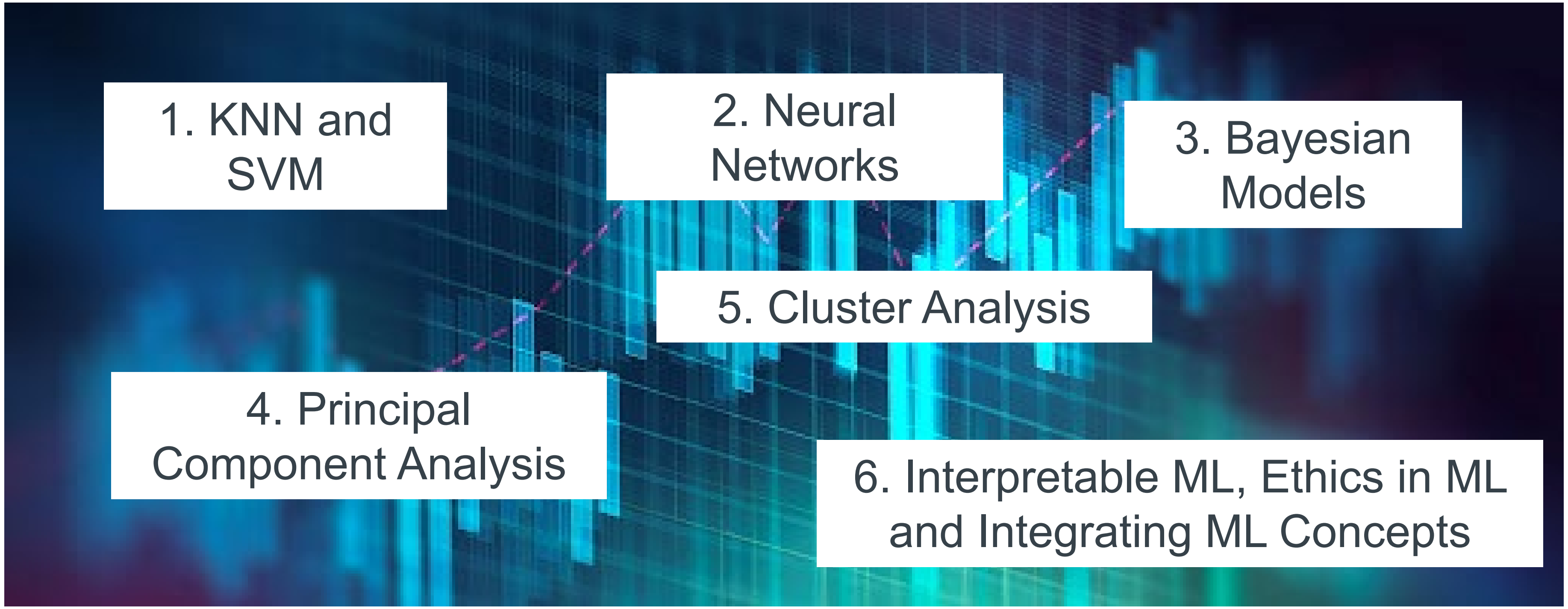
Introduction to Machine Learning

Machine Learning - II

Georgetown University McDonough School of Business

Course Content



1. KNN and SVM

2. Neural Networks

3. Bayesian Models

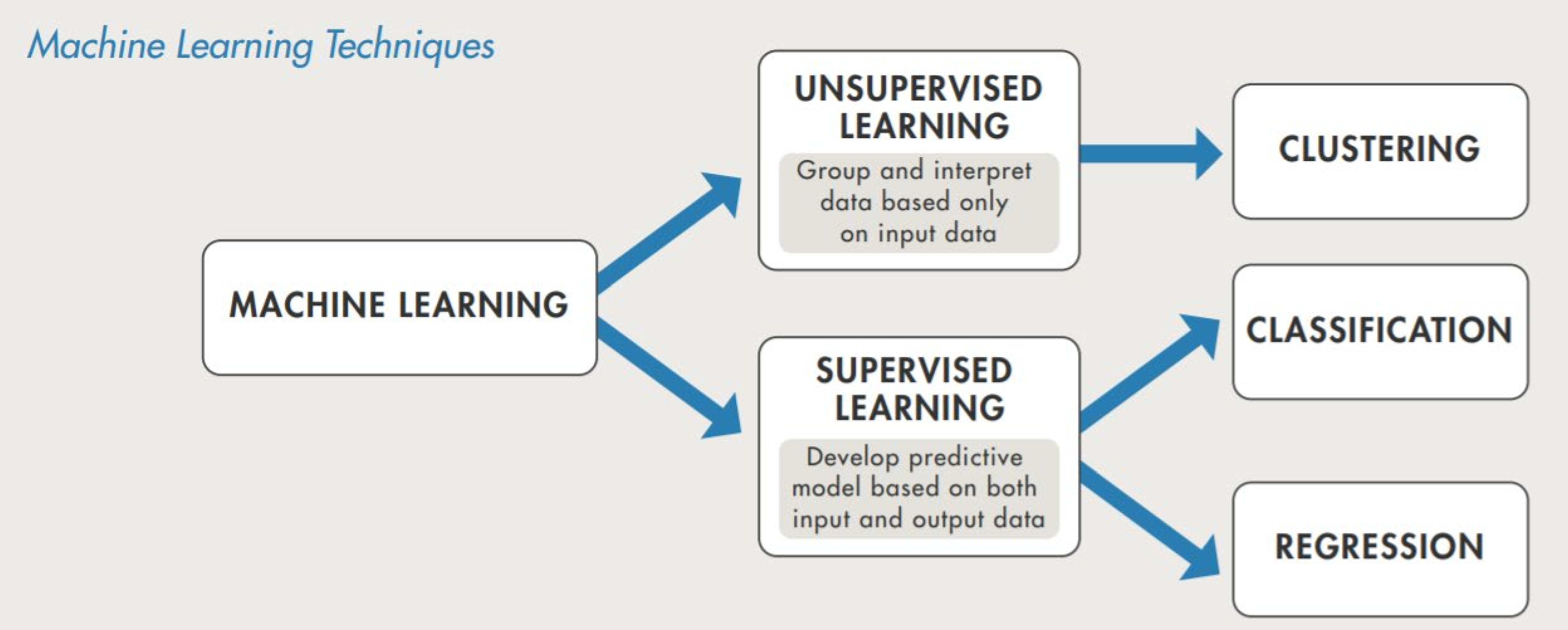
4. Principal Component Analysis

5. Cluster Analysis

6. Interpretable ML, Ethics in ML and Integrating ML Concepts

MACHINE LEARNING ALGORITHMS REFRESHER

How Machine Learning Works?



Machine Learning Techniques

MACHINE LEARNING

UNSUPERVISED LEARNING

Group and interpret data based only on input data

SUPERVISED LEARNING

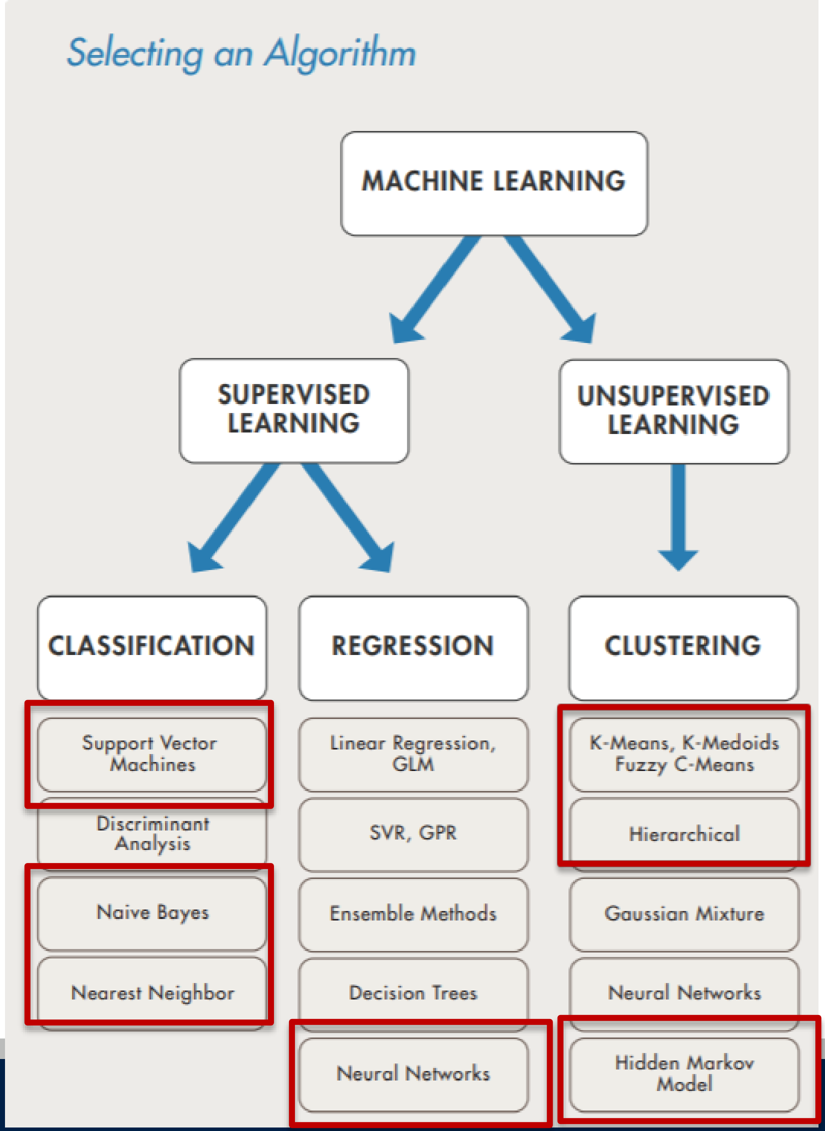
Develop predictive model based on both input and output data

CLUSTERING

CLASSIFICATION

REGRESSION

How Do You Decide which Algorithm to Use?



Selecting an Algorithm

MACHINE LEARNING

SUPERVISED LEARNING

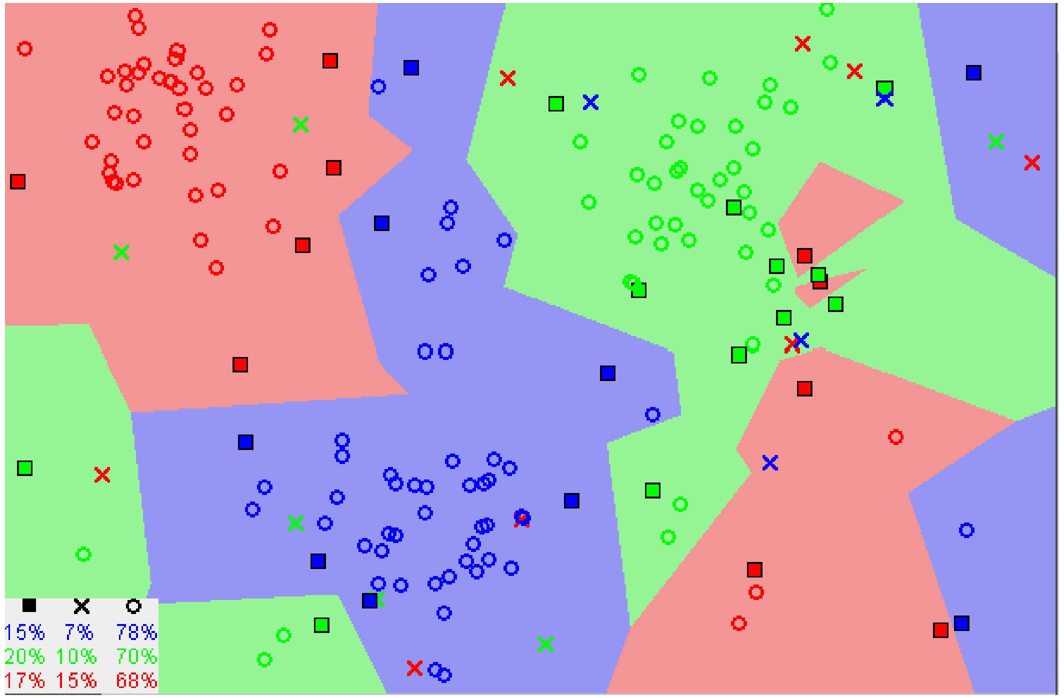
UNSUPERVISED LEARNING

CLASSIFICATION	REGRESSION	CLUSTERING
Support Vector Machines	Linear Regression, GLM	K-Means, K-Medoids Fuzzy C-Means
Discriminant Analysis	SVR, GPR	Hierarchical
Naive Bayes	Ensemble Methods	Gaussian Mixture
Nearest Neighbor	Decision Trees	Neural Networks
	Neural Networks	Hidden Markov Model

K-NEAREST NEIGHBORS

K-Nearest Neighbors

- Observations close to each other share similar characteristics
- Hinges on the idea of distance between observations
- Distance between observations can be calculated in several ways

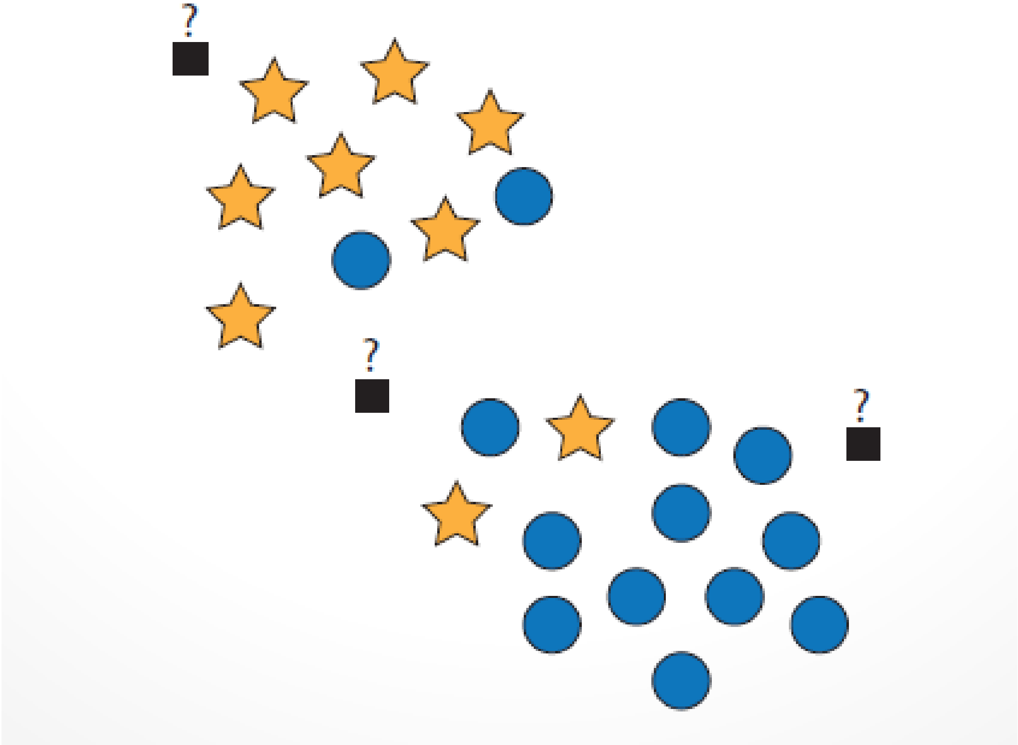


The value of a data point is determined by the data points around it

kNN: Intuition

Identify several cases that are most similar to a given observation

Use the information from those "neighbors" to classify or predict the new observation

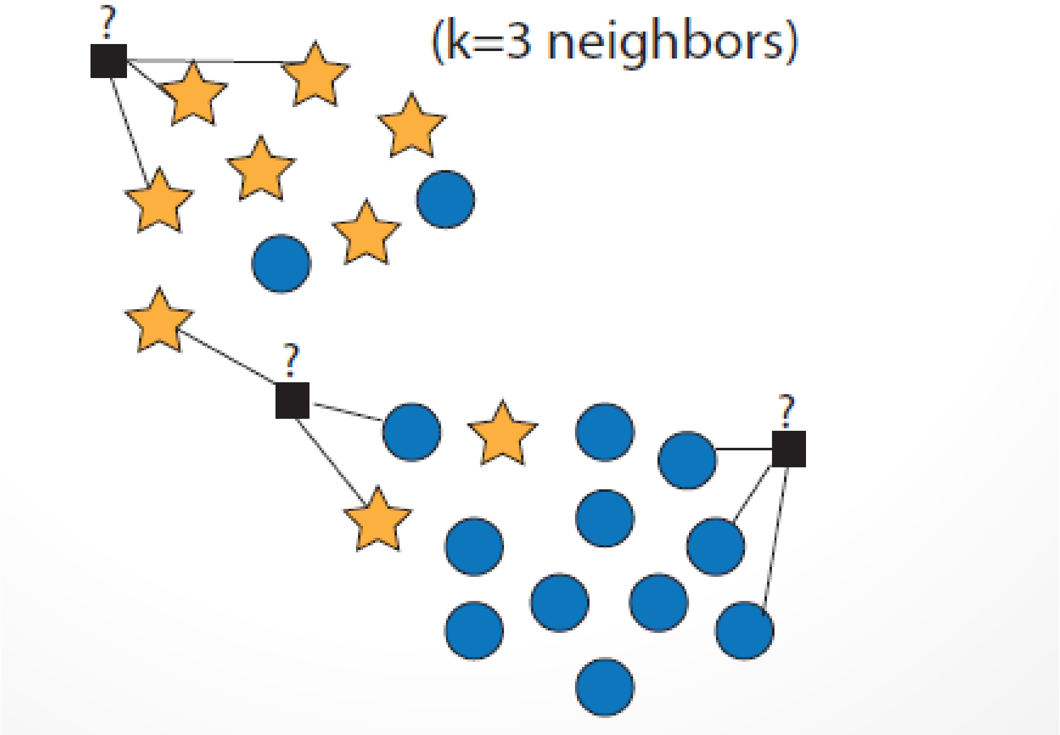


kNN: Intuition

Identify several cases that are most similar to a given observation

Use the information from those "neighbors" to classify or predict the new observation

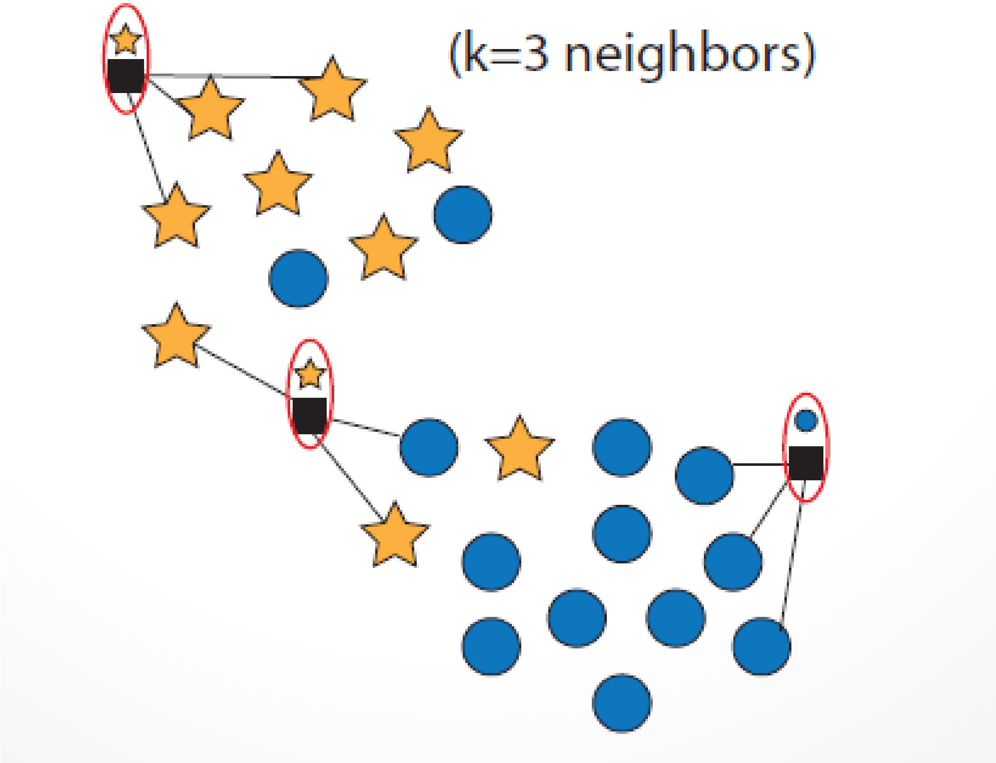
The nearest neighbors are defined by the value of "k"



kNN: Intuition

Identify several cases that are most similar to a given observation

Use the information from those "neighbors" to classify or predict the new observation



Considerations

- How should I measure similarity? Distance measures
  - Numeric attributes
  - Ordinal attributes
  - Categorical attributes
  - How do I combine these?
- How should I combine the results of neighbors?
  - Classification:
    - Majority rules (mode)?
    - Weight votes by nearness?
  - Regression/prediction:
    - Mean?
    - Median?
- How many neighbors should I use (value of k)?

DISTANCE FUNCTIONS

Building Distance Functions

- Numeric variables (includes some ordinal)
  - Some type of normalization or standardization is usually required
    - Standardize the variable before input to the model
    - OR standardize the distance in creating an overall distance matrix
  - Most common type of standardization
    - min-max normalization (feature scaling)

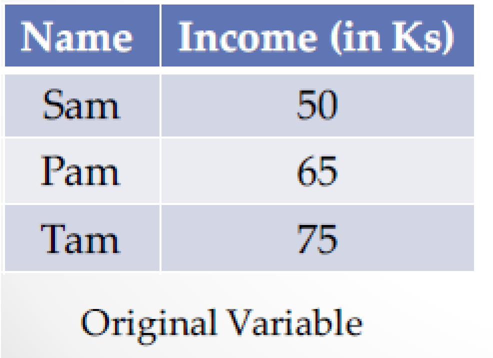
$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

    - z-score standardization

Building Distance Functions

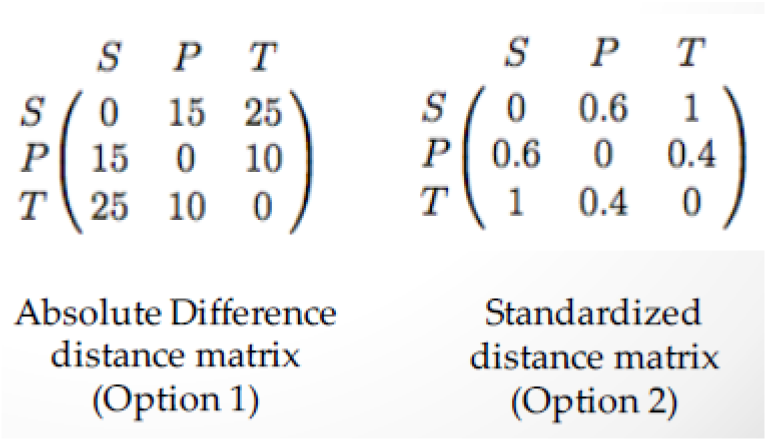
Options for computing distances: Absolute difference

- No standardization
- Standardize:
  - Divide by max distance (Scales the distance between 0 and 1)
  - Divide by standard deviation for that variable (How many standard deviations apart are the observations?)
  - Can even divide by the standard deviation for the distance!



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

Original Variable



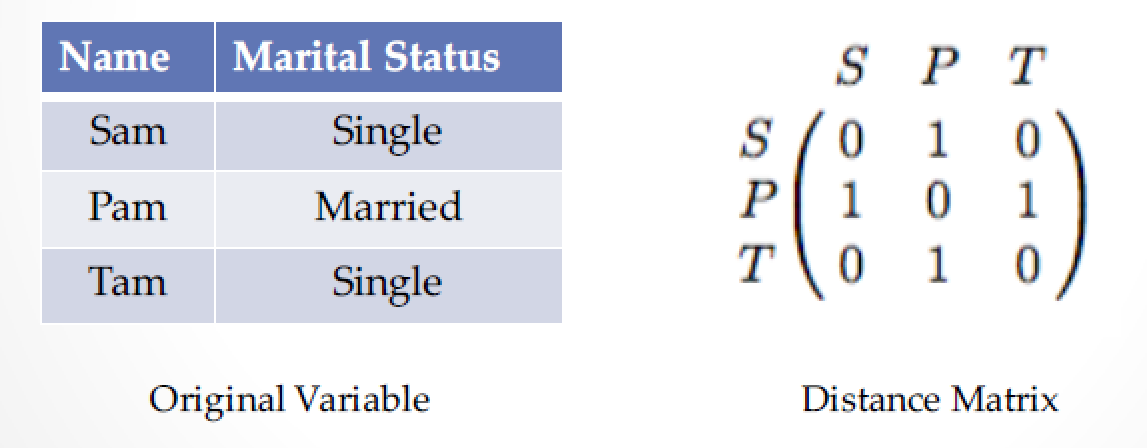
$$\begin{array}{c|ccc} & S & P & T \\ S & 0 & 15 & 25 \\ P & 15 & 0 & 10 \\ T & 25 & 10 & 0 \end{array} \qquad \begin{array}{c|ccc} & S & P & T \\ S & 0 & 0.6 & 1 \\ P & 0.6 & 0 & 0.4 \\ T & 1 & 0.4 & 0 \end{array}$$

Absolute Difference distance matrix (Option 1)

Standardized distance matrix (Option 2)

Building Distance Functions

- Categorical variables (includes some ordinal):
  - Distance=0 if matching, 1 otherwise



Name	Marital Status
Sam	Single
Pam	Married
Tam	Single

Original Variable

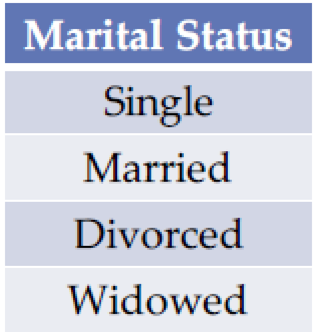
$$\begin{array}{c} \\ S \\ P \\ T \end{array} \begin{array}{ccc} S & P & T \\ \hline \end{array}$$

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

Distance Matrix

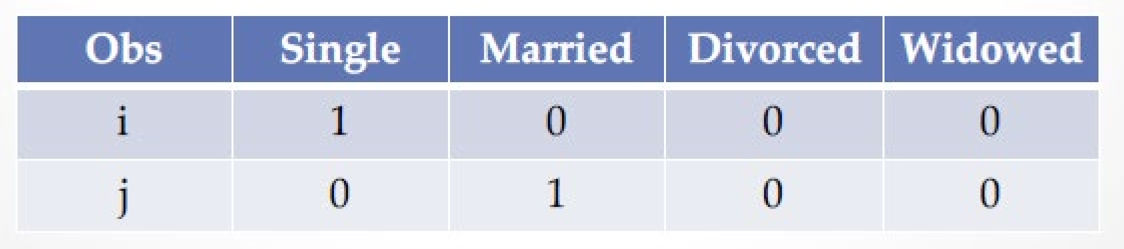
Building Distance Functions

- Suppose we have a categorical variable with four categories:



Marital Status
Single
Married
Divorced
Widowed

- Software default is usually to create 4 dummy variable (one-hot encoding) and treat them like numeric variables



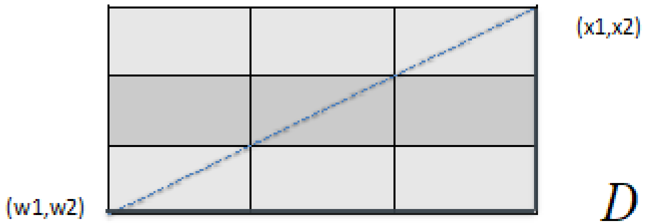
Obs	Single	Married	Divorced	Widowed
i	1	0	0	0
j	0	1	0	0

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

Building Distance Functions

- Used to build distance matrix
  - *Manhattan distance (L_1-norm)*
    - Measure of similarity in terms of the distance between a pair of objects
    - Distance between two points measured along axis at right angles
- Also used to merge the individual distances together



$$D=\sum_{i=1}^{d}|xi-wi|$$

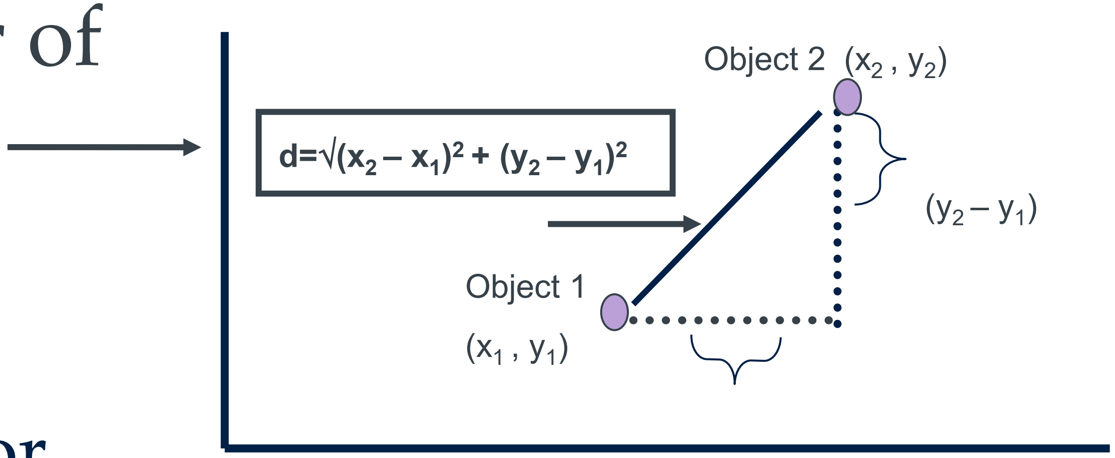
$$d_{x1}(i,j) + d_{x2}(i,j) + \dots + d_{x1}(i,j)$$

where

$d_{xk}(i,j)$ is the distance between observation i and j on variable x_k

Building Distance Functions

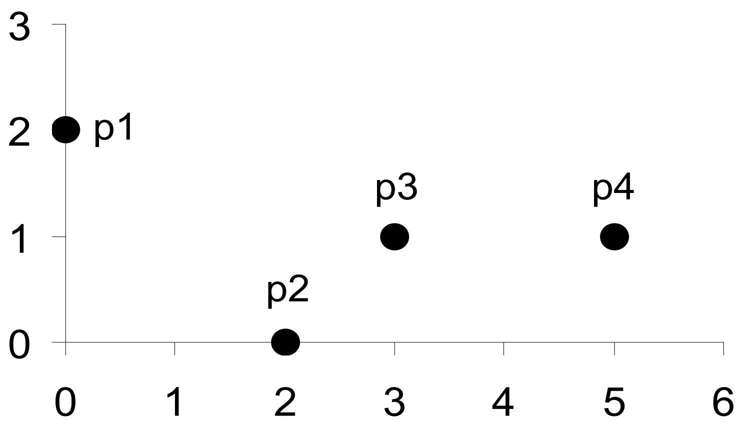
- Compute distance between pair of observations
  - *Euclidean distance (L_2-norm)*
    - Square root of the sum of the squared differences in values for each variable
- Merge the individual distances together



$$\sqrt{d_{x1}(i,j)^2 + d_{x2}(i,j)^2 + \ \dots \ + d_{xk}(i,j)^2}$$

where $d_{xk}(i,j)$ is the distance between observation i and j on variable x_k

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Building Distance Functions

- Compute distance between pair of observations (categorical variables only)
  – *Jaccard's Coefficient Measure*

$$J = \frac{a}{a+b}$$

  - *Jaccard's Distance Measure*

$$d_{ij} = \frac{b}{a+b} \ or\ 1 - J$$

  - *Dice's Coefficient Measure*

$$D = \frac{2a}{2a+b}$$

Gene expression levels under 17 variables (low=0, high=1)

	1 2 3 ……… 17
Gene i	0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 0 1
Gene j	0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 1 1

Match of 1s :	a =	0 1 1 0 0 0 0 0 1 0 0 1 1 1 0 0 1 = 7
Diff1 (mismatch):	b =	0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 0 = 5
Match of 0s :	c =	1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 = 5

J = 7/12 = 0.58

d_J = 0.42

D = 0.73

Building Distance Functions

- Compute distance between pair of observations
  - *Gower's Measure*
    - Find similarity between observations in a dataset consisting of mixed variables (continuous and categorical)
    - Uses Manhattan distance for continuous variables and Dice Coefficient for categorical variables

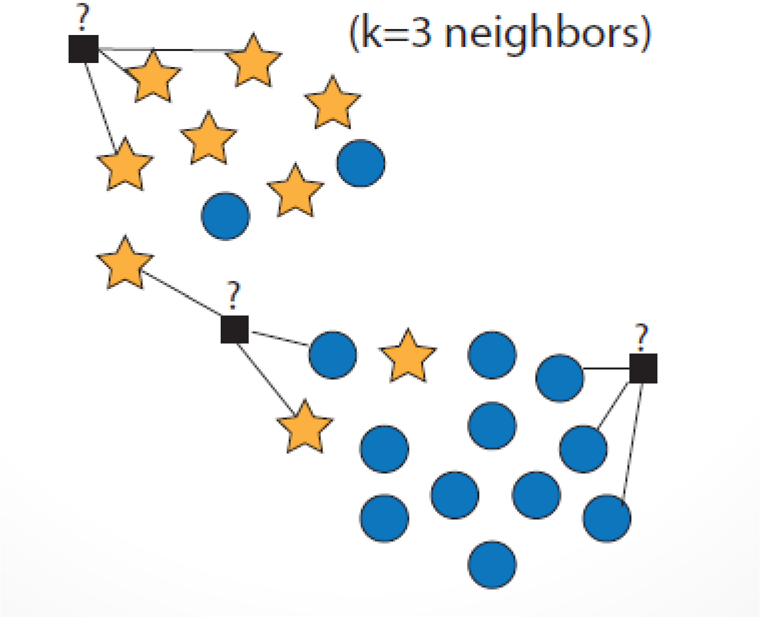
For a categorical variable k

$$d_{ijk} = \frac{\sum_k^n w_{ijk} d_{ijk}}{\sum_k^n w_{ijk}}$$

where d_{ij} is the distance between observation i and j;
d_{ijk} denotes the dice distance between i and j based on the kt_h variable;
w_{ijk} is usually 1 or 0 depending on if the k_{th} variable is categorical or not.

Combination Functions

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

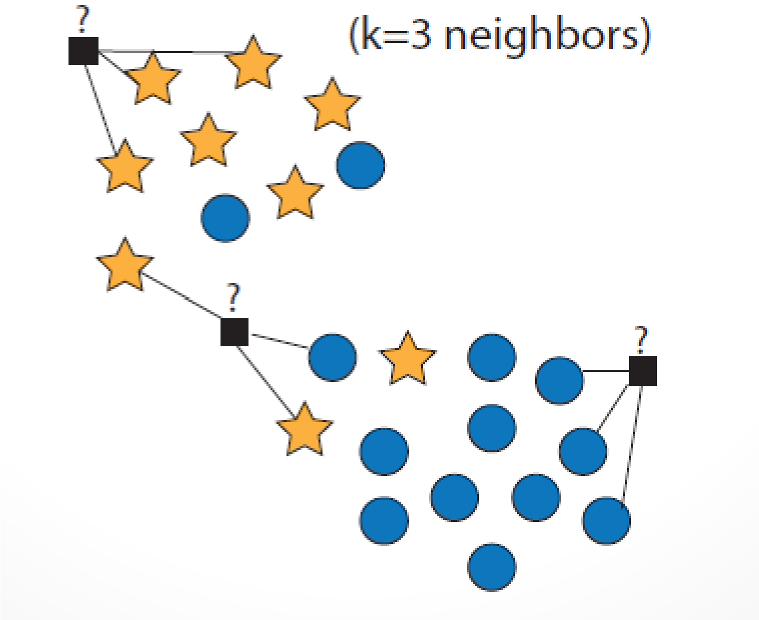


Combination Functions

- Numeric Target
  - Mean or median of the neighbor's 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
    - This can reduce the sensitivity to the parameter k

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

    - Add up the weighted votes to see which category has the most



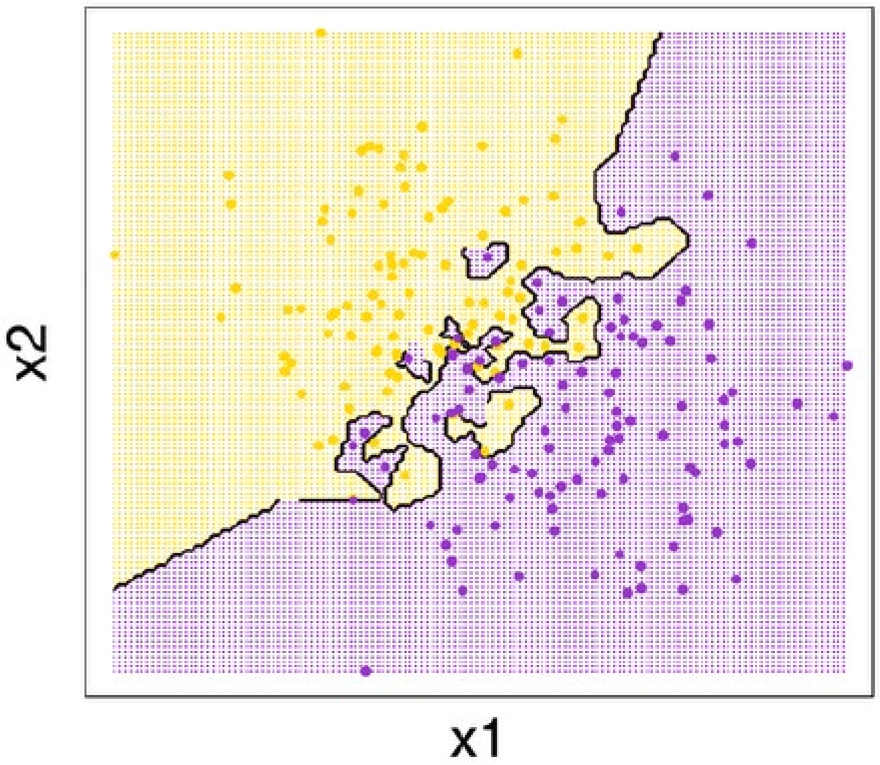
CHOOSING VALUE OF K

Similarity is Measured by Distance between Observations

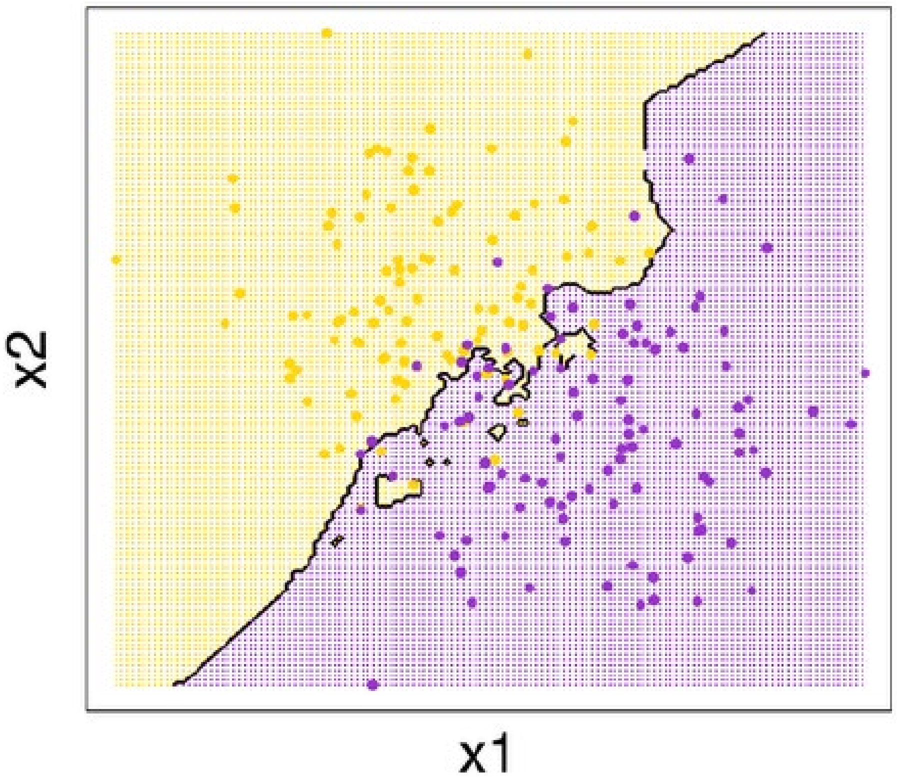
- With just one data point, you will have a kNN with k=1
- If you are around a group of 5 data points, each one in the group has an effect on the behavior of the data point and your value = mean of 5 data points = kNN with k=5
- kNN classifier determine class of a data point by majority voting principle or mean value of the neighbors for prediction

Choosing k

Binary kNN Classification (k=1)

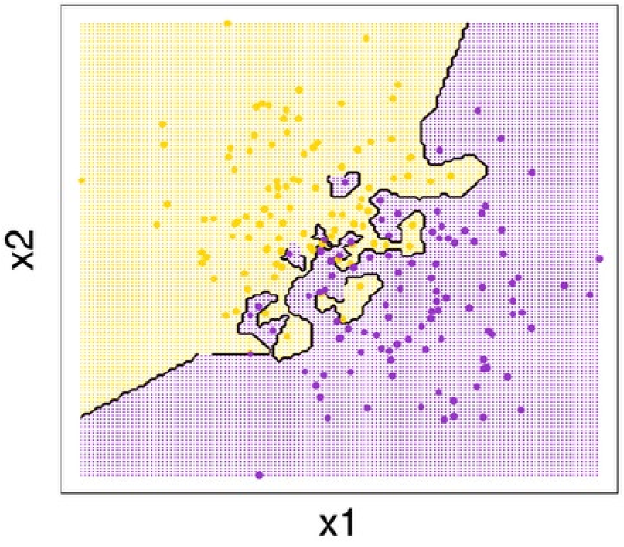


Binary kNN Classification (k=5)

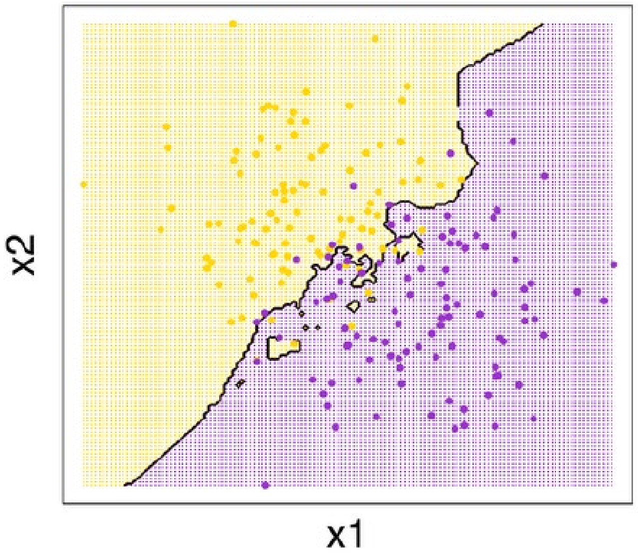


Choosing k

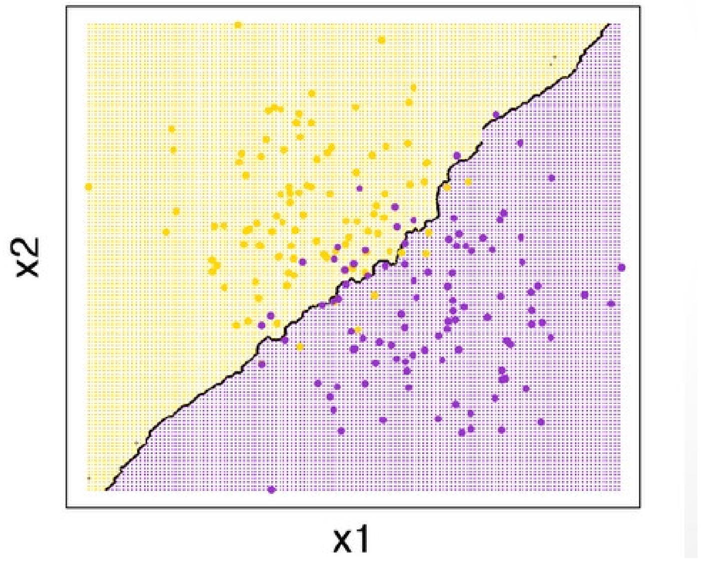
Binary kNN Classification (k=1)



Binary kNN Classification (k=5)



Binary kNN Classification (k=25)

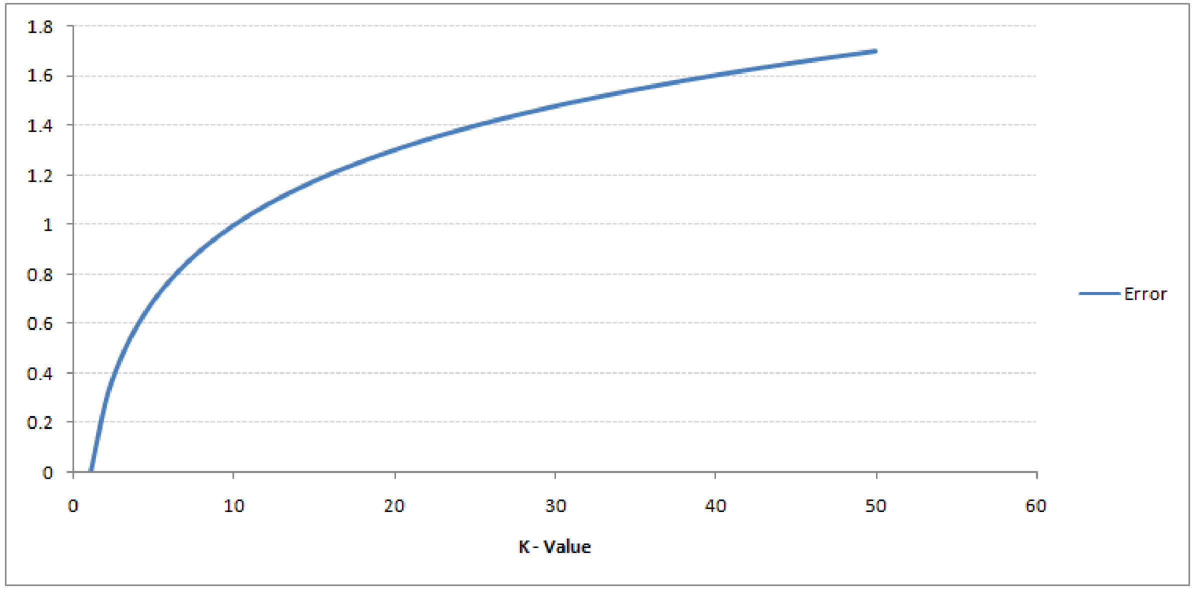


Choosing the Right Value of K

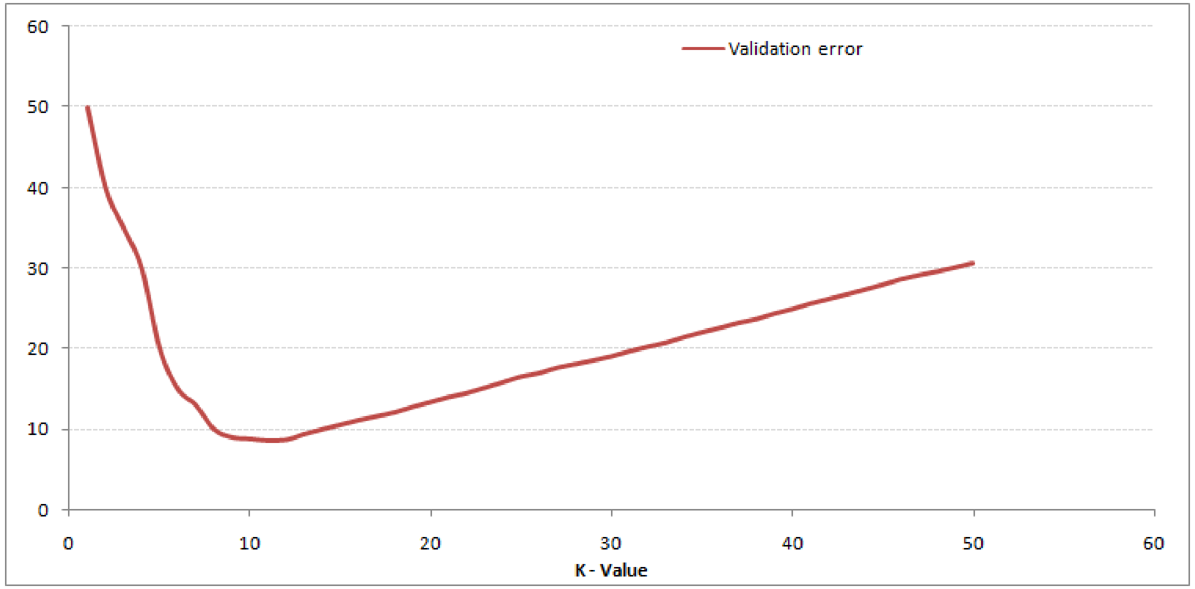
- Smaller values of k -> higher variance model
  - Tends toward overfitting
- Larger values of k -> higher bias model
  - Tends toward underfitting
- Best practice to tune this parameter with a validation set or with cross-validation
  - Run kNN several times with different values of k and choose k that reduces the RMSE in validation data
- Common practice to begin with k = $\sqrt{n}$, n= number of training samples

Choosing k

RMSE (y axis) for Training Set



RMSE (y axis) for Validation Set



kNN Algorithm

- Load the data
- Initialize k to your chosen number of neighbors
- For each example in the data:
  - Calculate the distance between the query example and the current example from the data
  - Add the distance and the index of the example to an ordered collection
- Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
- Pick the first k entries from the sorted collection
- Get the labels of the selected k entries
  - If regression, return the mean of the k labels
  - If classification, return the mode of the k labels

Advantages and Disadvantages of kNN

Advantage

- Highly unbiased, no prior assumption about underlying data
- The algorithm is simple and easy to implement
- There's no need to build a model, tune several parameters, or make additional assumptions
- The algorithm is versatile - It can be used for classification, regression, and search

Disadvantage

- The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase
- Results dependent on choice of distance function, no. of neighbors, k
- Does not produce a model: does not help us understand how features are related to classes

KNN in Practice

- Recommendation systems (products): Data consists of customers rating items (Netflix movies, Amazon.com products, etc.)
- Predict a customer's rating on a new product based on the way their nearest neighbors (other customers) rated that product.
- Search for semantically similar documents or observations
- Segmentation/Clustering
- Classification models with class variables (categorical outcome)
- …

Example: Predicting Red Wine Quality

The red wine dataset includes data on red *vinho-verde* wine samples, from the north of Portugal. The goal is to identify quality class based on physicochemical tests using kNN. Selected variables out of 12:

1. **Fixed acidity**: Most acids involved with wine are fixed or nonvolatile (do not evaporate easily)
4. **Residual sugar**: The amount of sugar remaining after fermentation stops, it is rare to find wines with less than 1 gram/liter; and wines with greater than 45 grams/liter are considered sweet.
5. **Chlorides**: The amount of salt in the wine
9. **pH**: Describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH
11. **Alcohol**: The percent alcohol content in the wine
12. **Quality**: Based on sensory data (score between 0- poor and 10 – excellent)

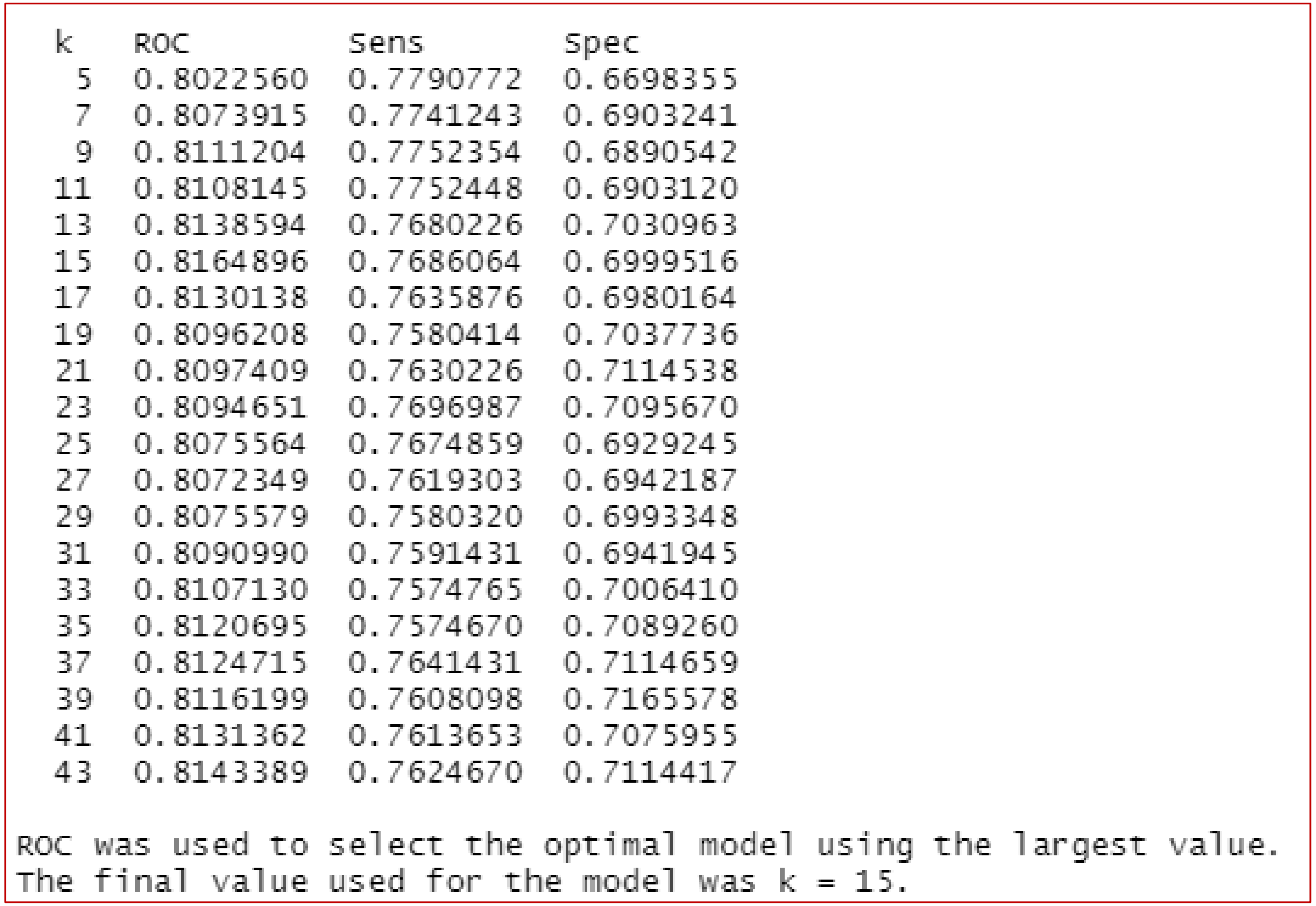
We collapsed the quality metric to two categories:
Low quality (Quality less than 5)
High quality (Quality 5 and over)

kNN Model Results

For the training data (70% of observations, the k value was tuned from 5-43 and ROC computed

Note that the sensitivity and specificity values are given as well

Model accuracy = 71%



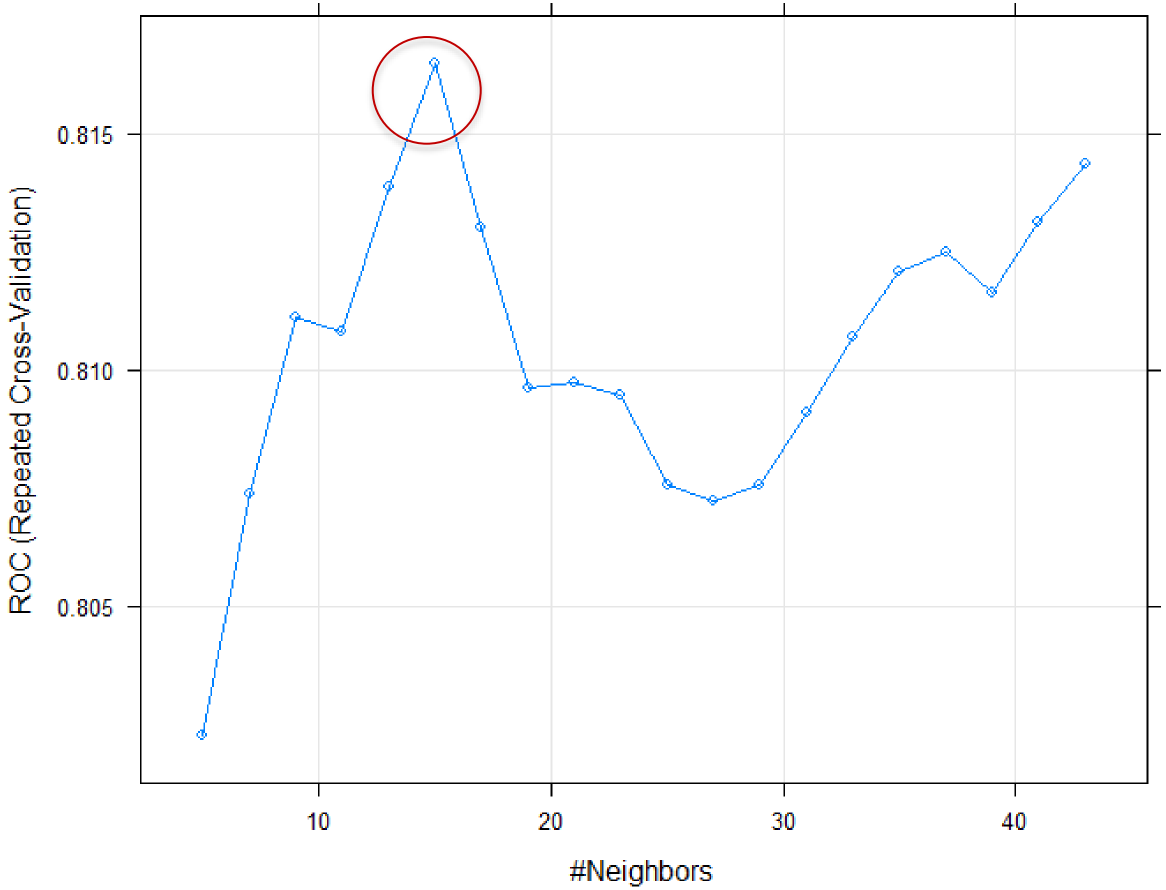
```
  k   ROC        Sens       Spec     
   5  0.8022560  0.7790772  0.6698355
   7  0.8073915  0.7741243  0.6903241
   9  0.8111204  0.7752354  0.6890542
  11  0.8108145  0.7752448  0.6903120
  13  0.8138594  0.7680226  0.7030963
  15  0.8164896  0.7686064  0.6999516
  17  0.8130138  0.7635876  0.6980164
  19  0.8096208  0.7580414  0.7037736
  21  0.8097409  0.7630226  0.7114538
  23  0.8094651  0.7696987  0.7095670
  25  0.8075564  0.7674859  0.6929245
  27  0.8072349  0.7619303  0.6942187
  29  0.8075579  0.7580320  0.6993348
  31  0.8090990  0.7591431  0.6941945
  33  0.8107130  0.7574765  0.7006410
  35  0.8120695  0.7574670  0.7089260
  37  0.8124715  0.7641431  0.7114659
  39  0.8116199  0.7608098  0.7165578
  41  0.8131362  0.7613653  0.7075955
  43  0.8143389  0.7624670  0.7114417

ROC was used to select the optimal model using the largest value.
The final value used for the model was k = 15.
```

kNN Model Results

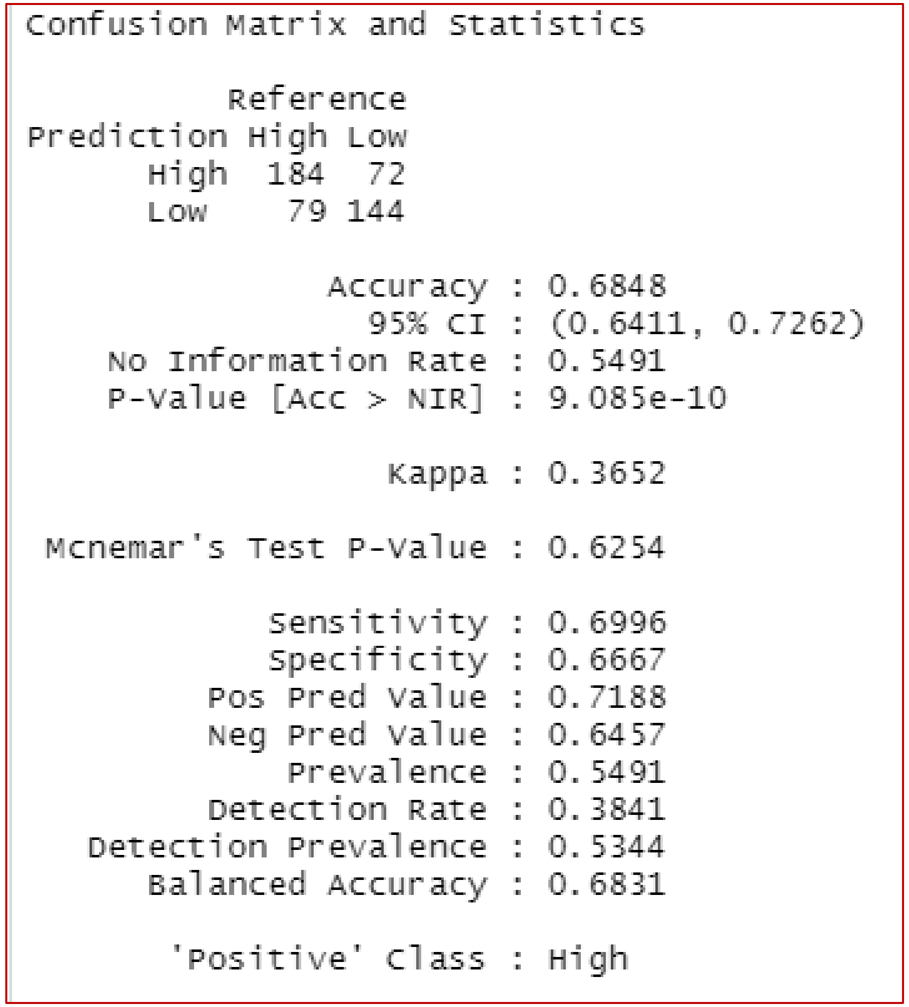
ROC for 10-fold cross validation

The final value used for the model was k=15



kNN Model Results

- Test Data (30% of data)
- The model results show that with 69% accuracy we can predict the quality of red wine using the attributes given



```
Confusion Matrix and Statistics

          Reference
Prediction High Low
      High  184  72
      Low    79 144

               Accuracy : 0.6848
                 95% CI : (0.6411, 0.7262)
    No Information Rate : 0.5491
    P-Value [Acc > NIR] : 9.085e-10

                  Kappa : 0.3652

 Mcnemar's Test P-Value : 0.6254

            Sensitivity : 0.6996
            Specificity : 0.6667
         Pos Pred Value : 0.7188
         Neg Pred Value : 0.6457
             Prevalence : 0.5491
         Detection Rate : 0.3841
   Detection Prevalence : 0.5344
      Balanced Accuracy : 0.6831

       'Positive' Class : High
```

kNN Summary

- The k-nearest neighbors (kNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems
- It’s easy to implement and understand but, has a major drawback of becoming significantly slower as the size of that data in-use grows
- kNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (k) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression)
- In the case of classification and regression, we saw that choosing the right k for our data is done by trying several k’s and picking the one that works best
- Finally, we looked at an example of how the kNN algorithm could be used in recommender systems, an application of kNN-search