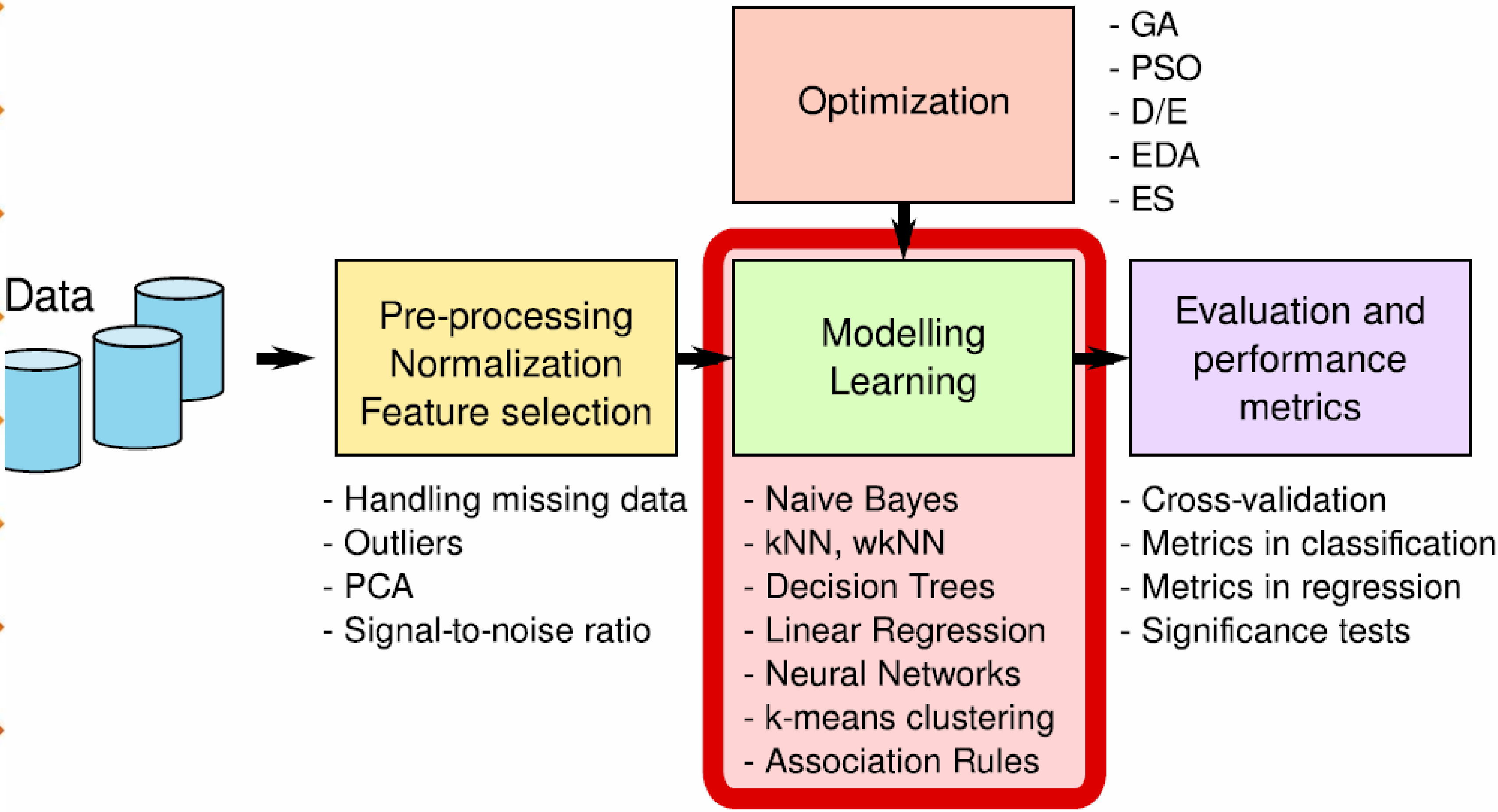
COMP809 Data Mining and Machine Learning

LECTURER: DR AKBAR GHOBAKHLOU
SCHOOL OF ENGINEERING, COMPUTER AND MATHEMATICAL SCIENCES

K-Nearest Neighbors (KNN)

Course Outline



K-Nearest Neighbours (KNN)

KNN is an algorithm that is an example of lazy learning.

- Non-parametric means that it makes no assumptions. The model is made up entirely from the data given to it rather than assuming its structure is normal.
- Lazy learning means that the algorithm makes no generalizations. This means that there is little training involved when using this method. Because of this, all of the training data is also used in testing when using KNN.

Instance-Based Classifiers

Set of Stored Cases

Atr1	……...	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- Store the training records
- Use training records to predict the class label of unseen cases

Unseen Case

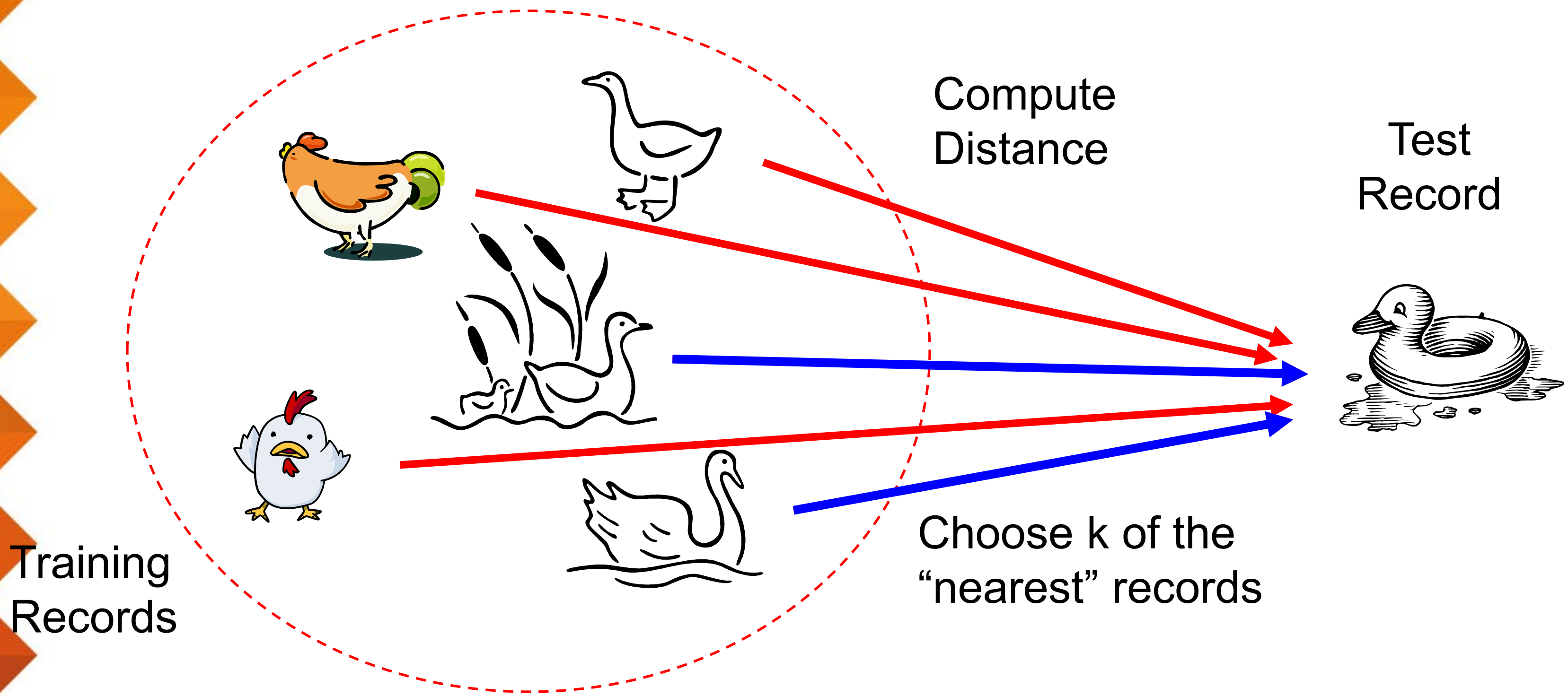
Atr1	……...	AtrN

Instance Based Classifiers

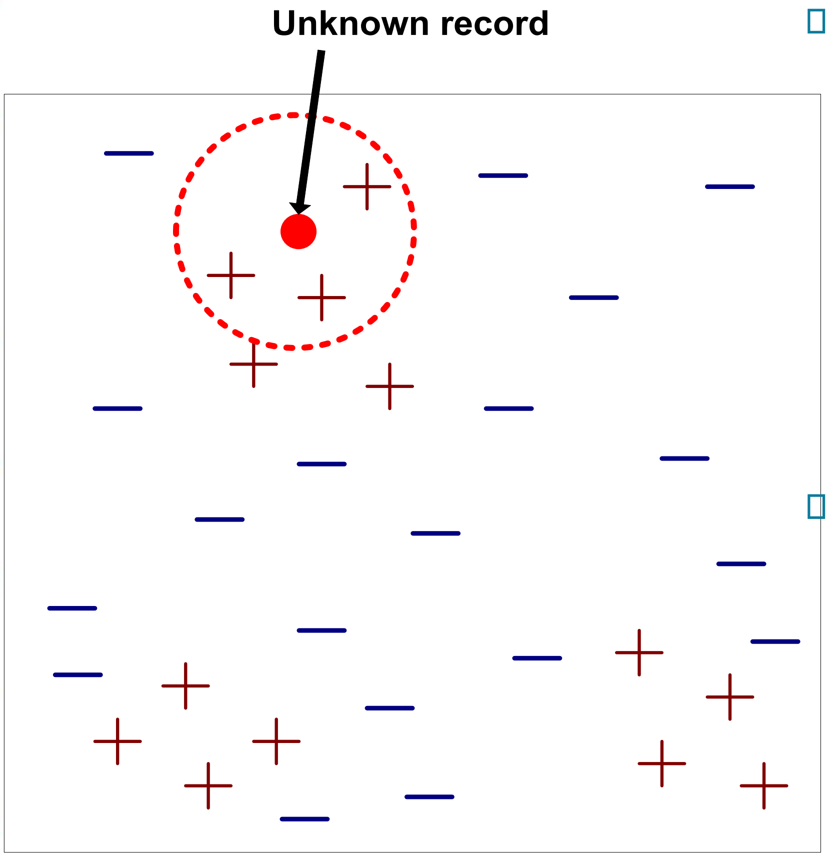
- Examples:
  - Rote-learner
    - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
  - Nearest neighbor
    - Uses k “closest” points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck

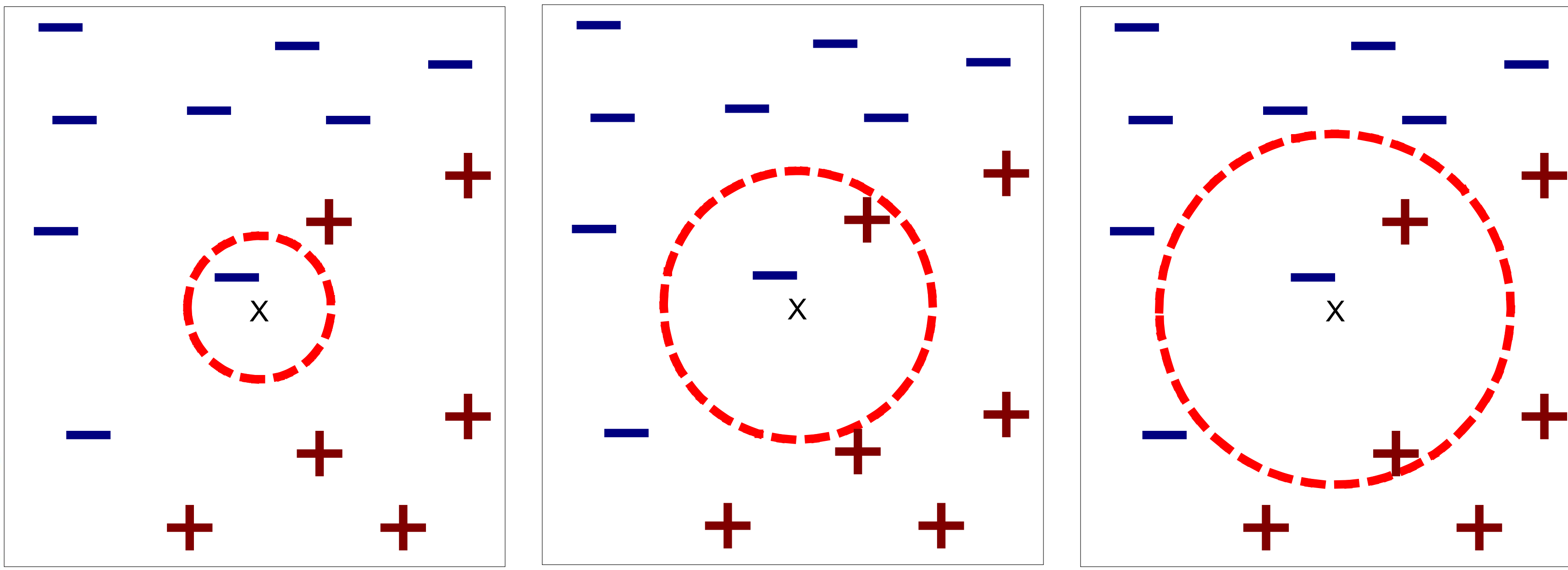


Nearest Neighbour Algorithm



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

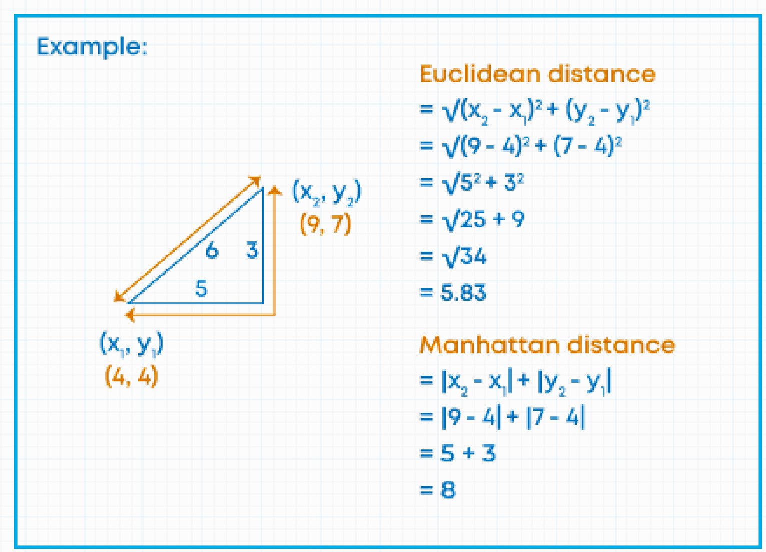
Definition of Nearest Neighbor

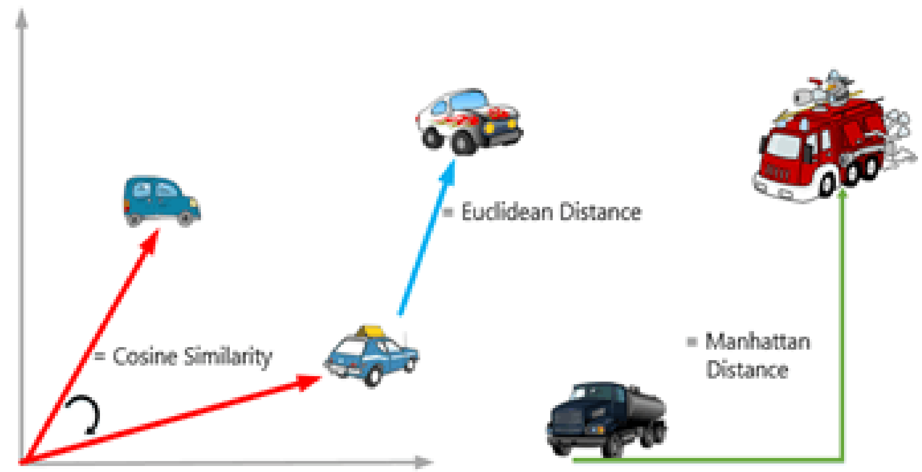


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

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

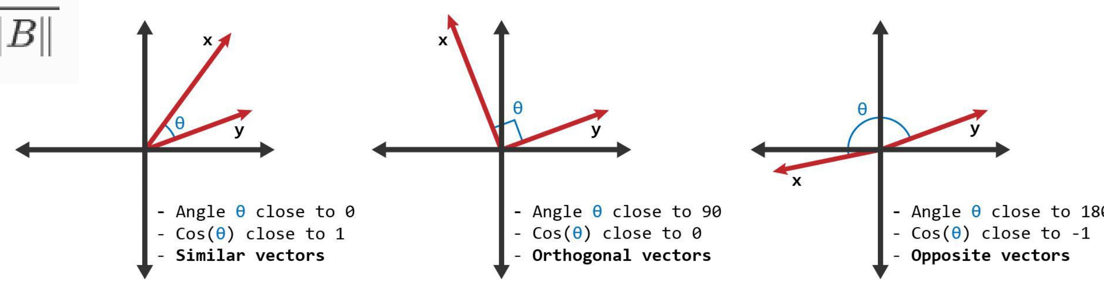
Similarity Distance Measures





Cosine similarity defined as the dot product of the vectors divided by their magnitude. For example, if we have two vectors, A and B, the similarity between them is calculated as:

$$similarity(A, B) = cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

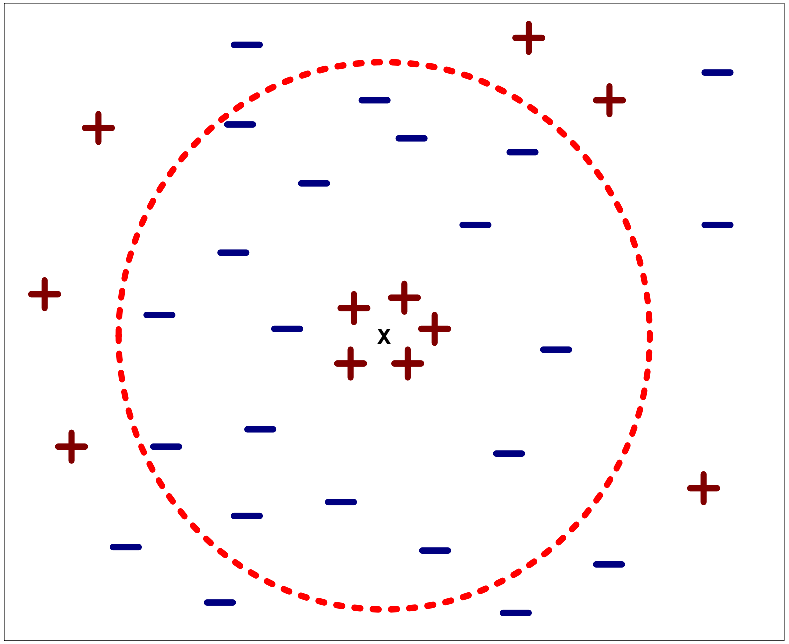


Nearest Neighbor Classification

- Compute distance between two points:
  - Euclidean distance $d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$
  - Manhattan distance $d(p,q) = \sum_i |p_i - q_i|$
  - q norm distance $d(p,q) = (\sum_i |p_i - q_i|^q)^{1/q}$
- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor Classification…

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



Nearest Neighbor Classification…

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 40kg to 150kg
    - income of a person may vary from $10K to $1M

Example

Name	Age	Gender	Income
Jim	53	M	$68,000
Mary	20	F	$61,000
John	49	M	$36,000

$$d(Jim, Mary) \approx \sqrt{(68000 - 61000)^2}$$
$$\approx \sqrt{(7000)^2}$$
$$\approx 7000$$

$$d(Jim, John) \approx \sqrt{(68000 - 36000)^2}$$
$$\approx \sqrt{(32000)^2}$$
$$\approx 32000$$

Nearest neighbor Classification…

- k-NN classifiers are lazy learners
  - It does **not build** models explicitly
  - Unlike eager learners such as decision tree induction and rule-based systems
  - Classifying unknown records are relatively expensive

Nearest Neighbour Algorithm: Nominal Data

- For nominal data, Euclidean distance cannot be used and a distance measure based on whether an exact match exists or not can be used

Applying Nearest Neighbour to Credit Scoring

- For the credit scoring example we will use k=3 and distance between two instances is based on a 0/1 scheme

Name	Debt	Income	Married?	Risk Actual	Probability: Good Risk	Probability: Poor Risk	Risk Predicted
Joe	High	High	Yes	Good	0.82	0.18	Good
Sue	Low	High	Yes	Good	0.69	0.31	Good
John	Low	High	No	Poor	0	1.0	Poor
Mary	High	Low	Yes	Poor	0	1.0	Poor
Fred	Low	Low	Yes	Poor	0	1.0	Poor

	Joe	Sue	John	Mary	Fred
Joe	0	1	2	1	2
Sue	1	0	1	2	1
John	2	1	0	3	2
Mary	1	2	3	0	1
Fred	2	1	2	1	0

Applying Nearest Neighbour to Credit Scoring

- To classify Sam (Debt=High, Income=High, Married = Yes), we find the 3 nearest neighbours to Sam, who are Joe, Sue, and Mary
- The risk values of the neighbours are *Good, Good and Poor,* which means that Sam is classified as a *Good* risk

Name	**Debt**	**Income**	**Married?**	**Risk Actual**	**Probability: Good Risk**	**Probability: Poor Risk**	**Risk Predicted**
Joe	High	High	Yes	Good	0.82	0.18	Good
Sue	Low	High	Yes	Good	0.69	0.31	Good
John	Low	High	No	Poor	0	1.0	Poor
Mary	High	Low	Yes	Poor	0	1.0	Poor
Fred	Low	Low	Yes	Poor	0	1.0	Poor

	Joe	**Sue**	**John**	**Mary**	**Fred**
Joe	0	1	2	1	2
Sue	1	0	1	2	1
John	2	1	0	3	2
Mary	1	2	3	0	1
Fred	2	1	2	1	0

Issues with k-NN

- Accuracy of prediction depends on the value of k and the choice of distance metric
- Most DM products choose a value of 10 as default for k
- Distance metric needs to be defined by the miner – this requires some care
- In datasets that contain numeric data, need to be careful with attributes that are defined on different scale ranges – eg *Income* and *Age*
- for nominal data (E.g. in age groupings) need to recognise that distance between pairs of age groupings are not always the same:
  d((21, 30), (51,60)) > d((21, 30), (41,50))

K-NN - Summary

- **Pros:**
  - Easy to use.
  - Quick calculation time.
  - Does not make assumptions about the data.
- **Cons:**
  - Accuracy depends on the quality of the data.
  - Must find an optimal k value (number of nearest neighbours).
  - Poor at classifying data points in a boundary where they can be classified one way or another.

References

- Chapter 4, Data Mining: Practical Machine Learning Tools and Techniques (3rd edition) / Ian Witten, Eibe Frank; Elsevier, 2011.
- Chapter 4, Introduction to Data Mining / Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, Inc, 2006.
- KNN Algorithm Using Python | How KNN Algorithm Works