THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Business Analytics (MM 5425)

L9. Clustering & Nearest Neighbor

Dr. Yue (Katherine) FENG

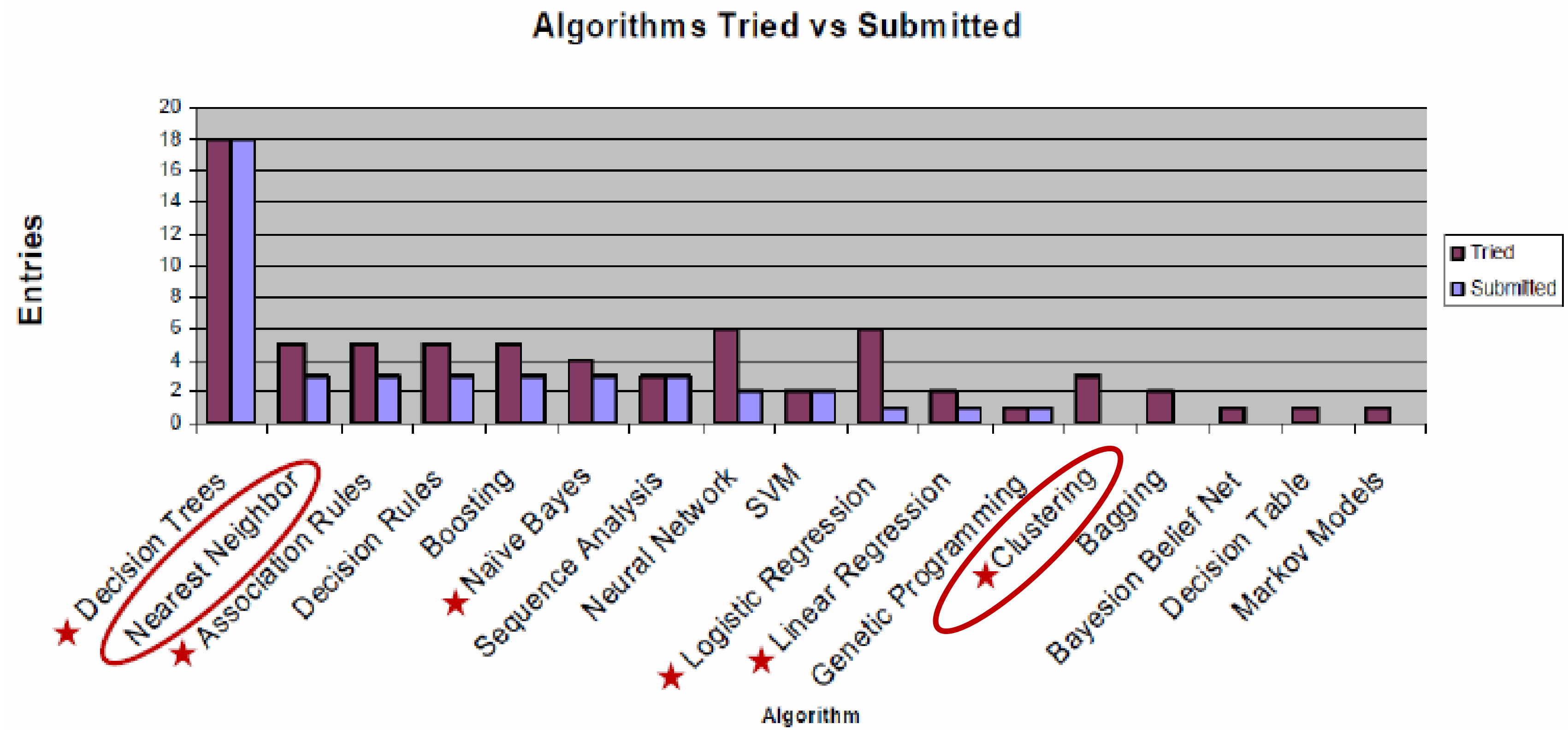
Recap: Unsupervised Learning

> How do I find things that occur together more than I might expect by chance?
  - **Associations (relationship between columns)**

> How do I find groupings of similar things?
  - **Clustering (relationship between rows)**

> **Key: how the data is constructed !**

Commonly Used Algorithms



Clustering

What is Clustering for?

> Example 1: Group people of similar sizes together to make "small", "medium" and "large" T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all

> Example 2: In marketing, segment customers according to their similarities.
  - To do targeted marketing

> Example 3: Given a collection of text documents, we want to organize them according to their content similarities.

What do We Mean by "Similar"?

- We describe a customer using 3 variables: {Age, Income, No. of purchases per day}

Customer 1:
{35, 700,000, 10.5}
Customer 2:
{45, 750,000, 8.2}
Customer 3:
{25, 300,000, 2.5}

Which pair of customers is more similar?

Key: Define similarity/distance metric between objects

Distance Measure: Euclidean Distance

> The Euclidean distance, d_{ij}, which between two records, *i* and *j*, with *k* attributes, is defined by

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ik} - x_{jk})^2}$$

The Euclidean distance between Customer 1 and 2:

$$d_{12} = \sqrt{(35 - 45)^2 + (700000 - 750000)^2 + (10.5 - 8.2)^2}$$

Normalizing Numerical Attributes

> Euclidean distance is highly influenced by the scale of each attribute, so that attributes with larger scales have a much greater influence over the total distance.

> Normalize numerical attributes, e.g., z-score

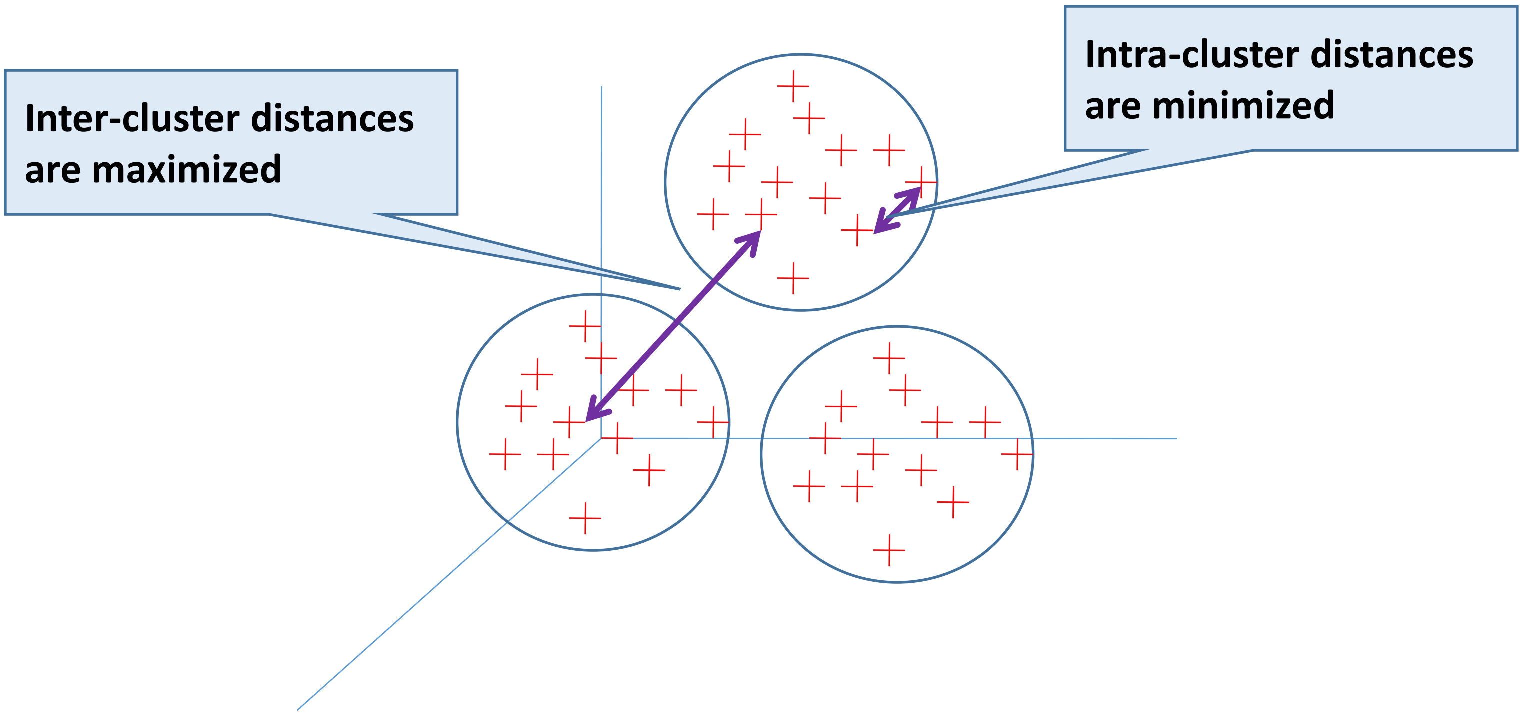
Sales dominating distance computation

Company	*Fixed*	*RoR*	*Cost*	*Load*	*Demand*	*Sales*	*Nuclear*	*Fuel Cost*
Arizona Public Service	1.06	9.2	151	54.4	1.6	9,077	0	0.628
Boston Edison Co.	0.89	10.3	202	57.9	2.2	5,088	25.3	1.555
Central Louisiana Co.	1.43	15.4	113	53	3.4	9,212	0	1.058
Commonwealth Edison Co.	1.02	11.2	168	56	0.3	6,423	34.3	0.7
Consolidated Edison Co. (NY)	1.49	8.8	192	51.2	1	3,300	15.6	2.044
Florida Power & Light Co.	1.32	13.5	111	60	−2.2	11,127	22.5	1.241
Hawaiian Electric Co.	1.22	12.2	175	67.6	2.2	7,642	0	1.652

Normalized(sales) = (9077-mean_sales)/std_sales

Clustering: Main Idea

> Create clusters of records to achieve:
  - Maximum similarity between records within a cluster
  - Maximum dissimilarity between records of different clusters



K-Means Clustering

> Most popular and simplest: ***K*-Means**

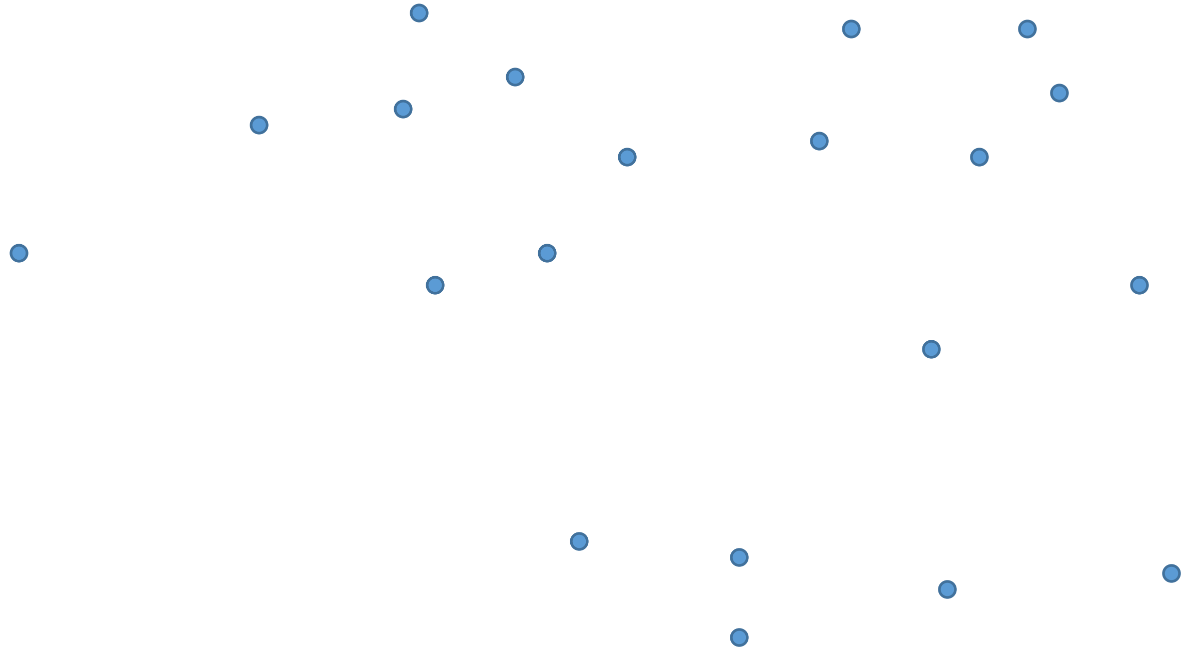
> Each cluster is associated with a centroid (center point).

> Each point is assigned to the cluster with the closest centroid.

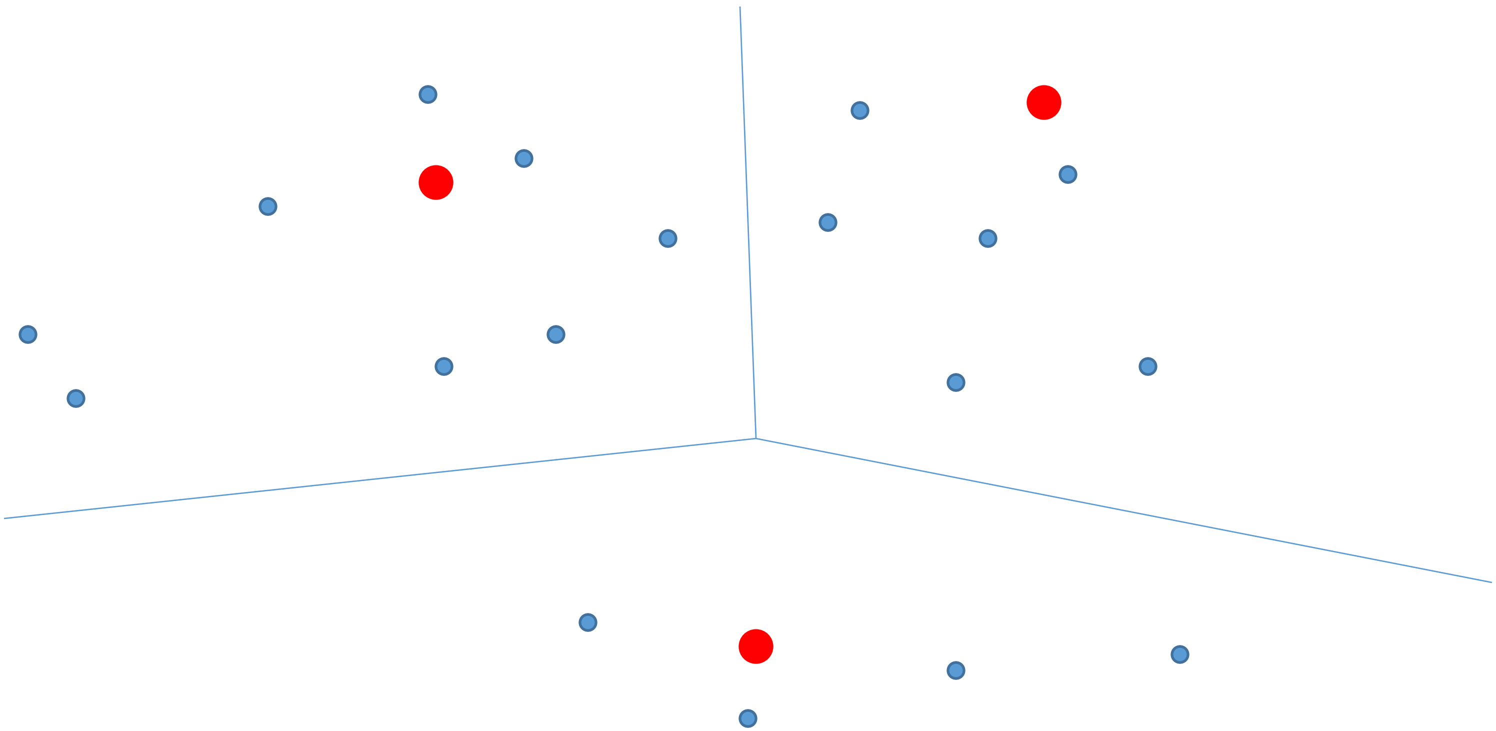
> Number of clusters ***k*** must be specified.

> **Objective:** minimize the sum of squared distances (SSD) to the *k* centers.

An Example: *3*-Means (*k*=3)

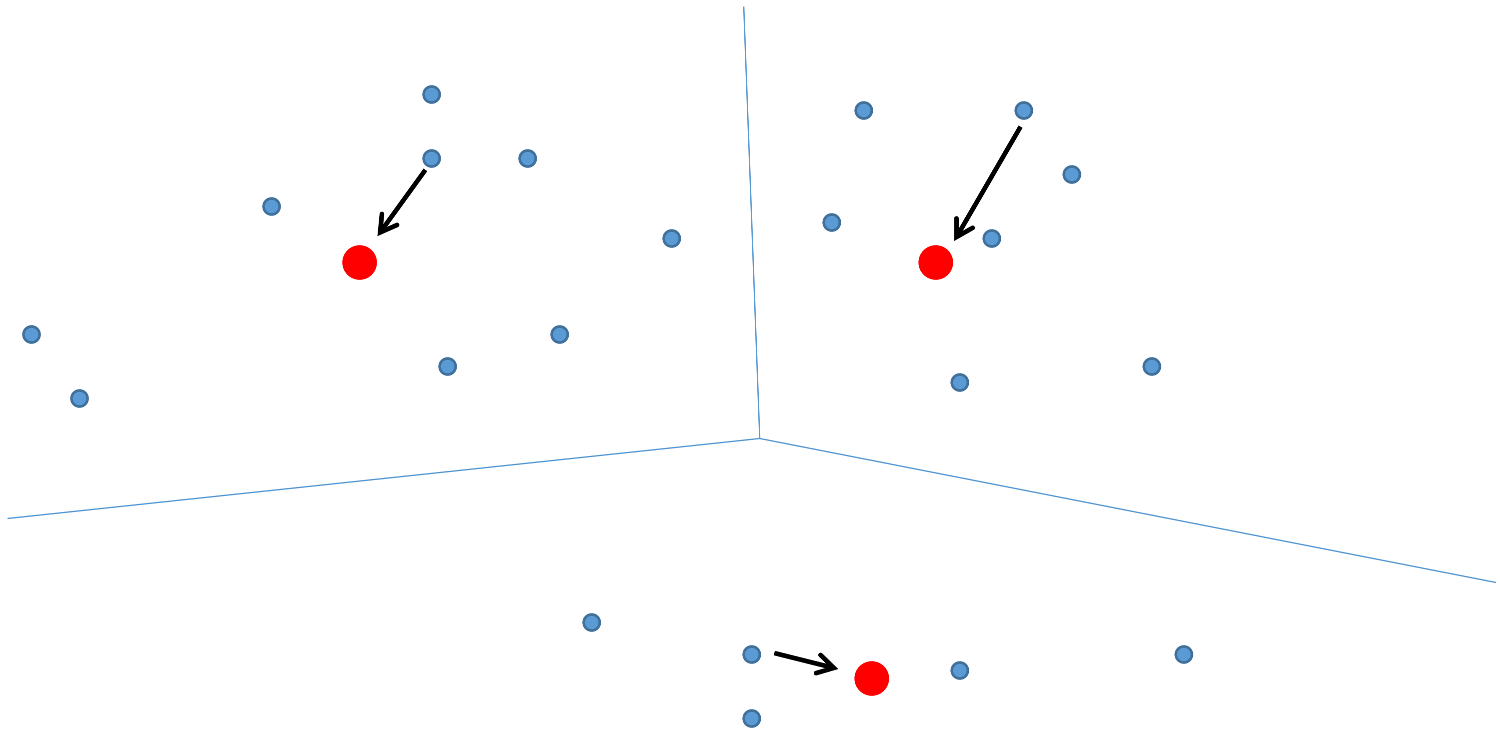


An Example: 3-Means (k=3)



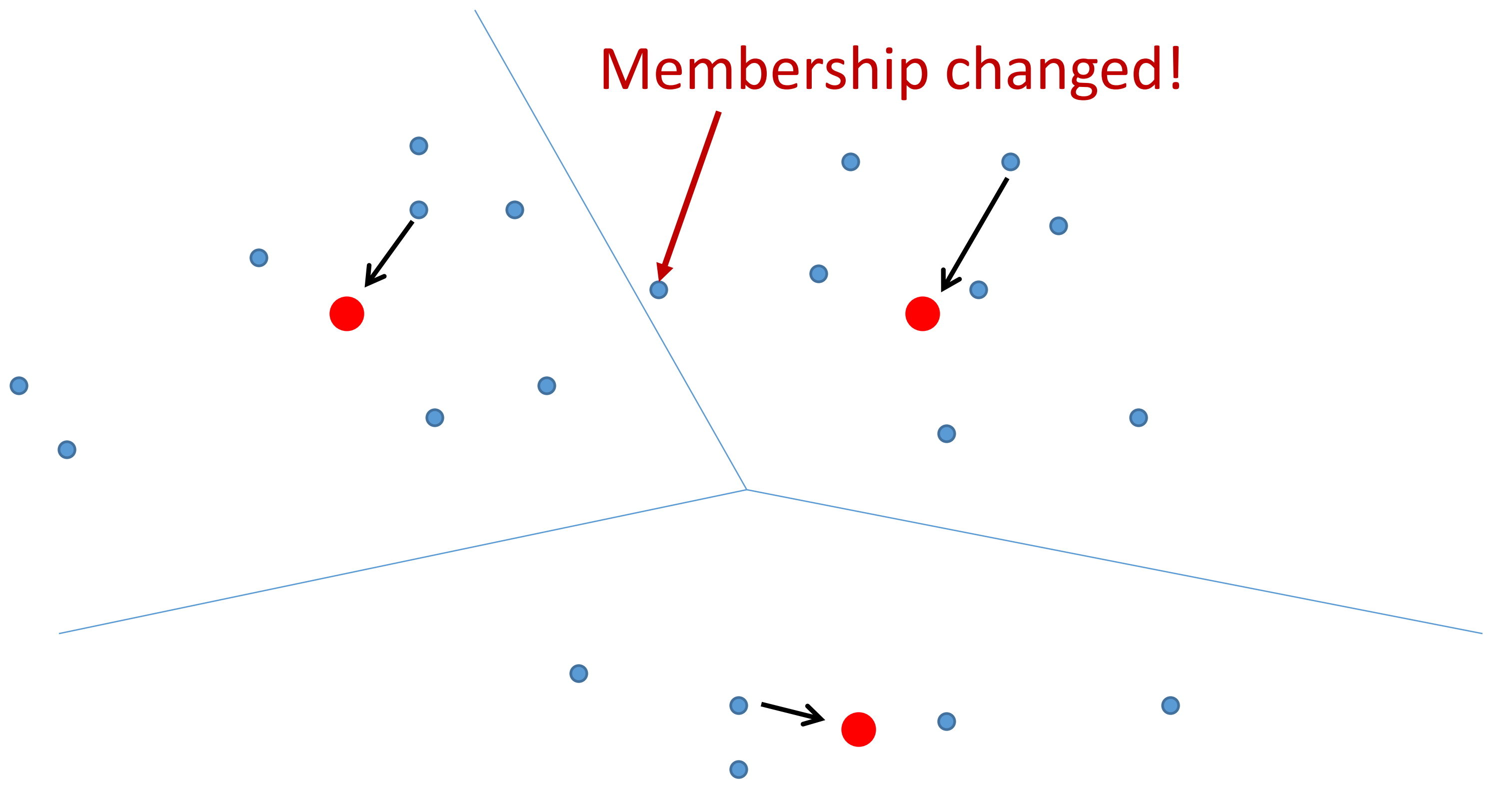
Randomly assign *k* centroids
Assign each example to the closest centroid

An Example: 3-Means (k=3)



Compute new centroids (note: the new centroids may not be any points in the data.)

An Example: 3-Means (k=3)



Assign each example to the closest centroid

K-Means Algorithm

1. Decide on a value for *k*.
2. Randomly initialize the *k* cluster centers.
3. Decide the cluster memberships of the N objects by assigning them to the **closest** cluster centers.
4. Re-estimate the *k* cluster centers, by **averaging** examples in each cluster.
5. If none of the N objects changed membership in the last iteration, exit. Otherwise goto **step 3**.

Several Issues

> Sensitive to the selection of initial center

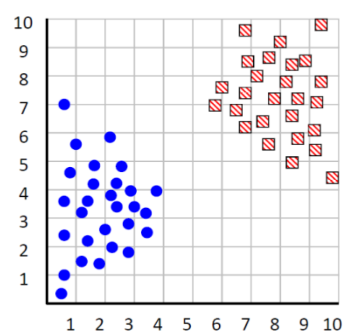
> Sensitive to noisy data and outliers

> Need to choose *k*, the number of clusters, in advance

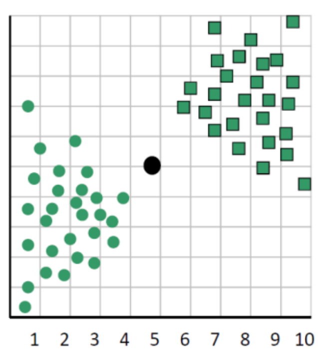
> **Question: How do I choose *k* for *k*-means?**

How to Choose the Number of Clusters (*k*)?

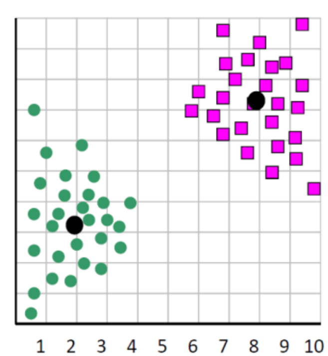
- In general, this is an unsolved problem.
- One approach: use sum of squared distances (SSD) as the objective and select the best *k*



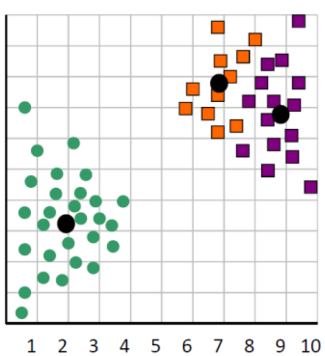
k = 1, SSD = 873



k = 2, SSD = 173.1

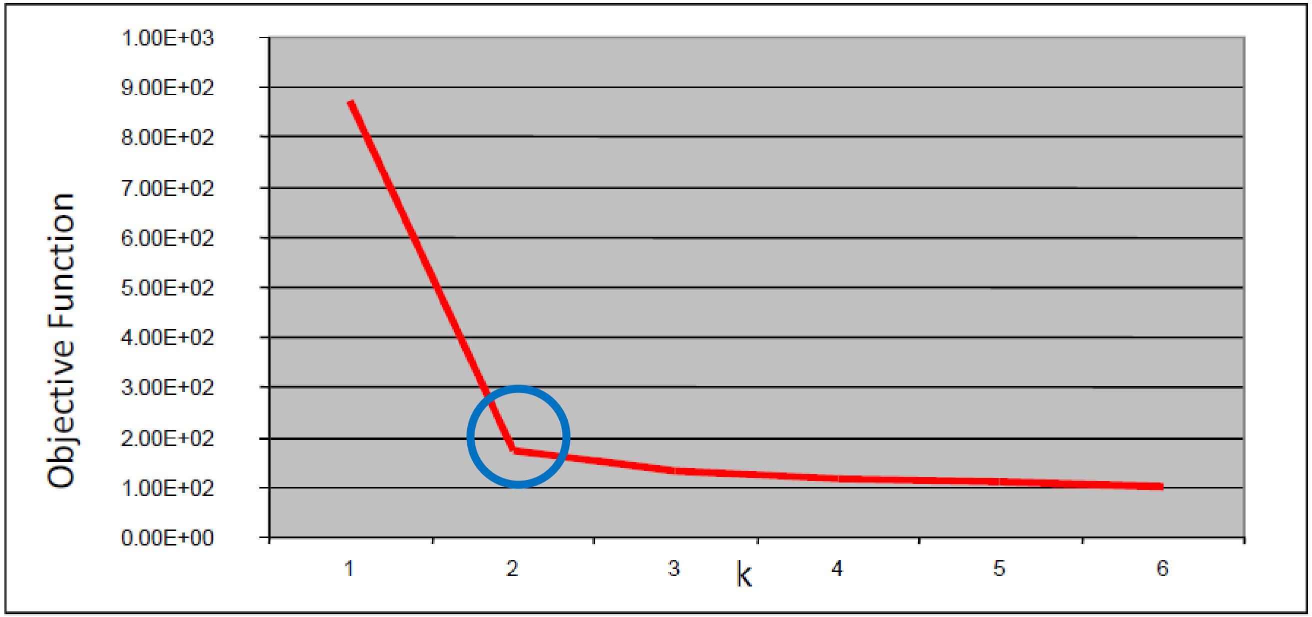


k = 3, SSD = 133.6



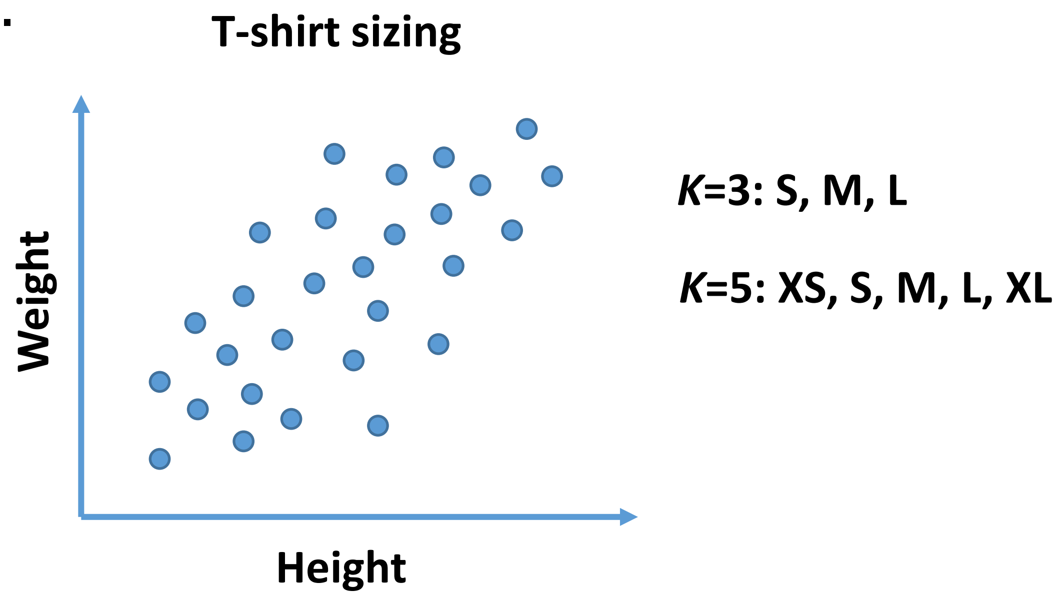
How to Choose the Number of Clusters (*k*)? -- Elbow Method

- Plot SSD values for incremental *k*s.
- Select the *k* where SSD has the large reduction (reduction is less thereafter).
- We also call this technique as "elbow method".



Practical Use: Choose the Value of *K*

- Sometimes you are running *K*-means to get clusters for some later/downstream use.
- Evaluate *K*-means based on how well it performs for that later purpose.

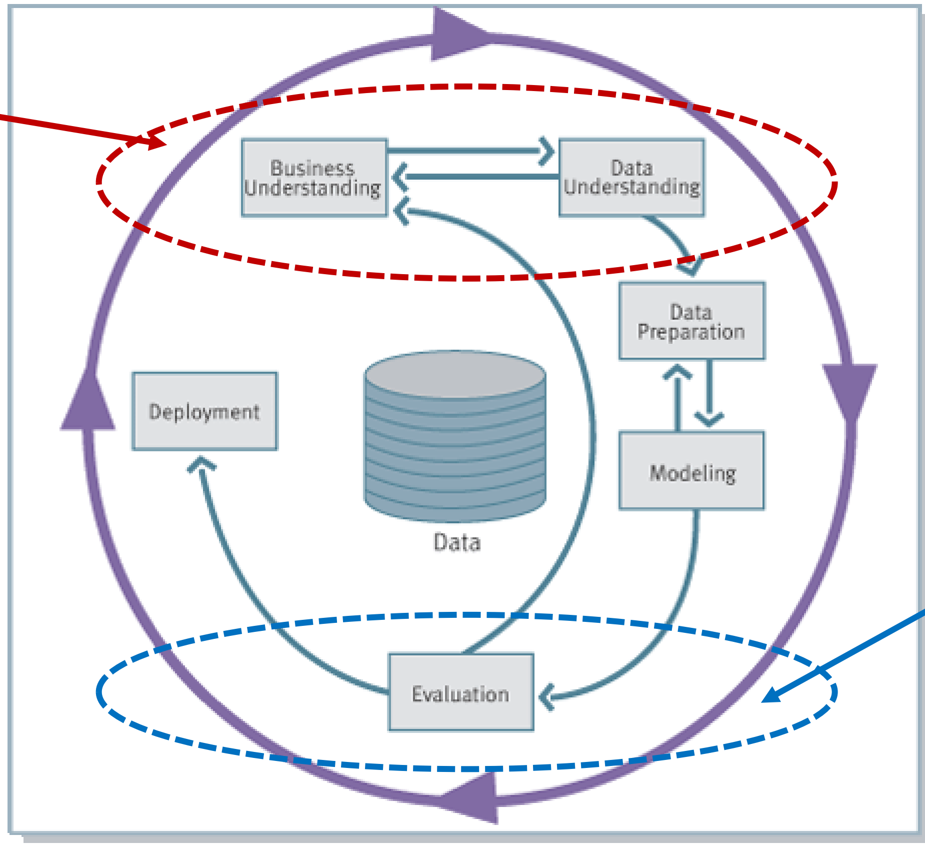


Evaluating Clusters

- What does it mean to say that a cluster is “good”?
- Clusters should have members that have a high degree of similarity (e.g., SSD).
- Domain knowledge evaluation.
- Check the centroids’ values to get some insights.

Supervised vs. Unsupervised Learning

Generally, supervised learning requires more effort and creativity in the formulation of the business problem as one of our standard problems (e.g., classification, regression).



Evaluation is much more straightforward for supervised learning, but requires more effort and creativity for unsupervised learning.

Nearest Neighbor

Recap: Similarity and Distance

> The key concept of clustering is similarity, formulated as a numeric distance between data instances.

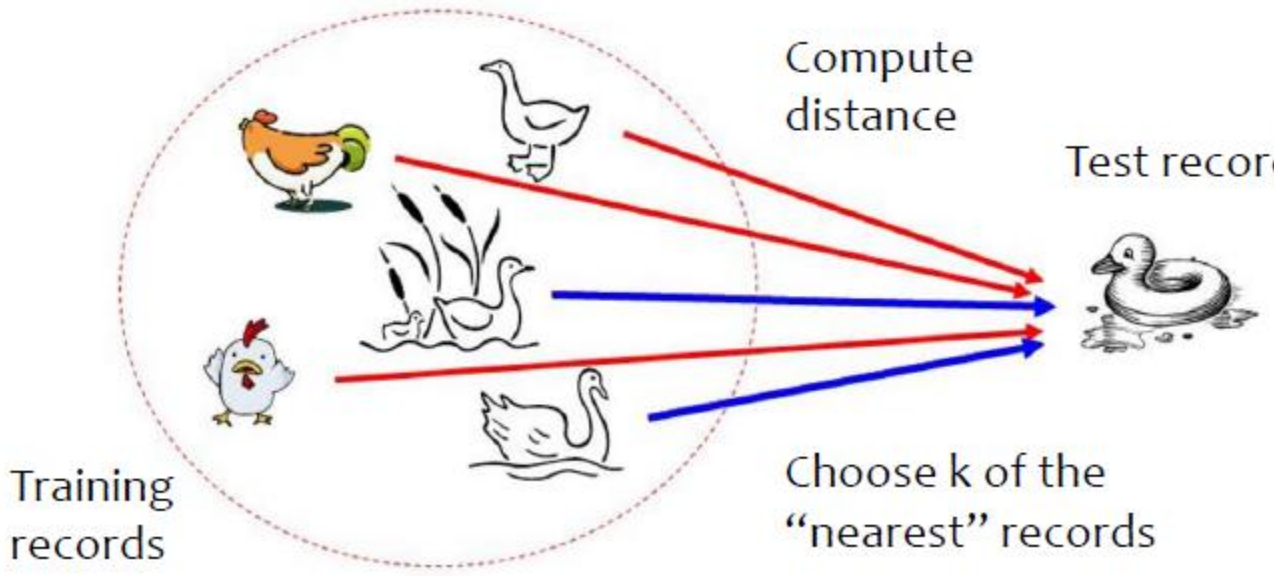
> Clustering: groups data instances based on similarity (unsupervised learning)

> Another application based on similarity for classification/regression (supervised learning)

- ***K*-Nearest Neighbor (KNN)**

Nearest Neighbor Classification: The Idea

> If it walks like a duck, quacks like a duck, then it's probably a duck.



An Example

No response

Response

No response

Response

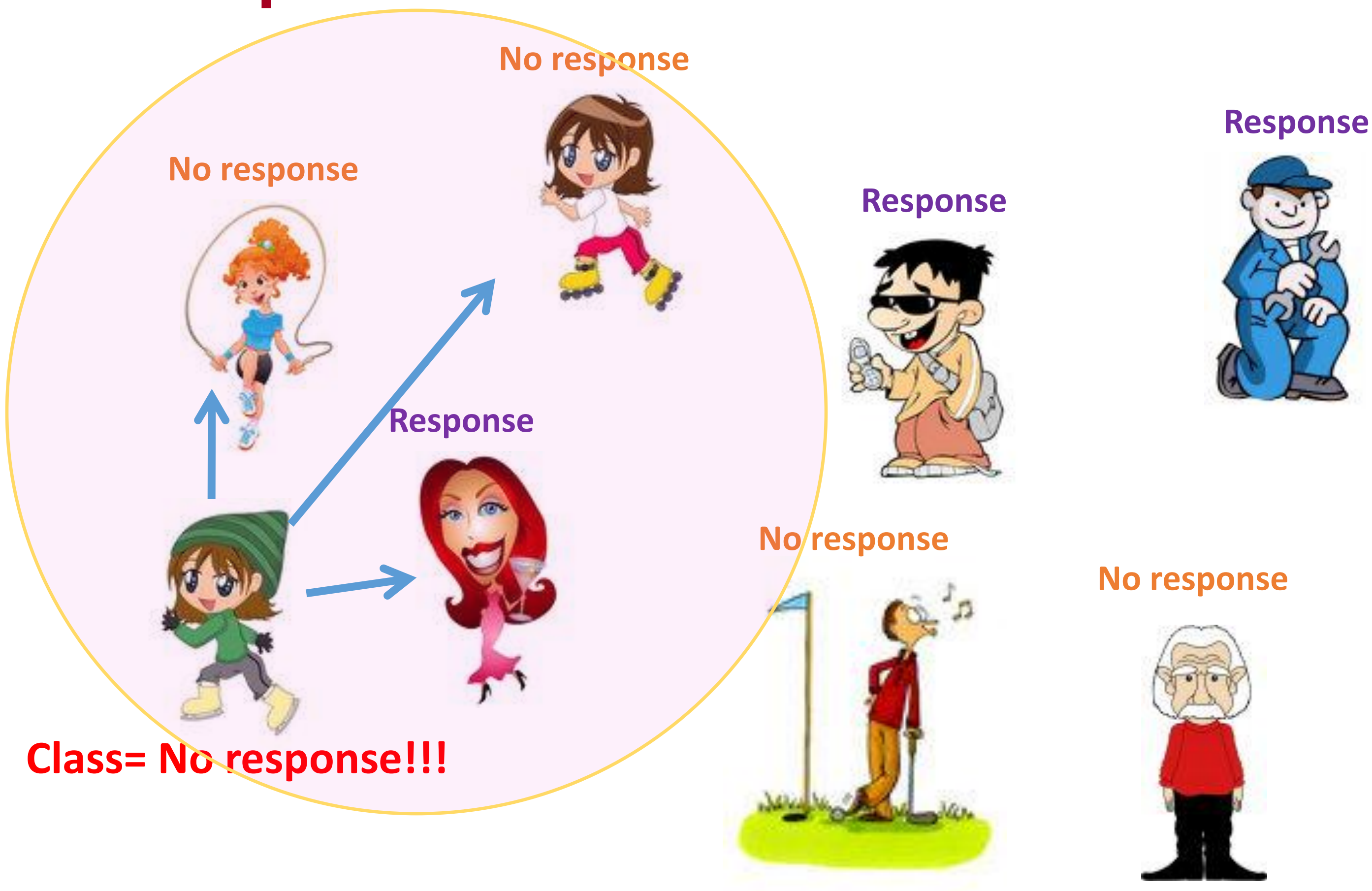
Response

No response

No response

Response or Not???

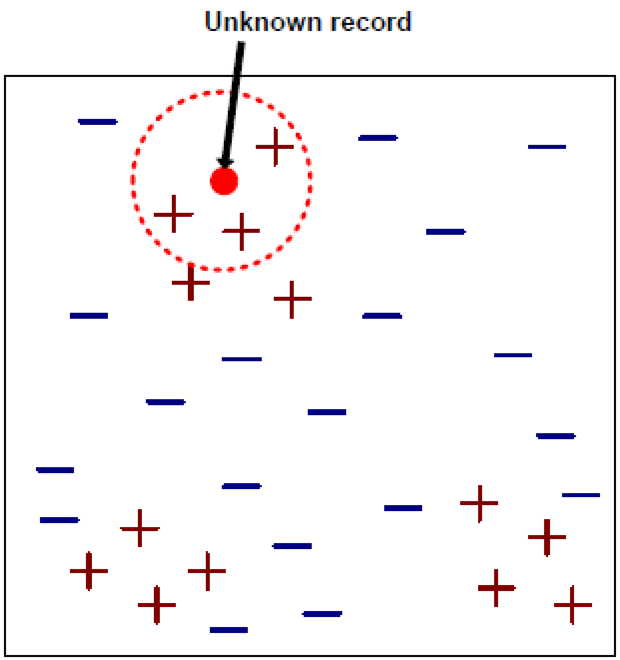
An Example


No response
No response
Response
Response
Response
No response
No response
Class= No response!!!

K-Nearest Neighbor Classification (KNN)

> To classify an unknown example:

1. Calculate distances between the example and all examples in training data
2. Identify *k* nearest neighbors
3. Use class labels of *k* nearest neighbors to determine the class label of unknown example (e.g., by taking majority vote, proportion, or weighted average)



Two Problems in KNN Algorithm

> How to calculate the distances between examples?

> How to choose the value of *K* - the number of nearest neighbors to be retrieved?

Distance

> "Similarity" is measured by **Euclidean distance** between two examples.

Rachel:
Age=41
Income=215K
No. of credit cards=3

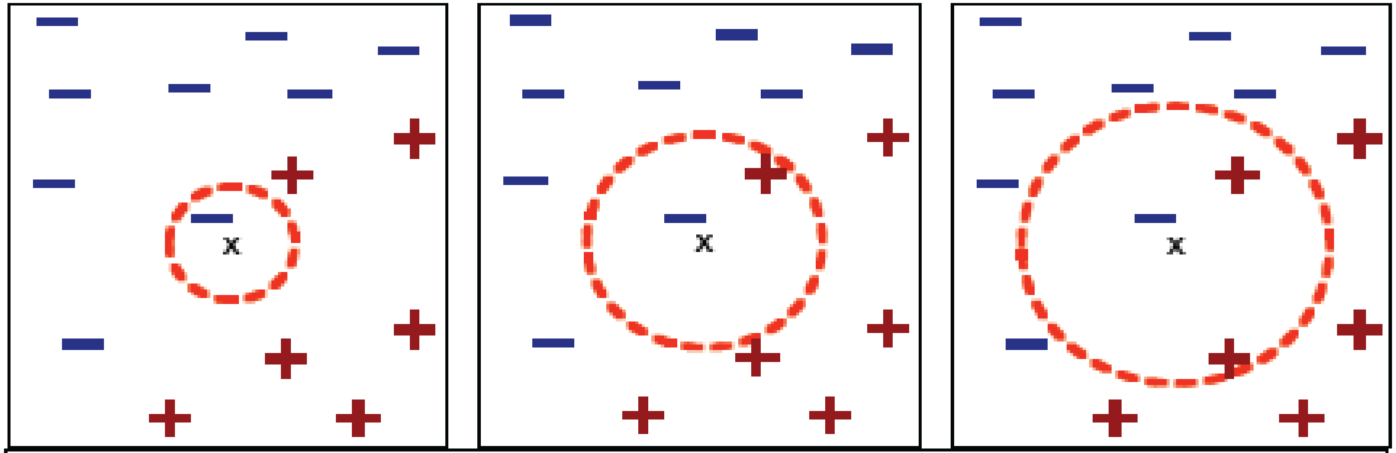
John:
Age=35
Income=95K
No. of credit cards=2

$$Distance(\text{John, Rachel}) = \sqrt{(35-41)^2 + (95\text{K} - 215\text{K})^2 + (2-3)^2}$$

> Attributes have to be scaled to prevent distance measures from being dominated by one of the attributes

- Feature normalization (z-score): rescale features to have zero mean and unit variance.

Changing *K* for Nearest Neighbors



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Changing *K* in the algorithm may change the predicted class label of the example.

The Selection of *K*

> If *K* is too small, sensitive to noise points

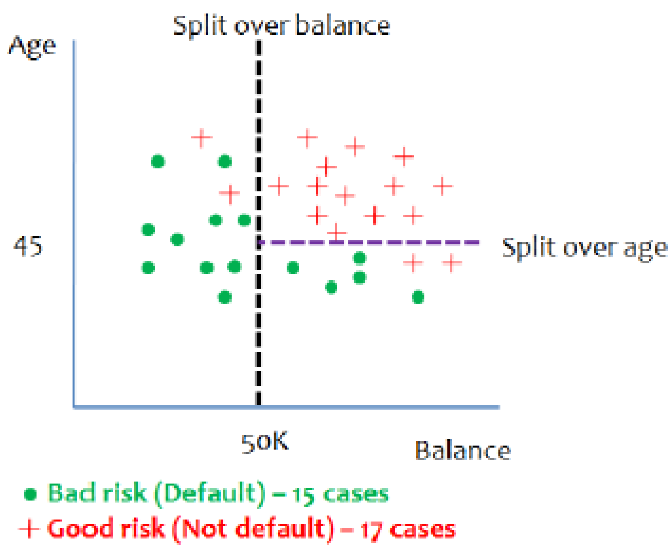
> If *K* is too large, include too many neighbors which may not be relevant

> Think about two extreme cases:

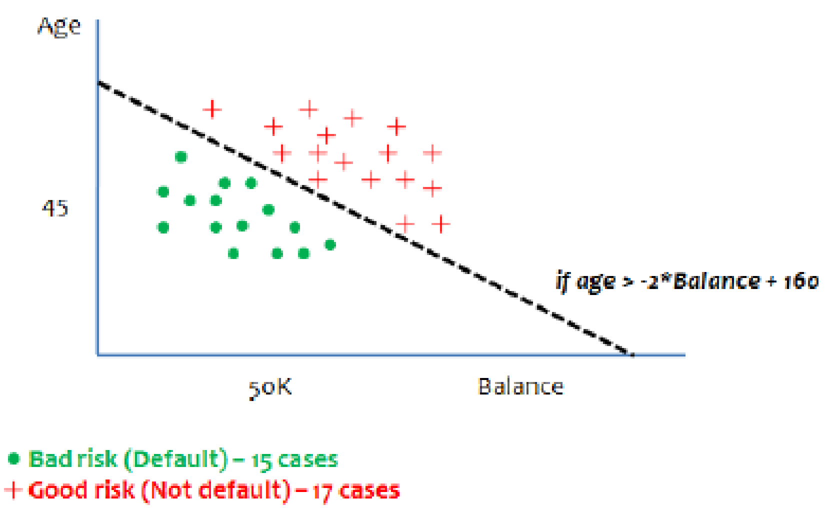
- *K*=1
- *K*=N where N is the number of examples in the training data

> Use **weighted average**, put more weight on closer neighbors

Geometric Interpretation

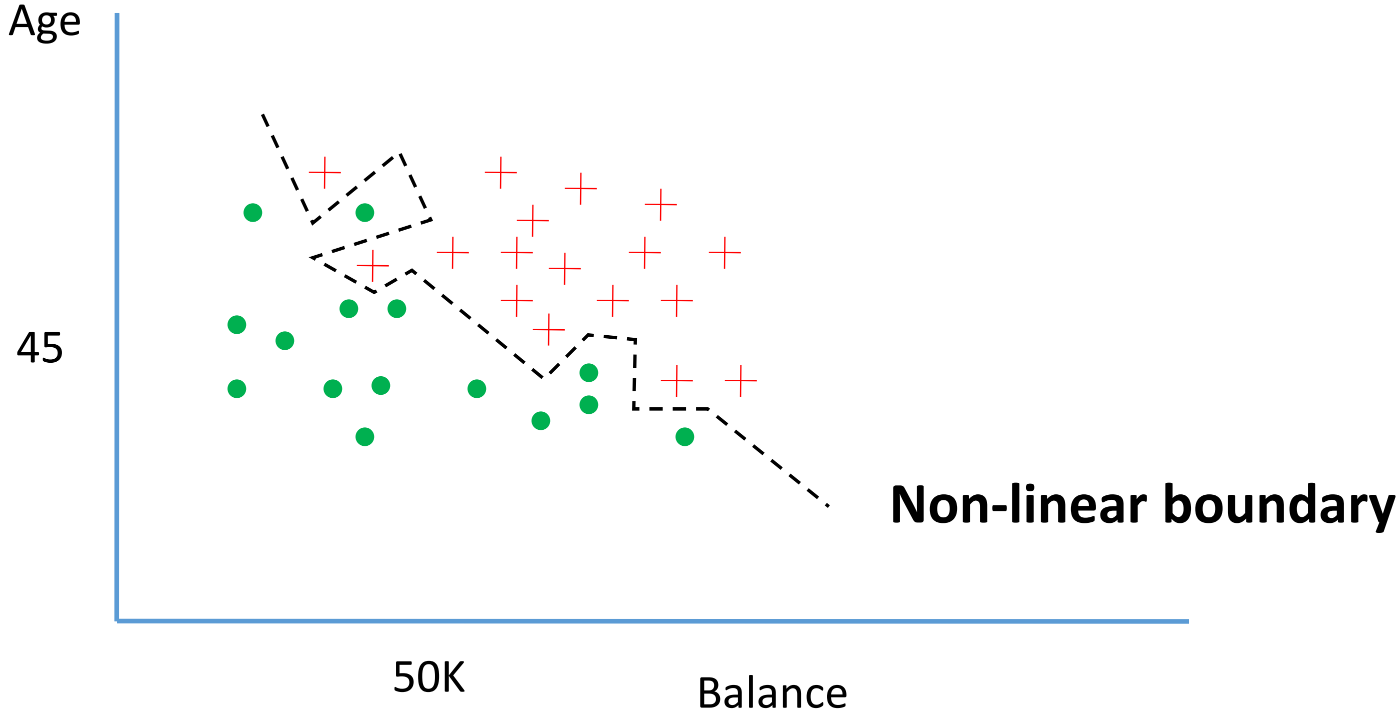


Decision tree classifier



Logistic regression classifier

How Does 1-NN Classifier Partition Space?



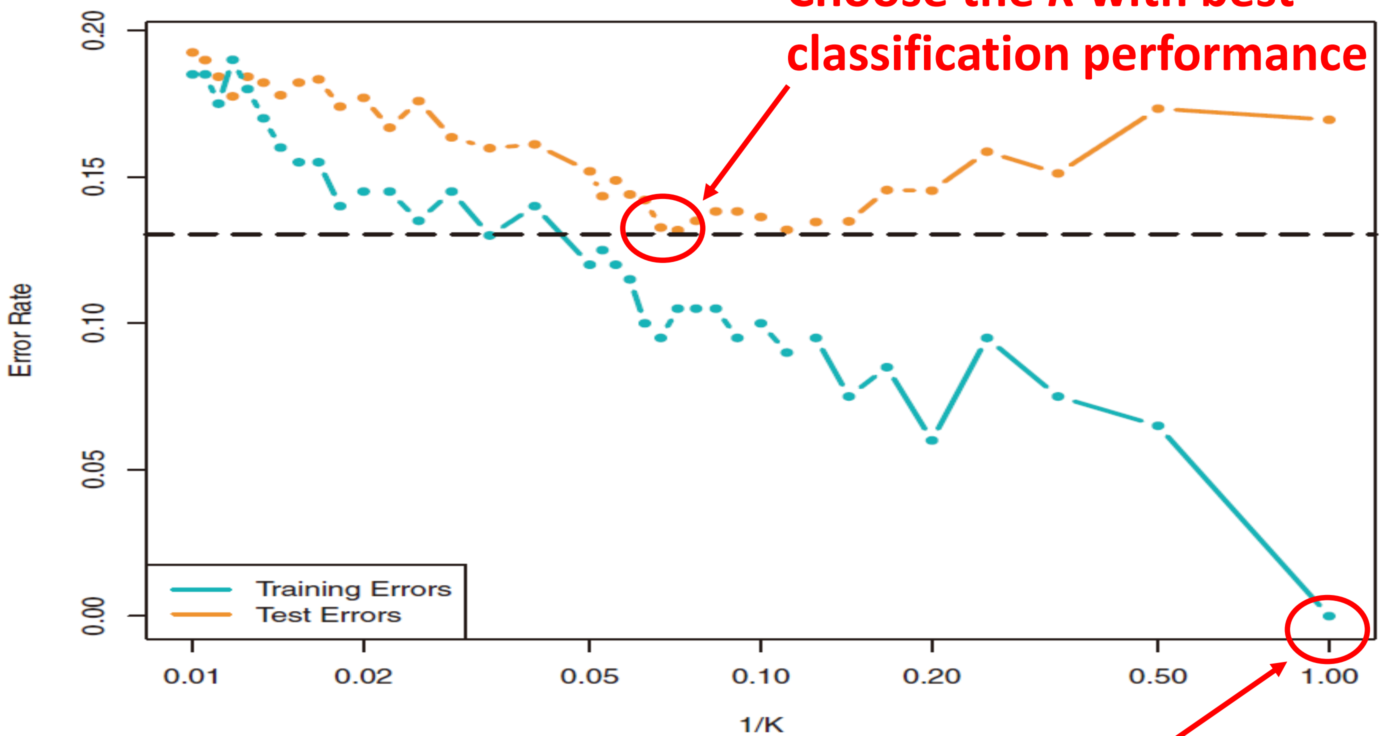
● Bad risk (Default) – 15 cases

+ Good risk (Not default) – 17 cases

The Selection of *K*

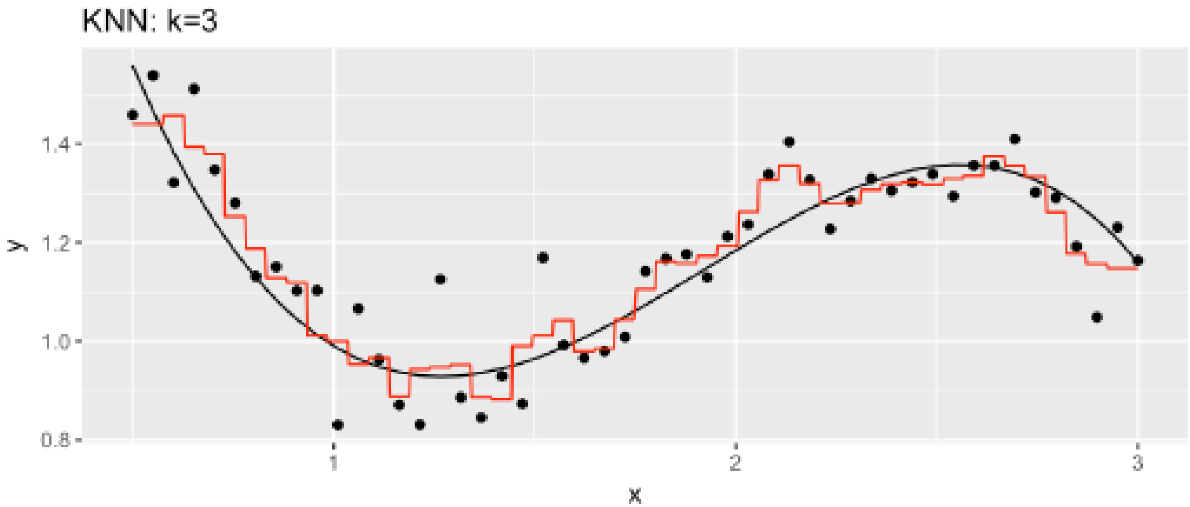
We want to balance between overfitting (*K*=1) and no learning (*K*=N).

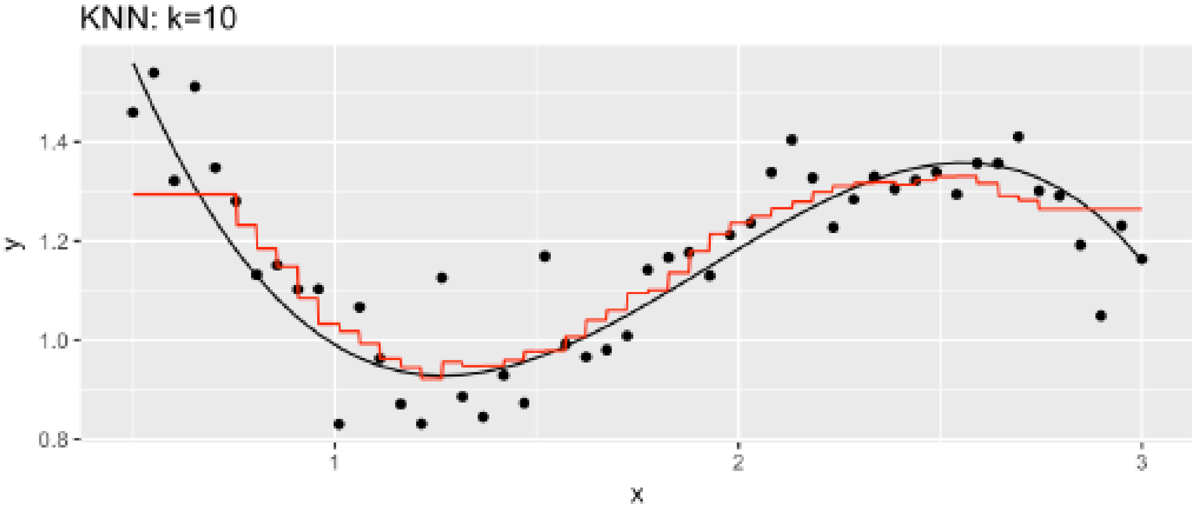
Choose the *K* with best classification performance



When *K*=1, zero error! Training data have been memorized...

KNN for Regression

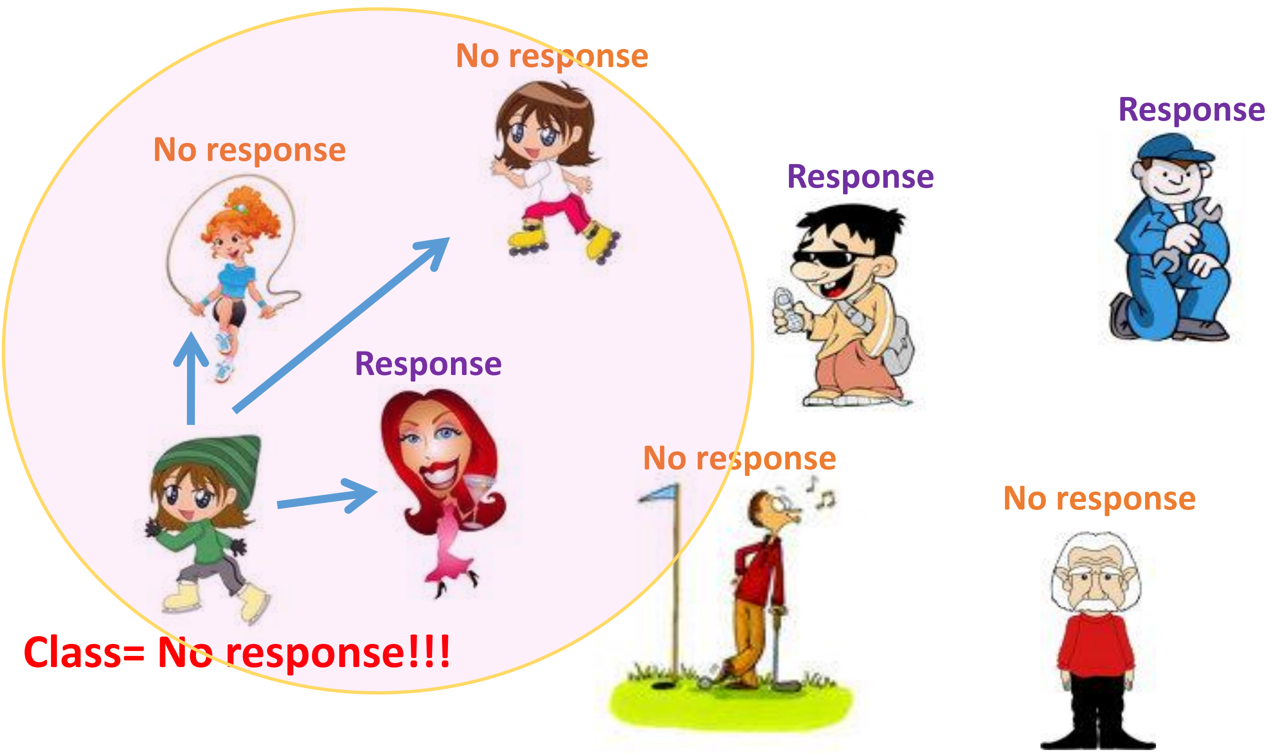




x: value of attribute

y: value of target variable

KNN: Memory-based Learning!



- **No model** is built: Store all training examples
- **Any processing is delayed until a new instance is classified**

Strength of KNN

> Simple to implement and use

> Comprehensible – easy to explain the prediction

> Robust to noisy data by averaging k-nearest neighbors (overfitting control)

> Some appealing applications

- Collaborative filtering for recommender systems

Weakness of KNN

> Takes (much) longer time to classify a new example

- KNN does not build models explicitly
- Need to calculate and compare distance from new example to all examples in training data
- Prohibitively expensive for large number of examples

Thank You !