# Chapter 1

# PCA Algorithm

Principal component Analysis.

# Principal Component Analysis

- In ML applications,…
  - data frequently has high dimensionality
- Ex: ***Facial images***
  - $256 \times 256 \sim 256^2 \times 1$ vector

$2^{16}$ pixels!

# Principal Component Analysis

- In ML applications,…
  - data frequently has **very high dimensionality**!
- Ex: ***Facial images***
  - $256 \times 256 \sim 256^2 \times 1$ vector

# Principal Component Analysis

- **High Dimensionality** ⇒ Analysis is Difficult!
  - How to **reduce dimensionality**?
  - Without losing (much) information ?

PCA.

# Principal Component Analysis

- This is termed Dimensionality Reduction
  - Extract relevant features
  - "**Feature Extraction**"

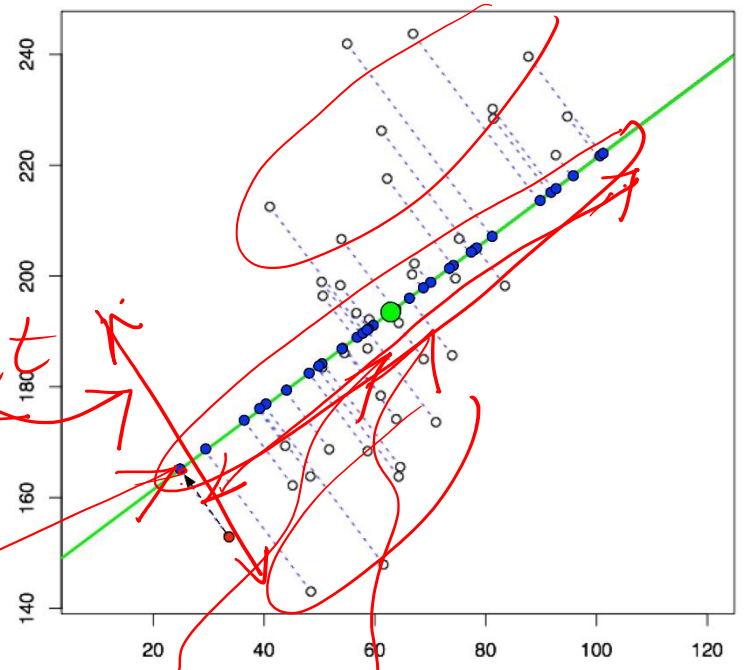
Key steps in ML

# Principal Component Analysis

- This is termed **Dimensionality Reduction**
  - Extract relevant features
  - "**Feature Extraction**"

# Principal Component Analysis

- Data has ***large spread*** along certain directions
  - Principal Directions
  - Data can be **projected** along this direction




Spread is NOT significant

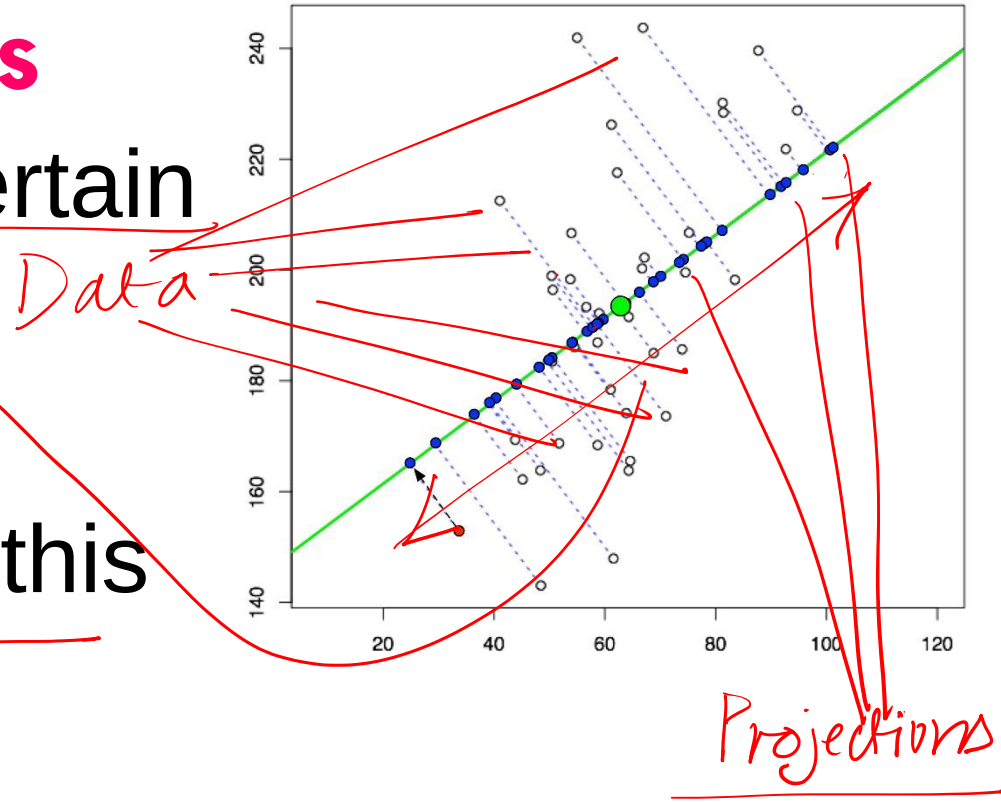
Principal Direction

Maximum spread.

Projections

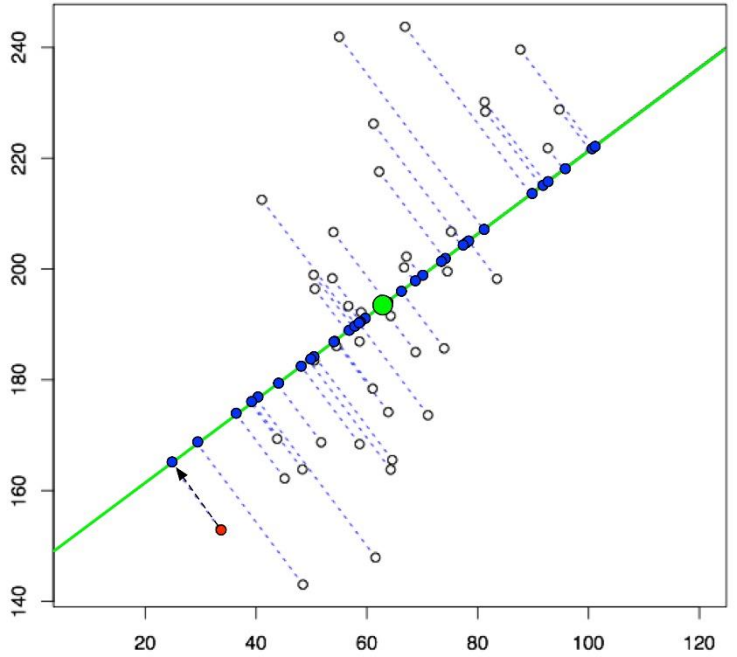
# Principal Component Analysis

- Data has ***large spread*** along certain directions
  - i.e. **Principal directions**
  - Data can be **projected** along this direction



# Principal Component Analysis

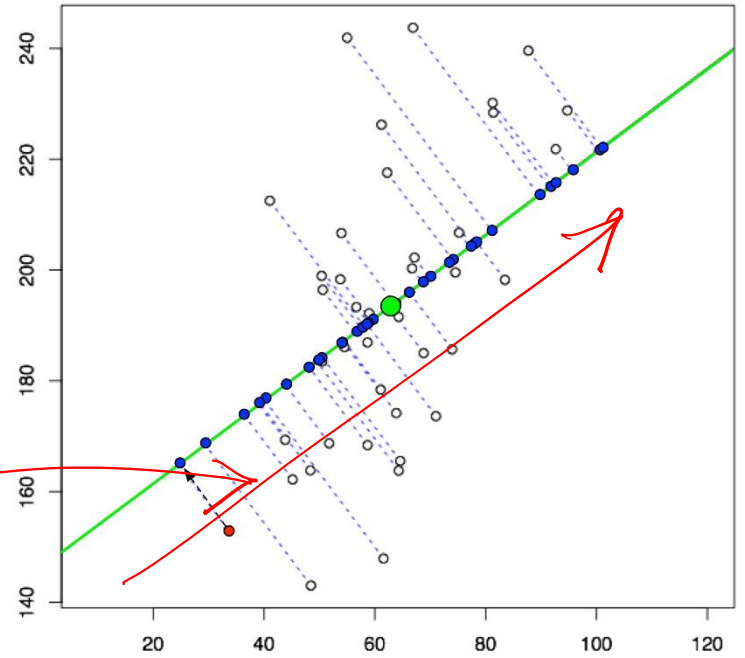
- This **dimensionality reduction**
  - enables **efficient classification and learning**




Low complexity processing

# Principal Component Analysis

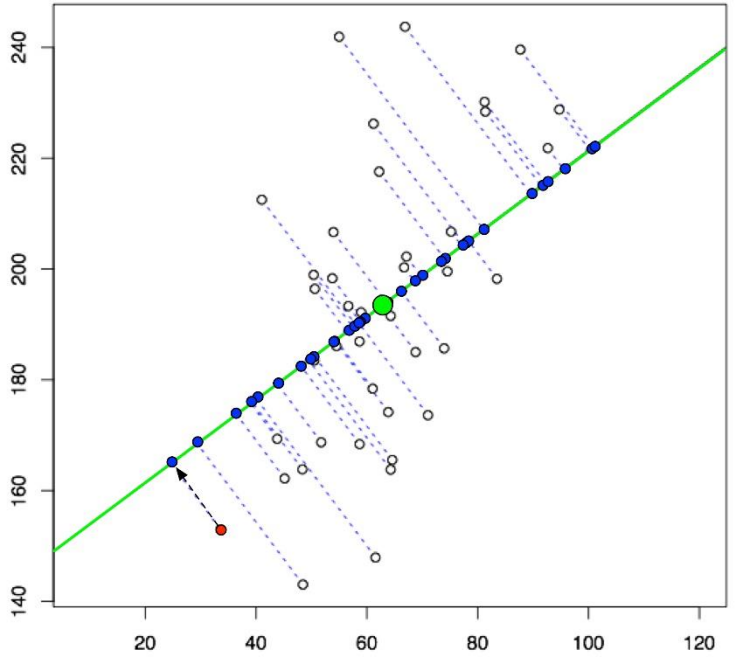
- PCA helps determine these PRINCIPAL DIRECTIONS




Direction of Largest Principal component

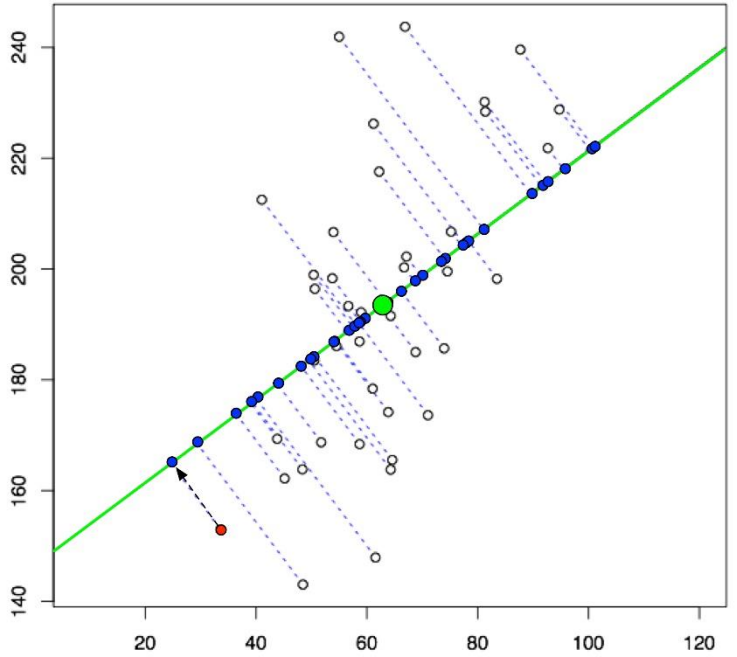
# Principal Component Analysis

- PCA helps determine these **principal directions**



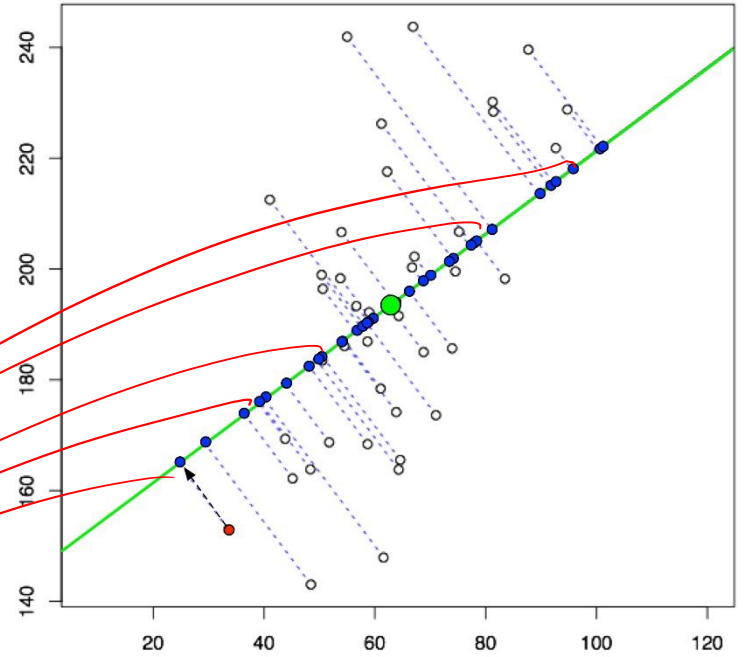
# Principal Component Analysis

- Projection of data along these ***principal directions***
  - Yields the PRINCIPAL COMPONENTS of data



# Principal Component Analysis

- Projection of data along these ***principal directions***
  - Yields the **principal components** of data




Principal components

# PCA Procedure

- The **PCA procedure** can be described as follows
- Consider the data set

$$\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_N$$

N Data points

- Each is of size $M \times 1$

# PCA Procedure

- The **PCA procedure** can be described as follows
- Consider the data set

$$\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_N$$

- Each is of size $M \times 1$

# PCA Procedure

- Estimate the **mean** as follows

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^{N} \tilde{x}_i$$

Sample mean

Sample mean

- ***Subtract the mean*** from each of the ***data vectors*** to obtain

$$\bar{x}_i = \tilde{x}_i - \bar{\mu}$$

centered Data

# PCA Procedure

- Estimate the **mean** as follows

$$\bar{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{x}}_i$$

- ***Subtract the mean*** from each of the ***data vectors*** to obtain

$$\bar{\mathbf{x}}_i = \tilde{\mathbf{x}}_i - \bar{\boldsymbol{\mu}}$$

# PCA Procedure

- The **covariance estimate** is given as

$$R = \frac{1}{N} \sum_{i=1}^{N} \bar{x}_i \bar{x}_i^T$$

# PCA Procedure

- The **covariance estimate** is given as

$$\mathbf{R} = \frac{1}{N-1}\sum_{i=1}^{N} \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T$$

Estimate.

computed

Captures the spread of Data.

Covariance estimate.

Estimate of Data covariance matrix

$R = E\{\bar{x}\bar{x}^T\}$

Since Data is centered.

Definition

# PCA Procedure

- Let $\bar{\mathbf{b}}$ be the direction of the **largest principal component**

# PCA Procedure

- To **maximize spread**…
- choose $\bar{\mathbf{b}} = \bar{\mathbf{v}}_1$, **eigenvector** corresponding to maximum eigenvalue of R

$$R\,\bar{V}_1 = \lambda_{max}\,\bar{V}_1$$

Eigenvector for maximum eigenvalue

maximum Eigenvalue

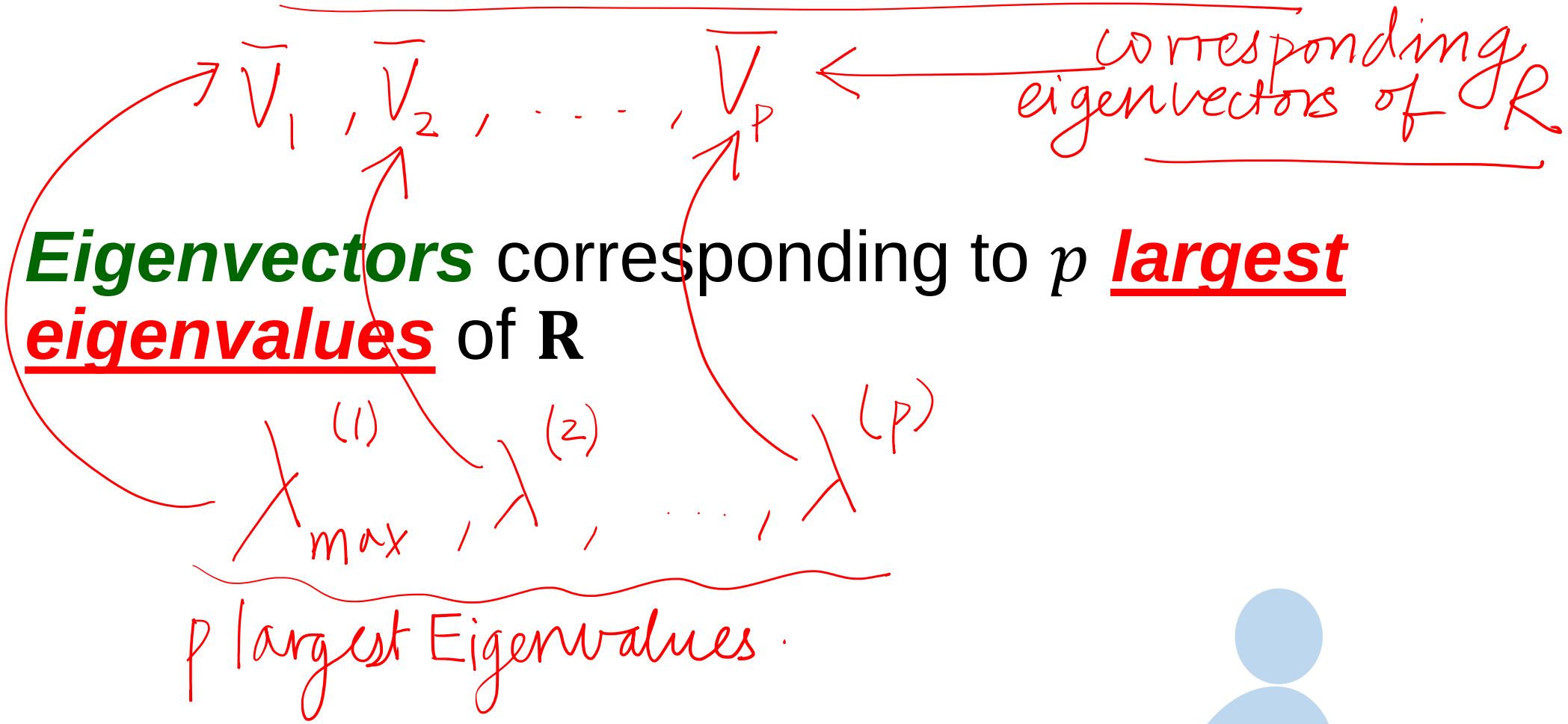
# PCA Procedure

- To **maximize spread**…
- choose $\bar{\mathbf{b}} = \bar{\mathbf{v}}_1$, **eigenvector** corresponding to ***maximum eigenvalue***!

$$\mathbf{R}\bar{\mathbf{v}}_1 = \lambda_{\max}\bar{\mathbf{v}}_1$$

# PCA Procedure

- Thus, the **principal directions** are




- ***Eigenvectors*** corresponding to $p$ ***largest eigenvalues*** of $\mathbf{R}$

# PCA Procedure

- Thus, the **principal directions** are

$$\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \ldots, \bar{\mathbf{v}}_p$$

- ***Eigenvectors*** corresponding to $p$ ***largest eigenvalues*** of $\mathbf{R}$

# PCA Procedure

- **Principal components** can be found as

$$\tilde{\mathbf{V}} = \begin{bmatrix} \bar{V}_1 & \bar{V}_2 & \cdots & \bar{V}_p \end{bmatrix}$$

Principal components

$$\begin{matrix} \bar{V}_1^T \bar{x}_i \\ \bar{V}_2^T \bar{x}_i \\ \vdots \\ \bar{V}_p^T \bar{x}_i \end{matrix}$$

Projections of $\bar{x}_i$ along Principal Directions

$$\check{\mathbf{x}}_i = \begin{bmatrix} \bar{V}_1^T \\ \bar{V}_2^T \\ \vdots \\ \bar{V}_p^T \end{bmatrix} \bar{x}_i$$

Principal components

$p \times 1$ vector

# PCA Procedure

- **Principal components** can be found as

$$\tilde{\mathbf{V}} = [\bar{\mathbf{v}}_1 \quad \bar{\mathbf{v}}_2 \quad \dots \quad \bar{\mathbf{v}}_p]$$

$$\check{\mathbf{x}}_i = \begin{bmatrix} \bar{\mathbf{v}}_1^T \\ \bar{\mathbf{v}}_2^T \\ \vdots \\ \bar{\mathbf{v}}_p^T \end{bmatrix} \bar{\mathbf{x}}_i = \tilde{\mathbf{V}}^T \bar{\mathbf{x}}_i$$

Principal components

# PCA Procedure

- Find the **principal direction** corresponding to

$$\mathbf{R} = \begin{bmatrix} 1 & 2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & -2 \end{bmatrix}$$

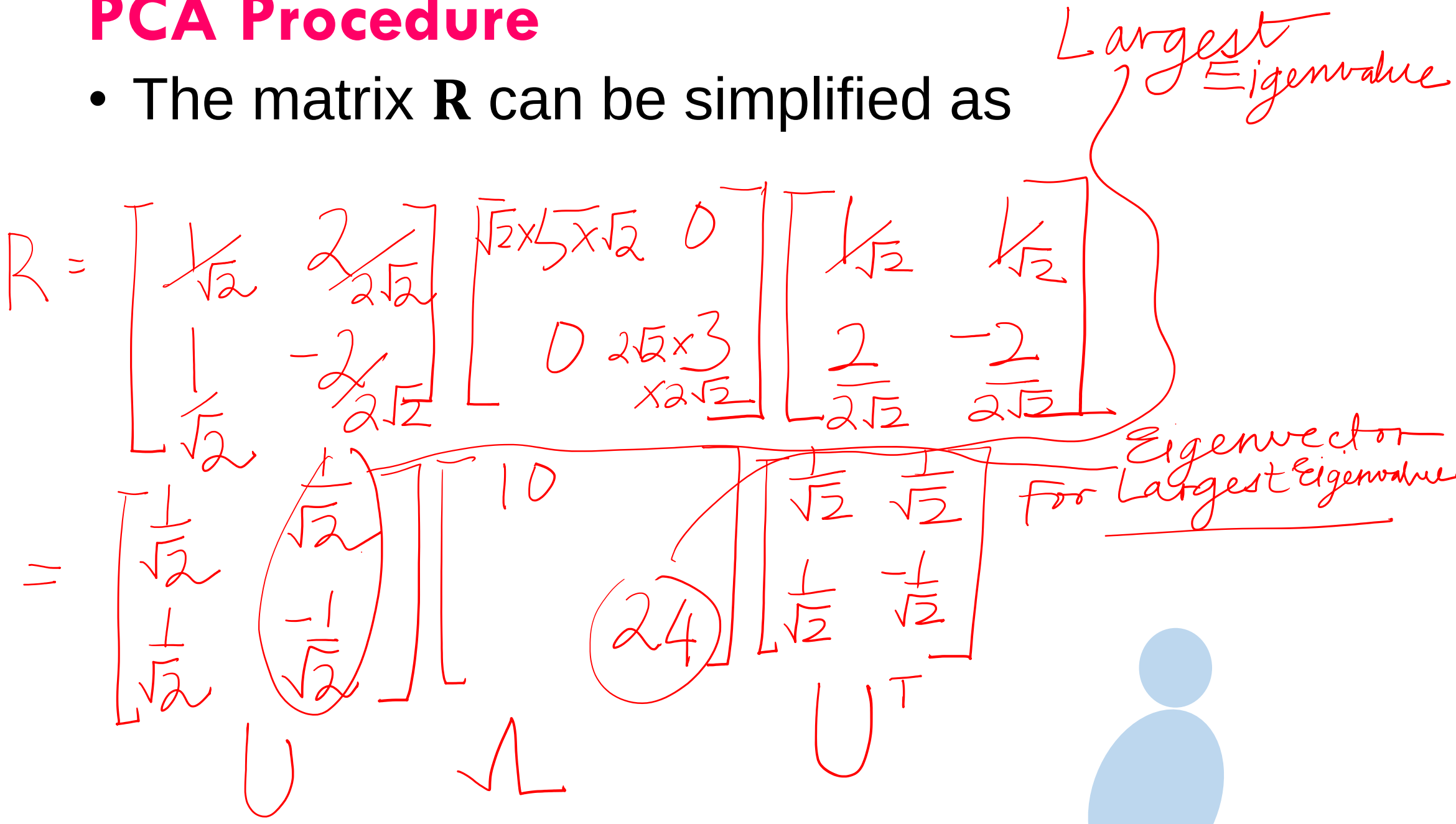
$U \Lambda U^T$

$UU^T = U^T U = I$

$U$ is an orthogonal matrix

# PCA Procedure

- The matrix **R** can be simplified as



# PCA Procedure

Direction of maximum Spread:

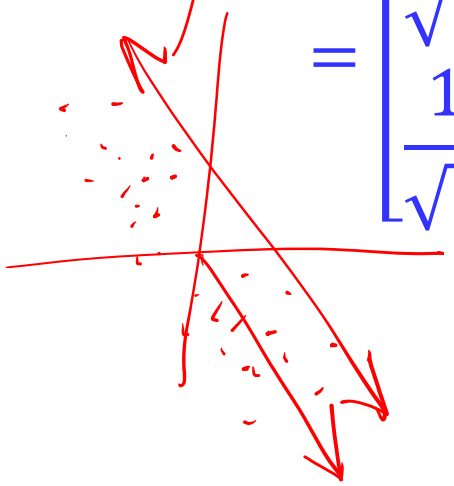
$$\mathbf{R} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{2}{2\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{2}{2\sqrt{2}} \end{bmatrix} \begin{bmatrix} 5 \times \sqrt{2} \times \sqrt{2} & 0 \\ 0 & 3 \times 2\sqrt{2} \times 2\sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{2}{2\sqrt{2}} & -\frac{2}{2\sqrt{2}} \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 10 & 0 \\ 0 & 24 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$ ← Principal Direction

Largest eigenvalue.

$\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \vec{x}$ → Principal component



# SVD and PCA

- We now demonstrate the relation between **SVD (Singular Value Decomposition)** and **PCA**

# SVD and PCA

- Consider now the **data matrix** $\mathbf{X}$ defined as

$$\mathbf{X} = \frac{1}{\sqrt{N-1}} \begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \vdots \\ \bar{x}_N^T \end{bmatrix}$$

Data matrix

Data vectors -

# SVD and PCA

- Consider now the **data matrix** $\mathbf{X}$ defined as

$$\mathbf{X} = \frac{1}{\sqrt{N-1}} \begin{bmatrix} \bar{\mathbf{x}}_1^T \\ \bar{\mathbf{x}}_2^T \\ \vdots \\ \bar{\mathbf{x}}_N^T \end{bmatrix}$$

# SVD and PCA

- Consider now **SVD** of **X** given as

$$\mathbf{X} = U \Sigma V^T$$

$\sigma_i \geq 0$

$\sigma_1 \geq \sigma_2 \geq \cdots$

$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{bmatrix}$

Singular values.

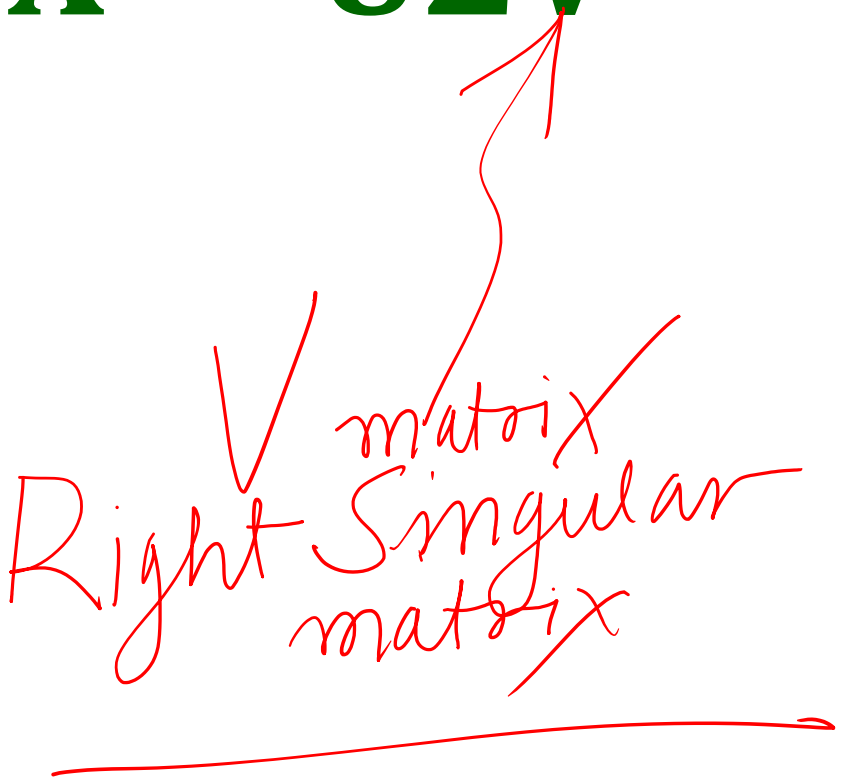
$U U^T = U^T U = I$

$V V^T = V^T V = I$

# SVD and PCA

- Consider now **SVD** of $\mathbf{X}$ given as

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$



# SVD and PCA

- **Eigenvectors** of $\mathbf{R}$ are **right singular vectors** of $\mathbf{X}$
- Hence, the **principal directions** can also be obtained as

$\bar{V}_1, \bar{V}_2, \ldots, \bar{V}_p$ ← columns of $V$ matrix

Right singular vectors corresponding to $p$ largest singular values.

# SVD and PCA

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p$

- **Eigenvectors** of $\mathbf{R}$ are **right singular vectors** of $\mathbf{X}$
- Hence, the **principal directions** can also be obtained as
  - $p$ **dominant right singular vectors** of $\mathbf{X}$ from its **SVD**

$$\mathbf{V} = \begin{bmatrix} | & | & & | & | & \\ \bar{v}_1 & \bar{v}_2 & \cdots & \bar{v}_p & \bar{v}_{p+1} & \cdots \\ | & | & & | & | & \end{bmatrix}$$

$\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \ldots, \bar{\mathbf{v}}_p$

Principal Directions

```python
from sklearn import datasets
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.mixture import GaussianMixture
from sklearn.metrics import confusion_matrix

```

SCIKIT-LEARN
Python functions
for ML

Subdibrary
Decomposition

Inbuilt
DATASETS
For plotting

PCA module
Principal Component
Analysis -

```
1  from sklearn import datasets
2  from sklearn.decomposition import PCA
3  import matplotlib.pyplot as plt
4  from sklearn.mixture import GaussianMixture
5  from sklearn.metrics import confusion_matrix
6
```

Metrics Sublibrary

confusion matrix

To characterize performance of ML Algorithm

Gaussian mixture Module For classification

```
7
8   irisset = datasets.load_iris()
9   X = irisset.data
10  Y = irisset.target
11
```

DATA

For verification
Labels

Data

Labeled

IRIS
One of most Basic
& Popular datasets for ML

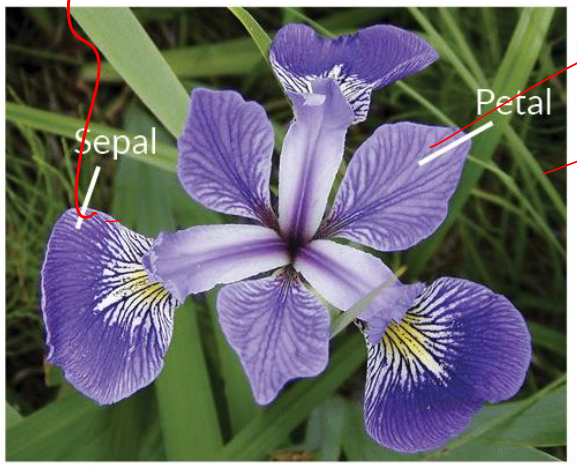
# SVD and PCA

- This data sets consists of 3 different types of irises'
  - Versicolor
  - Setosa
  - Virginica

Sepal

Petal

IRIS Flowers



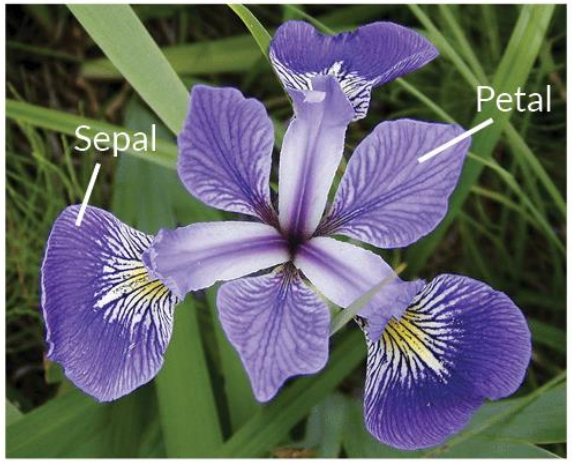

Iris Versicolor

Iris Setosa

Iris Virginica

# SVD and PCA

- This data sets consists of 3 different types of irises’
  - **Setosa**
  - **Versicolour**
  - **Virginica**




**Iris Versicolor**

**Iris Setosa**

**Iris Virginica**

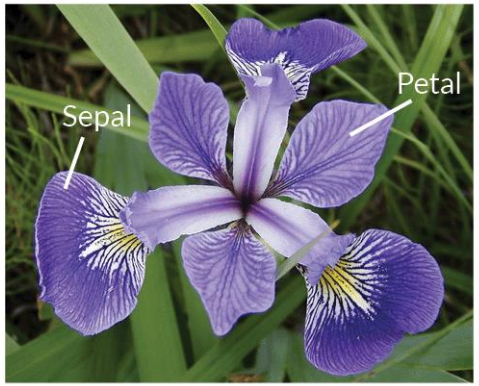
# SVD and PCA

- The features are
  - Sepal Length
  - Sepal width
  - Petal Length
  - Petal width

Features
Attributes

4 Features
use PCA to compress
to 2 Features.

- Stored in a 150x4 numpy.ndarray



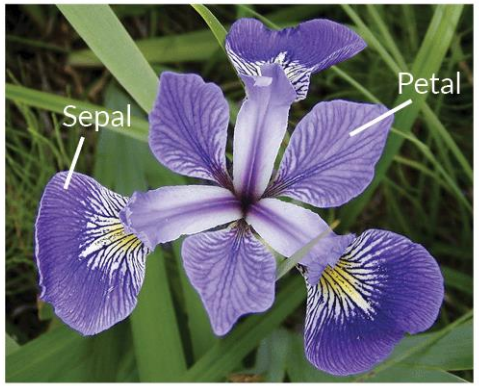
Iris Versicolor


Iris Setosa


Iris Virginica

# SVD and PCA

- The features are
  - **Sepal Length**
  - **Sepal Width**
  - **Petal Length**
  - **Petal Width.**
- Stored in a 150x4 numpy.ndarray



Iris Versicolor


Iris Setosa


Iris Virginica

```
11
12  pca = PCA(n_components=2);
13  Xp = pca.fit(X).transform(X)
14
```

Fit and Transform Data

Projections to determine Principal components

number of principal components = 2

2

```
14
15  plt.figure(1)
16  plt.scatter(Xp[:, 0], Xp[:, 1], c=Y, cmap='jet',s=10)
17  plt.suptitle('Original Clusters')
18  plt.grid(1,which='both')
19  plt.axis('tight')
20  plt.show()
21
```

1st Figure

2D Scatter plot With original labels.

True labels = colors

Grid

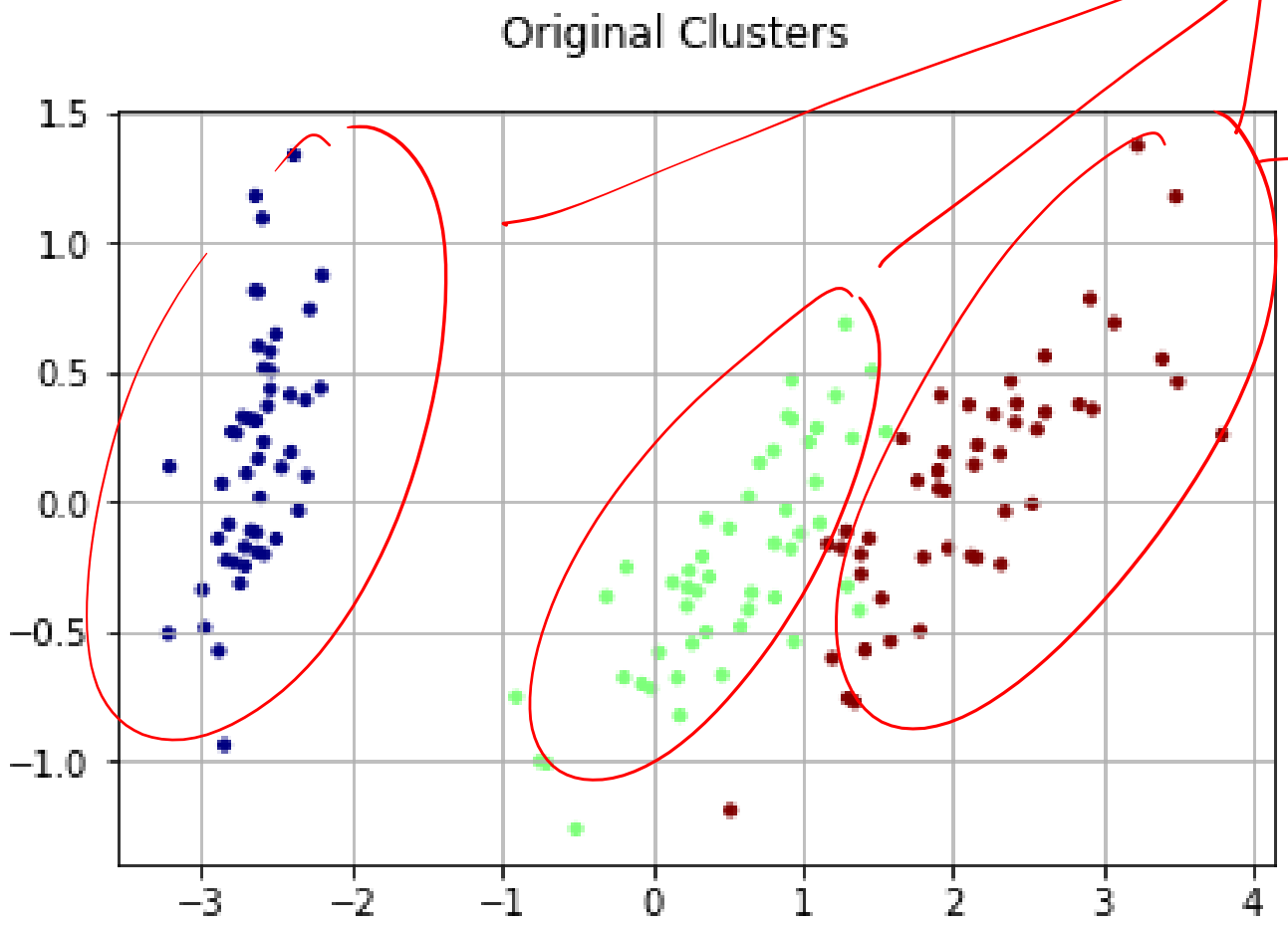
Removes Extra Space

By shortening axis

Remove extra space in plot

Original clusters with true labels.

Show




3 Original clusters.

True Labels. From data.

```
21
22  GMM = GaussianMixture(n_components = 3)
23  GMM.fit(Xp)
24  y_predG = GMM.predict(Xp)
25
```

Gaussian mixture model

For probabilistic clustering

Fit GMM to given data
PCA compressed data

Unsupervised
No labels!

Predict labels.

```
21
22  GMM = GaussianMixture(n_components = 3)
23  GMM.fit(Xp)
24  y_predG = GMM.predict(Xp)
25
```

Number of components = 3.

Estimates GMM parameters.

$\mu, \Sigma$

Predicted Labels.

```
26  plt.figure(2)
27  plt.scatter(Xp[:, 0], Xp[:, 1], c=y_predG, cmap='jet',s=10)
28  plt.suptitle('GMM Clusters')
29  plt.grid(1,which='both')
30  plt.axis('tight')
31  plt.show()
```

grid

2nd Figure

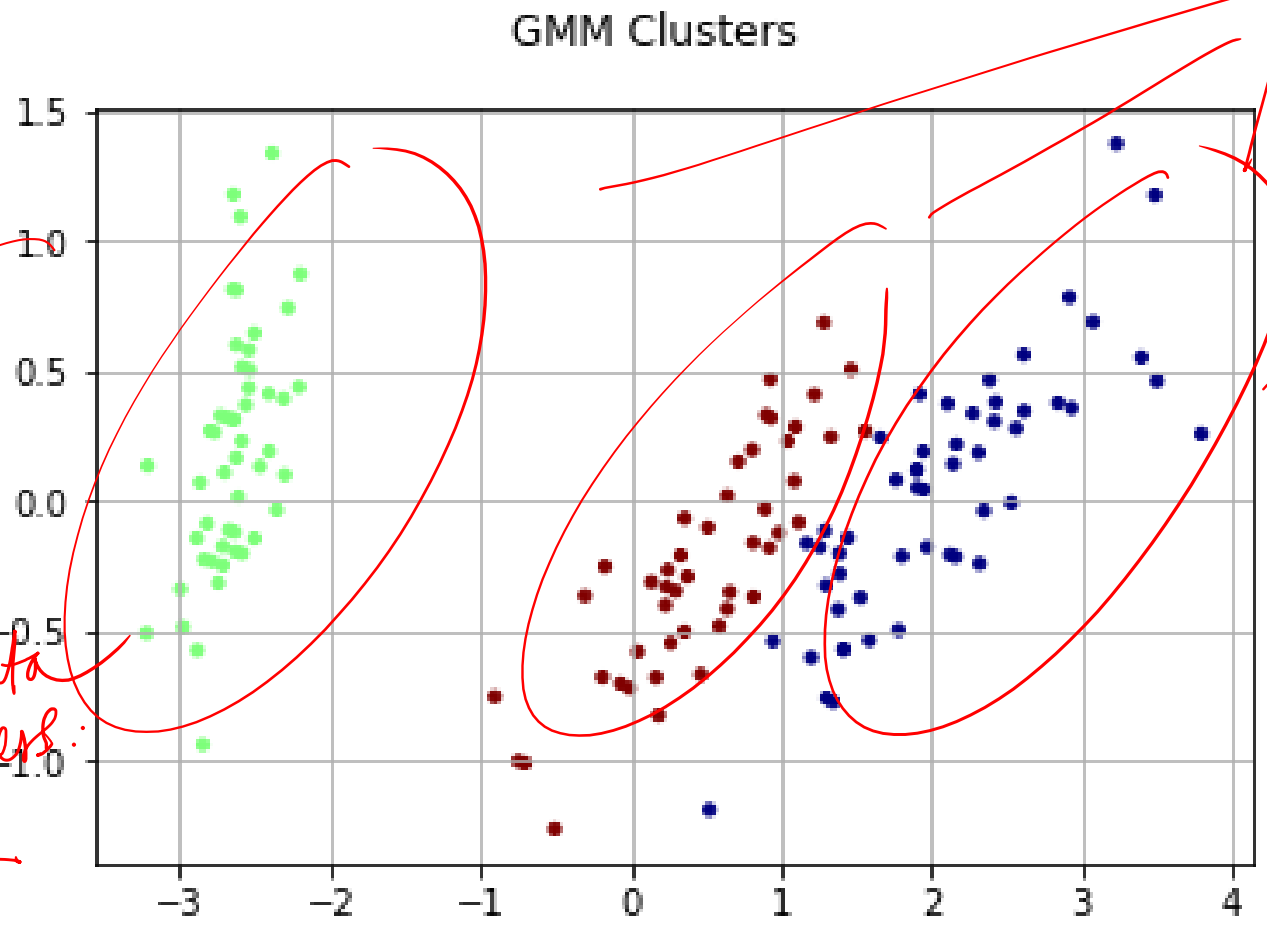
2D Scatter plot

Predicted Labels.

Title.

Show plot

axis tight
To shorten axes
remove extra space




GMM works with PCA compressed data TO yield clusters.

Clusters From GMM

Closely match original clusters.

```
32
33  cmat = confusion_matrix(Y, y_predG)
```

```
In [3]: cmat
Out[3]:
array([[ 0, 50,  0],
       [ 3,  0, 47],
       [50,  0,  0]], dtype=int64)

In [4]:
```

confusion matrix

No misclassified

cluster

No misclassification

how many points in each cluster misclassified.

3 have been misclassified

Instructors may use this white area (14.5 cm / 25.4 cm) for the text. Three options provided below for the font size.

Font: Avenir (Book), Size: 32, Colour: Dark Grey

Font: Avenir (Book), Size: 28, Colour: Dark Grey

Font: Avenir (Book), Size: 24, Colour: Dark Grey

Do not use the space below.