KLASA
By Rajut Indonesia

Applied Predictive Analysis 1.0

13 November 2021

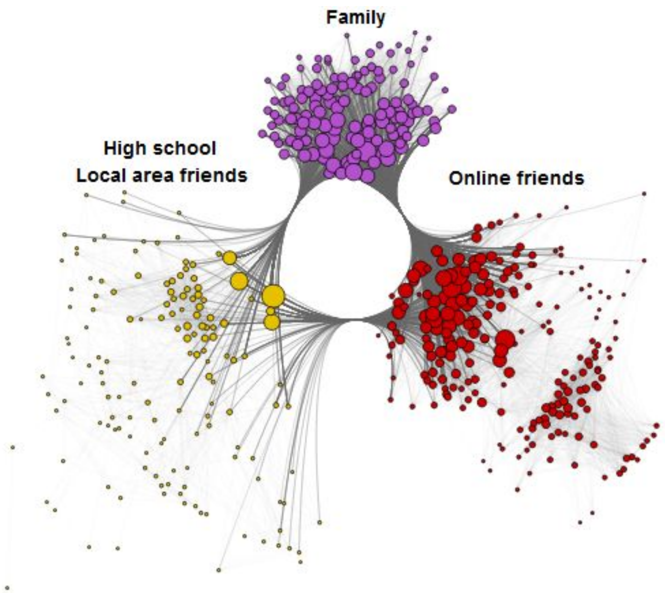
Data, Probability & Random Variables

Type of Data

- Numerical
  - Discrete, *e.g., population*
  - Continuous, *e.g., speed*
  - Interval, *e.g., temperature, time*
  - Ratio, *e.g., age, distance*
- Categorical
  - Nominal, *e.g., nationality, gender*
  - Ordinal, *e.g., socioeconomic status, grading system*
- Multimedia
  - Text
  - Image
  - Audio
  - Video
- Others
  - Geospatial, *e.g., Vector, Raster*
  - Biological, *e.g., DNA sequence*

	Nominal	Ordinal	Interval	Ratio
Categories	●	●	●	●
Rank order		●	●	●
Equal spacing			●	●
True zero				●

- Data representation
  - Cross-sectional
  - Temporal
  - Graph / network



Probability Theory

Applicable for any problems involving **uncertainty**,

a. model uncertainty

b. data uncertainty, *i.e.*, *sampling*

Example: a fair die

Sample space $\Omega = \{1,2,3,4,5,6\}$

- $\Pr(A)$, the **probability** that the event A is true

  $0 \le \Pr(A) \le 1$

  $A = \{1,3,5\}$ $\quad$ $\Pr(A) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$

  $B = \{5,6\}$ $\quad$ $\Pr(B) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$

  $\bar{B} = \{1,2,3,4\}$ $\quad$ $\Pr(\bar{B}) = \Pr(\Omega) - \Pr(B) = 1 - \frac{1}{3} = \frac{2}{3}$

- $\Pr(A,B)$, the **joint probability** of events A and B both happening

  $A \cap B = \{5\}$, $\quad$ $\Pr(A \cap B) = \Pr(A,B) = \frac{1}{6}$

  $\Pr(A,B) = \Pr(A)\Pr(B)$, if A and B are independent events

- $\Pr(A|B)$, the **conditional probability** of event happening given that event has occurred

  Bayes rule: $\Pr(A|B) = \frac{\Pr(A,B)}{\Pr(B)} = \frac{\Pr(A)\Pr(B|A)}{\Pr(B)}$

  $\Pr(A)$, Prior; $\Pr(B|A)$, Likelihood; $\Pr(B)$, Evidence

  Example: two fair dice

  Sample space $\Omega = 6 \times 6 = 36$ possible outcomes

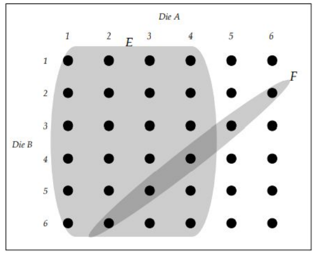
  What is the probability that the dice add up to 8, $\Pr(A)$, given that the first die gives a value that is ≤ 4, $\Pr(B)$?

  $A = \{(2,6),(3,5),(4,4),(5,3),(6,2)\}$ $\quad$ $\Pr(A) = \frac{5}{36}$

  $B = \{1,2,3,4\}$ $\quad$ $\Pr(B) = \frac{4}{6} = \frac{2}{3}$

  $B|A = \{(2,6),(3,5),(4,4)\}$ $\quad$ $\Pr(A \cap B) = \frac{3}{5}$

  $\Pr(A|B) = \frac{\Pr(A)\Pr(B|A)}{\Pr(B)} = \frac{5/36 \cdot 3/5}{2/3} = \frac{1}{8}$



Random Variables

A **random variable**, X, represents some unkown quantity of interest, *e.g., way a die will land when someone rolls it.*

The set of possible values is known as the **sample space**, $\Omega = \{1,2,3,4,5,6\}$.

The set of outcomes from a given Ω is called an **event**, seeing an odd number, $X = \{1,3,5\}$

A random variable has a **probability distribution**

- **Discrete random variables (counting)**

  $X \in \Omega$, a finite or countably infinite sample space.

  $p(x) = \Pr(X = x)$, **Probability Mass Function (PMF)**, where $0 \leq p(x) \leq 1$ and $\sum_{x \in \Omega} p(x) = 1$

  Cumulative Distribution Function (CDF), the sum of the probabilities of achieving that value and each successive lower values.

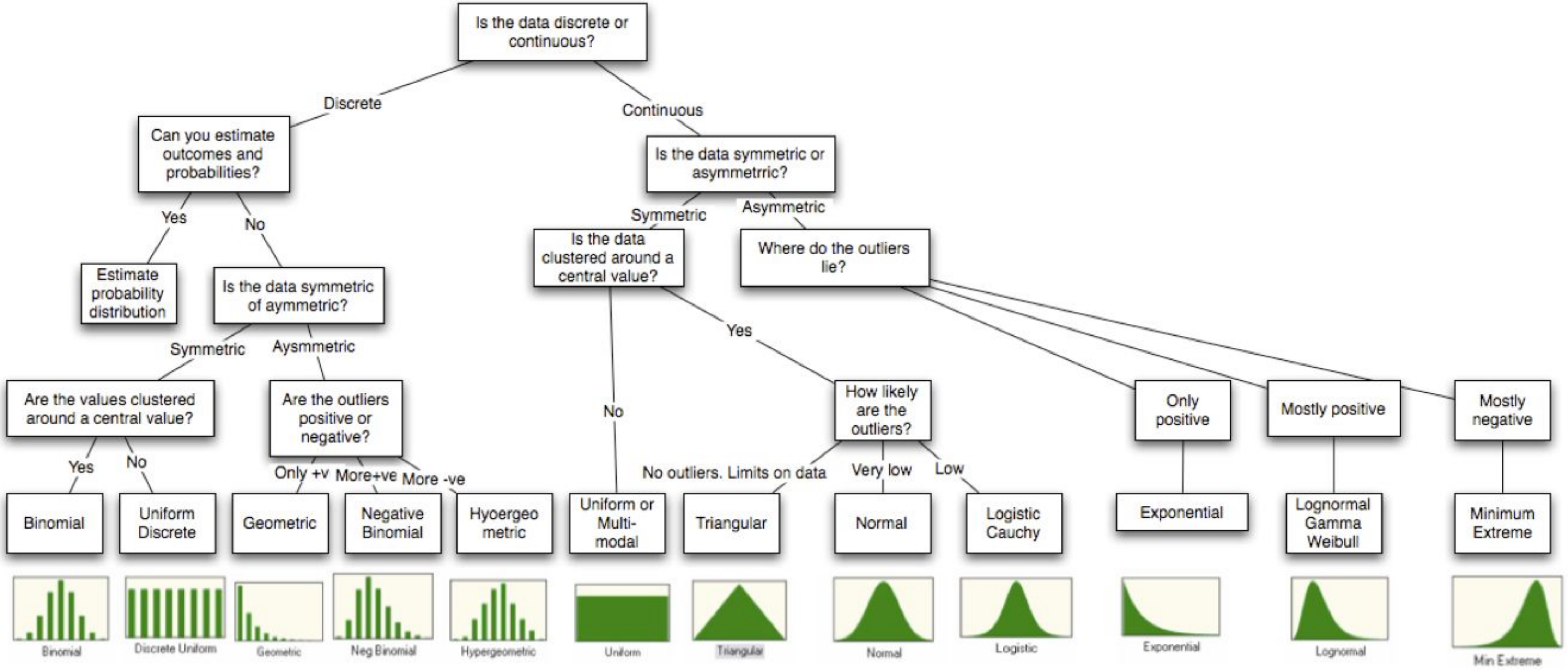
- **Continuous random variables (measuring)**

  $X \in \mathbb{R}$, a real-valued quantity.

  $F(x) = \Pr(a \leq X \leq b) = \int_a^b f(x)\,dx$, **Probability Density Function (PDF)**, where $f(x) \geq 0$ and $\int_{-\infty}^{\infty} x\,dx = 1$

  Cumulative Distribution Function (CDF), the area under the PDF curve to the left of the value in question.

Probability Distribution



Summary Statistics

- **Expected value (Mean), μ**

  Discrete: $\mathbb{E}[X] = \sum_{x\in\Omega} x \cdot p(x)$

  Continuous: $\mathbb{E}[X] = \int_{\Omega} x \cdot p(x)\, dx$

- **Variance, σ^2**

  A measure of the "spread" of a distribution

  $\mathbb{V}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

- **Covariance**

  The degree to which two r.v. X and Y are (linearly) related

  $\mathrm{Cov}[X,Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$

- **Correlation, ρ**

  The normalized measure (with a finite lower and upper bound) to which two r.v. X and Y are (linearly) related

  $\mathrm{corr}[X,Y] = \frac{\mathrm{Cov}[X,Y]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}}$

- **Skewness**

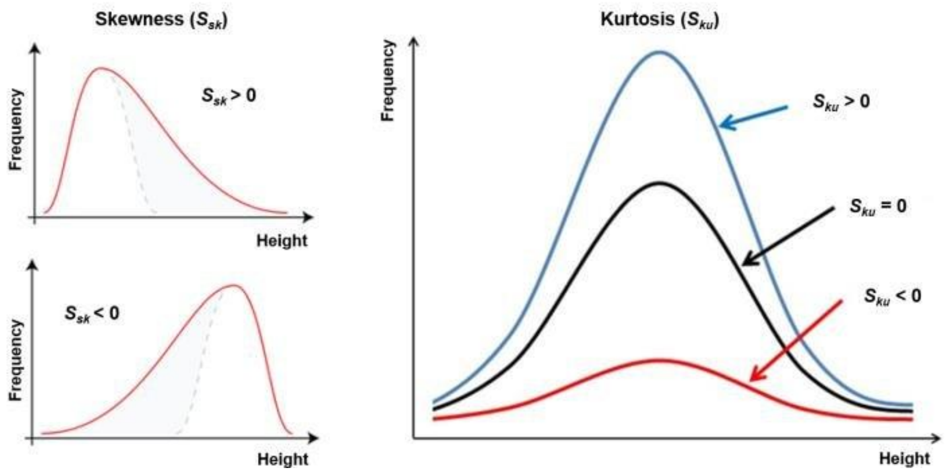
  How much the distribution deviates from the normal distribution

  $\mathbb{S}[X] = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right]$

- **Kurtosis**

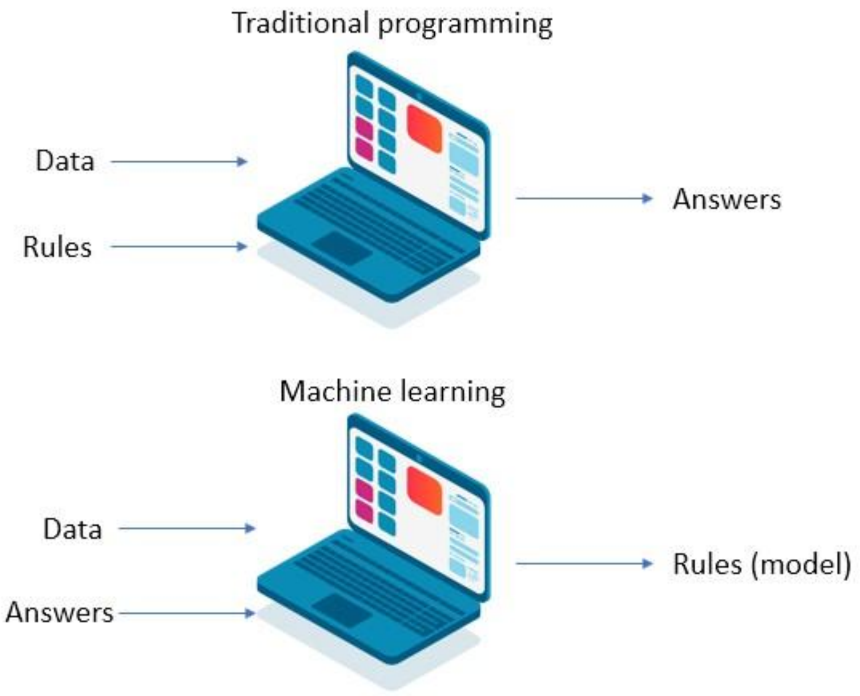
  A measure of the combined size of the tails relative to whole distribution

  $\mathbb{K}[X] = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]$



Introduction to Machine Learning

Basic Concepts



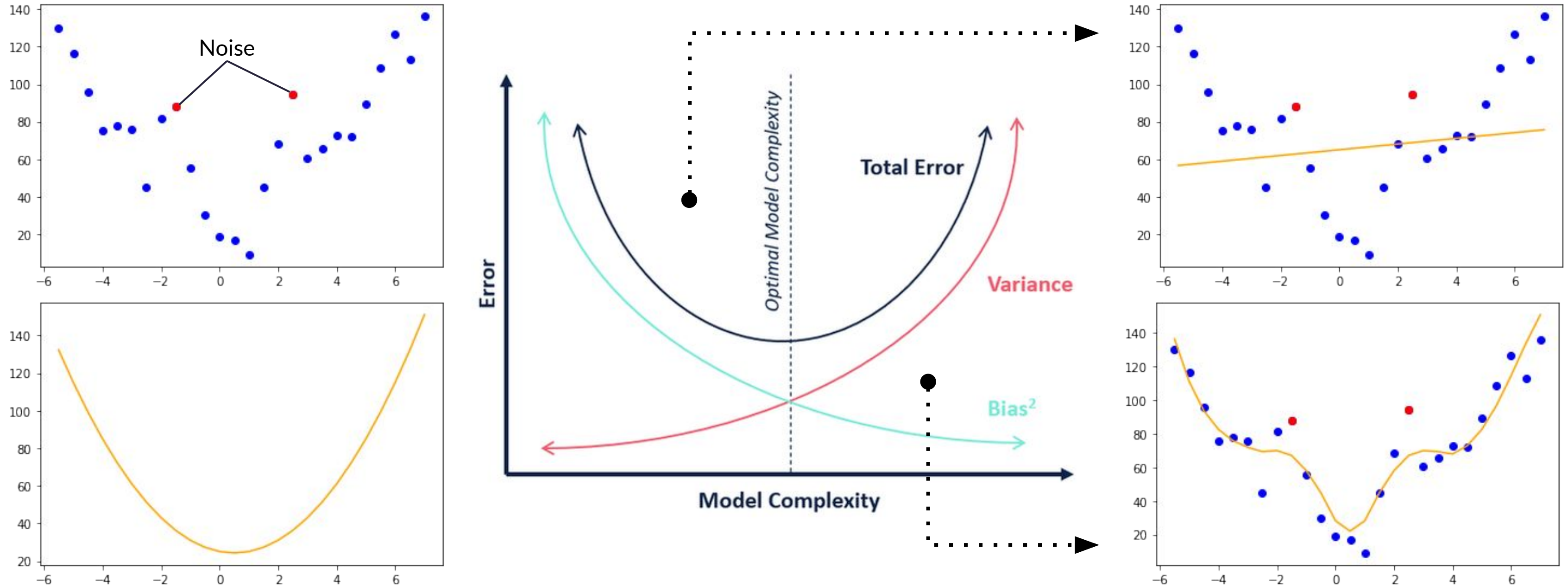
- Components of Machine Learning

- **Representation** or **Mapping Function**
  a. Parametric: estimate θ using $\mathcal{D}$
     Model has a fixed number of parameters, *e.g., linear regression, logistic regression, naïve bayes, ANN*
     + simplify the function to a known form
     + faster computation
     - require strong assumption
  b. Non-parametric: estimate $\Pr(\theta|\mathcal{D})$
     The number of parameters grows with the amount of training data, *e.g., kNN, SVM, decision tree*
     + no need strong assumption about the nature of data distribution
     - high computational cost

- **(Learning) Objective Function**
  A function that is being optimized during learning / training
  e.g., Least square, Likelihood, Logistic loss, Hinge loss, Cross-entropy, Gini impurity

- **Parameter Estimation** or **Algorithm**
  The process of quantifying uncertainty (i.e., parameters) about an unknown quantity estimated from a finite sample of data
  a. Analytical via Calculus & Algebra
     e.g., LS, ML, ERM, MAP, Bayes
  b. Numerical via Optimization methods
     e.g., Gradient descent, Newton's, BFGS, Lagrange multipliers, C4.5

Basic Concepts

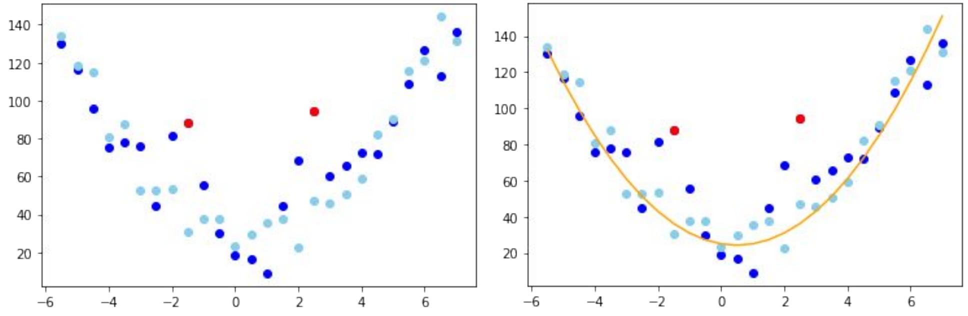


Underfitting

Overfitting

$$\mathrm{MSE}'^{*} = \mathbb{E}\left\{\mathrm{Bias}_{\mathcal{D}}\left[\hat{f}(x;\mathcal{D})\right]^2 + \mathrm{Var}_{\mathcal{D}}\left[\hat{f}(x;\mathcal{D})\right]\right\} + \sigma^2$$

- Strategies:
  a. Regularization, e.g., L1 norm, L2 norm
  b. k-fold Cross-validation
  c. Early stopping
  d. Dimensionality reduction
  e. Add more samples



No free lunch theorem!
"All models are wrong, but some models are useful." — George Box (1987)

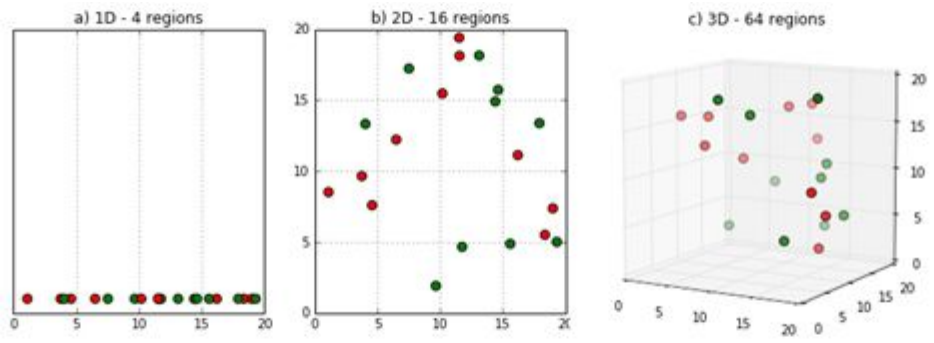
Type of Learning

- Supervised learning
  - Regression, *i.e., Y continuous*
  - Classification, *i.e., Y discrete or categorical*
- Unsupervised learning
  - Dimensionality reduction, *e.g., PCA, ICA, Factor analysis, AE*
  - Clustering, *e.g., Hierarchical, k-Means, DBSCAN, GMM*
  - Anomaly detection, *e.g., One-class SVM, Isolation forest, LOF*
  - Recommender system, *e.g., Content-based, CF*
  - Association rule
  - Topic modeling, *e.g., latent Dirichlet allocation*
  - Synthetic data generation, *e.g., VAE, GAN*
- Semi-supervised learning
- Reinforcement learning
- Active learning

- Learning approaches
  - **Discriminative**
    Learning a decision boundary in order to make prediction of unseen data.
    a. Supervised: $\Pr(Y|X;\theta)$
       e.g., linear regression, logistic regression, ANN, SVM, decision tree, kNN, ensembles
    b. Unsupervised: $\Pr(Z|X;\theta)$
       e.g., PCA, One-class SVM, Isolation forest
  - **Generative**
    Learning the probability distribution of training data to return a probability for a given example.
    a. Supervised: $\Pr(X,Y|\theta) = \Pr(Y|X;\theta)\Pr(X)$
       e.g., naïve bayes, Gaussian discriminant analysis, deep belief network
    b. Unsupervised: $\Pr(X|Z;\theta)$ OR $\Pr(X,Z|\theta)$
       e.g., latent Dirichlet allocation, VAE, GAN

Unsupervised Learning

Dimensionality Reduction



Curse of dimensionality

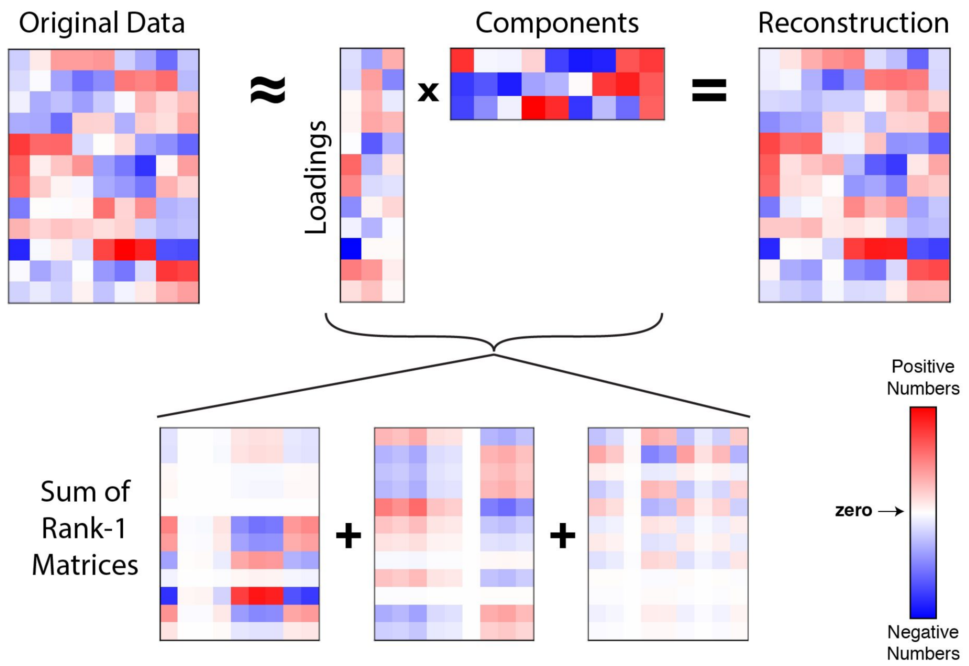
1. Less density (sparse)
2. Higher distance

- Principal Component Analysis (PCA)
- Factor Analysis
- Isomap
- t-Distributed Stochastic Neighbor Embedding (tSNE)
- Uniform Manifold Approximation and Projection (UMAP)
- Locally Linear Embedding (LLE)
- Kernel PCA
- Autoencoders

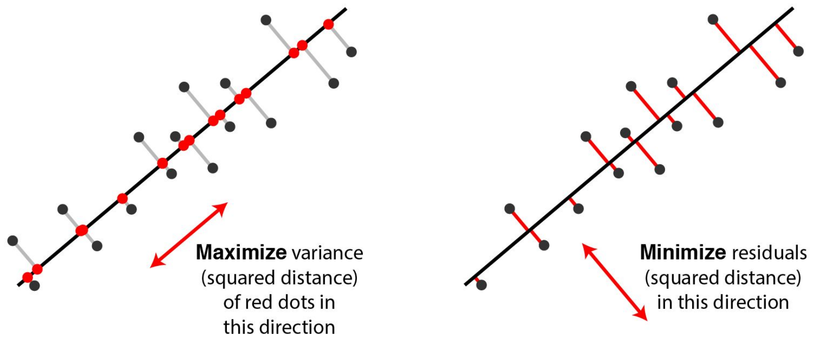
Principal Component Analysis

Principal Component Analysis (PCA):

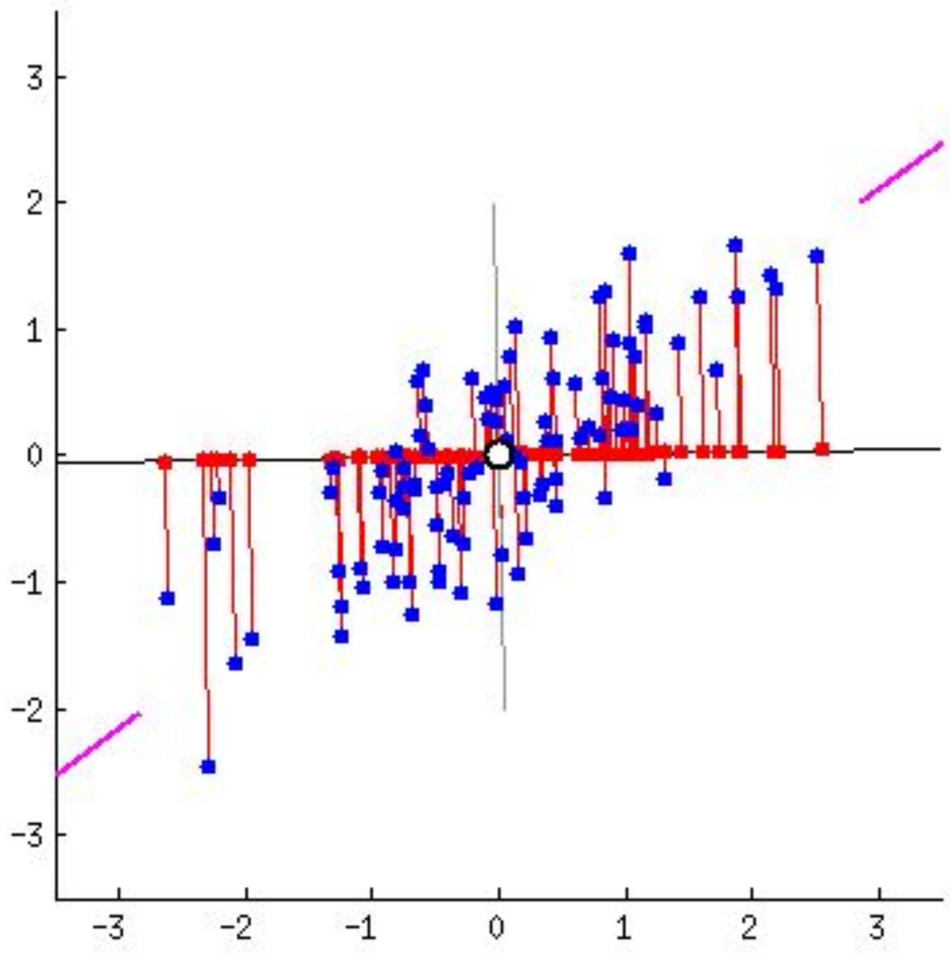
a. Linear dimensionality reduction technique for data visualization and/or speeding machine learning algorithms.
b. Extracting information from a high-dimensional space by projecting it into a lower-dimensional sub-space.
c. Using an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables (orthogonal to each other).
d. There is no guarantee that the new dimensions (principal components) are interpretable.
e. First principal component has a maximum variance.



Principal Component Analysis



1. The variation of values along this line should be maximal.
2. The reconstruction error (the connecting red line) should reach minimum



PCA Hands-On

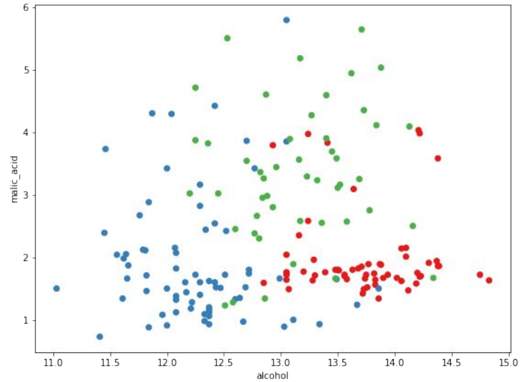
Wine Dataset

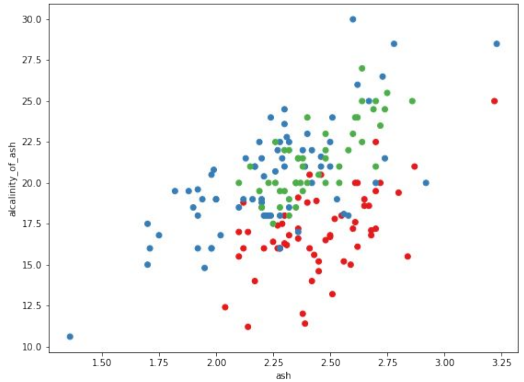
```python
import pandas as pd
from sklearn.datasets import load_wine
# instantiating
wine = load_wine()
# creating dataframe
df = pd.DataFrame(data=wine["data"], columns=wine["feature_names"])
# checking first six rows of dataframe
df.head()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69

```python
# Dimension of data
df.shape
(178, 13)
```

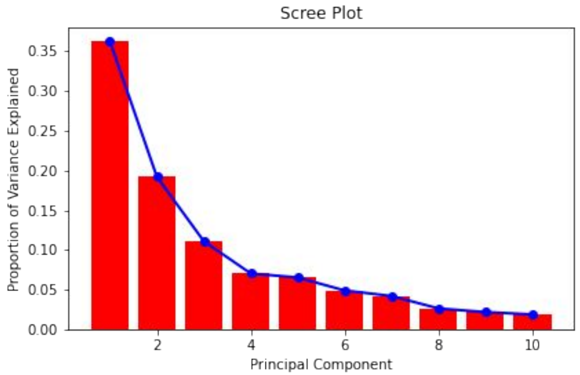
```python
# Data preprocessing
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(df)
scaled_df = scaler.transform(df)
```

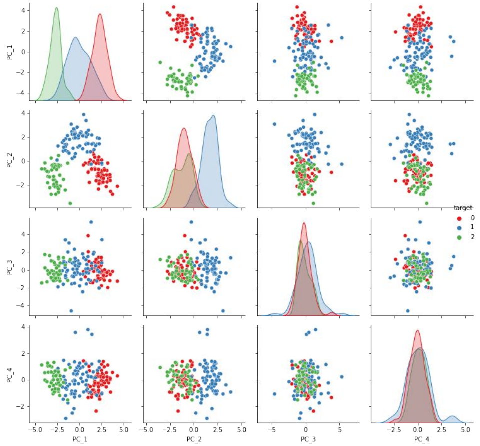





```python
# Running PCA
from sklearn.decomposition import PCA
# Let's say, components = 10
pca = PCA(n_components=10)
```

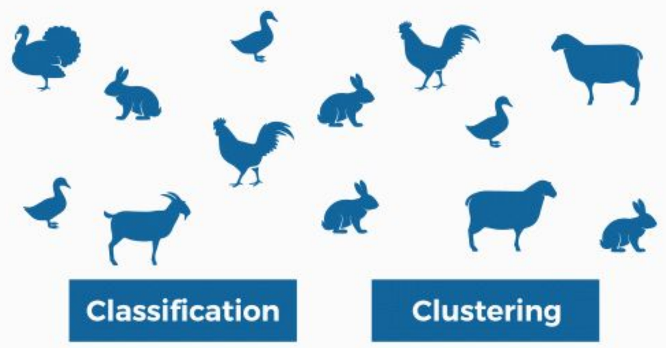
```python
pca.fit(scaled_df)
pca.explained_variance_ratio_[:4].sum()
0.735989990758993
```





```python
pca = PCA(n_components=4)
pca.fit(scaled_df)
x_pca = pca.transform(scaled_df)
df_pca = pd.DataFrame(data=x_pca, columns=["PC_1", "PC_2", "PC_3", "PC_4"])
df_pca["target"] = wine["target"]
```

Clustering



What do we need for clustering?

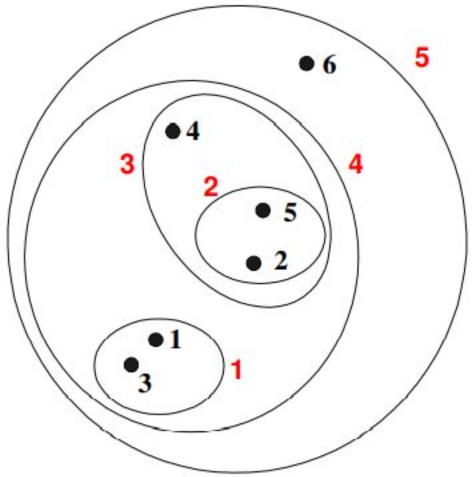
1. Proximity measure (similarity OR distance)
   e.g., Euclidean distance, Manhattan distance, Minkowski distance, Jaccard distance
2. Criterion function
   e.g., Intra-cluster cohesion / compactness (SSE), Inter-cluster separation / isolation
3. Algorithm

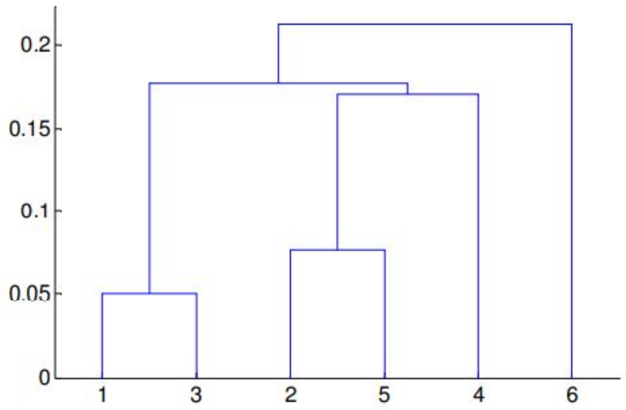
- Hierarchical
  - *e.g., Agglomerative, Divisive*
- Partitional
  - Distance-based:
    e.g., K-Means, k-Medoids, etc.
  - Density-based
    e.g., DBSCAN, etc.
  - Model-based
    e.g., Gaussian mixture, SOM
  - Graph-based
    e.g., Spectral clustering

Hierarchical clustering

Hierarchical algorithms find successive clusters using previously established clusters:

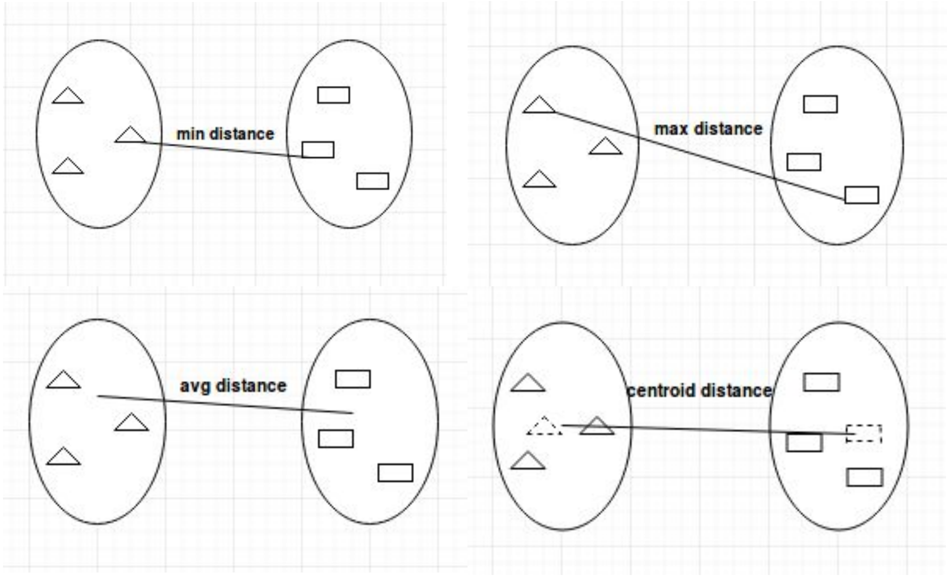
a. Agglomerative: begin with each element as a separate cluster and merge them into successively larger clusters
b. Divisive: begin with the whole set and proceed to divide it into successively smaller clusters





Four ways to measure inter-cluster distance:

a. Single-linkage (minimum distance)
b. Complete-linkage (maximum distance)
c. Average-linkage
d. Centroid-linkage
e. Ward's method

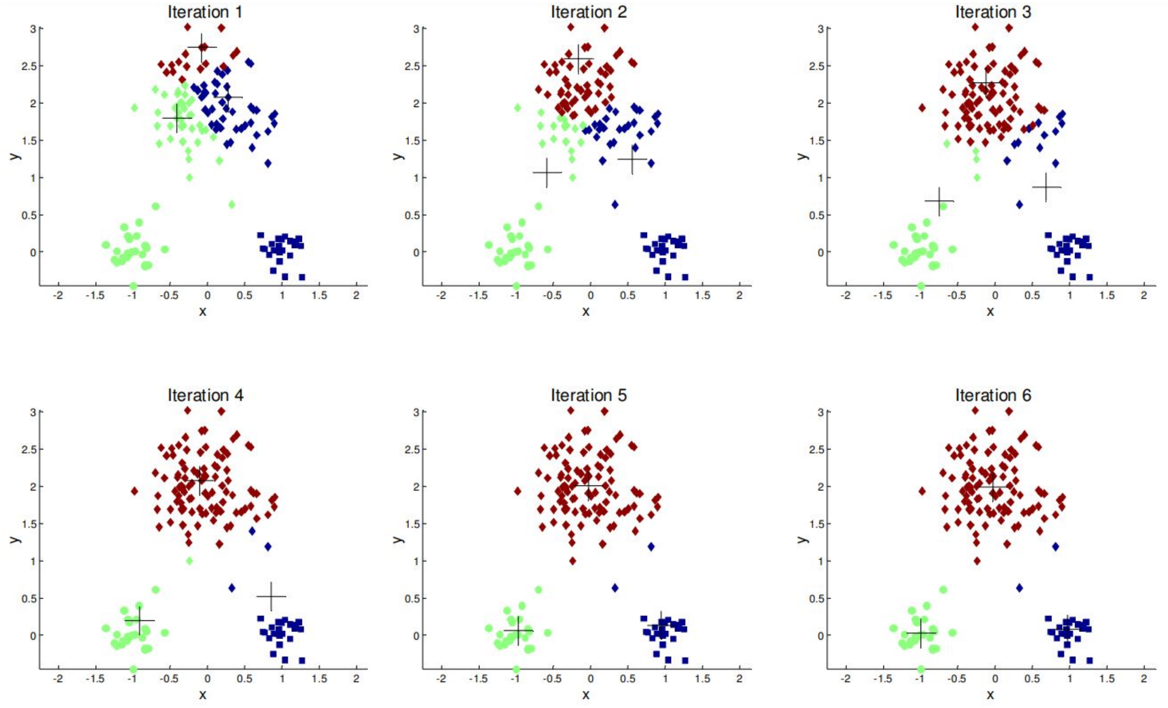


k-Means clustering

Partitional clustering: each instance is placed in exactly one of k non-overlapping clusters (normally required K as parameter).

k-Means clustering:

a. Each cluster is associated with a centroid (center point)
b. Each point is assigned to the cluster with the closest centroid
c. The objective is to minimize the sum of distance of the points to their respective centroid



k-Means variations:

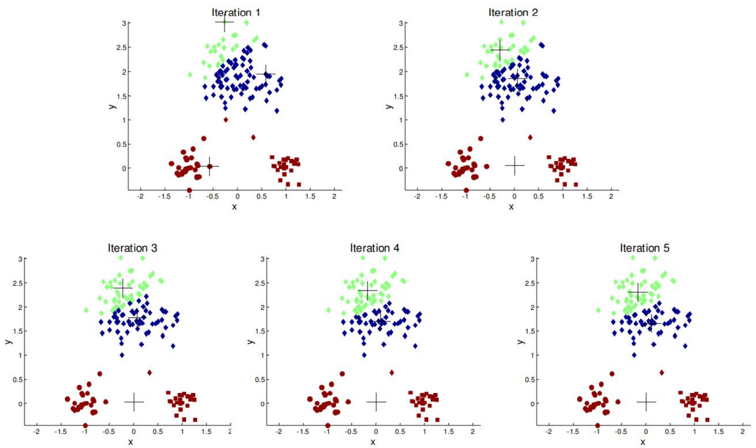
a. k-Medoids (the centroid of the cluster is defined to be one of the points in the cluster)
b. k-Centers (the objective is to minimize the maximum diameter (total distance between any two points in the cluster)

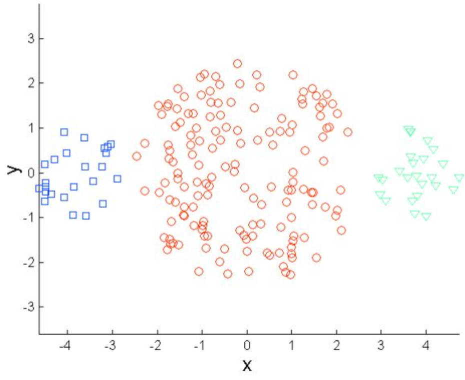
Strengths and Weaknesses

- Hierarchical clustering
  - **Strengths**
    a. No need to specify the number of clusters in advance
    b. Maps nicely onto human intuition for some domains
  - **Weaknesses**
    a. Do not scale well (longer time computation)
    b. Local optimum
    c. Interpretation of the results is very subjective

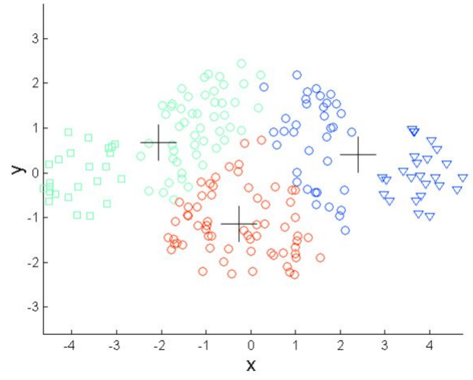
- k-Means
  - **Strengths**
    a. Easy to understand and to implement
    b. Intuitive objective function
    c. More efficient in terms of computation
  - **Weaknesses**
    a. Local optimum (if SSE is used)
    b. Only applicable for problems where mean/mode is defined
    c. Need to specify the number of clusters in advance
    d. Sensitive to outliers
    e. Sensitive to initial centroids
    f. Not suitable for discovering clusters with different sizes, densities, and non-convex shapes

Examples of k-Means Limitations

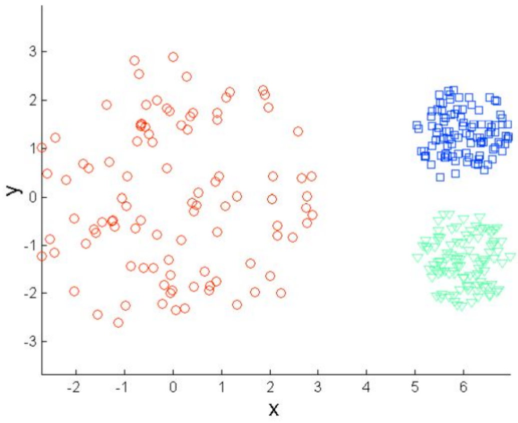




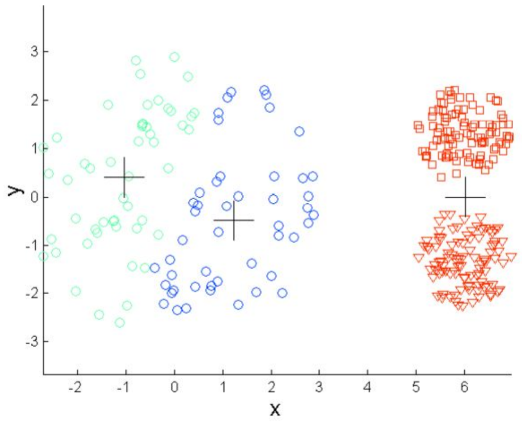
Original Points



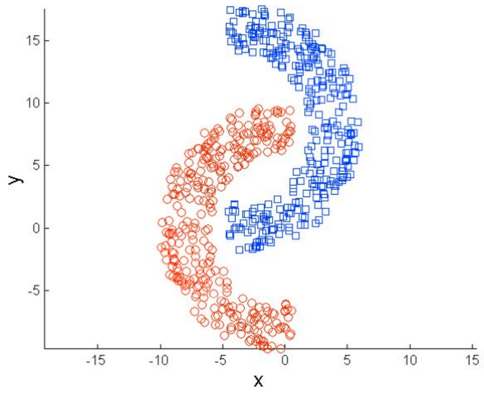
K-means (3 Clusters)



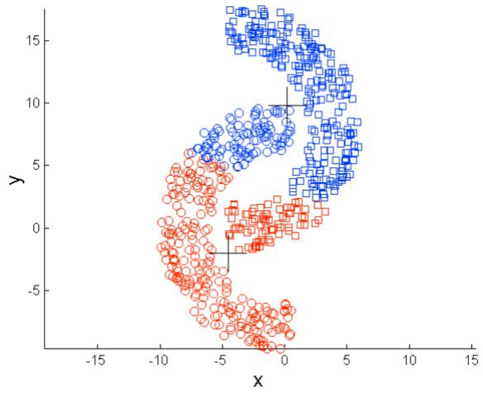
Original Points



K-means (3 Clusters)



Original Points



K-means (2 Clusters)

Performance Evaluation

- **Known Cluster Labels**

  Small sample size or large number of cluster

  a. Rand Index: [0,1]
  b. Adjusted Rand Index: [-1,1]
  c. Normalized Mutual Information: [-1,1]
  d. Adjusted Mutual Information: [-1,1]

  Large sample size or small number of cluster

  a. Homogeneity: [0,1]
  b. Completeness: [0,1]
  c. V-measure: [0,1]
  d. Fowlkes-Mallows: [0,1]

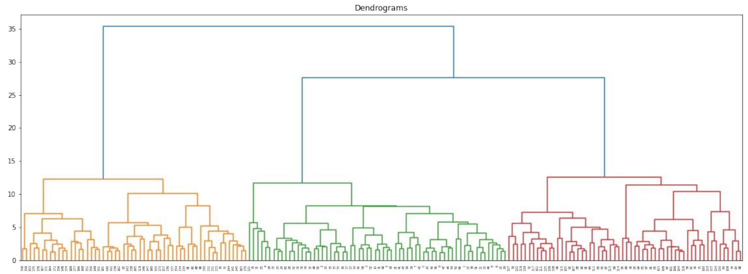
- **Unknown Cluster Labels**

  a. Silhouette Coefficient: [-1,1]
  b. Calinski-Harabasz Index / Variance Ratio Criterion
  c. Davies-Bouldin Index: [0,∞]

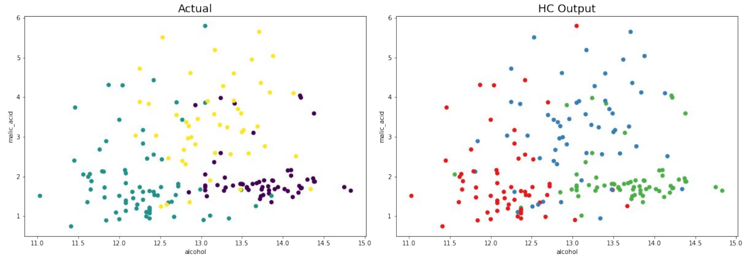
Clustering Hands-On

Wine Dataset

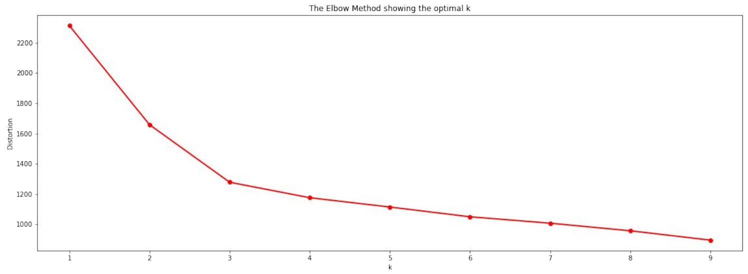
```python
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
```



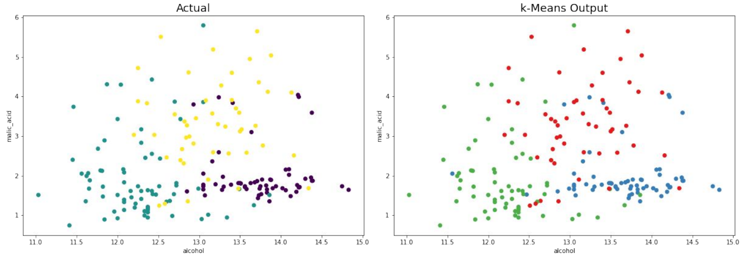

```python
# Train and predict cluster using Hierarchical clustering
hc_model = AgglomerativeClustering(n_clusters=3, affinity="euclidean", linkage="ward")
df["hc_output"] = hc_model.fit_predict(scaled_df)
# Insert the ground truth
df["target"] = wine["target"]
# Measure the model performance: 1.0 is perfect, around 0.0 is bad
from sklearn.metrics import adjusted_rand_score
adjusted_rand_score(df["target"], df["hc_output"])
0.7899332213582837
```




```python
from sklearn.cluster import KMeans
distortions = []
K = range(1,10)
for k in K:
    kmean_model = KMeans(n_clusters=k)
    kmean_model.fit(scaled_df)
    distortions.append(kmean_model.inertia_)
```




```python
kmean_model = KMeans(n_clusters=3)
kmean_model.fit(scaled_df)
df["kmean_output"] = kmean_model.predict(scaled_df)
adjusted_rand_score(df["target"], df["kmean_output"])
0.8974949815093207
```



Task

Compare the performance between:

1. k-Means clustering for Cancer dataset from sklearn
2. k-Means clustering for ***n*** Principal Components of Cancer dataset from sklearn

Evaluation:

1. Adjusted Rand Index (ARI)
2. Time computation (*hint*: use `%%timeit` in the top of the cell)

Thank You

Appendices

Bias-Variance Trade-off

$$y = f(x) + \varepsilon$$

$$\begin{aligned}
\mathbb{E}\left[\left(y - \hat{f}(x;\mathcal{D})\right)^2\right] &= \mathbb{E}\left[\left(f(x) + \varepsilon - \hat{f}(x;\mathcal{D})\right)^2\right] \\
&= \mathbb{E}\left[\left(f(x) - \hat{f}(x;\mathcal{D})\right)^2\right] + \mathbb{E}[\varepsilon^2] + 2\mathbb{E}\left[\left(f(x) - \hat{f}(x;\mathcal{D})\right)\varepsilon\right] \\
&= \mathbb{E}\left[\left(\left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right) + \left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)\right)^2\right] + \mathbb{E}[\varepsilon^2] + 2\mathbb{E}\left[\left(f(x) - \hat{f}(x;\mathcal{D})\right)\right]\mathbb{E}[\varepsilon] \\
&= \mathbb{E}\left[\left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)^2\right] + 2\mathbb{E}\left[\left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)\right]\mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)\right] + \mathbb{E}[\varepsilon^2] \\
&= \left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)^2\right] + 2\left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)\mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)\right] + \mathbb{E}[\varepsilon^2] \\
&= \left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)^2\right] + 2\left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right) + \mathbb{E}[\varepsilon^2] \\
&= \left(f(x) - \mathbb{E}[\hat{f}(x;\mathcal{D})]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x;\mathcal{D})] - \hat{f}(x;\mathcal{D})\right)^2\right] + \mathbb{E}[\varepsilon^2] \\
&= \mathrm{Bias}[\hat{f}(x;\mathcal{D})]^2 + \mathrm{Var}[\hat{f}(x;\mathcal{D})] + \sigma^2
\end{aligned}$$