Advanced Machine Learning

K-Nearest Neighbours

Instructor: Rossano Schifanella

@UDD

Distance or Similarity Measures

- Many data mining and analytics tasks involve the **comparison of objects in terms of their distance or similarity**, e.g.:
  - **k-Nearest Neighbors** search, classification, and prediction
  - **Clustering**
- **Many real-world applications** rely on the computation similarities or distances among objects
  - **Personalization**
  - **Recommender systems**
  - **Document categorization**
  - **Information retrieval**
  - **Target marketing**

Measuring Distance

- In order to compare similar items, we need a way to **measure the distance** between objects (e.g., records)
- Often requires the representation of objects as **feature vectors**

An Employee DB

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

Feature vector corresponding to Employee 2: <M, 51, 64000.0>

Term Frequencies for Documents

	T1	T2	T3	T4	T5	T6
Doc1	0	4	0	0	0	2
Doc2	3	1	4	3	1	2
Doc3	3	0	0	0	3	0
Doc4	0	1	0	3	0	0
Doc5	2	2	2	3	1	4

Feature vector corresponding to Document 4: <0, 1, 0, 3, 0, 0>

Data Matrix and Distance Matrix

- **Data matrix**
  - Conceptual representation of a table
    - Cols = features; rows = data objects
  - n data points with p dimensions
  - Each row in the matrix is the vector representation of a data object

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

- **Distance (or Similarity) Matrix**
  - n data points, but indicates only the **pairwise distance** (or similarity)
  - **A triangular matrix**
  - **Symmetric**

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Properties of a distance measure

1. **$d(x,y) \geq 0$**

   non-negativity or separation axiom

2. **$d(x,y)=0 \Leftrightarrow x=y$**

   identity of indiscernibles

3. **$d(x,y) = d(y,x)$**

   symmetry

4. **$d(x,z) \leq d(x,y) + d(y,z)$**

   sub-additivity or triangle inequality

Distance Measures

Euclidean distance:

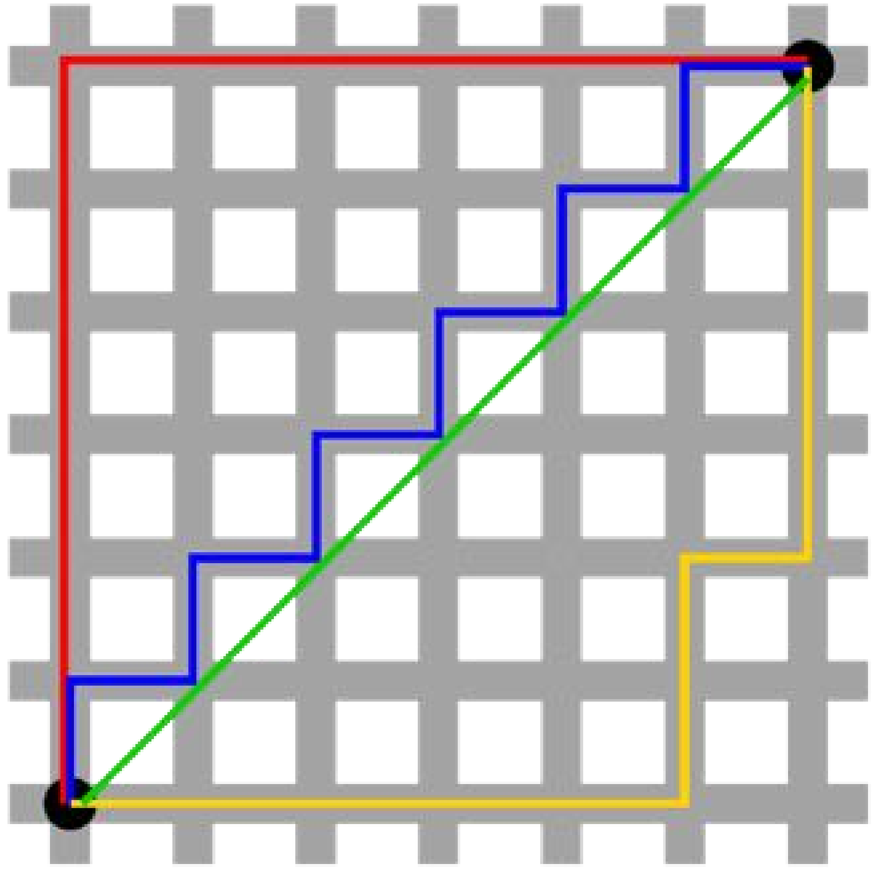
$$d(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+\ldots+|x_{ip}-x_{jp}|^2}$$

Manhattan distance:

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\ldots+|x_{ip}-x_{jp}|$$

They are specific cases of the **Minkowski** distance

$$d(i,j)=(|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+\ldots+|x_{ip}-x_{jp}|^q)^{1/q}$$



The path is irrelevant

Vector-Based Similarity Measures

- In some situations, **distance measures provide a skewed view of data**
  - E.g., when the data is **very sparse** and 0's in the vectors are not significant
- In such cases, typically **vector-based similarity measures** are used
  - **Cosine similarity**
  - **Dot product**
  - **Norm**

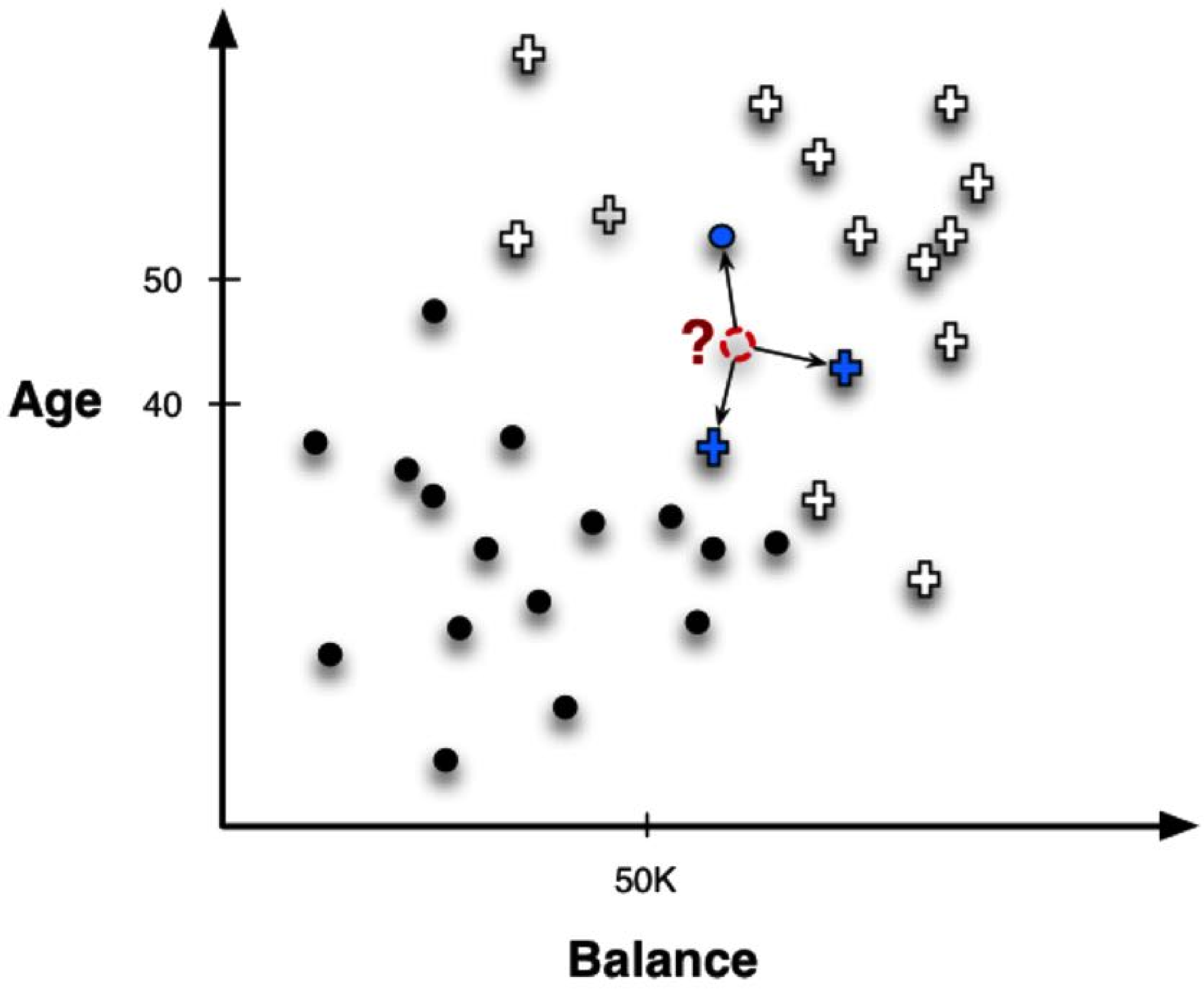
Correlation as Similarity

- In cases where there could be **high mean variance across data objects** (e.g., movie ratings), Pearson Correlation coefficient is a good option
- **Pearson Correlation**

$$r = \frac{\sum_{i=1}^{n}\left(\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2 \sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}}$$

- Often used in recommender systems based on **Collaborative Filtering**

Nearest Neighbor Classifiers



K-Nearest-Neighbor Strategy

- Given **object x**:
  - **Find the k most similar objects to x (k-nearest neighbors)**
    - **Variety of distance or similarity measures** can be used
    - This requires comparison between x and all objects in the database (**expensive!**)

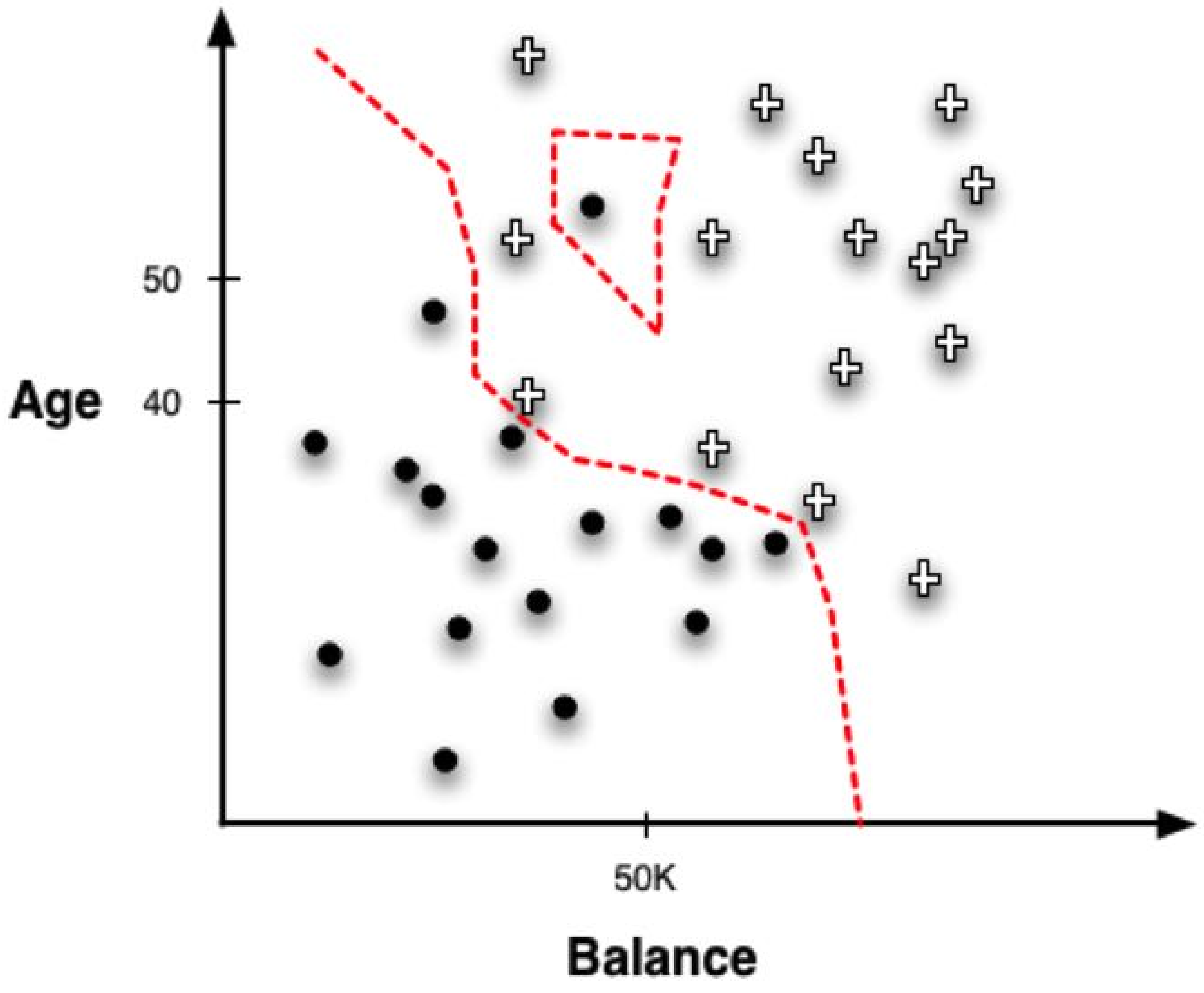
K-Nearest-Neighbor Strategy

- **Classification**
  - Find the class label for each of the k-neighbor
  - Use a **voting** or **weighted voting** approach to determine the **majority class** among the neighbors (**a combination function**)
    - **Distance-based**
      - closer neighbors get higher weights
    - **Heuristic**
      - weight for each neighbor is based on domain-specific characteristics of that neighbor
  - Assign the majority class label to x

K-Nearest-Neighbor Strategy

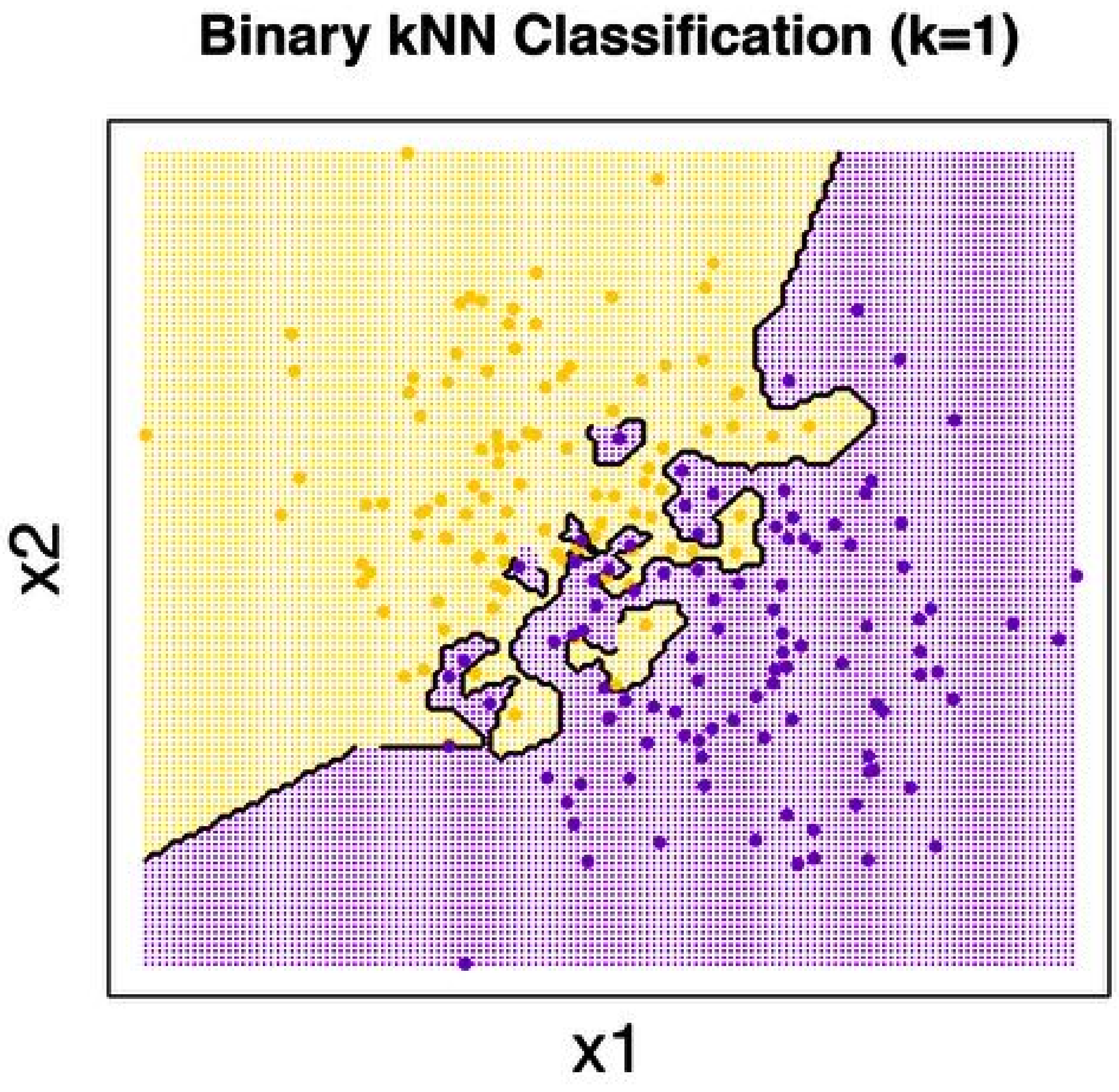
- **Prediction**:
  - Identify the **value of the target attribute** for the k-neighbors
  - Return the **weighted average** as the predicted value of the target attribute for x

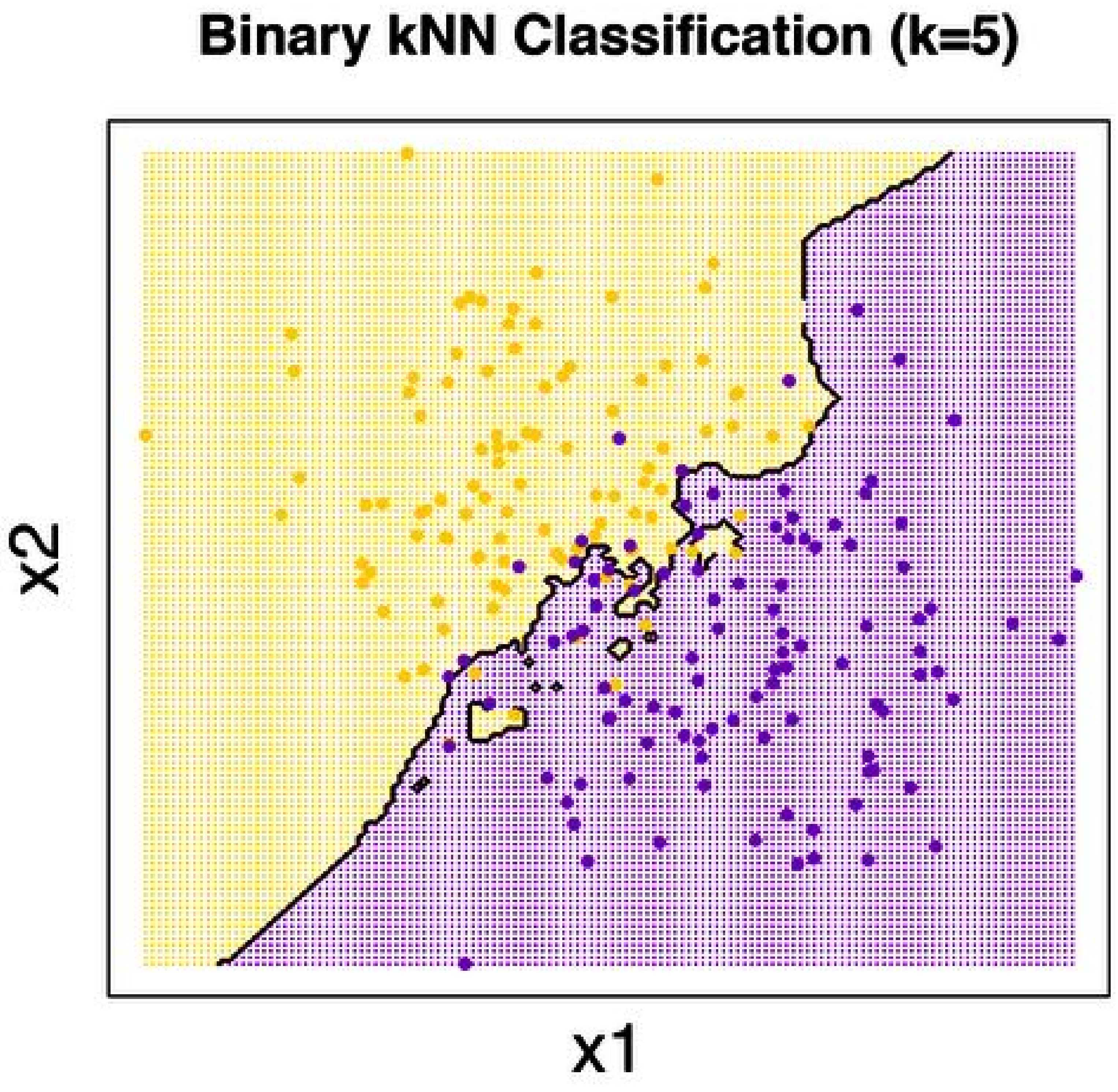
Geometric Interpretation

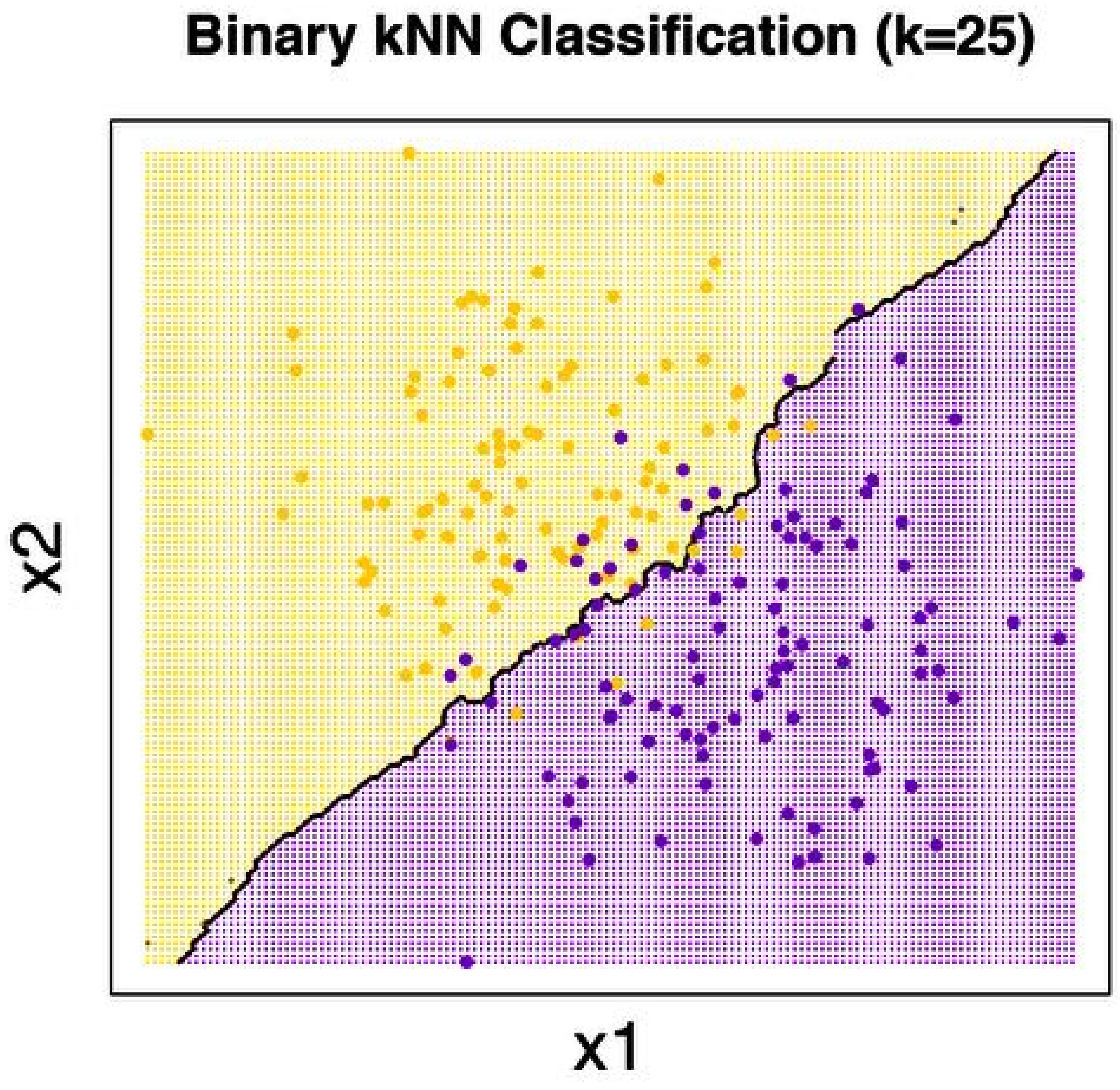


Boundaries created by a 1-NN classifier.

Model Complexity







k-NN Properties

- **Memory-based**, no explicit training or model, **"lazy learning"**
  - defers all of the work until new instance is obtained
- Less data preprocessing and model evaluation, but **more work has to be done at classification time**
- In its basic form, one of the most **simple** machine learning methods
- Gives the **maximum likelihood estimation of the class posterior probabilities**.
- Can be used as a **baseline method**.

k-NN Algorithm Advantages

- **Easy to understand and program**
- **Explicit reject option**
  - if there is no majority agreement
- **Easy handling of missing values**
  - restrict distance calculation to subspace
- Extremely **flexible** classification scheme
- Well suited for
  - **Multi-classes**
  - Records with **multiple or ambiguous class labels**
- Can sometimes be the best method!

k-NN Algorithm Disadvantages

- **Affected by local structure**
- **Sensitive to noise, irrelevant features**
- **Computationally expensive**
- **Large memory requirements**
- More frequent classes dominate result (if distance not weighed in)
- **Curse of dimensionality**:
  - high number of dimensions and low number of training samples
    - "nearest" neighbor might be very far
    - in high dimensions "nearest" becomes meaningless

Questions?

@rschifan

schifane@di.unito.it

http://www.di.unito.it/~schifane