

Clustering Algorithms & Use-Cases

Analytiks

Unsupervised: Clustering

- Segregate groups with similar traits, assign them into clusters.
- An Unsupervised learning algorithm - no need for target variable!
- All elements within a group are more similar among them than they are to the others

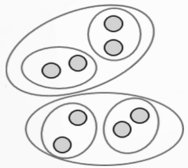
Some Types:

K-Means

Hierarchical

Examples:

Customer Profiling	Anomaly Detection	Land Use (Geospatial Images)
Identifying Groups of High Claimants (Insurance)	City Planning: Groups of Houses	Pre-processing for Classification



Key Criterion:

High Inter-Class Similarity

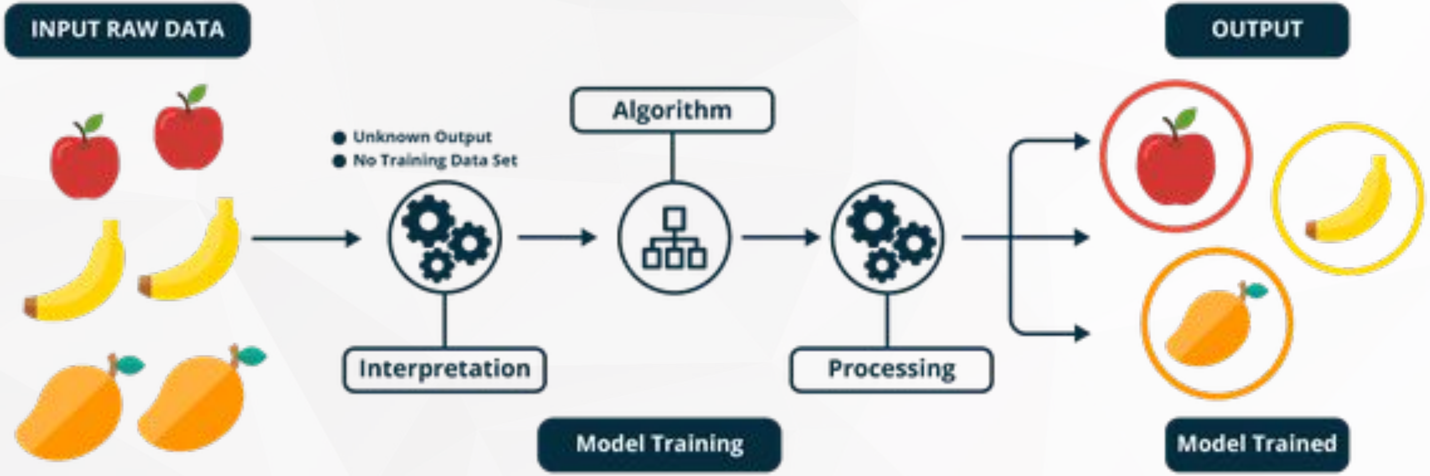
Low Intra-Class Similarity

Look for:

Number of clusters that produce tighter clusters

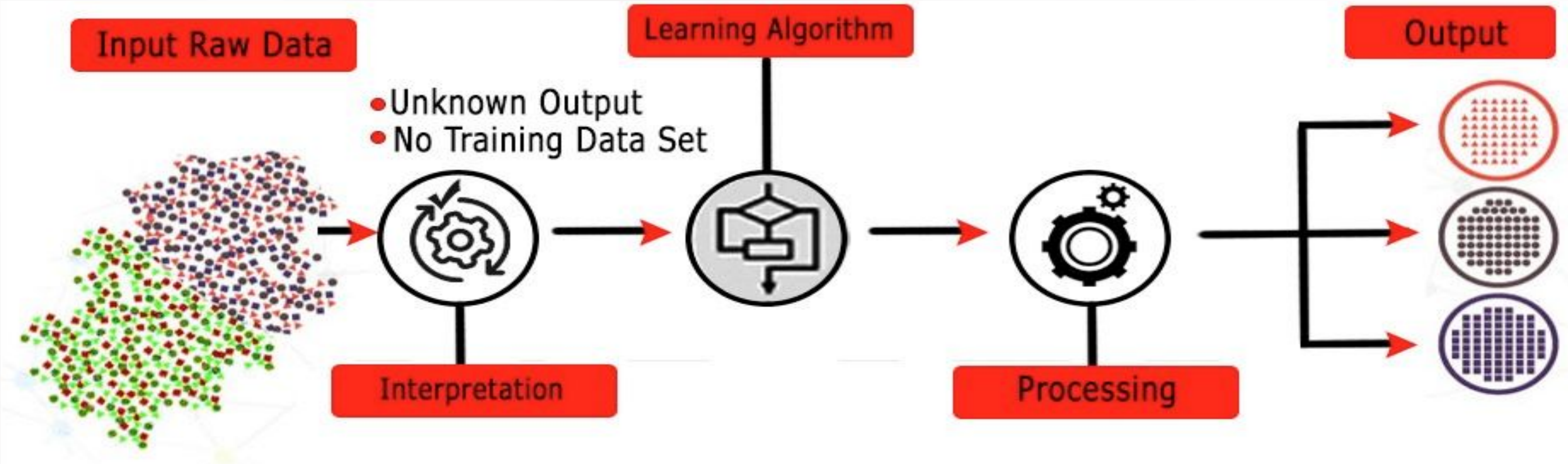
Number of clusters that are actionable / makes sense to the business

How does a Machine Learn?



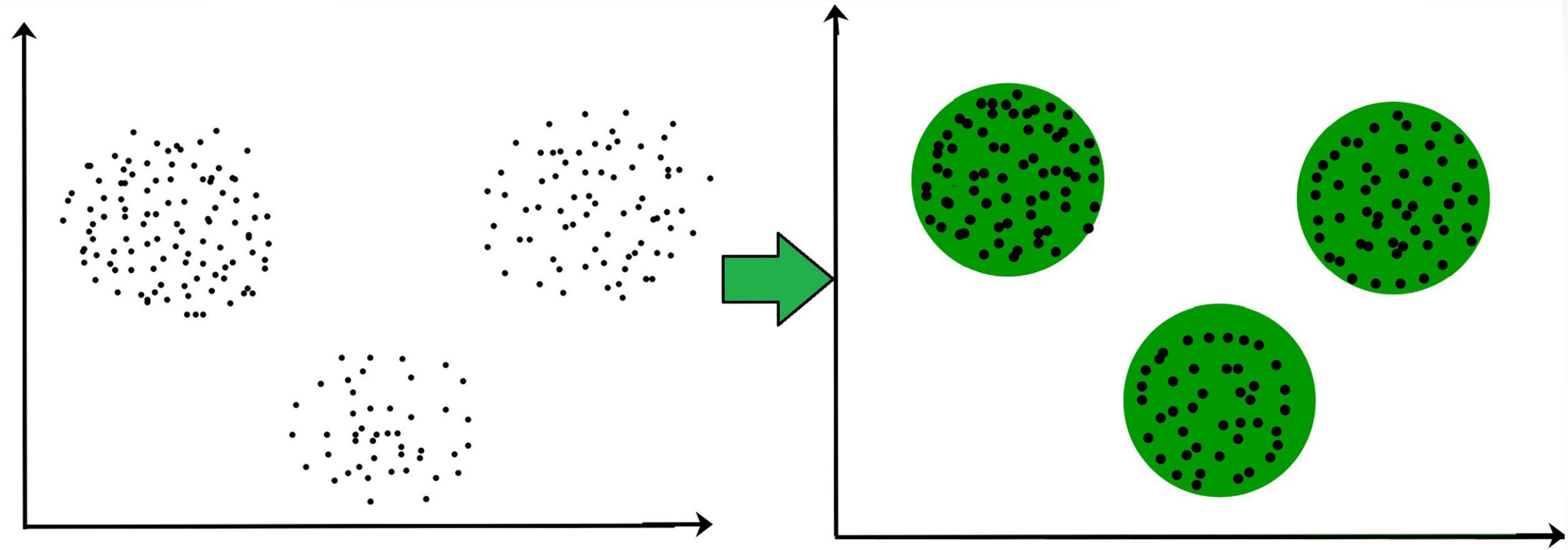
NO LABEL!

How does a Machine Learn?



NO LABEL!

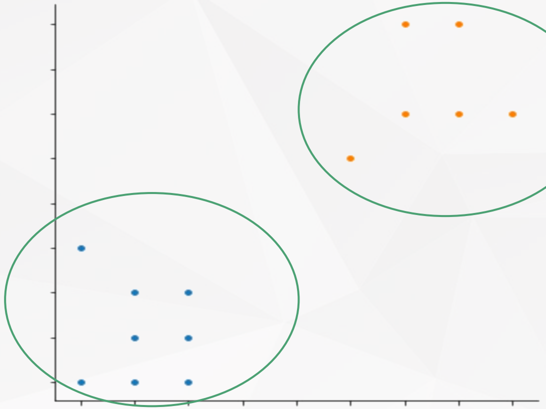
Clustering: Illustration



Two Main Types of Clustering Algorithms

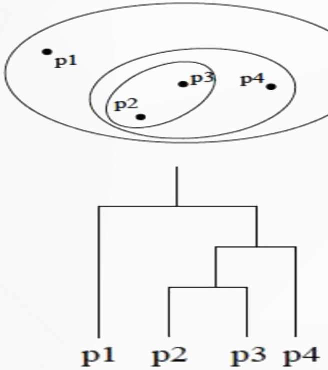
Partition Based

- Data points are divided into finite number of partitions
- Each data point assigned to one subset



Hierarchical

- Data points are organized into nested clusters
- Organized into a hierarchical tree called a dendogram



Clustering Algorithms

Clustering

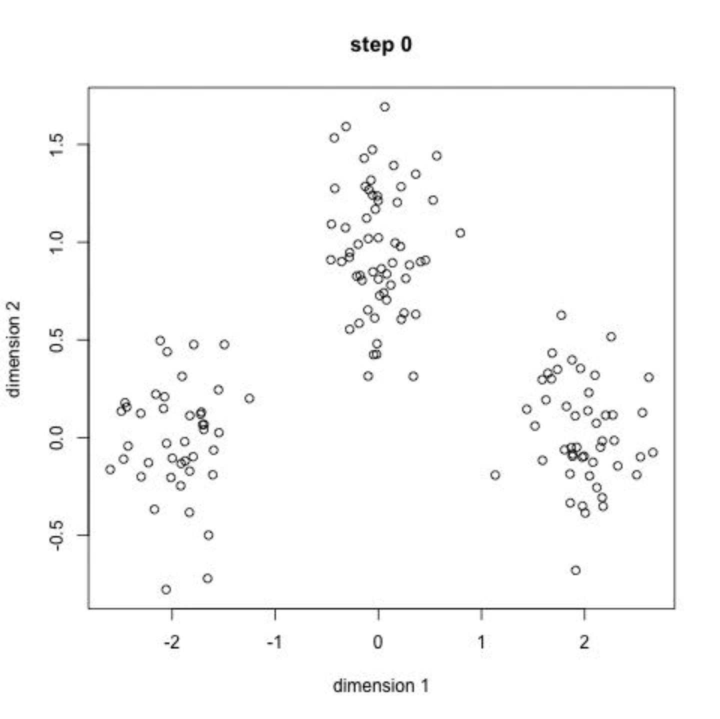
Algorithms

- K-Means
- Hierarchical Clustering

K-Means Clustering: Overview

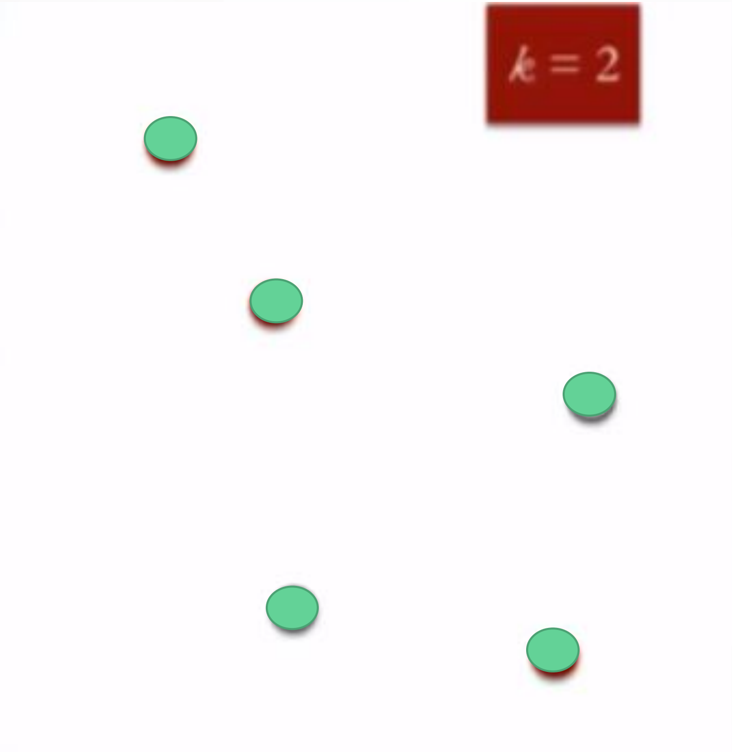
- Popular partition-based algorithm
  - Organizes the n objects into k partitions (k < n)
  - k - number of partitions or clusters

- Centroid-based technique
  - Cluster similarity is measured in regard to the mean value of the objects in a cluster
  - Mean value: seen as the cluster's centroid or center of gravity



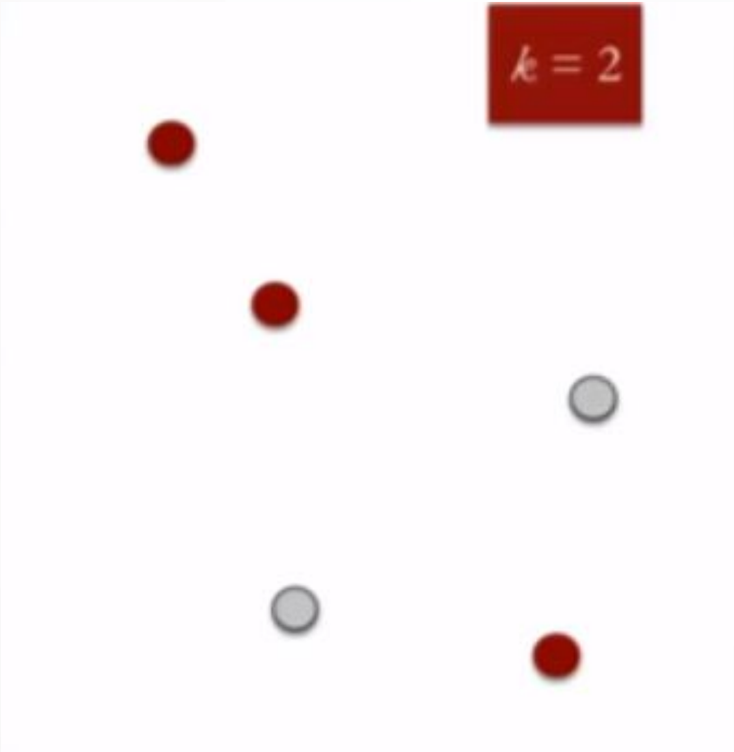
K-Means Clustering: How it Works

1. Specify the desired number of clusters k



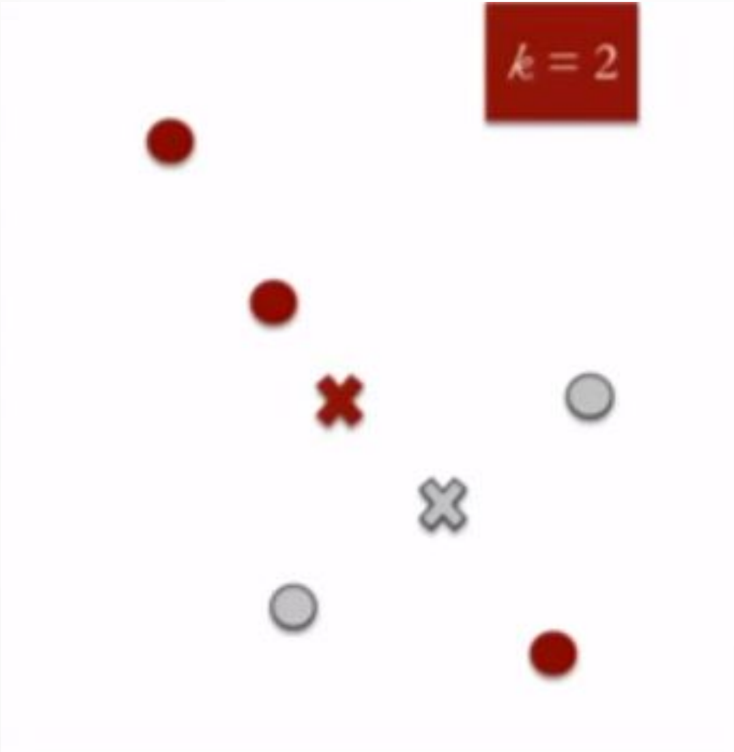
K-Means Clustering: How it Works

2. Randomly assign each data point to a cluster



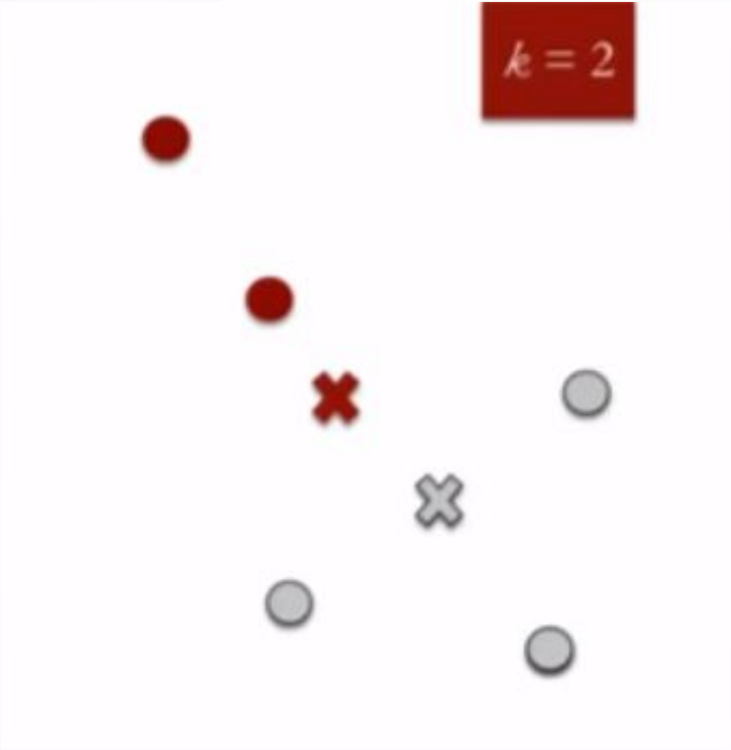
K-Means Clustering: How it Works

3. Use assigned points to compute for the centers/means



K-Means Clustering: How it Works

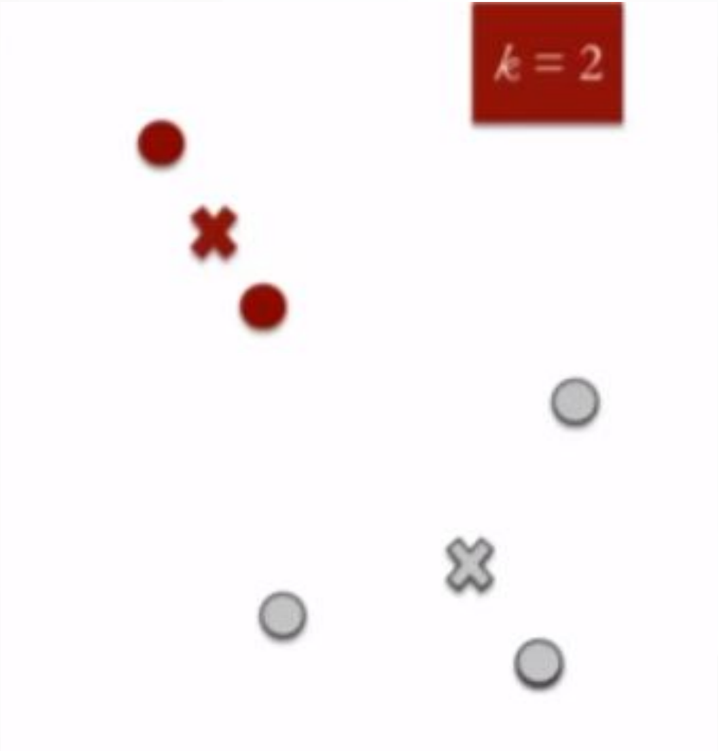
4. Re-assign each point to the closest cluster centroid



K-Means Clustering: How it Works

5. Re-compute cluster centroids

6. Repeat steps 4 and 5 until no improvements are possible
    a. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.



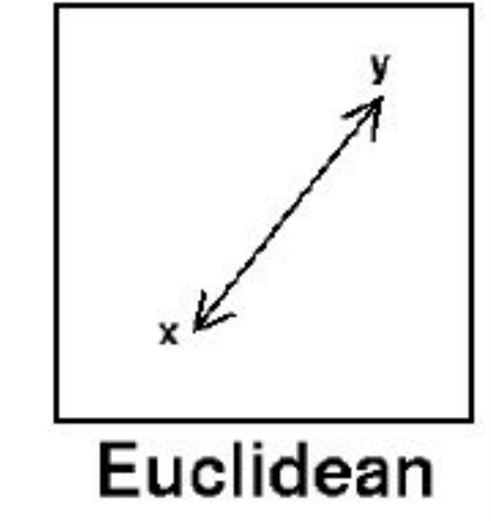
K-Means Clustering: Similarity Function

- Square Error Criterion
  - Commonly used
  - Also known as: Squared Euclidean Distance

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

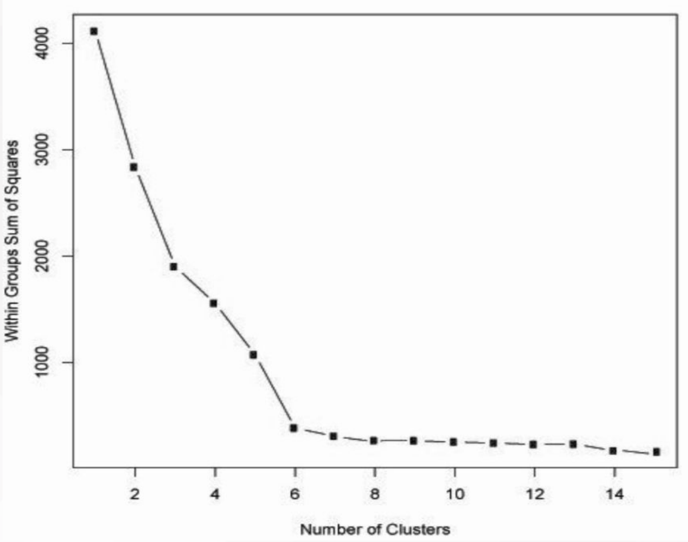
- Where:
  - k - number of clusters
  - p - point/object in the data
  - m_i - mean of the ith cluster

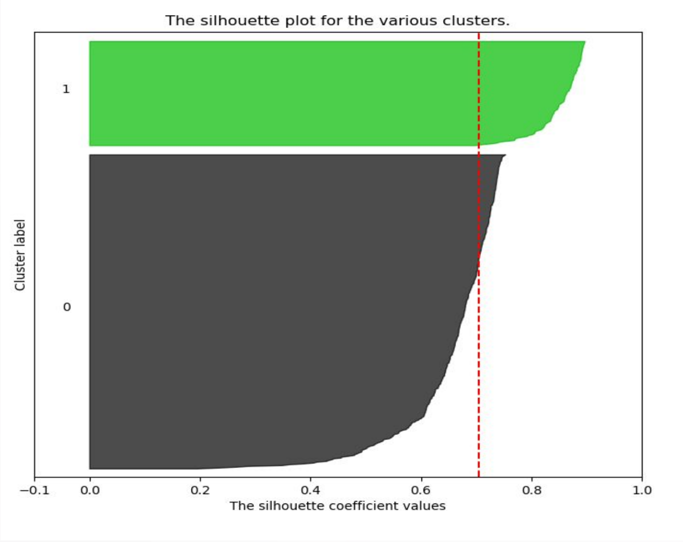
- Euclidean Distance Formula: $\sqrt{(a^2 + b^2)}$
- Squared Euclidean Distance Formula: $(a^2 + b^2)$



K-Means Clustering: Choosing k

- Use prior knowledge / Domain Knowledge
- Elbow method: Try for different values of k’s
  - Sum of Squared Errors
  - Silhouette method

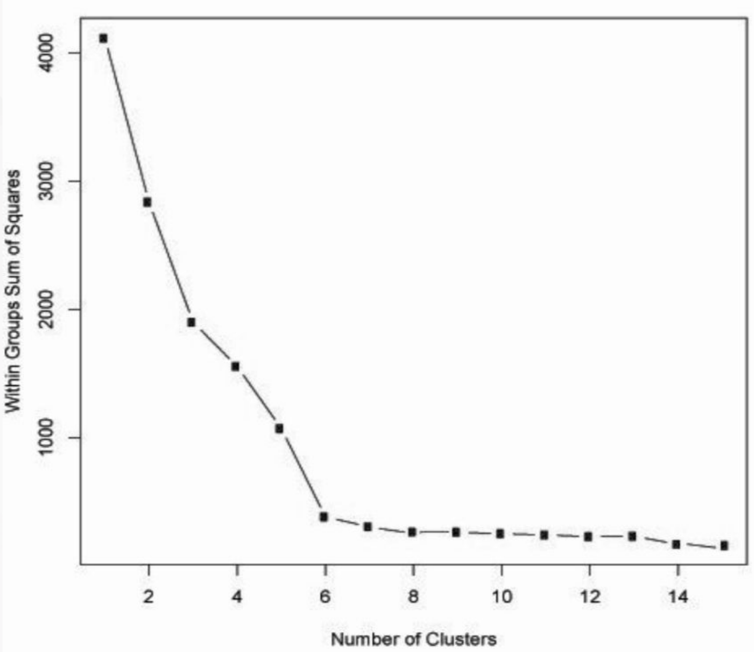




K-Means Clustering: Elbow Method

- Steps:
  - Run k-means clustering on the dataset for a range of values of k (i.e. k from 1 to 10)
  - For each value of k calculate the sum of squared errors (SSE)
  - Plot a line chart of the SSE for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best.

- The goal is to choose a small value of k that still has a low SSE, and the elbow usually represents where we start to have diminishing returns by increasing k.

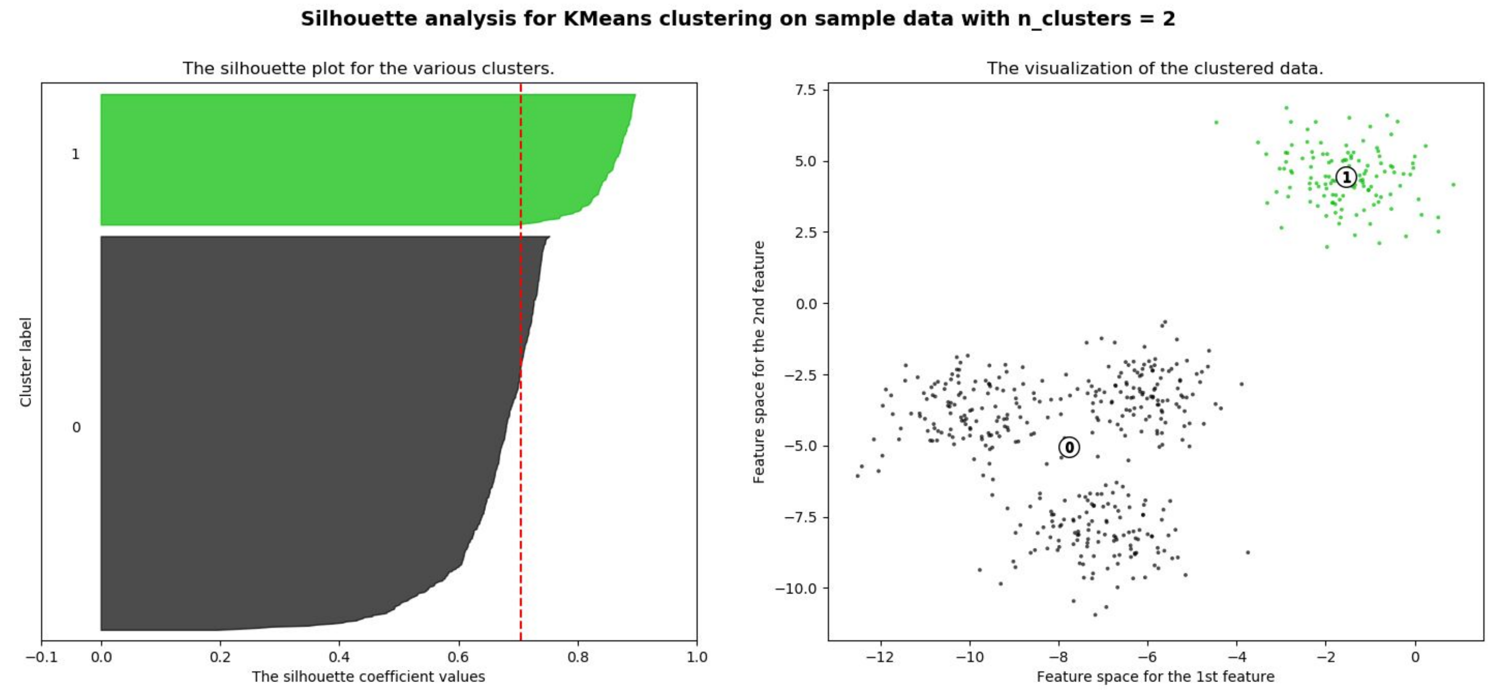


K-Means Clustering: Silhouette Analysis

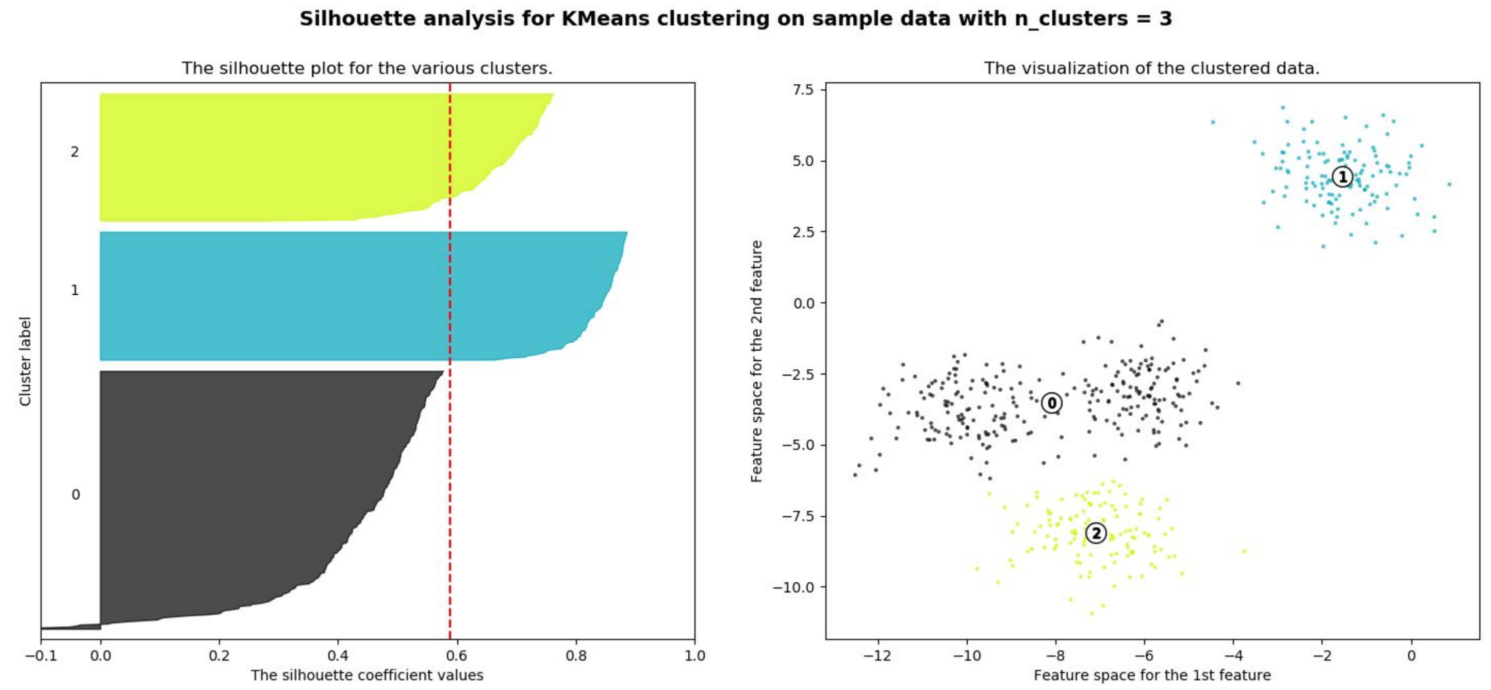
- A way to measure how close each point in a cluster is to the points in its neighboring clusters.
- Values lies in the range of [-1, 1]
    - +1: indicates that the sample is far away from its neighboring cluster and very close to the cluster its assigned
    - -1: indicates that the point is close to its neighboring cluster than to the cluster its assigned.
    - 0: means its at the boundary of the distance between the two cluster.

- The higher the value, the better is the cluster configuration.

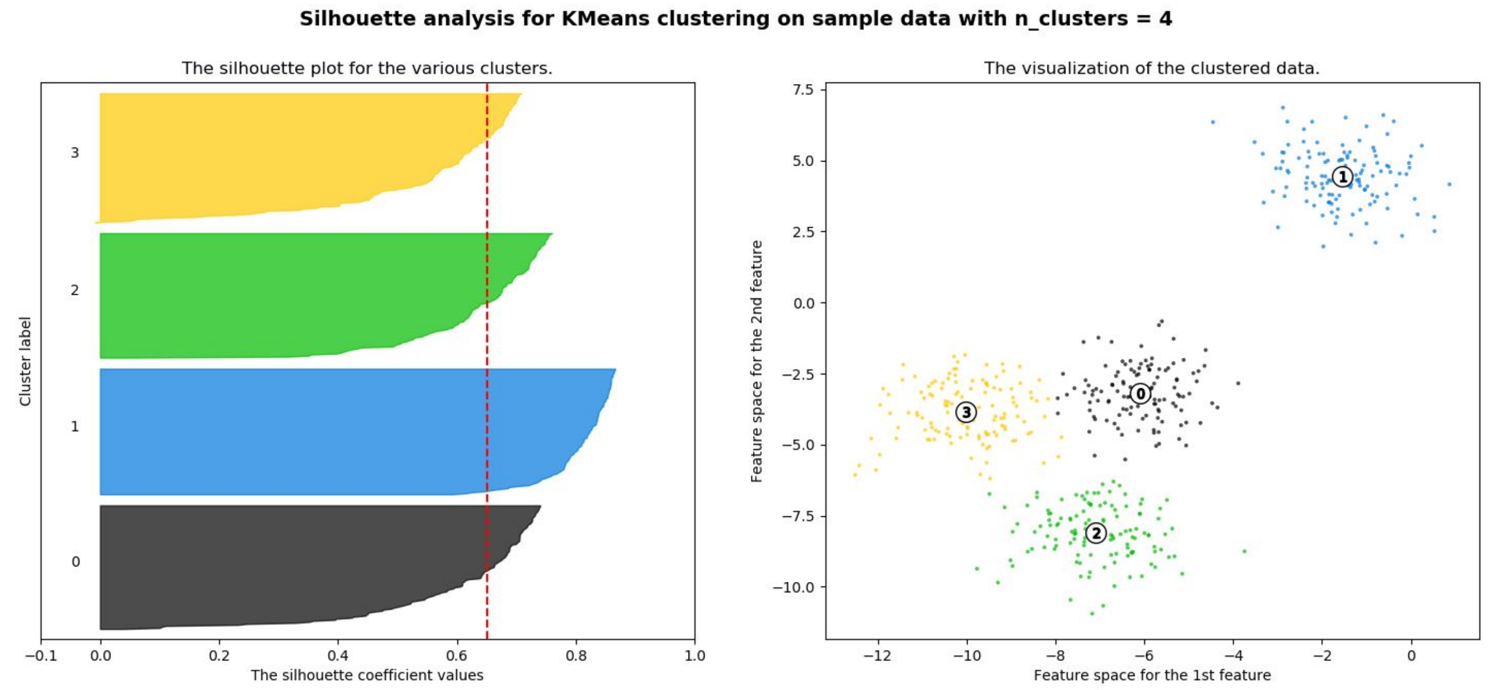
K-Means Clustering: Silhouette Analysis



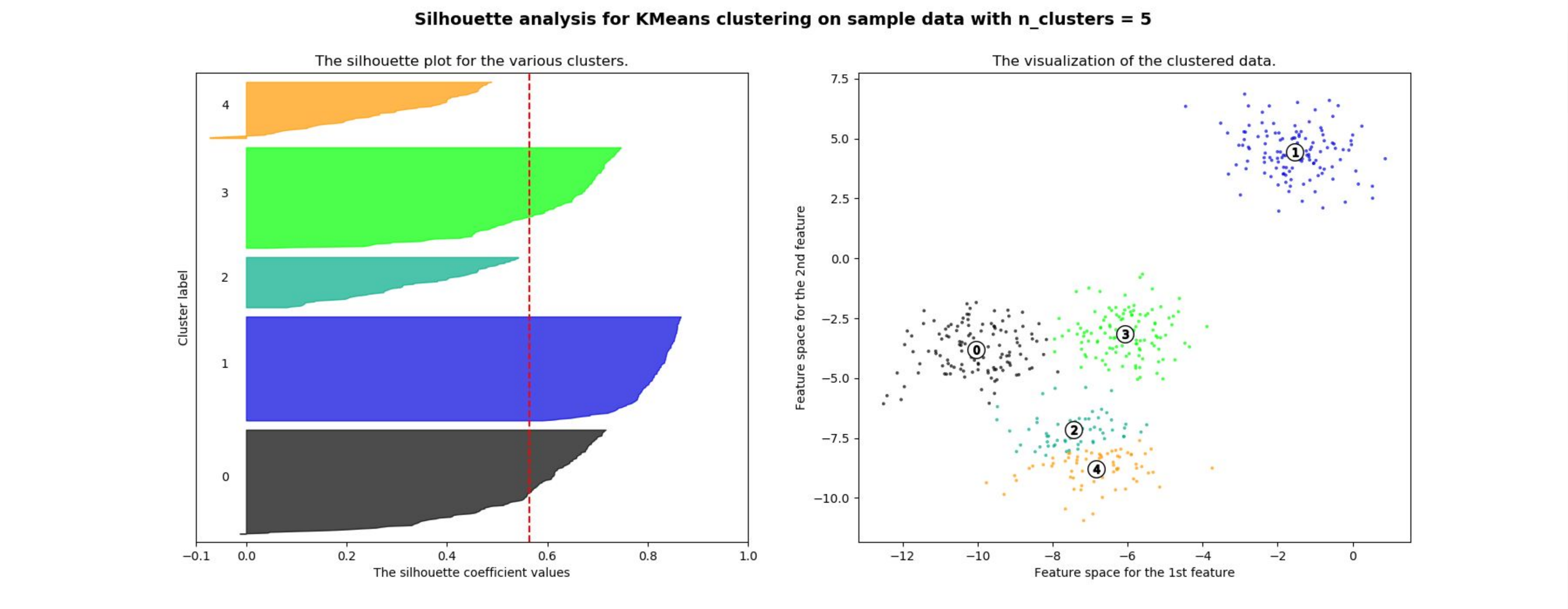
K-Means Clustering: Silhouette Analysis



K-Means Clustering: Silhouette Analysis



K-Means Clustering: Silhouette Analysis



K-Means Clustering: Silhouette Analysis

K-Means Clustering: Silhouette Analysis

N_Clusters	Average Silhouette Score
2	0.7050
3	0.5882
4	0.6505
5	0.5638
6	0.4505

K-Means Clustering: Pros and Cons

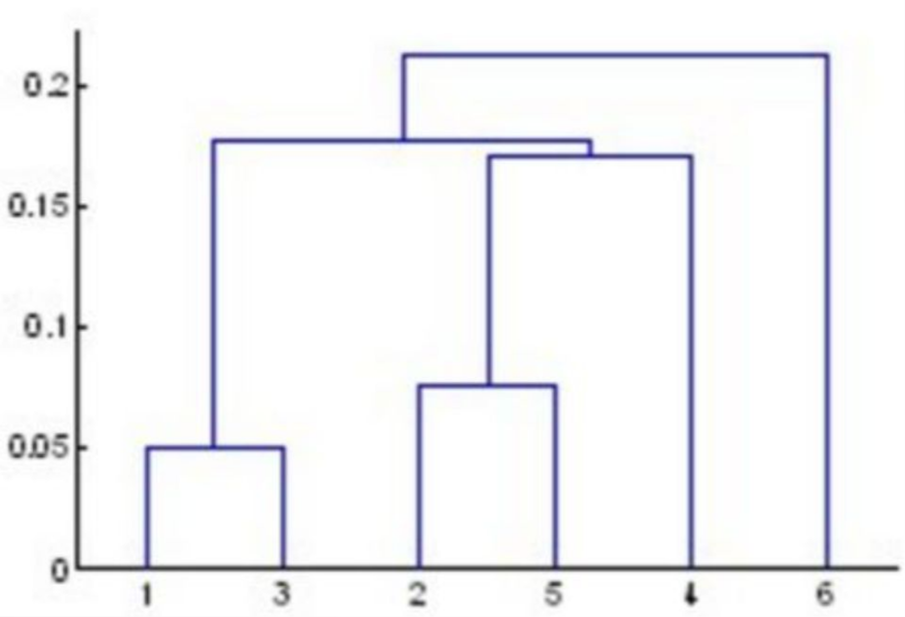
+	-
- Relatively scalable and efficient in processing large datasets - Produce tighter clusters than hierarchical clustering, especially if the clusters are globular	- Only applicable when the mean of a cluster is defined - Not applicable for categorical variables - Variation: k-modes - Need to specify k - Not suitable for clusters with very different sizes and density and non-globular - Different initial partitions can result in different final clusters - Sensitive to outliers

LAB:
K-Means Clustering

Analytiks

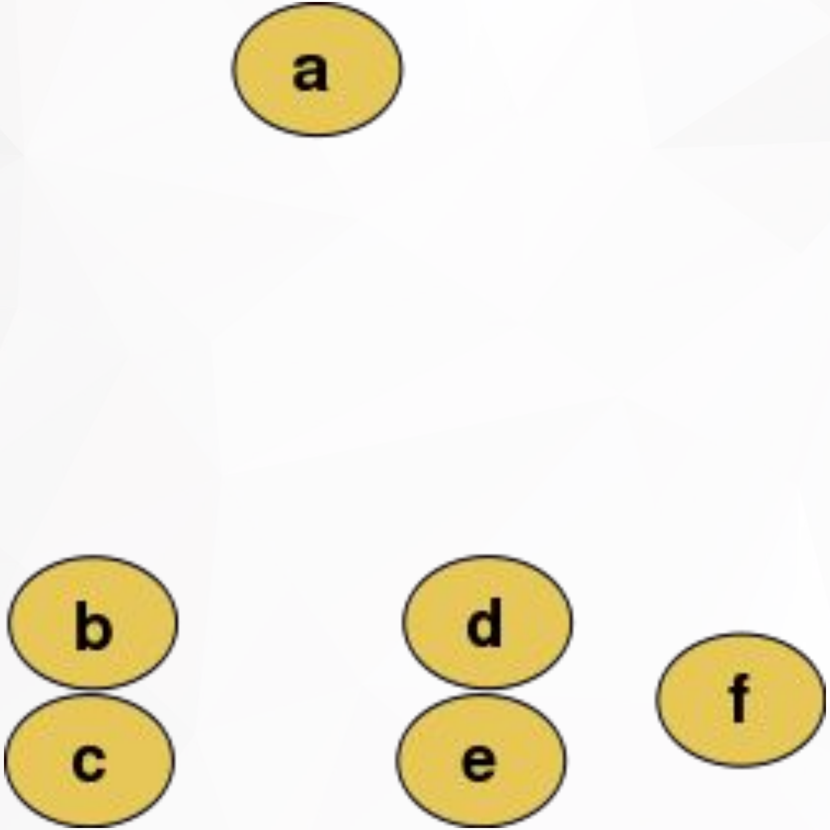
Hierarchical Clustering: Overview

- Algorithm that builds hierarchy of clusters.
- Shows the sequences of merges/splits
- Approaches:
  - Agglomerative:
    - Bottom up Approach
  - Divisive
    - Top down Approach
- Resulting Output: Dendrogram



Hierarchical Clustering: How it Works

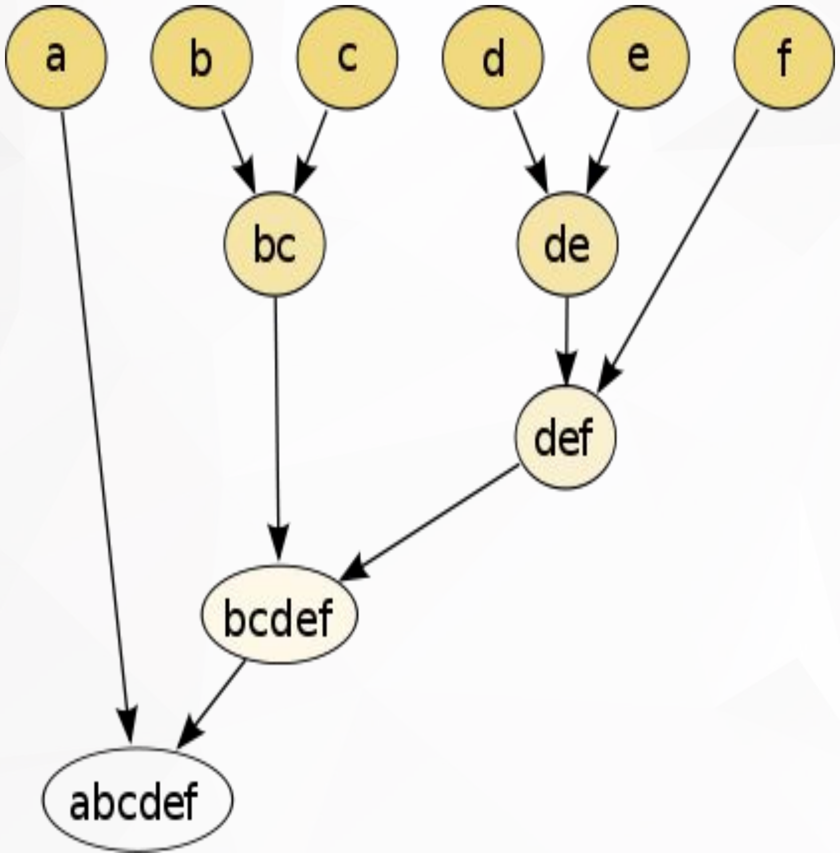
1. For instance, a , b ,c, d, e,f are 6 customers, and we wish to group them into clusters.



Hierarchical Clustering: How it Works

2. Sequentially group these students and we can stop the process at any number of clusters we want.

3. Example:
   a. 2 Clusters - (a, bcdef)
   b. 3 Clusters - (a, bc, def)
   c. 4 Clusters - (a, bc, de, f)

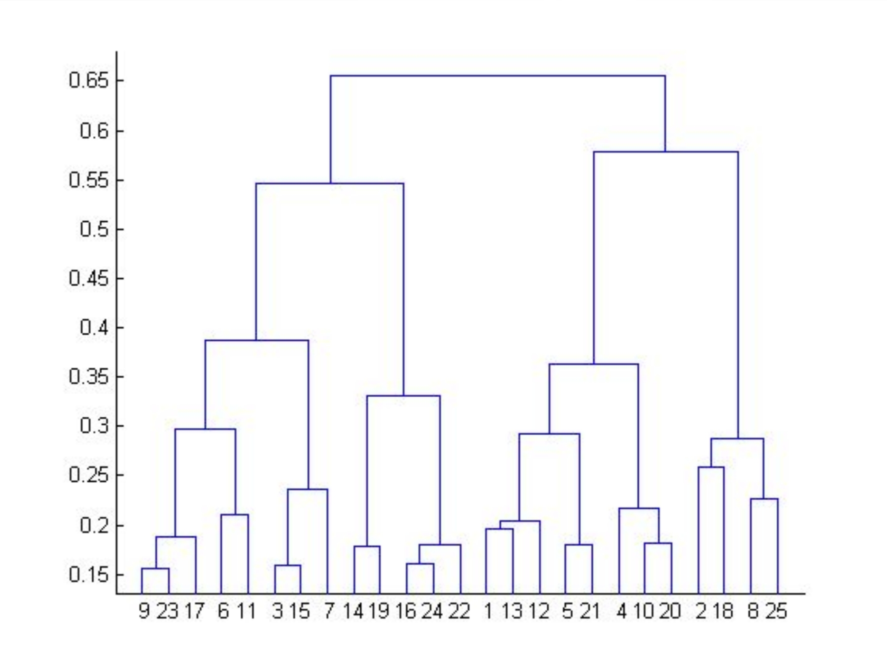


Hierarchical Clustering: Example

1. At the bottom, we start with 25 data points, each assigned to separate clusters.

2. Two closest clusters are then merged, until only one cluster is present at the top.

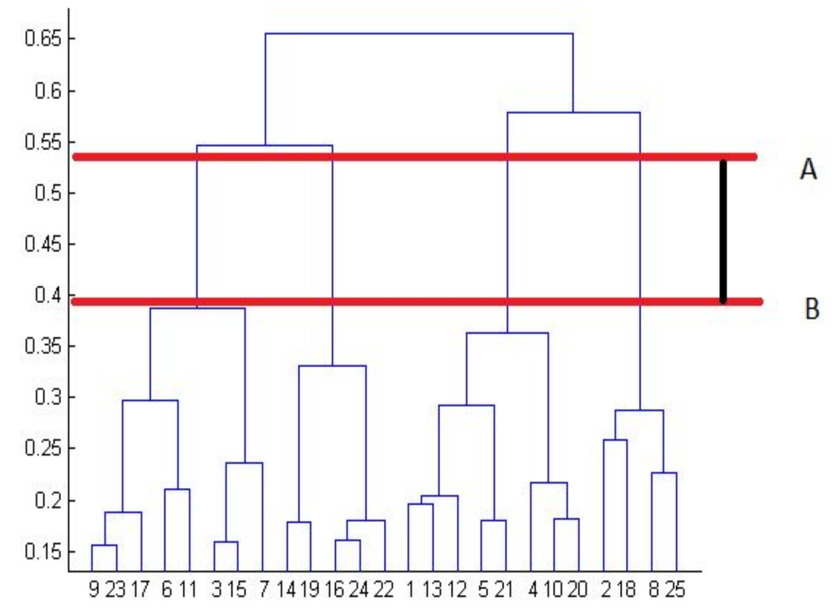
3. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.



Hierarchical Clustering: Example

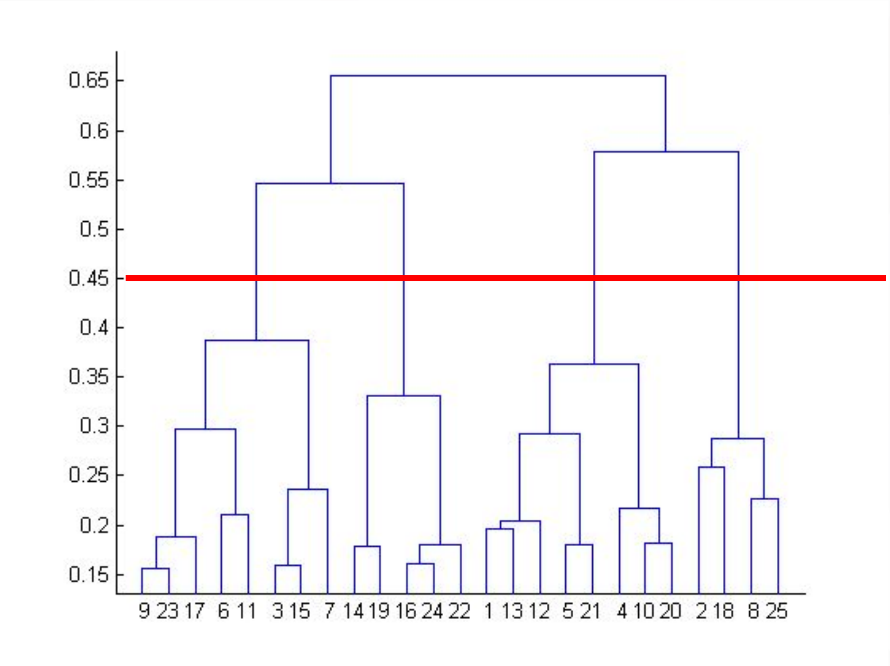
4. The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram.

5. Typically, the best choice for number of clusters is where the difference is most significant.



Hierarchical Clustering: Example

6. The number of clusters is the number of vertical lines in the dendrogram cut by a horizontal line



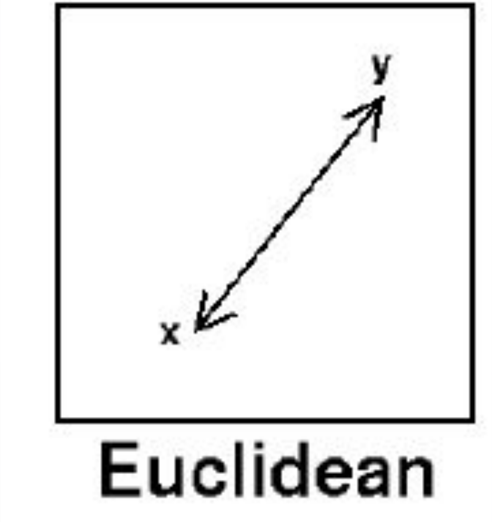
Hierarchical Clustering: Distance Functions

- Square Error Criterion
    - Commonly used
    - Also known as: Squared Euclidean Distance

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |\boldsymbol{p} - \boldsymbol{m}_i|^2$$

- Where:
    - k - number of clusters
    - p - point/object in the data
    - m_i - mean of the ith cluster

- Euclidean Distance Formula: $\sqrt{(a^2 + b^2)}$
- Squared Euclidean Distance Formula: $(a^2 + b^2)$



Hierarchical Clustering: Pros and Cons

+	-
• No prior information about the number of clusters required • Easy to implement and gives best result in some cases • May correspond to meaningful taxonomies	• Can be computationally expensive • Can suffer from: ○ Sensitivity to noise and outliers ○ Breaking large clusters ○ Difficulty handling clusters with varying sizes and convex shapes • Can be difficult to identify the correct number of clusters by the dendrogram

LAB: Hierarchical Clustering

Analytiks

Business Use Case: Customer Profiling & Segmentation

Customer Segmentation

- Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics.
- a company tailors offerings to segments that are the most profitable and serves them with distinct competitive advantages.
- Helps companies:
  - Develop marketing campaigns
  - Craft pricing strategies to extract maximum value from high and low profit customers
  - Basis for allocating resources for product development, marketing, service and delivery programs

Customer Segmentation

Customer Segmentation requires managers to:

1. Divide the market into meaningful and measurable segments according to customers' **needs**, their **past behaviors** or their **demographic profiles.**
2. Determine the profit potential of each segment by analyzing the revenue and cost impacts of serving each segment.
3. Target segments according to their profit potential and the company's ability to serve them in a proprietary way.
4. Invest resources to tailor product, service, marketing and distribution programs to match the needs of each target segment.
5. Measure performance of each segment and adjust the segmentation approach over time as market conditions change decision making throughout the organization.

Source : Bain & Company Management Tools: Customer Segmentation
https://www.bain.com/insights/management-tools-customer-segmentation

RFM is a method used for analyzing customer value:

Recency – How recently did the customer purchase?
Frequency – How often do they purchase?
Monetary Value – How much do they spend?

Clustering for a Customer Segmentation Use Case

1. Collect features that you want / need / believe to identify characteristics of a cluster
2. Prepare dataset - Each row is a unique customer with features that capture customer demographic, behavior, preferences, history, value*, etc.
3. Remove outliers
4. Normalize data
5. Perform Clustering
6. Test Clustering Using Classification
7. Analyze Clusters to make them actionable

LAB:
Customer Segmentation
Use Case : Retail

Analytiks

Exercise: Customer Segmentation Use Case : Insurance Survey

Analytiks

Advanced Topic: Unsupervised Feature Extraction - PCA

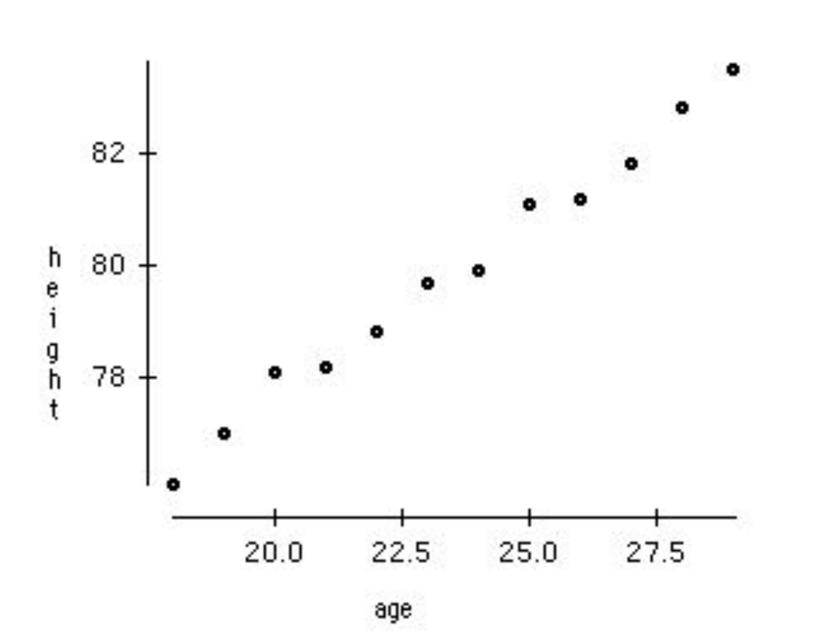
Principal Component Analysis (PCA)

Algorithms

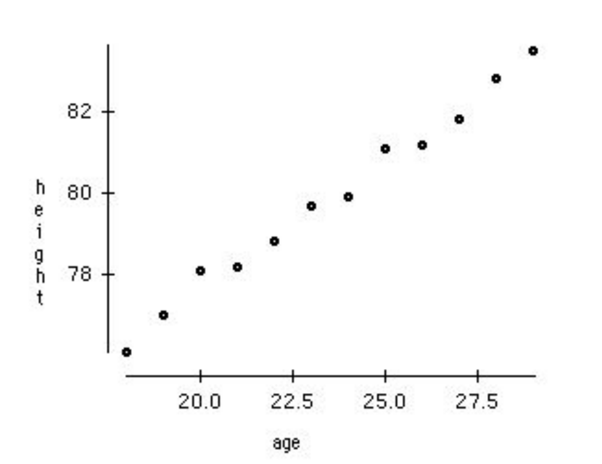
- A method of Feature Extraction
- extracts ***low dimensional*** set of features ***from a high dimensional*** data set with a motive to ***capture as much information as possible***
- Used for model improvement as well as visualization

PCA for Visualization Example

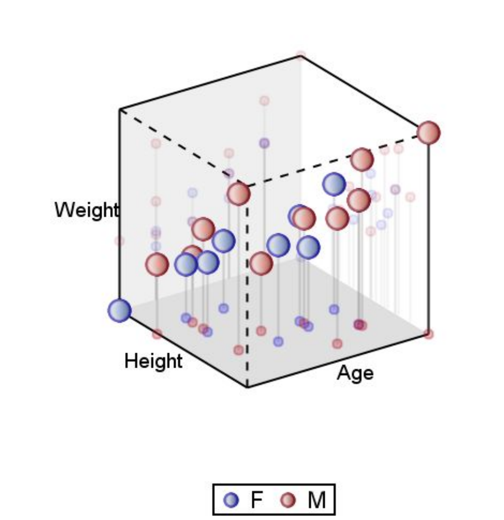
Given 2 variables: Age & Height



PCA for Visualization Example



Plotting 2 variables

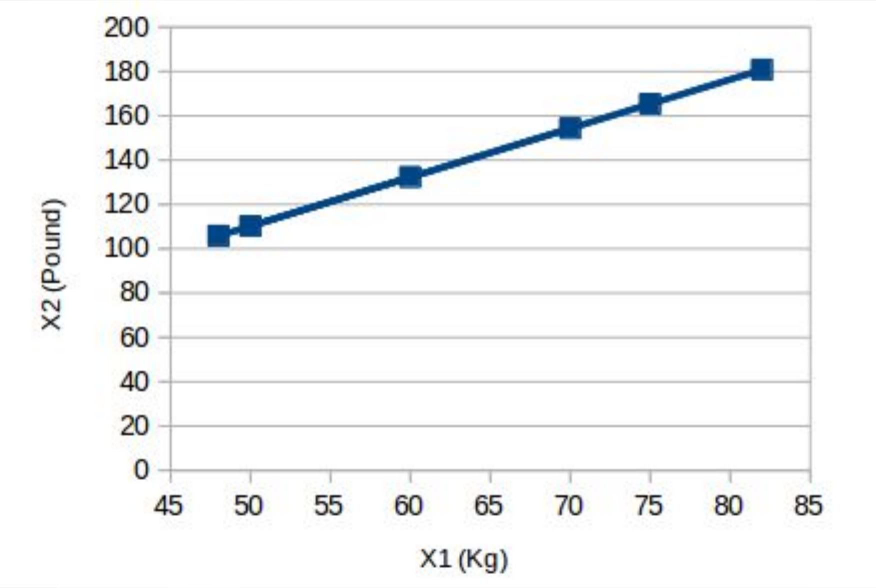


Plotting 3 - 4 variables

How do we plot more variables?

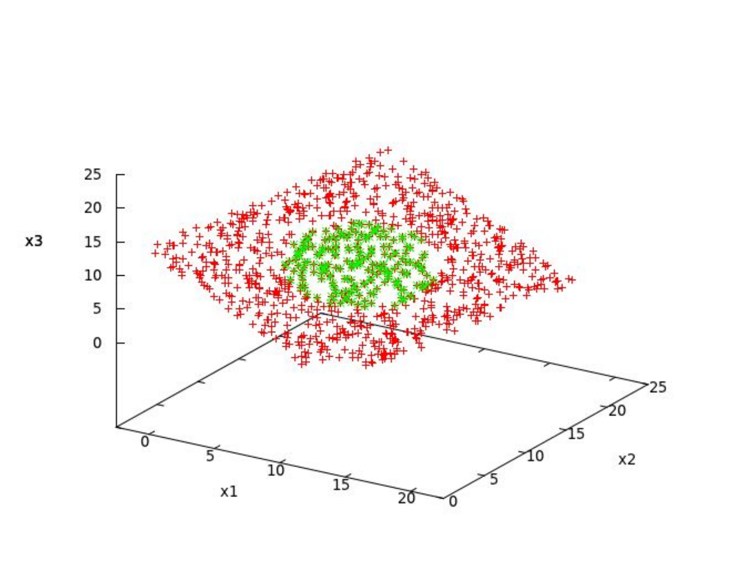
Principal Component Analysis

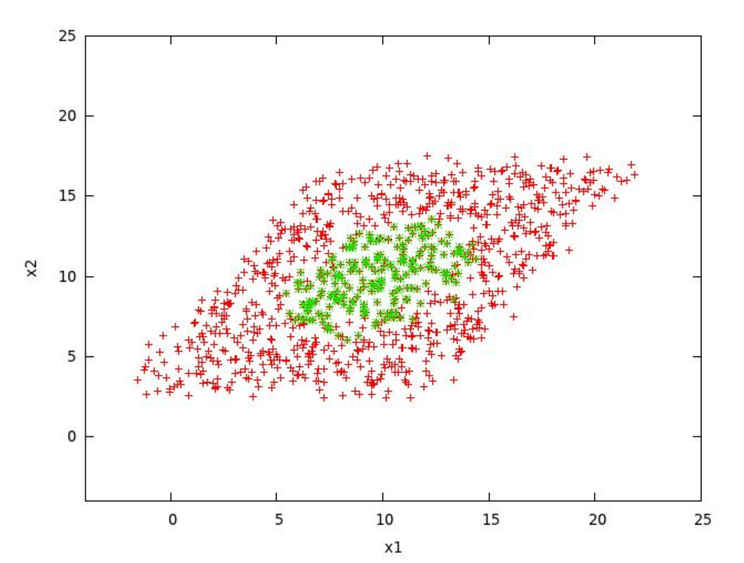
- Objective: Reduce the number of dimensions in a dataset while retaining most information



Principal Component Analysis

Suppose we have data with 3 features, simulating a square with a circle in the middle

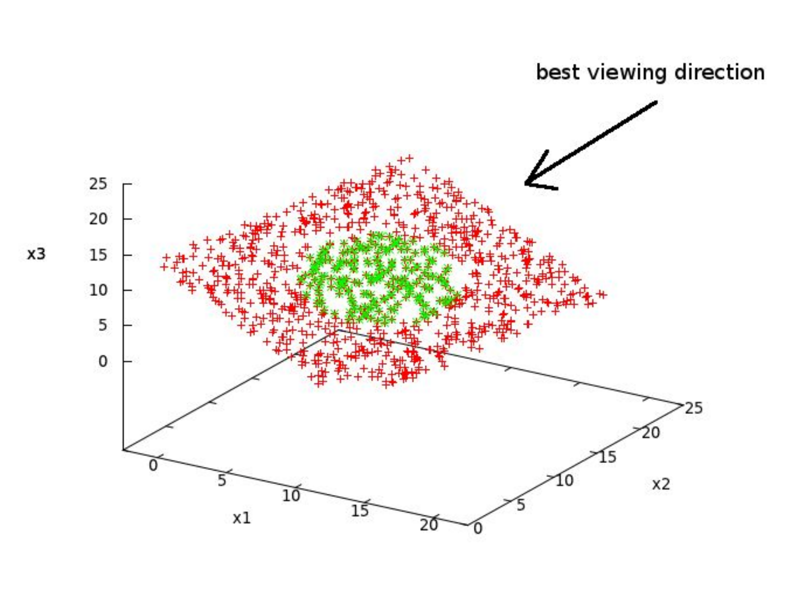


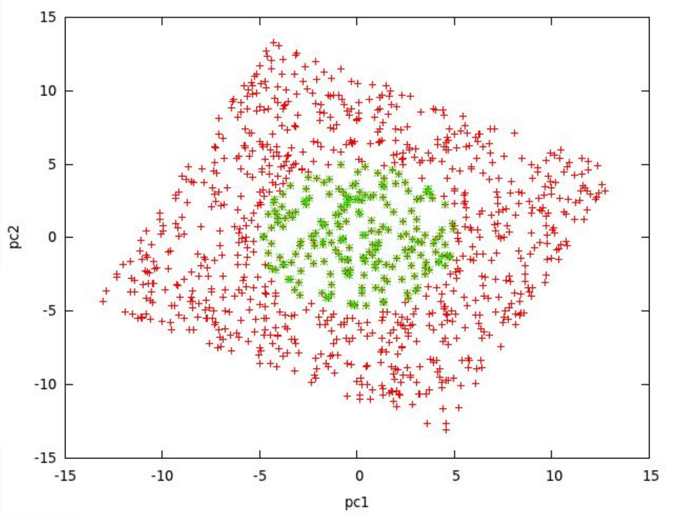


But projecting just using 2 of the variables, it appears skewed

Principal Component Analysis

The best projection is when we're looking directly / perpendicular to the plane

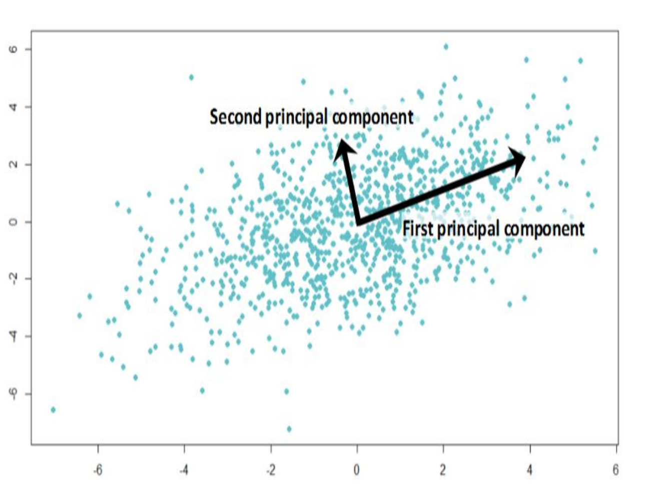




- With PCA, you can see the square and the circle clearly
- PCA essentially "rotates" the data, putting it on a new set of axes/principal components : PC 1, PC 2, PC 3...

Principal Components

- A **principal component** is a ***normalized linear combination*** of the original predictors in a data set.
- The first principal component is a linear combination of original predictor variables which ***captures the maximum variance*** in the data set → The larger the variability, the larger the information captured by the component
- The second and succeeding principal components capture the remaining variation without being related to the previous component (Correlation = 0)
- **Explained Variance -** how much information (variance) can be attributed to each of the principal components.
- E.g. if PCA1 has Explained variance of 72% and PCA2 has explained variance of 23%, PCA1 and PCA2 capture 95% of the information



Reminder: Standardize before PCA

- The original variables might be in different scales
- Performing PCA on non-standardized variables will lead to insanely large loadings for variables with high variance;
- This will lead to dependence of a principal component on the variable with high variance
- Solution: The principal components are supplied with standardized version of original predictors/input variables

LAB:
PCA: Iris Dataset

analytiks

HW:
Clustering Exercise

analytiks