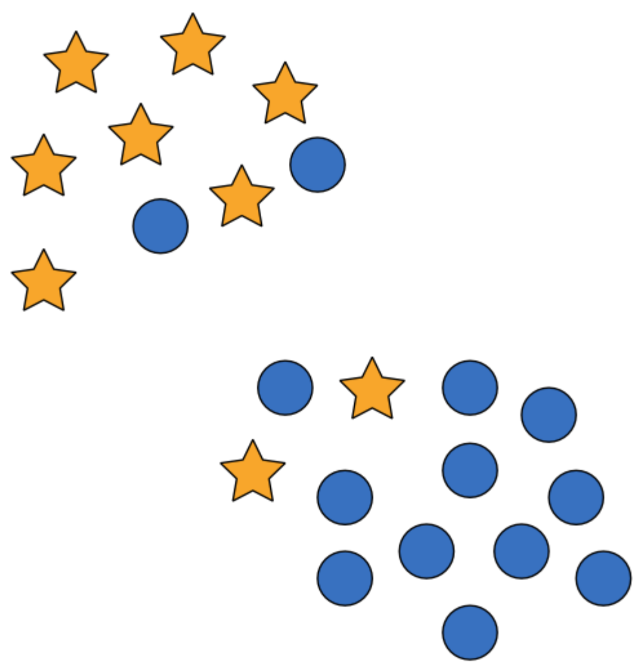
k-Nearest Neighbor (kNN) Methods

Intuition

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

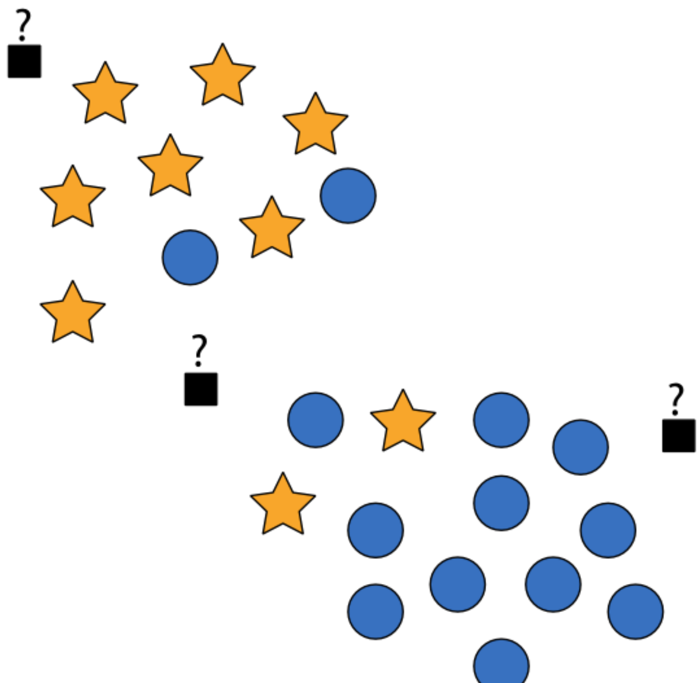
Intuition

- Identify several cases that are most similar to a given observation.
- Use the information from those 'neighbors' to classify/predict the new observation.



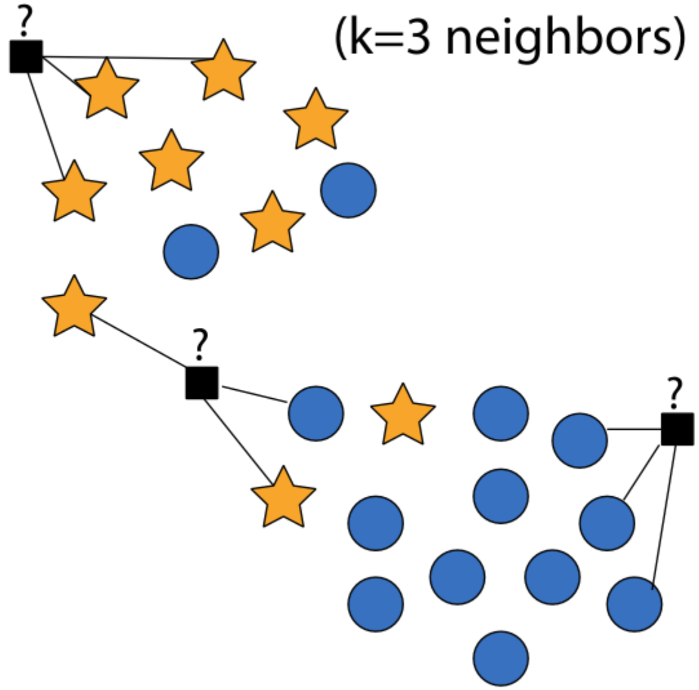
Intuition

- Identify several cases that are most like a given observation.
- Use the information from those ‘neighbors’ to classify/predict the new observation.



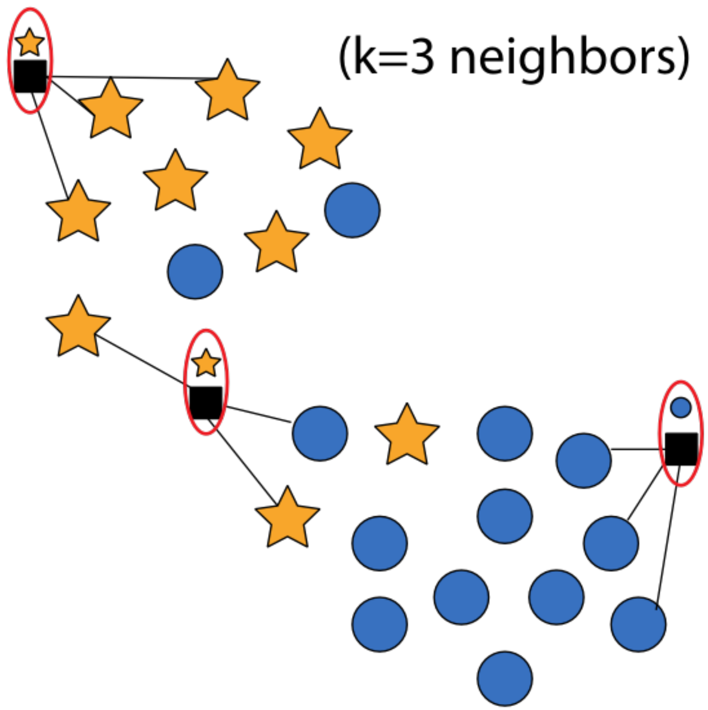
Intuition

- Identify several cases that are most like a given observation.
- Use the information from those 'neighbors' to classify/predict the new observation.



Intuition

- Identify several cases that are most like a given observation.
- Use the information from those ‘neighbors’ to classify/predict the new observation.



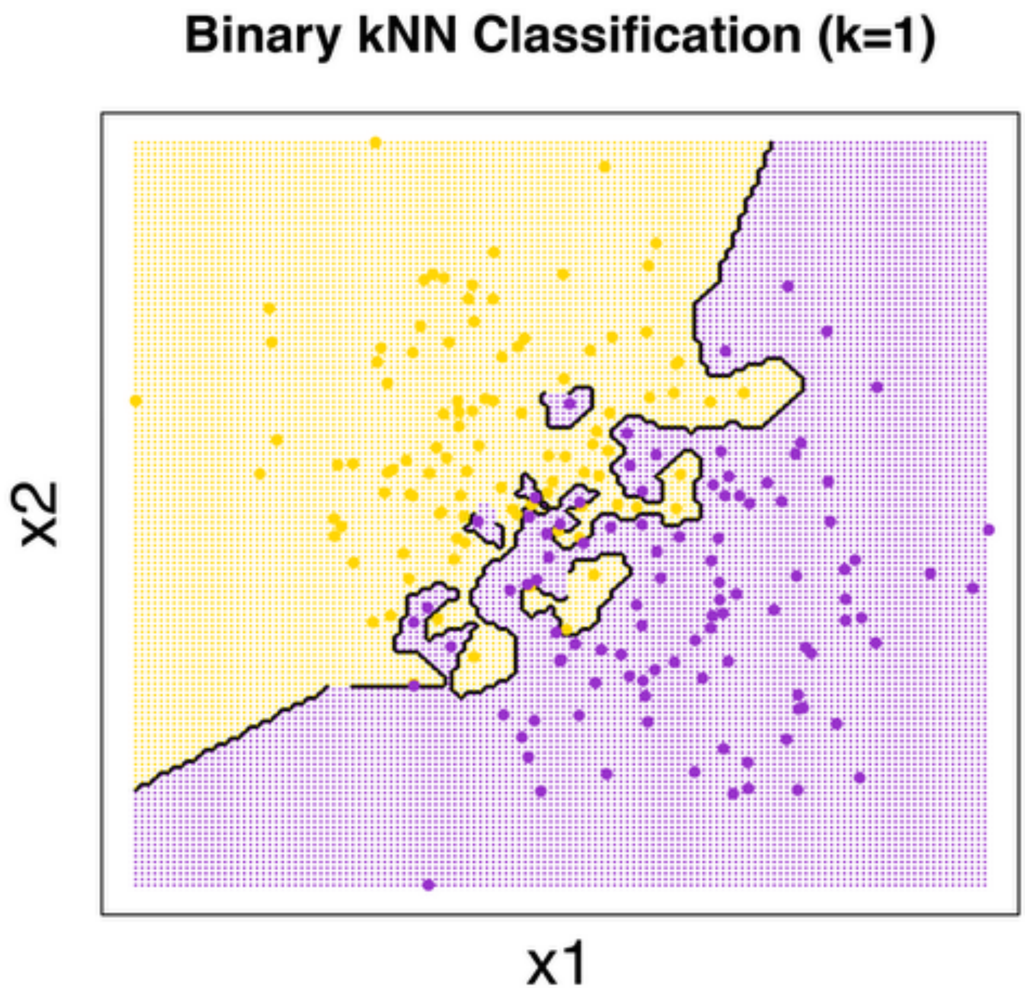
Considerations

- How should I measure nearness?
  - Numeric Attributes?
  - Ordinal Attributes?
  - Categorical Attributes?
  - How do I combine these?
- How should I combine the results of neighbors?
  - Classification:
    - Majority rules?
    - Weight votes by nearness?
  - Prediction:
    - Mean?
    - Median?
- How many neighbors should I use?

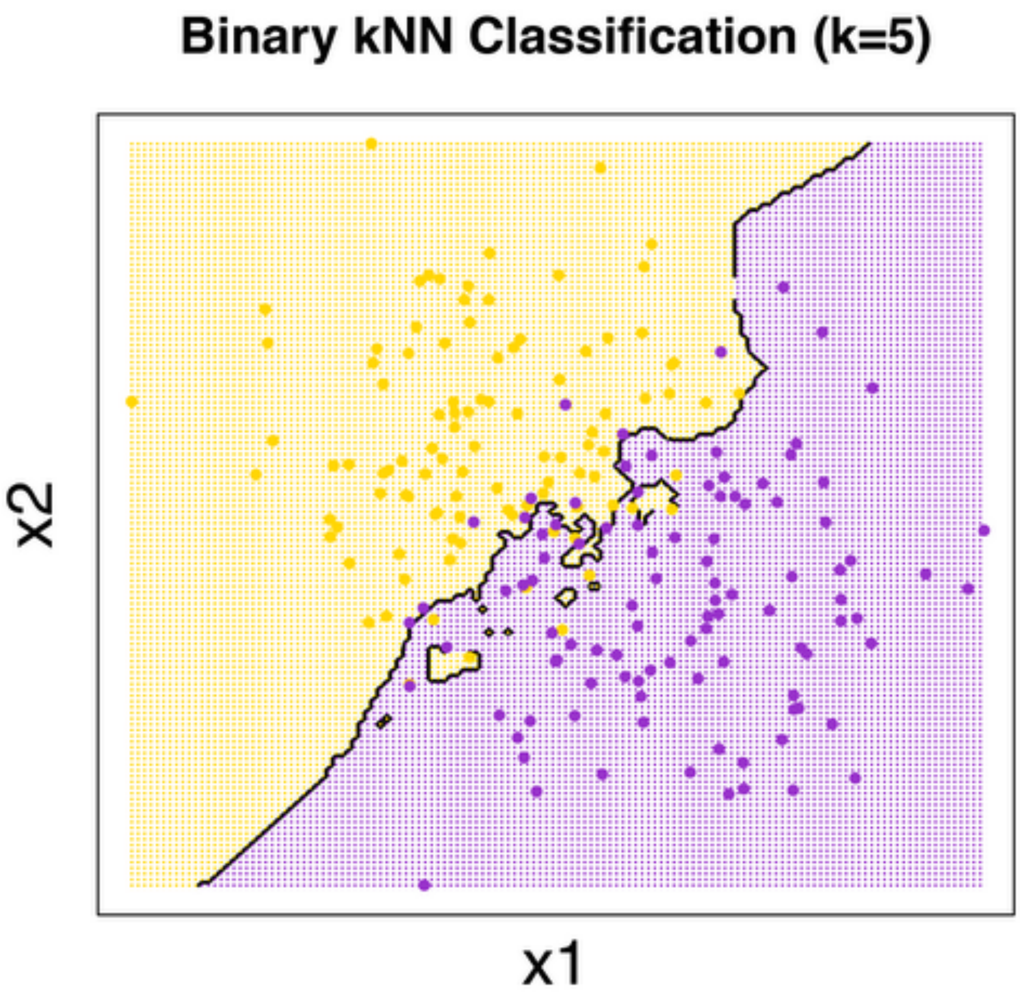
Considerations

- The methodology is simple but FLEXIBLE for creativity
- Using default kNN function in R or SAS will save time but lose flexibility.
- Consider creating your own distance matrices that use your own intuition.

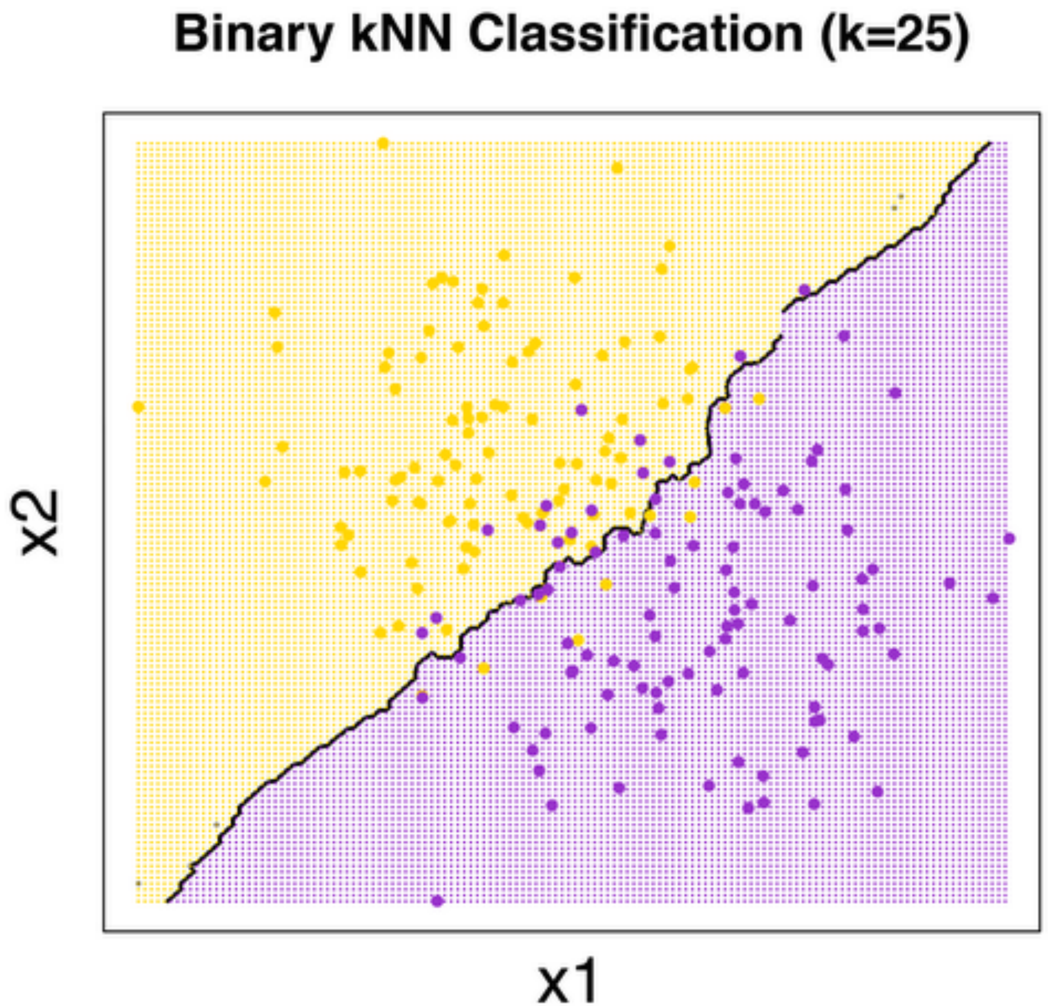
Choosing k



Choosing k



Choosing k



Choosing k

- **Smaller** values of k => higher variance model

  (tends toward **overfitting**)
- **Larger** values of k => higher bias model

  (tends toward **underfitting**)

- Common practice to begin with $k = \sqrt{n}$ where n is the number of training examples
- Best to tune this parameter with a validation set or with cross-validation.

Advantages of kNN

- Generally a **good predictive model**
- **Easy to explain**, intuitive, understandable
- Applicable to **any type of data**
- **Nonparametric**: Makes no assumptions about the underlying distribution of the data.
- **Immune to the effect of outliers** or high leverage observations
- Large/representative training set is only assumption

Disadvantages of kNN

- **Computationally expensive in classification phase**
- **Requires storage for the training set**
    - (The training set IS the model!)
- Results **dependent on choice of distance** function, combination function, and number of neighbors, k.
- Susceptible to noise
- Require **lots of data preprocessing** and consideration for distance metrics
- Does not produce a model. **Does not help us understand how the features are related to the classes.**

Building Custom Distance Functions

• • •

(Self-Study)

Building Distance Functions

Numeric Variables (Includes some ordinal):

- Some type of normalization or standardization is usually required
  - Standardize the variable before input to the method
  - OR standardize the distance in creating an overall distance matrix.
- Most common types of standardization:
  - min/max normalization (feature scaling) $\dfrac{x - x_{min}}{x_{max} - x_{min}}$
  - z-score standardization $\dfrac{x - \bar{x}}{\sigma_x}$

Building Distance Functions

Options for Standardizing Distances:

1. Absolute difference (No standardization)
2. Divide by max distance(Scales distance between 0 and 1)
3. Divide by the std. deviation for that variable

    (How many std. deviations is the distance?)
4. Can even divide by the std. deviation for the distance!

Name	Income (in Ks)
Sam	50
Pam	65
Tam	75

Original Variable

$$\begin{array}{c|ccc} & S & P & T \\ \hline S & 0 & 15 & 25 \\ P & 15 & 0 & 10 \\ T & 25 & 10 & 0 \end{array}$$

Absolute Difference distance matrix (Option 1)

$$\begin{array}{c|ccc} & S & P & T \\ \hline S & 0 & 0.6 & 1 \\ P & 0.6 & 0 & 0.4 \\ T & 1 & 0.4 & 0 \end{array}$$

Standardized distance matrix (Option 2)

Building Distance Functions

Categorical Variables (Includes some ordinal):

- Simple Matching Distance = 0 if matching, 1 otherwise

Name	Marital Status
Sam	Single
Pam	Married
Tam	Single

Original Variable

$$\begin{array}{c c} & \begin{array}{ccc} S & P & T \end{array} \\ \begin{array}{c} S \\ P \\ T \end{array} & \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

Distance Matrix

Categorical Similarity Measures

With MANY categorical attributes, makes sense to consider them separately from your numeric attributes.

Obs	Marital Status	Insurance Status	Education	Cluster	Card Line	Home
i	M	Y	B.S.	A	Slate	Own
j	S	Y	PhD	B	Slate	Rent

Jacquard Similarity = 2 / 6

Jacquard Distance = 4 / 6

Many ***many*** more options.

Building Distance Functions

You can define anything you find reasonable!

Example: Zip Codes.

- $d(i,j) = 0$ if zip codes are identical
- $d(i,j) = 0.1$ if first three digits identical
- $d(i,j) = 0.5$ if first digits are identical
- $d(i,j) = 1$ if first digits different.
- or use the corresponding geographical distance (more work)

Building Distance Functions

- Let $d_{x_k}(i, j)$ be the distance between observation i and observation j on variable x_k
- Merge the individual distances together:
  - Manhattan (L_1-norm)

$$d(i, j) = d_{x_1}(i, j) + d_{x_2}(i, j) + \ldots + d_{x_p}(i, j)$$

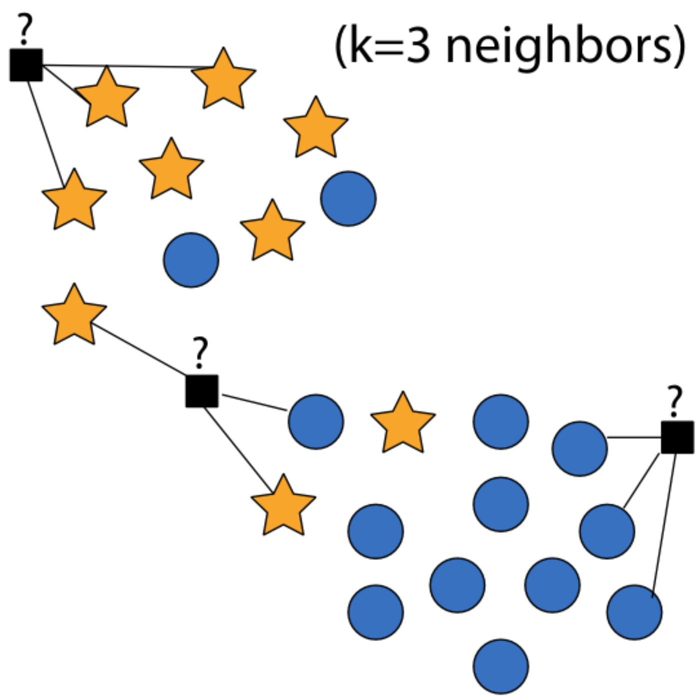
  - Euclidean (L_2-norm)

$$d(i, j) = \sqrt{d_{x_1}(i, j)^2 + d_{x_2}(i, j)^2 + \ldots + d_{x_p}(i, j)^2}$$

  - Can even weight distances based on a given variable's correlation or association with the target.

Combination Functions

- Now we have distances to each of k neighbors
- How do I combine that information to make a prediction for the given observation?



Combination Functions

Numeric Target

- Mean or Median of the neighbors' target value

Class Target

- Basic approach: democracy – majority rules
- Create probabilities for each class as the proportion of neighbors voting for each class.
- Weighted voting: nearer neighbors have stronger votes

$$w_j = \frac{1}{d(i,j)^2}$$

  - This can reduce the sensitivity to the parameter k
  - Add up the weighted votes to see which category has the most