Data Mining:

Classification – Lazy Learner k NearestNeighbour

- Adapter from Data Mining Concepts and Techniques
- Chapter 9. Classification Advanced Methods: Basic Concepts and Methods

Lazy vs. Eager Learning

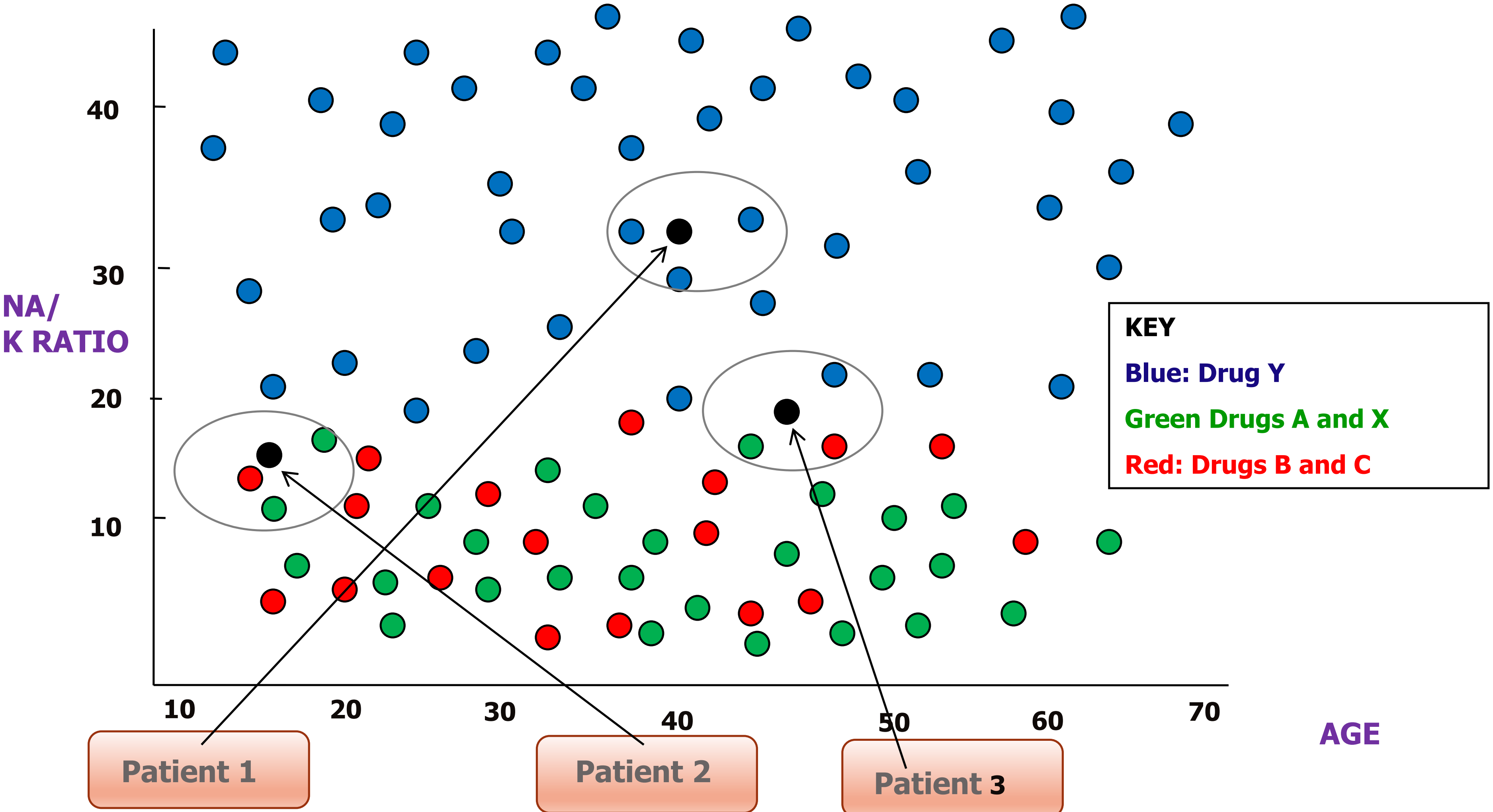
- Lazy vs. eager learning
  - **Lazy learning** (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
  - **Eager learning** (e.g. Decision Tree Algorithms CART, c4.5): Given a set of training tuples, constructs a classification model before receiving new (e.g., test) data to classify
- Lazy: less time in training but more time in predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form an implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

2

k-Nearest Neighbour Algorithm

- Most often used for Classification
- This is an example of an instance-based learning, in which the training set data set is stored, so that a classification for a new unclassified record may be found simply by comparing it to similar records in the training set
- Example: Classifying the type of drug a patient should be prescribed, based on certain patient characteristics such as age and a patients sodium/potassiums ration

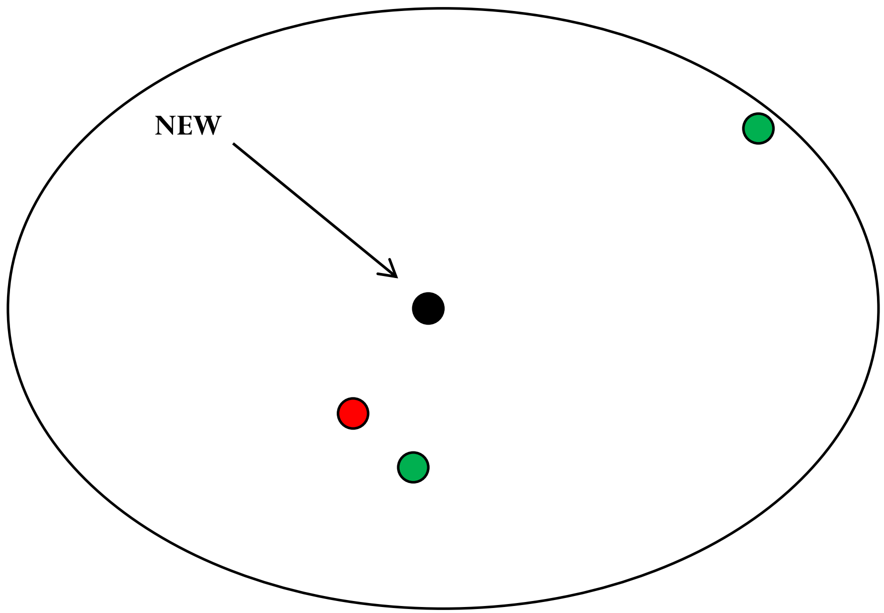
k-Nearest Neighbour Algorithm Example



Scatter plot of sodium/potassium ratio against age with drug overlay.
A training set of 200 patients

k-Nearest Neighbour Algorithm Example

patient 2

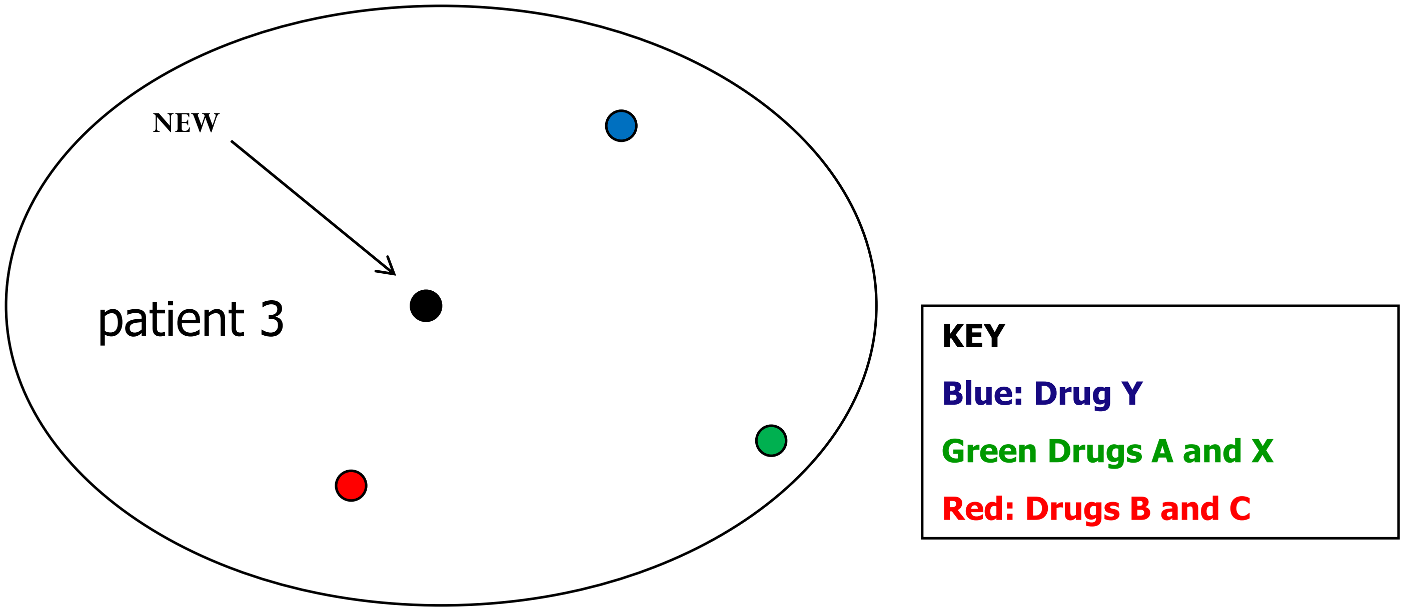


KEY
Blue: Drug Y
Green Drugs A and X
Red: Drugs B and C

A close up of the training data points (3 nearest neighbours) to new **patient 2**

- Let k= 1 for our k-nearest neighbour algorithm, so that new patient 2 would be classified according to whichever single (one) observation it was closest to
- What if k=3? Or k=2?

k-Nearest Neighbour Algorithm Example



A close up of the training data points (3 nearest neighbours) to new patient 3

- For k=1 what would the k-nearest algorithm choose?
- k=2? k=3?

k-Nearest Neighbour Algorithm Issues

- How many neighbours (k) should we consider?
- How do we measure distance?
- How do we combine the information from more than one observation?
- Should all attributes be weighted equally, or should attributes have more influence than others?

K-Nearest Neighbour Algorithm

- Distance Function required with following properties

1. $d(x,y) \geq 0$, and $d(x,y) = 0$ if and only if $x = y$
2. $d(x,y) = d(y,x)$
3. $d(x,z) \leq d(x,y) + d(y,z)$

- Most Common distance function

$$d_{\text{Euclidean}}(\mathrm{x},\mathrm{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

where **x** = *x*1, *x*2, . . . , *xm*, and **y** = *y*1, *y*2, . . . , *ym* represent the *m* attribute values of two records

- Give that patient A x_1 =20 years and x_2 = 12Na/K while patient B y_1=30 years old and y_2 = 8 Na/K, calculate the Euclidean distance.

k-Nearest Neighbour Algorithm

- When measuring distance, certain attributes have very large values can can overwhelm the influence of other attributes
- Must normalise the attribute values to avoid this impact

For continuous variables

Min–max normalization:

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Z-score standardization:

$$X^* = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

Note: for **Categorical Variable** Euclidean distance metric not appropriate. Must convert to numerical.

$$\text{different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

k-Nearest Neighbour Algorithm

- Given the following data which patient B OR C is **more similar** to a 50 year old male (Patient A)?

Patient	Age	Age_{MMN}	Age_{Zscore}	Gender
A	50	$\frac{50-10}{50}=0.8$	$\frac{50-45}{15}=0.33$	Male
B	20	$\frac{20-10}{50}=0.2$	$\frac{20-45}{15}=-1.67$	Male
C	50	$\frac{50-10}{50}=0.8$	$\frac{50-45}{15}=0.33$	Female

- For Z-Score
  - Assumes age range is 50, the minimum is 10, the mean is 45 and the standard deviation is 15.

k-Nearest Neighbour Algorithm – Combination Function

- **Simple Unweighted Function**
  - Decide on value of k; how many records will have a voice in classifying the new record
  - Compare the new record to the k nearest neighbours
  - The k records are chosen they all get one equal vote
- **Weighted Voting**
  - Closer or more similar records to the new record should be weighted more heavily then more distant neighbours
  - Closer neighbours have a larger voice in the classification decision than do more distant neighbours
  - Less likely to be ties
  - Votes of these records are then weighted according to the inverse square of their distances

k-Nearest Neighbour Algorithm – Combination Function Example

KEY
Blue: Drug Y
Green Drugs A and X
Red: Drugs B and C

- Given

Record	Target Variable Value	Age	Na/k	Age MMN	Na/k MNN
Unseen	**?**	**17**	**12.5**	**.05**	**.25**
A	**Red**	16.8	12.4	.0467	.2471
B	**Green**	17.2	10.5	.0533	.1912
C	**Green**	19.5	13.5	.0917	.2794

- Using Weighted Voting, determine the appropriate classification for the new record i.e. what drugs would you give this patient?
- Use $w \equiv \frac{1}{d(x_q, x_i)^2}$ for voting
- What drug combination would you give the new patient for k=1?, k=2?, K=3?

k-Nearest Neighbour Algorithm – Combination Function Example

KEY
Blue: Drug Y
Green Drugs A and X
Red: Drugs B and C

- Given

Record	Target Variable Value	Age	Na/k	Age MMN	Na/k MNN
Unseen	**?**	**17**	**12.5**	**.05**	**.25**
A	**Red**	16.8	12.4	.0467	.2471
B	**Green**	17.2	10.5	.0533	.1912
C	**Green**	19.5	13.5	.0917	.2794

- What are the distances of records A, B, and C from the new record (note k=3)?

$$d(\text{new}, A) = \sqrt{(0.05 - 0.0467)^2 + (0.25 - 0.2471)^2} = 0.004393$$

$$d(\text{new}, B) = \sqrt{(0.05 - 0.0533)^2 + (0.25 - 0.1912)^2} = .058893$$

$$d(\text{new}, C) = \sqrt{(0.05 - 0.0917)^2 + (0.25 - 0.2794)^2} = 0.051022$$

- Using Weighted Voting, determine the appropriate classification for the new record i.e. what drugs would you give this patient?
- Use $w \equiv \frac{1}{d(x_q, x_i)^2}$ for voting

Choosing The Value Of k

- Not an obvious solution
- Small value of k
  - Possibly that the classification may be unduly affected by outliers
  - Danger of overfitting i.e. the algorithm memorises the training set at the expense of generality
  - Large value of k
    - Will tend to overlook locally interesting behaviour.
  - Possible solution
    - Try various values of k with different randomly selected training sets and choose the value of k that minimises Classification Error (Maximises Accuracy)