Finding Linear Discriminants
4) Subspace
5) Project Data

Iris Dataset



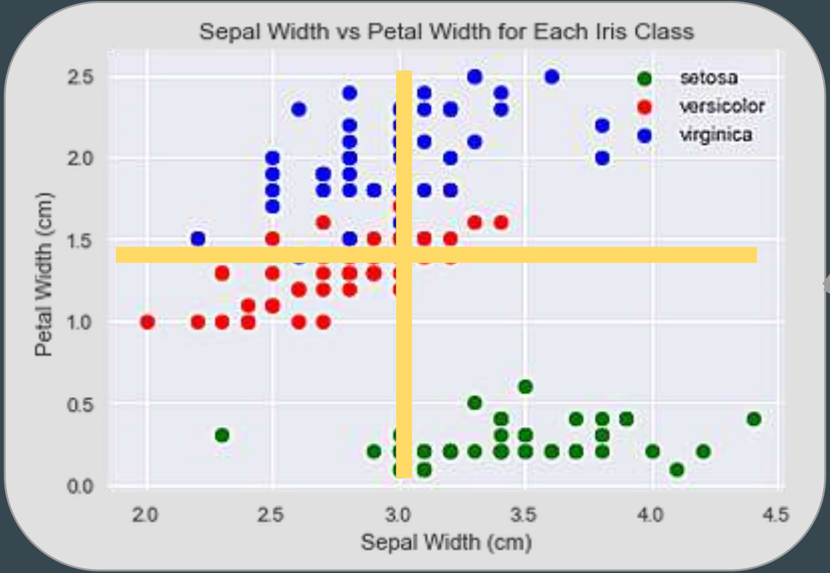
Step 3: Finding Linear Discriminants

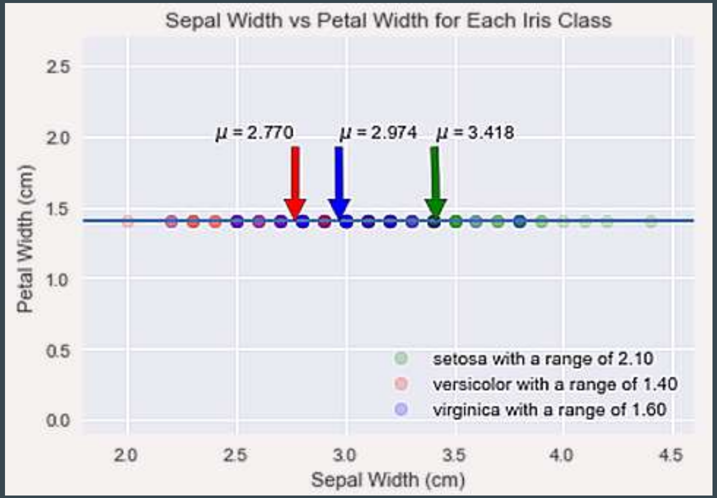
Finding W using eigenvectors and eigenvalues

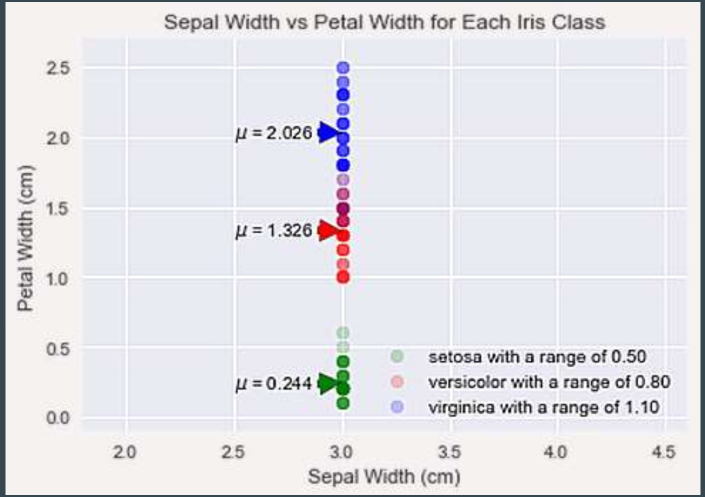


Now, remember…

Iris Dataset







The Math of Finding Linear Discriminants

Working towards:

$$y = W^T * x$$

We have so far:

S_B and S_W

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T\mu_1 - w^T\mu_2)^2 = w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w = w^T S_B w$$

Projected means

$$\tilde{s}_i^2 = \sum_{y\in\omega_i}(y - \tilde{\mu}_i)^2 = \sum_{x\in\omega_i}(w^T x - w^T\mu_i)^2 = $$
$$= \sum_{x\in\omega_i} w^T(x - \mu_i)(x - \mu_i)^T w = w^T S_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_W w$$

Scatter of the projection

http://research.cs.tamu.edu/prism/lectures/pr/pr_l10.pdf

The Math of Linear Discriminants

$$y = W^T * x$$

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

Fisher's Criterion

Derivative & set = 0

$$S_W W = \lambda S_B W$$

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w}$$

=

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

Eigenvector

Matrix

Eigenvalue

Eigenvectors and Eigenvalues using I

In [86]:

```
A  = np.linalg.inv(S_W).dot(S_B)
print('W =\n', W)

eig_vals, eig_vecs = np.linalg.eig(np.linalg.inv(S_W).dot(S_B))

for i in range(len(eig_vals)):
    eigvec_sc = eig_vecs[:,i].reshape(2,1)
    print('\nEigenvector {}: \n{}'.format(i+1, eigvec_sc.real))
    print('Eigenvalue {:}: {:.2e}'.format(i+1, eig_vals[i].real))
```

```
A =
 [[  2.1996  -6.599 ]
 [ -5.3921  18.3021]]

Eigenvector 1:
[[-0.9583]
 [-0.2859]]
Eigenvalue 1: 2.31e-01

Eigenvector 2:
[[ 0.343 ]
 [-0.9393]]
Eigenvalue 2: 2.03e+01
```

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w}$$

$$A v = \lambda v$$

$$|A - \lambda \cdot I| = 0$$

solve for λ

Quadratic Equation

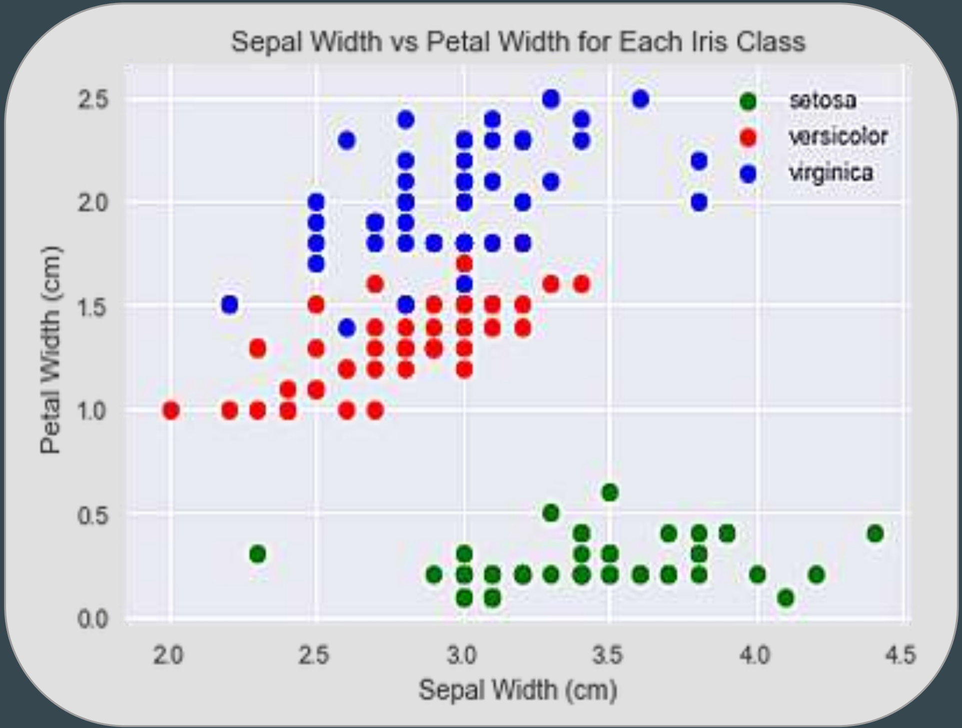
solve for v = W

YouTube: PatrickJMT

5 Steps to LDA

1) Means
2) Scatter Matrices
3) Finding Linear Discriminants
4) Subspace
5) Project Data

Iris Dataset



Step 4: Subspace

$$y = W^T * x$$

- Sort our Eigenvectors by decreasing Eigenvalue

- Choose the top Eigenvectors to make your transformation matrix used to project your data

```
Eigenvalues in decreasing order:


32.2719577997
0.27756686384
5.71450476746e-15
5.71450476746e-15
```

Choose top (Classes - 1) Eigenvalues

5 Steps to LDA

1) Means
2) Scatter Matrices
3) Finding Linear Discriminants
4) Subspace
5) Project Data

Iris Dataset

Step 5: Project Data

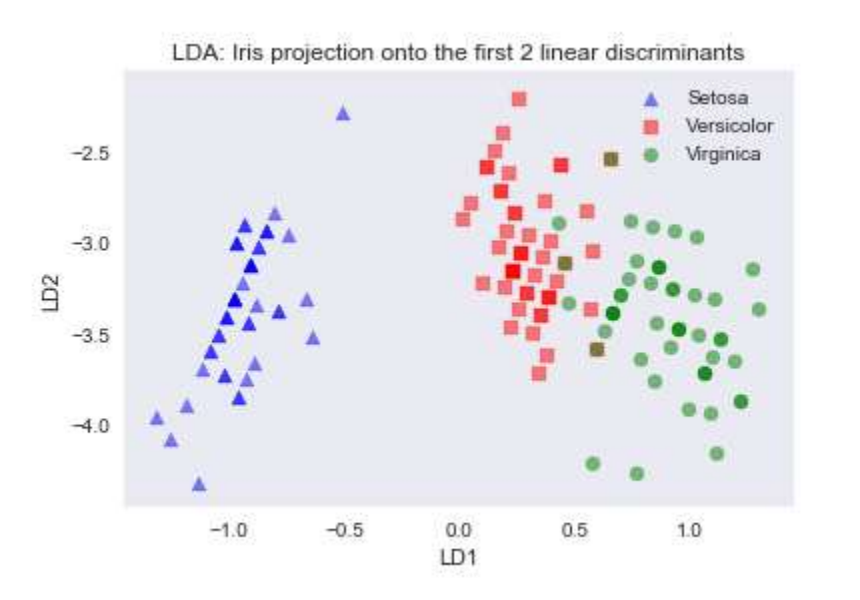
$$y = W^T * x$$

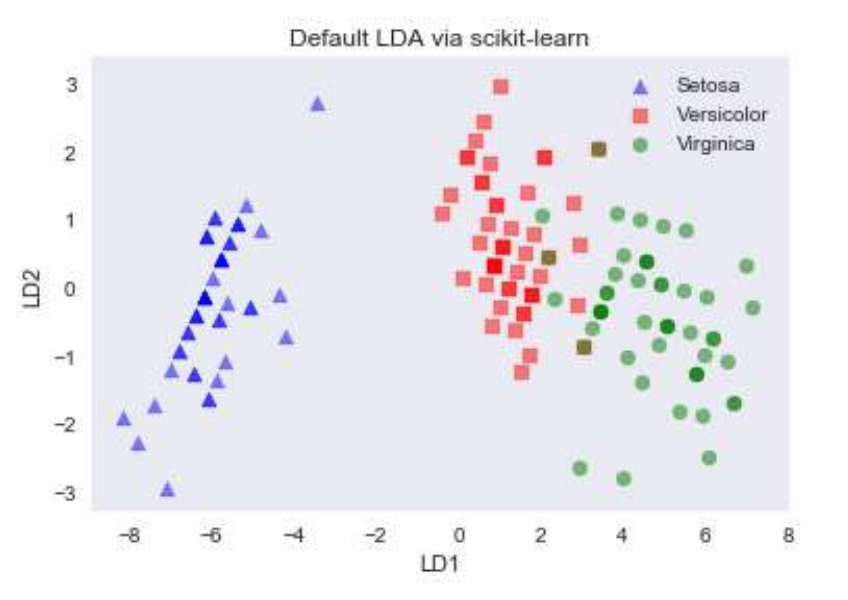
x_

)

https://memegenerator.net/instance/48422270/willy-wonka-well-now-that-was-anticlimactic

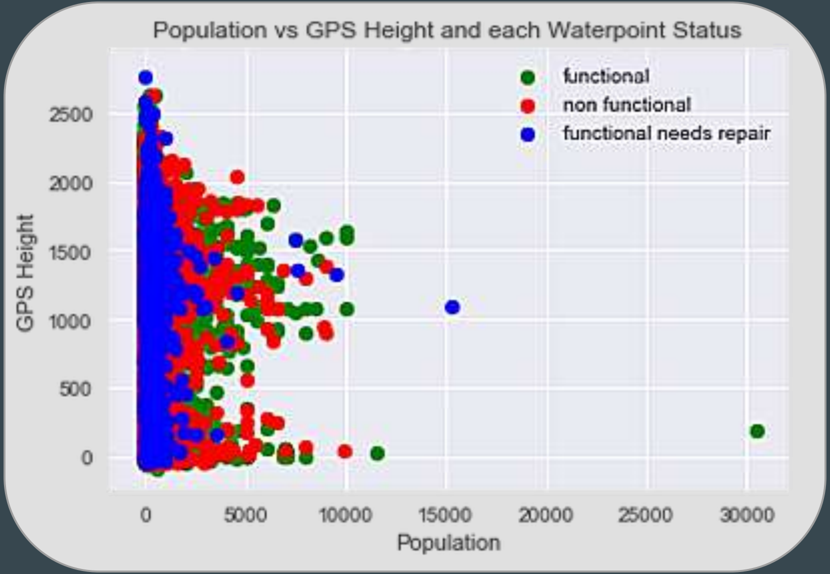
Results and scikit-learn

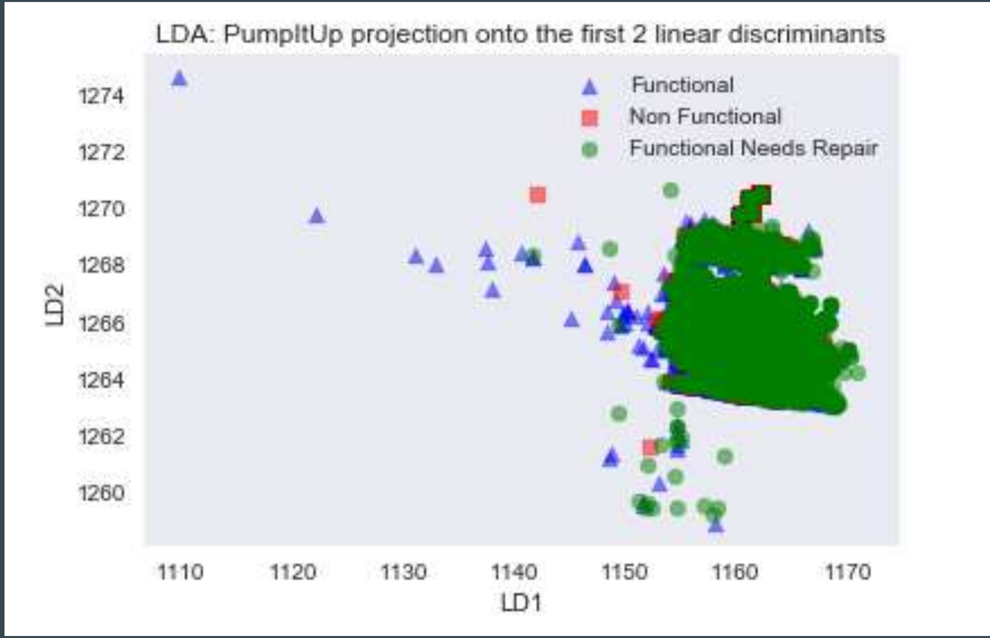




Disclaimer

Your (My) Dataset





Thanks!

www.jaclynkokx.com

jaclynkokx@gmail.com

@JackieKokx