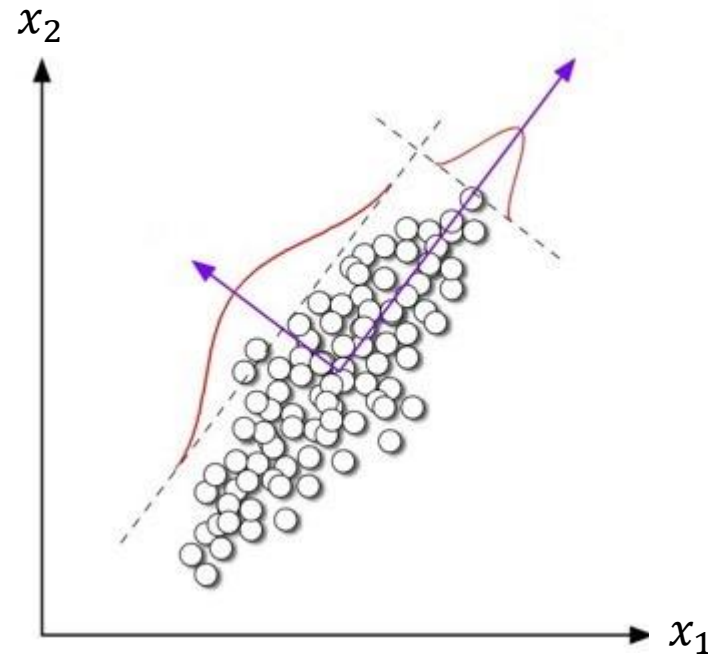


Class -23

Principal Component Analysis (PCA)

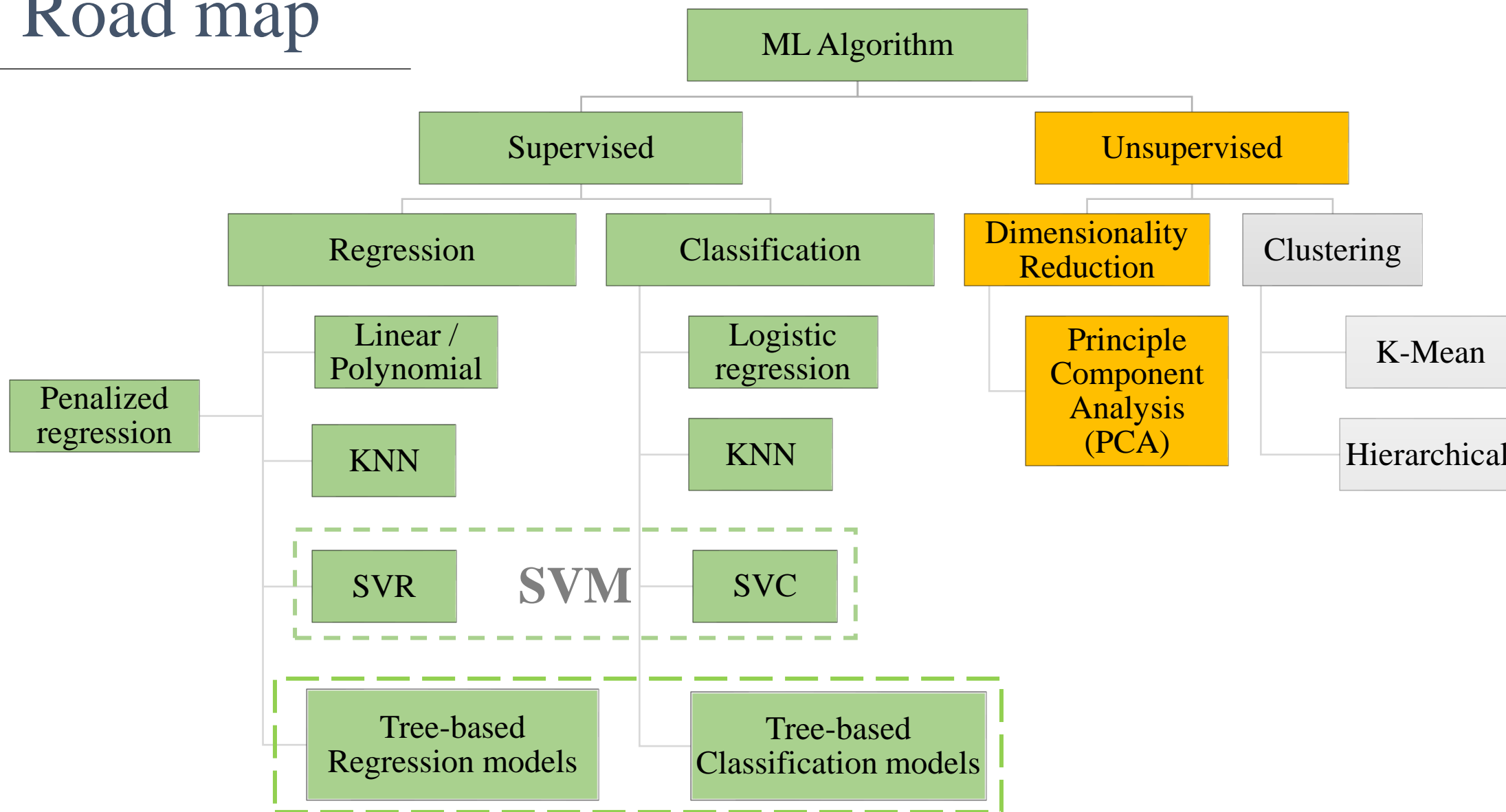


Prof. Pedram Jahangiry





Road map





Topics

Part I

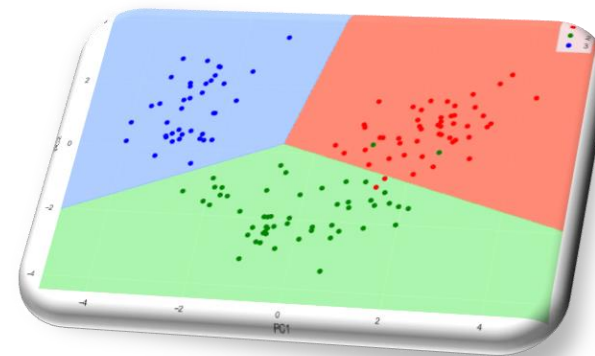
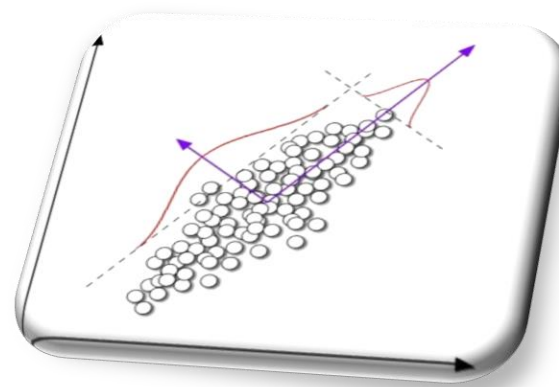
1. Unsupervised Machine Learning
2. Principal Component terminology

Part II

1. Principal Components Analysis (PC)
2. Scree plot

Part III

1. Why PCA? Pros and Cons!
2. Applications of PCA

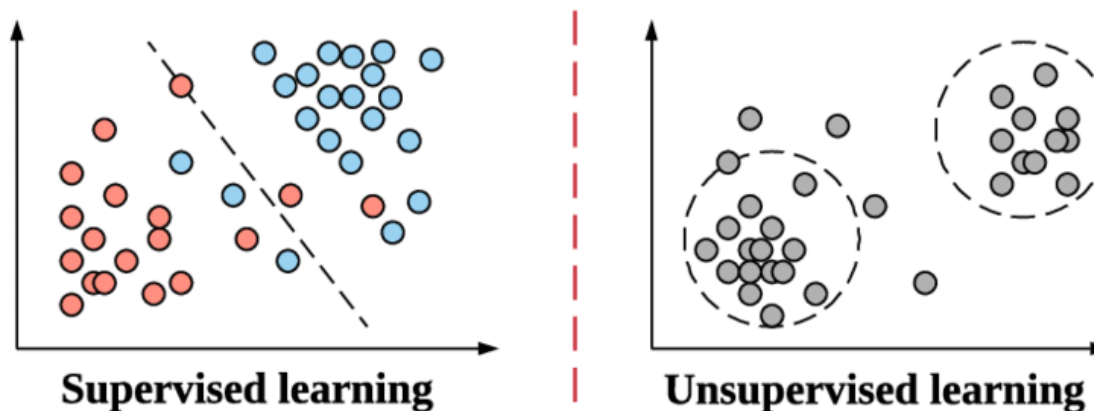


Part I

1. Unsupervised Machine Learning
2. PC terminology

→ Unsupervised Learning

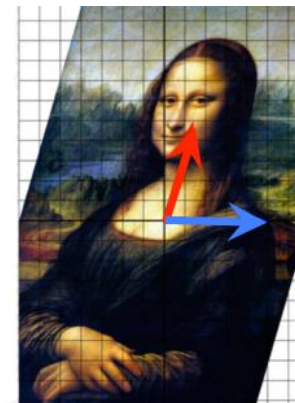
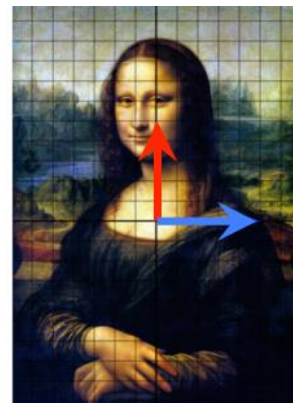
- **Unsupervised learning** is a machine learning technique that does not use labeled data (no target variable)
- The **goal** is to discover the underlying patterns and find groups of samples that behave similarly.
- The two main types of unsupervised learning algorithms are:
 - 1) **Dimension reduction algorithm**
 - Principal Component Analysis
 - 2) **Clustering techniques**
 - K-Mean
 - Hierarchical





Eigen things!

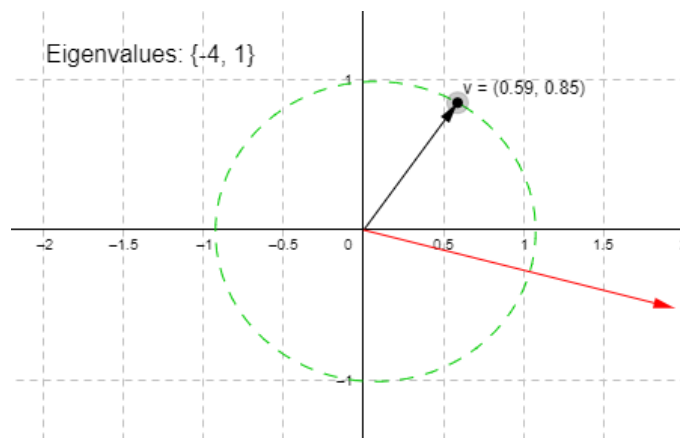
- **Eigenvector** does not change direction in a transformation.
- In this mapping, the **blue arrow** is eigenvector (why?)
- Its **eigenvalue** = 1 (why?)
- For a square matrix A, an Eigenvector and Eigenvalue are defined as follow:



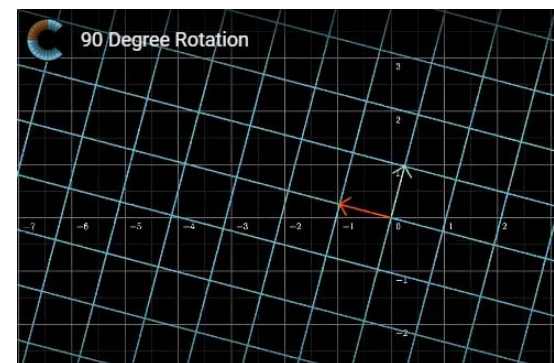
Transformation
matrix Eigenvalue

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

Eigenvector



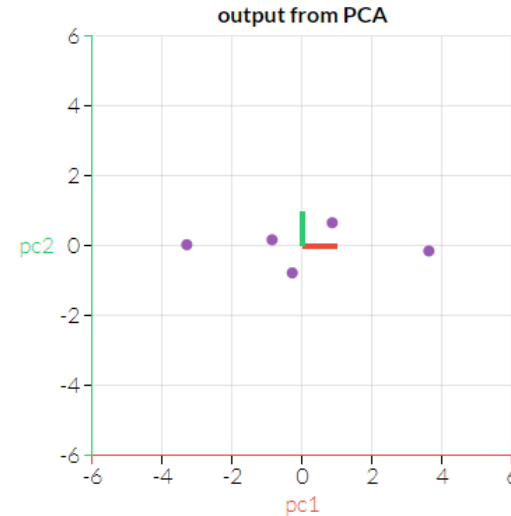
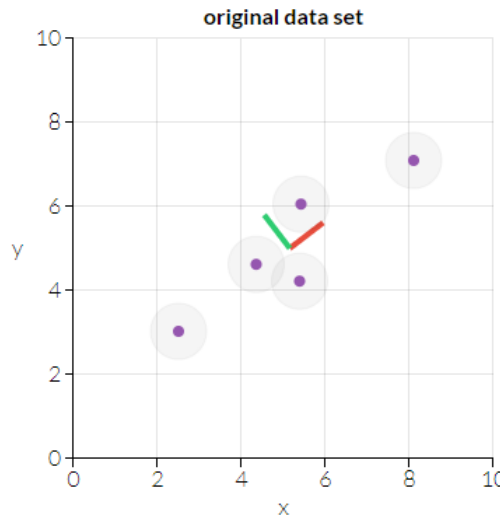
What is a transformation matrix?





PC terminology (connecting PC and Eigenthings)

- The **eigenvectors** and **eigenvalues** of a **covariance matrix** represent the “core” of a PCA
- Consider the following data points:



If you want to reduce the dimension of the data to 1, which PC would you drop?



- We want to find a **direction**(s) in the data set that **explain the most variation**. So, our transformation matrix will be the **covariance matrix**.
- The **principal components** (**eigenvectors** of the covariance matrix) determine the **directions** of the new feature space, and the **eigenvalues** determine their **magnitude**. In other words, the eigenvalues explain the variance of the data **along the new feature axes**.

→ PC terminology (PC definition)

In summary:

- **Principal components** are vectors that define a **new coordinate system** in which the first axis goes in the direction of the highest variance in the data.
- The second axis is orthogonal to the first one and goes in the direction of the second highest variance in the data.

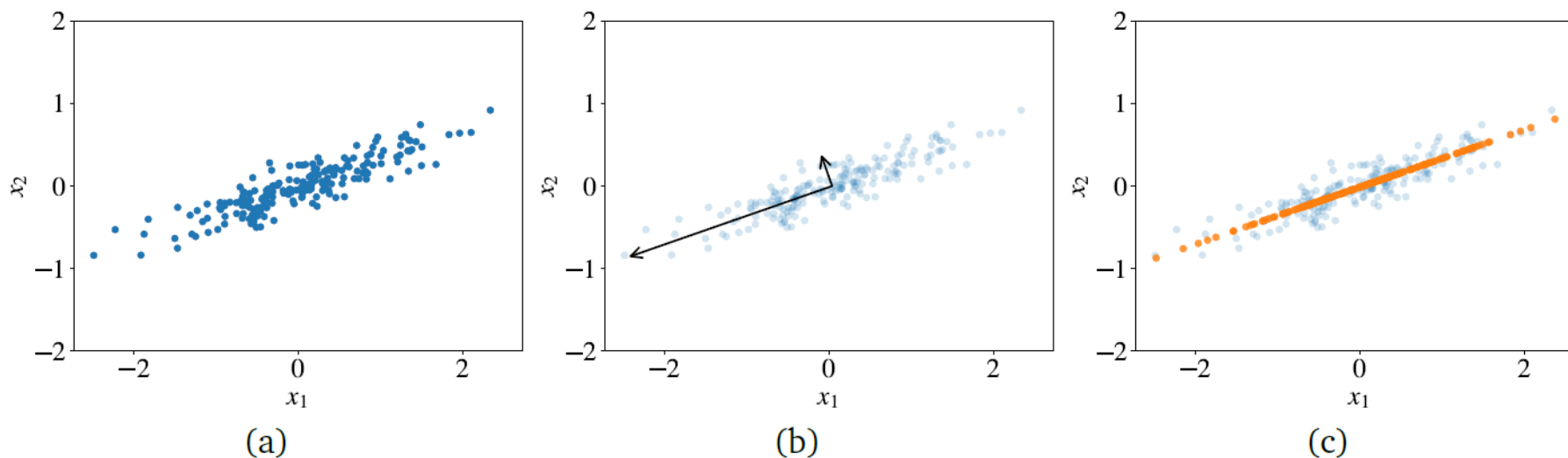
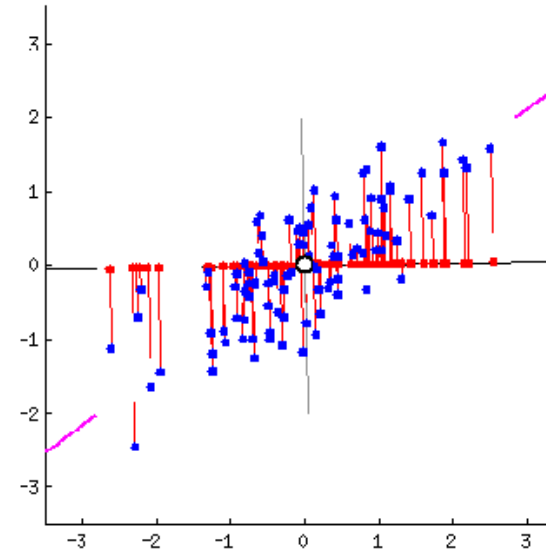
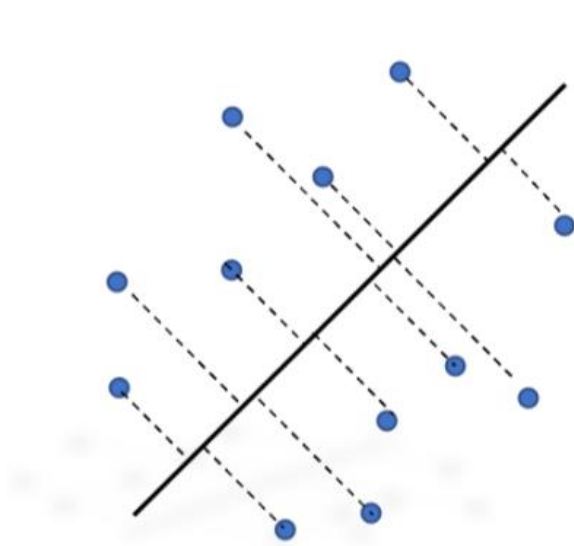


Figure 9.7: PCA: (a) the original data; (b) two principal components displayed as vectors; (c) the data projected on the first principal component. | [Source: The hundred-page machine learning book](#)

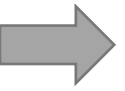
→ PC terminology (the objective function)

- **Projection errors:** The perpendicular distance (Euclidian) between the data point and a Principal Component.
- **Spread:** Variation of the data along Principal component.
- In PCA the goal is to **minimize the projection errors** (or equivalently maximize the spreads)



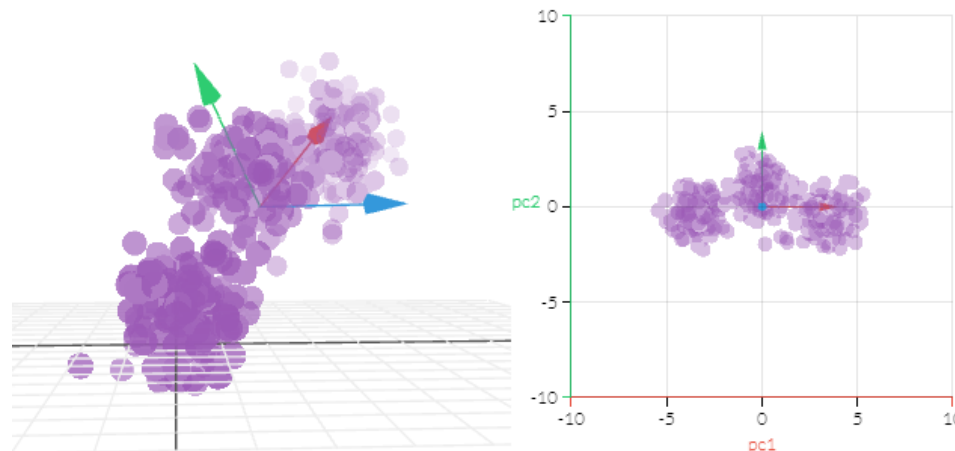
Part II

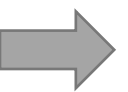
1. Principal Component Analysis (PCA)
2. Proportion Variance Explained
3. Scree plot



Principal Component Analysis (PCA)

- **Dimension reduction** aims to represent a dataset with many typically correlated features by a smaller set of features that still does well in describing the data.
- When **many features** in a dataset, **visualizing** the data or **fitting models** to the data may become extremely complex and “noisy”.
- **Principal components analysis (PCA)** is used to summarize or transform highly correlated features of data into a few main, uncorrelated **components**.
- The PCA algorithm **orders the eigenvectors from highest to lowest** according to their **usefulness** in explaining the **total variance** in the initial data (i.e., eigenvalues)





PCA details

- The PC1 of a set of features X_1, X_2, \dots, X_p is the **normalized** linear combination of the features that has **the largest variance**.

$$PC_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p \quad \text{where } \sum_{j=1}^p \phi_{j1}^2 = 1$$

- The elements $\phi_{11}, \dots, \phi_{p1}$ are referred to as **loadings** of PC1.
- Note that the X features are **standardized (why?)**
- The loading vector ϕ_1 defines a direction in feature space along which the data vary the most i.e., maximizing the variance in that direction!

$$\underset{\phi_{11}, \dots, \phi_{p1}}{\text{maximize}} \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \quad \text{subject to } \sum_{j=1}^p \phi_{j1}^2 = 1$$

USA arrests data: Biplot

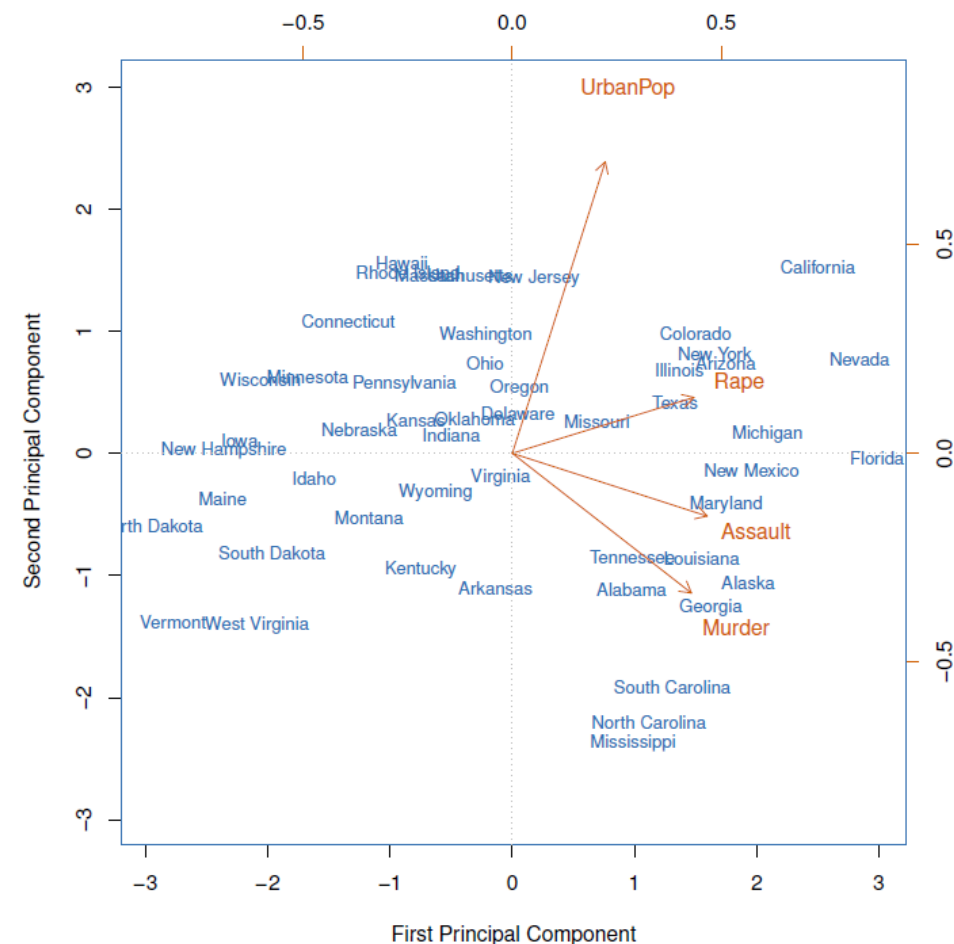
- USA arrests data contains the number of arrests per 100k residents for each of the 50 states.
- The **features** are murder, assault, rape and urban population.
- PCA was performed after **standardizing** each feature! The loadings are as follow:

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

- Biplot** displays both the PC **scores** and PC **loadings**.

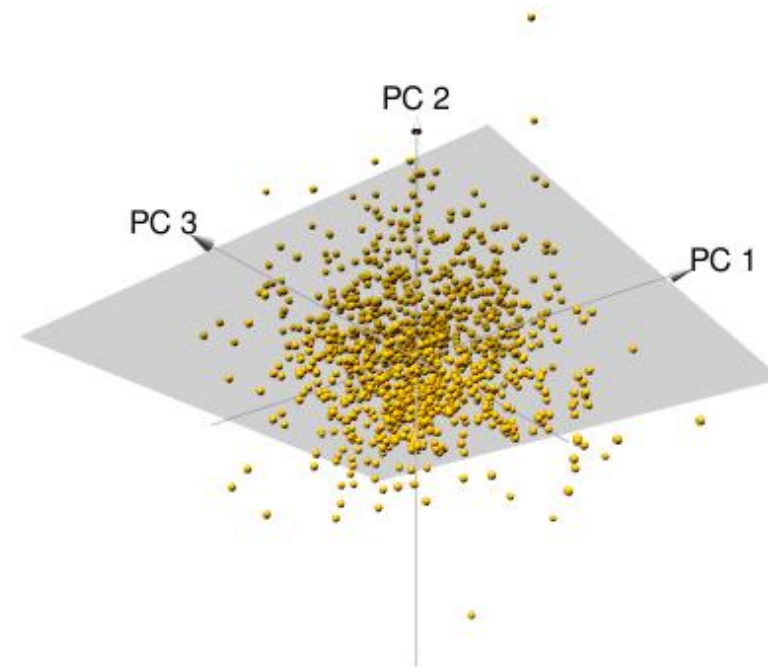
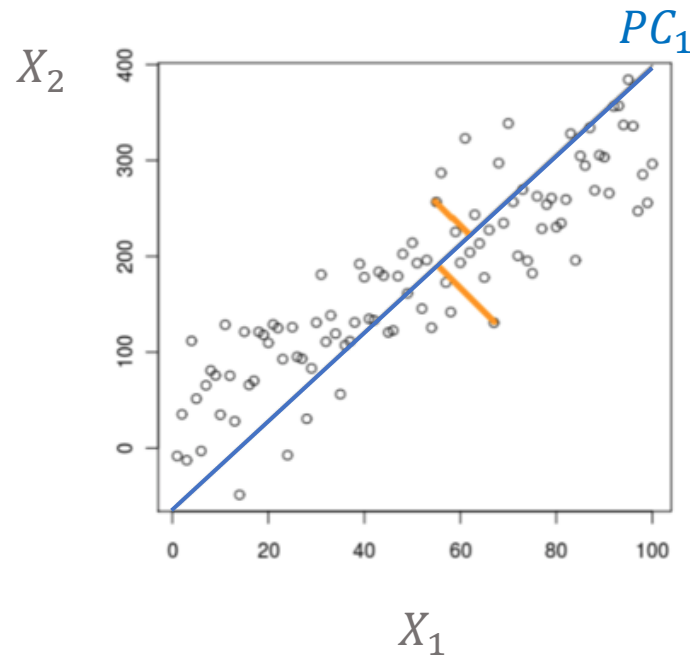
Source: ISLR first edition

	Murder	Assault	UrbanPop	Rape
California	0.2782682	1.262814	1.7589234	2.067820
Florida	1.7476714	1.970778	0.9989801	1.138967
New Hampshire	-1.3059321	-1.365049	-0.6590781	-1.252564



➔ Another interpretation of PCA

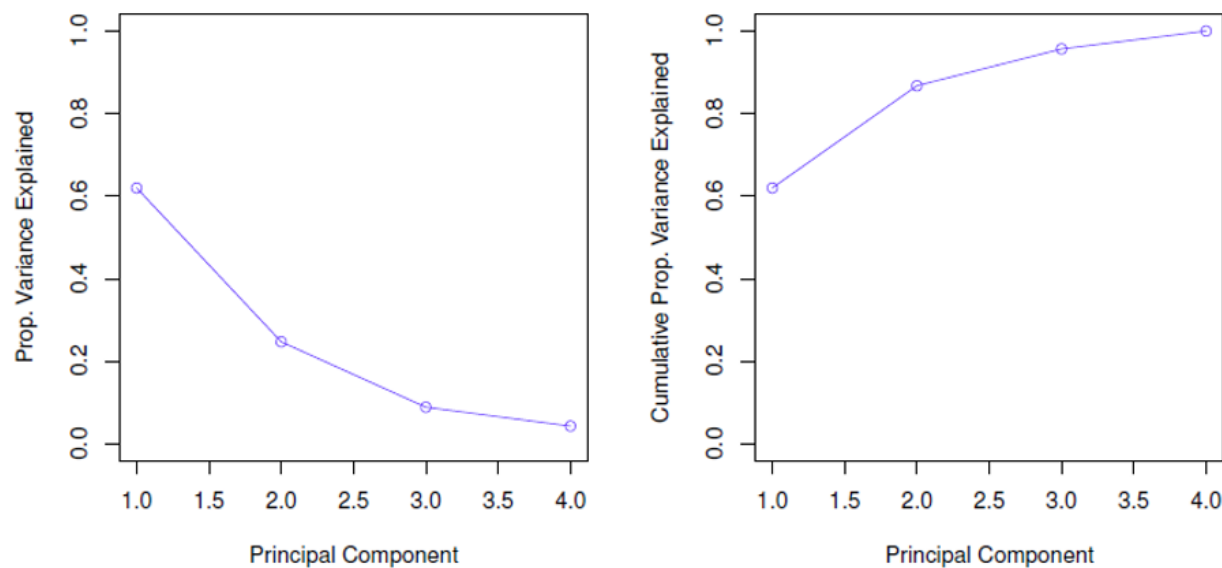
- PCA find the **hyperplane closest** to the observations!
- What is the difference between **PCA** and linear regression then?





Scree plot

- **Scree plot** shows the proportion of total variance in the data explained by each principal component. This is also called **Proportion Variance Explained (PVE)**
- The **PVEs sum to one**. Sometimes they are displayed as **cumulative PVEs**.



- What is the **optimal number of PCs** here? Why cannot we use **cross validation**?

Part III

Why PCA? Pros and cons!

Applications of PCA



PCA's Pros and Cons

Pros:

- Reducing the number of features to the **most relevant predictors** is very useful in general.
- Dimension reduction **facilitates** the data visualization in two or three dimensions.
- **Before** training another supervised or unsupervised learning model, it can be performed as part of EDA to **identify patterns** and detect **correlations**.
- Machine learning models are **quicker to train**, tend to reduce overfitting (by avoiding the curse of dimensionality), and are easier to interpret if provided with lower-dimensional datasets.

Cons:

- Hard to interpret!

Applications of PCA (Example from CFA II reading 7)

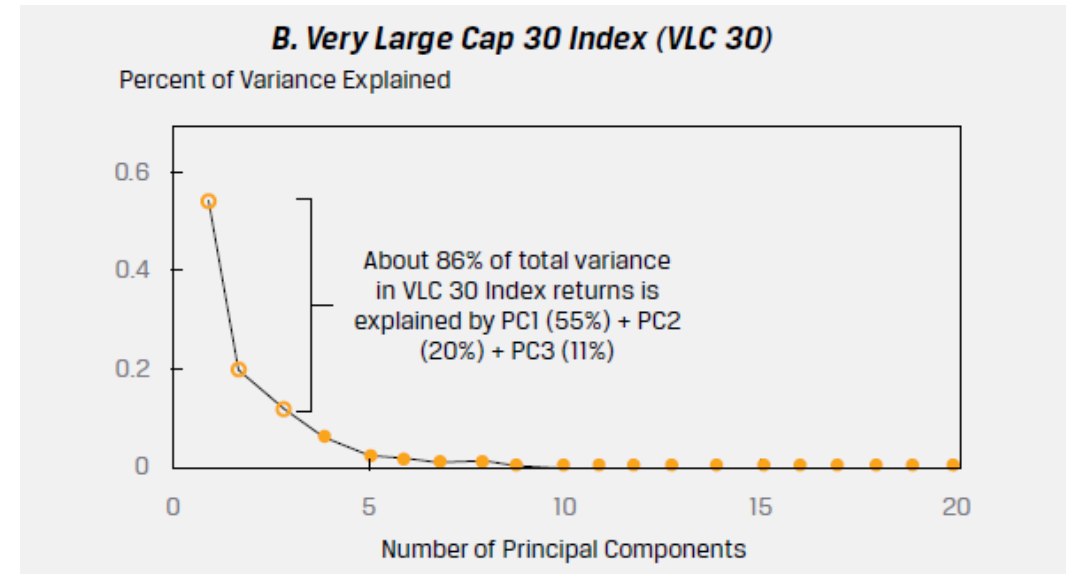
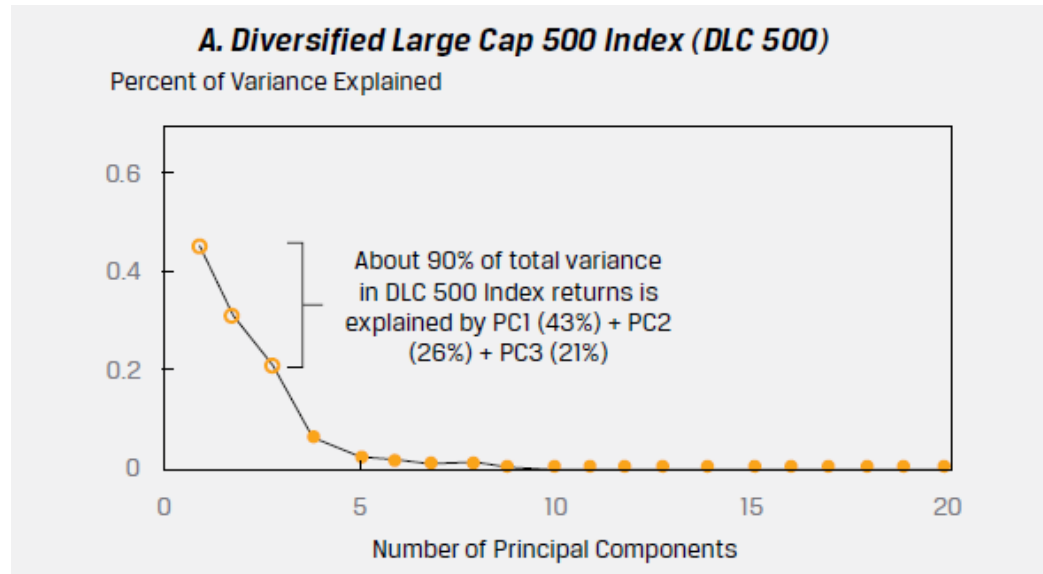
- Consider a hypothetical Diversified Large Cap (**DLC**) 500 and Very Large Cap (**VLC**) 30:
 - DLC 500 can be thought of as a diversified index of **500 large-cap companies** covering all economic sectors
 - VLC 30 is a more concentrated index of the **30 largest publicly traded companies**.
- The dataset consists of index prices and more than **2,000** fundamental and technical **features**
- **Multi-collinearity** among the features is **inevitable!**
- To mitigate the problem, PCA can be used to capture the information and **variance** in the data.





Applications of PCA (Example from CFA II reading 7)

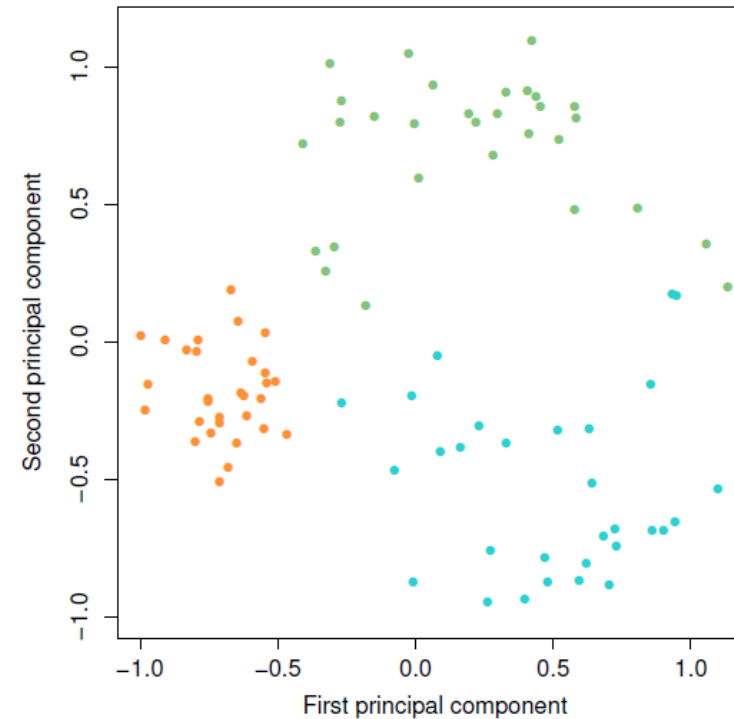
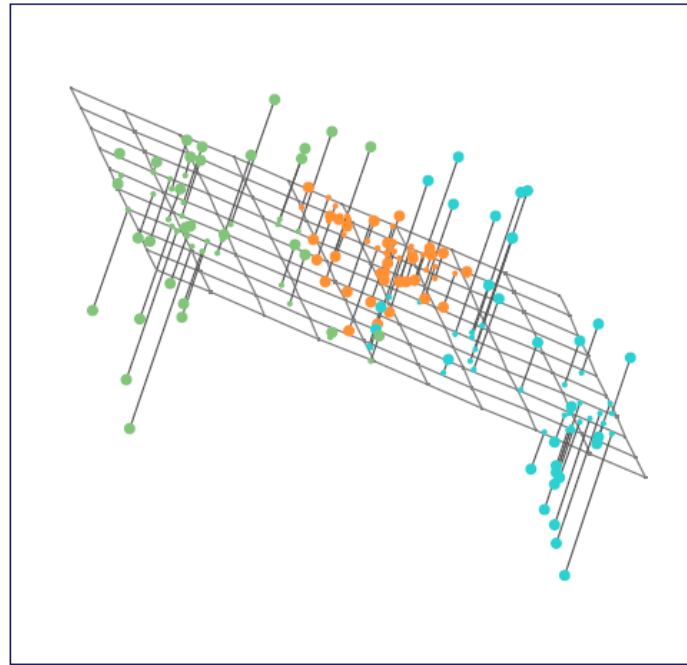
- The following **scree plots** show that of the **20 principal components** generated, **the first 3 together** explain about 90% of the variance of DLC 500 and 86% of the VLC 30.





Applications of PCA (Data visualization)

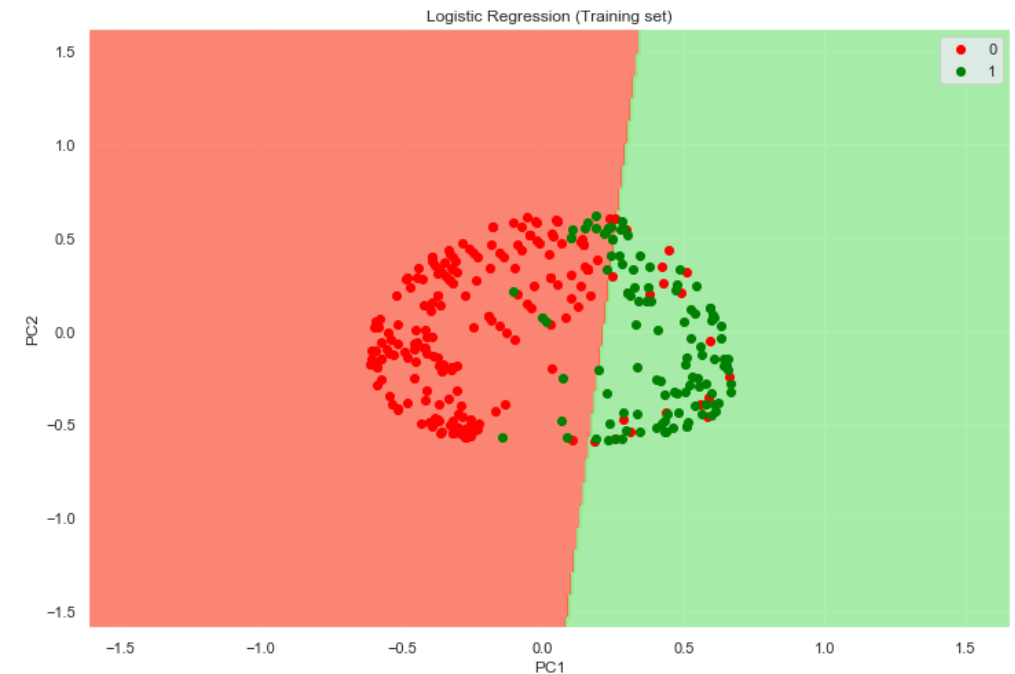
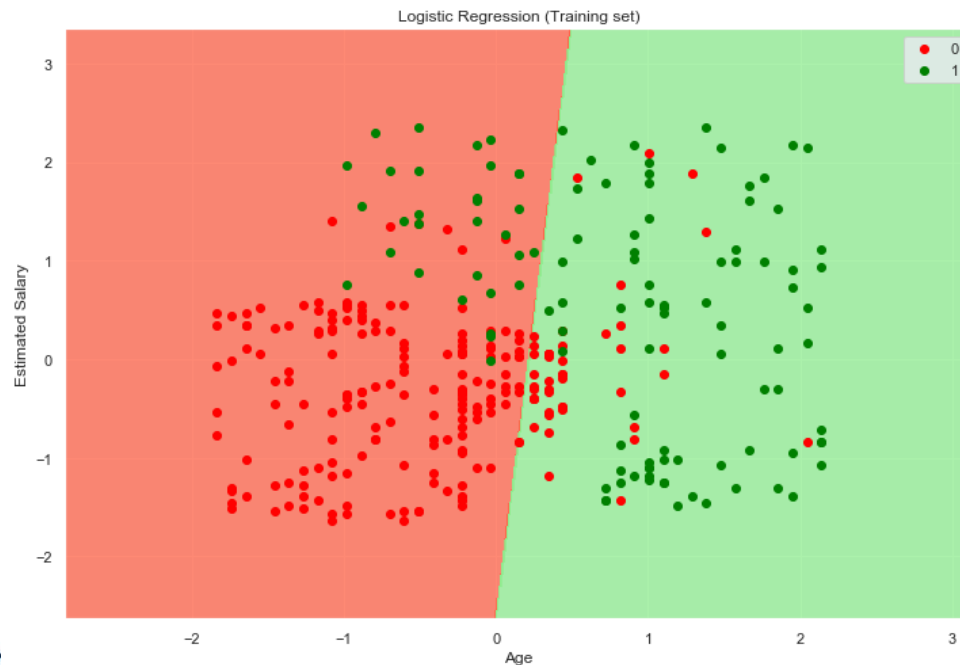
- PCA can be fed into other unsupervised or supervised learning models!
- Using PCA with an unsupervised model like K-Mean **clustering**:





Applications of PCA (Kernel PCA)

- What if we want to use a linear classifier (regressor) but the data is **non-linearly separable**?
- Solution: **Kernel PCA**
- Kernel PCA uses a kernel function to project dataset into a higher dimensional feature space, where it is **linearly separable**. It is like the idea of Support Vector Machines.
- Using PCA with a supervised learning model like logistic regression:



Students' questions

1. test