Data Mining: Classification

CS4821 – CS5831

Laura Brown

Some slides adapted from: A. Moore, E. Alpaydin, G. Piatetsky-Shapiro; Han, Kamber, & Pei; C.F. Aliferis; S. Russell; D. Klein; L. Kaebling; A. Mueller; P. Smyth; C. Volinsky; Tan, Steinbach, & Kumar; J. Taylor; G. Dong;

Supervised Learning

- Training Examples

$$\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$$

  - Identical independent distributed (i.i.d.) assumption
- Binary classification $\mathcal{Y} = \{-1, +1\}$
- Multi-class classification $\mathcal{Y} = \{1, 2, \ldots, C\}$
- Regression $\mathcal{Y} \in \mathbb{R}$

Classification Process

- Given a collection of records (training set)

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$ where

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$$

- Each record contains a vector of attributes, and a class label, $y \in \mathcal{Y}$
- Use the data, $\mathcal{D}$, to find a model for the class label as a function of the attributes

$$\hat{f}(\boldsymbol{x}): \mathbb{R}^p \longmapsto \mathcal{Y}$$

- Use the model, $\hat{f}$, to predict class for new data

$$\hat{y} = \hat{f}(\boldsymbol{x_n})$$

Split Data

training set

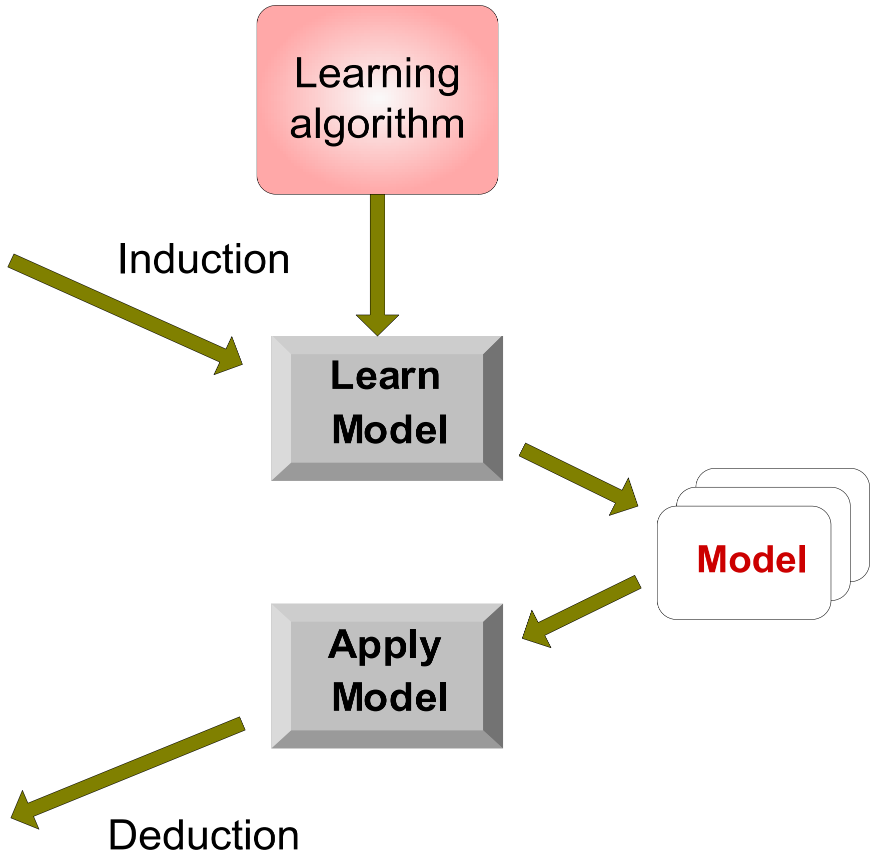
$$X = \begin{pmatrix} 1.1 & 2.2 \\ 6.7 & 0.5 \\ 2.4 & 9.3 \\ 1.5 & 0.0 \\ 0.5 & 3.5 \\ 5.1 & 9.7 \\ 3.7 & 7.8 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

test set

Classification Process

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Classification Models / Algorithms

- Memory-based reasoning
- Rule-based methods
- Probabilistic-based
  - Naïve Bayes
  - Bayesian Networks
- Regression-based
  - Logistic regression
  - Neural Network
- Discriminative
  - Decision Trees
  - Support Vector Machines (SVMs)
  - Neural Networks

LAZY LEARNER

Instance-based Learners

K Nearest Neighbors (KNN)

Lazy Learners

- Lazy learning (instance-based learners)
  Simply stores training data (or performs only minor preprocessing) and waits for test samples
- Eager learning (upcoming methods)
  Given a set of training samples, construct a classification model before receiving test data
- Lazy
  - Less time less time in training a model, may be longer time in predicting class of test sample

Nearest Neighbors (NN)

- Each training sample is a vector
- Remember (keep) all the training data
- When queried for new test sample's class
  - Find the nearest point(s), and return class based on the neighbor(s)

Not to be confused with Neural Networks (NN or ANN);
Nearest Neighbor is often abbreviated with kNN or KNN

What is “Nearest”?

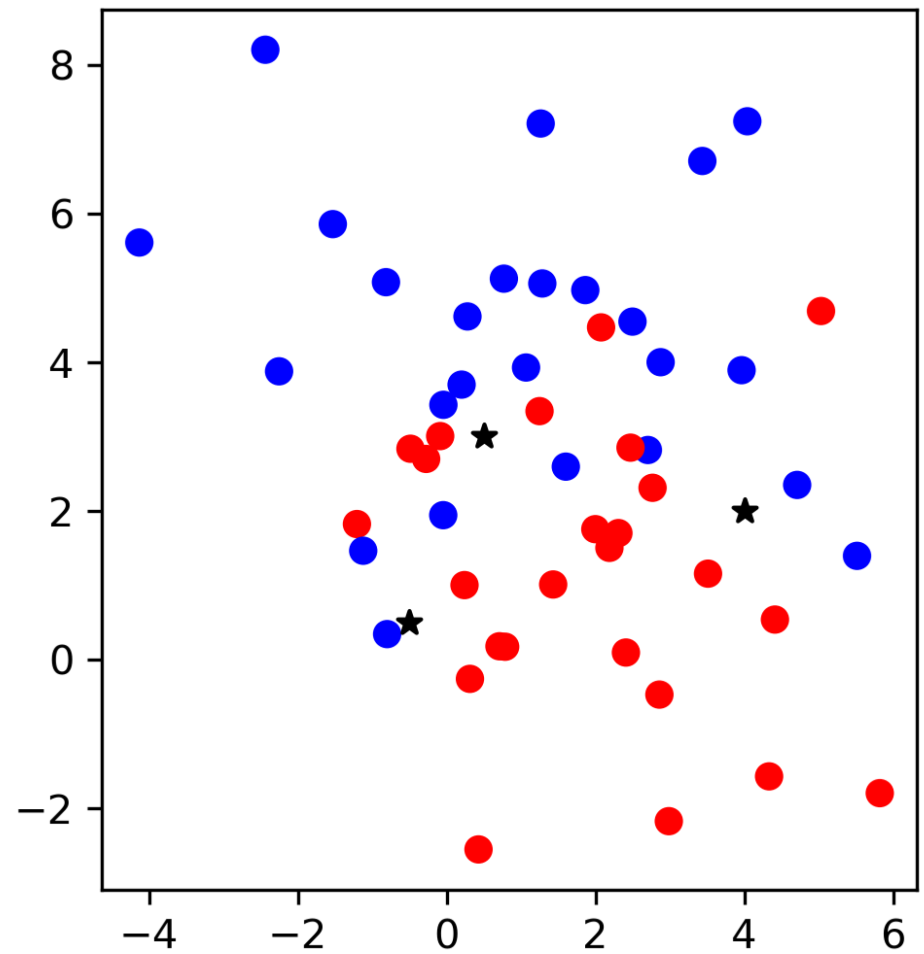
- Need to calculate or measure the distance between the test sample and the input records
- Often, Euclidean distance is used
  - Other distance measures can be used depending on the problem.

Nearest Neighbors

- 2-class classification (red/blue)
- 2 features
- 3 new samples (stars)

- Prediction for new $\boldsymbol{x}$

$$\hat{f}^{1-NN}(\boldsymbol{x}) = y_i,$$
$$i = argmin_j \, \|\boldsymbol{x}_i - \boldsymbol{x}\|_2$$

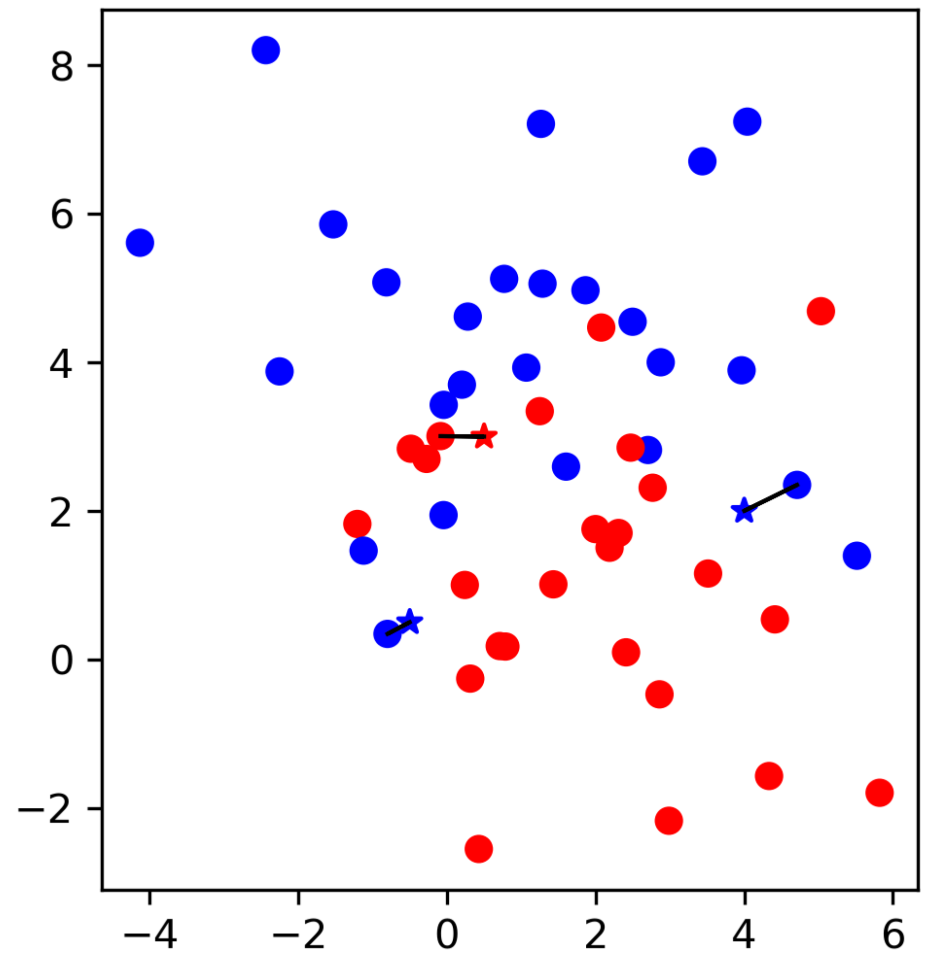


Nearest Neighbors

- 2-class classification (red/blue)
- 2 features
- 3 new samples (stars)

- Prediction for new ***x***

$$\hat{f}^{1-NN}(\boldsymbol{x}) = y_i,$$
$$i = argmin_j \|\boldsymbol{x}_i - \boldsymbol{x}\|_2$$



Noisy Data?

If the boundary between data classes is not clear, how is the prediction made with 1-NN?

- Expand to consider the *k* nearest neighbors
  - Predict given the majority values of the *k* nearest neighbors

- For example, with *k*=3 with classes of "+", "-", and "+", model would predict: "+"

KNN Algorithm

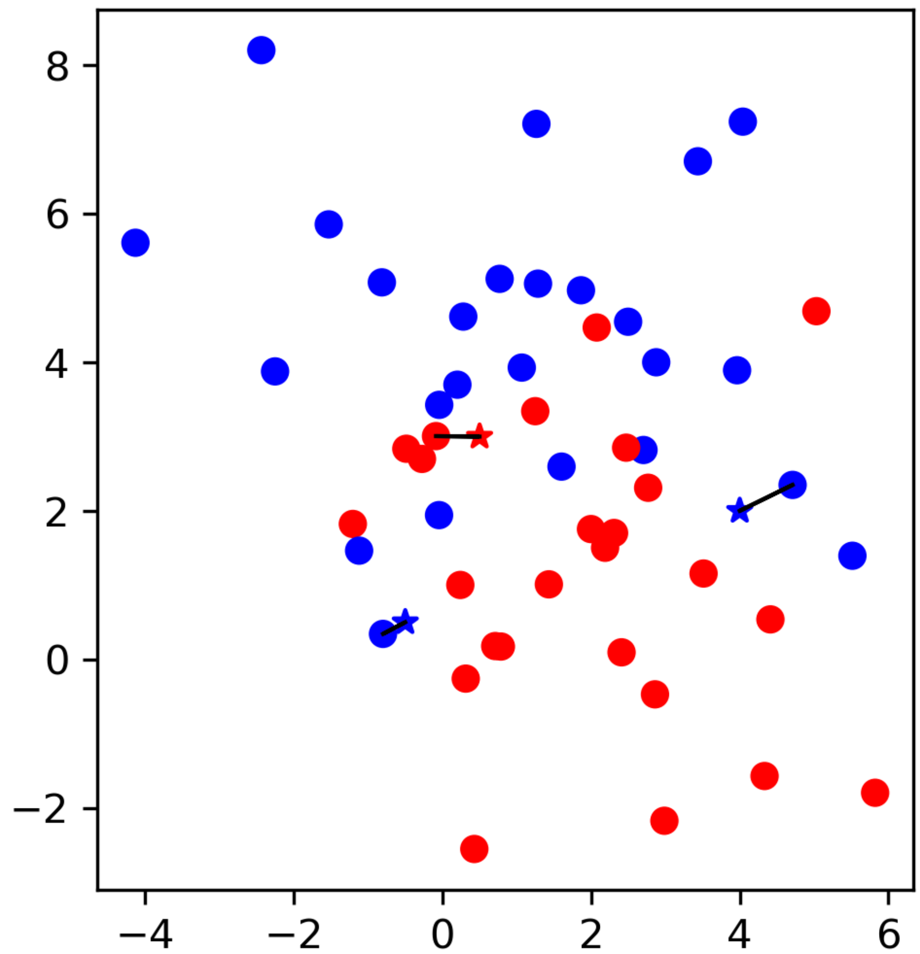
Given: training sample $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,n}$ distance function, *k*, and new input x

- Find the *k* closest examples with respect to the distance function, $\{j_1, \ldots, j_k\}$
- Return majority of class labels

KNN

- 2-class classification (red/blue)
- 2 features
- 3 new samples (stars)

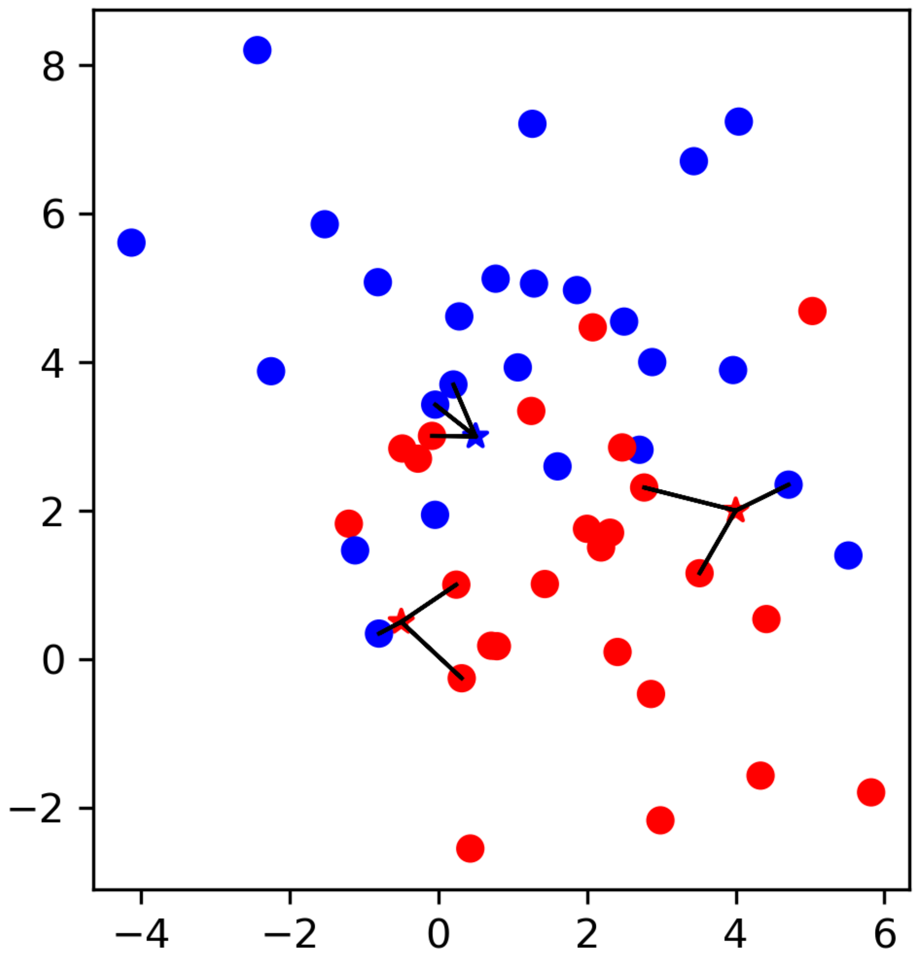
- Prediction for new ***x***
  k = 1



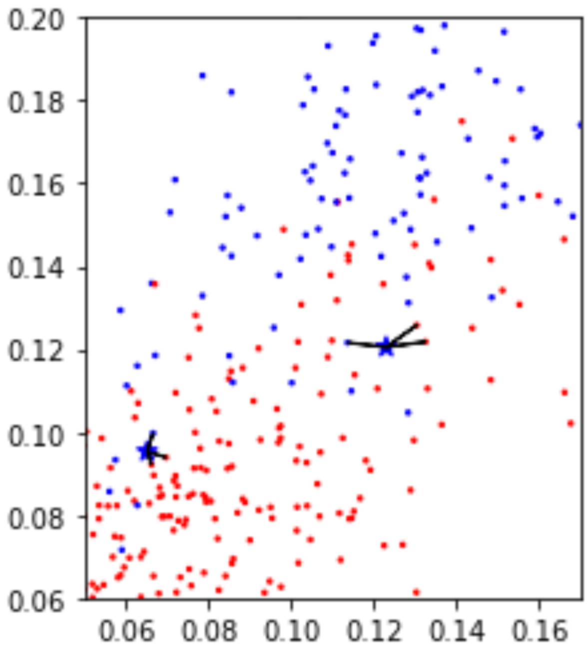
KNN

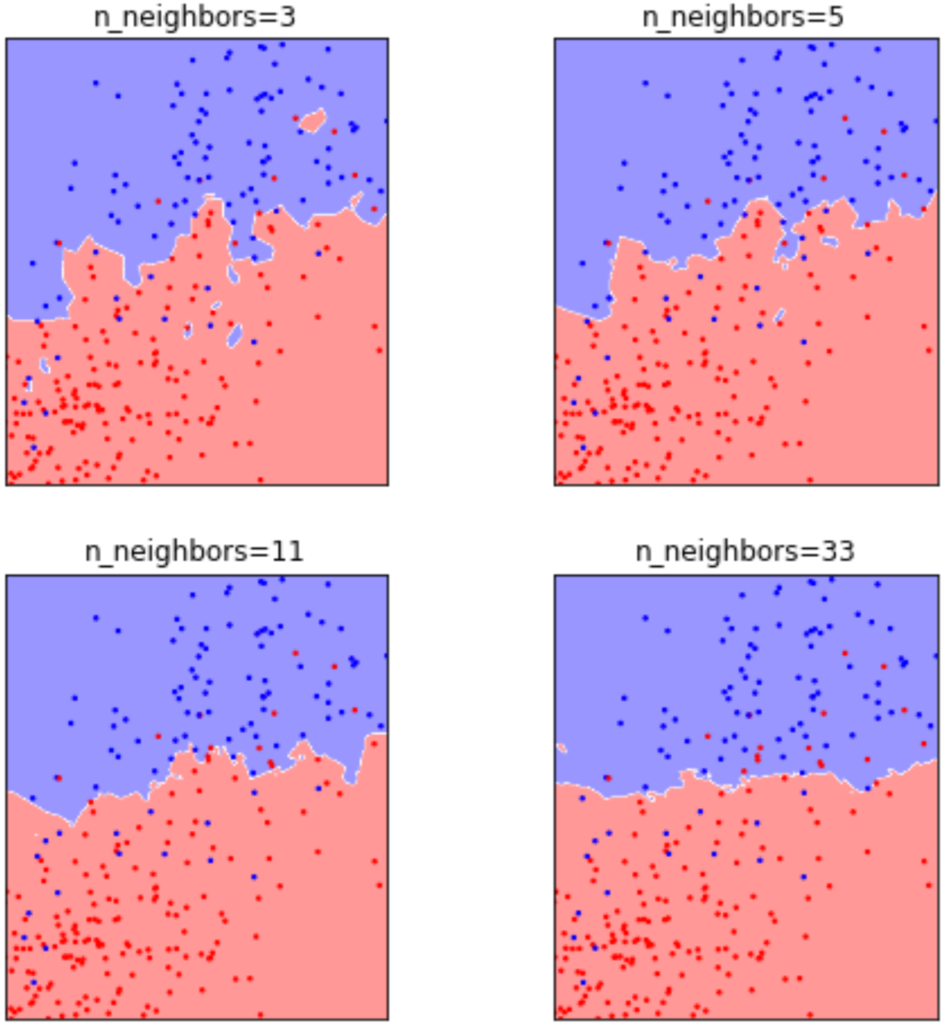
- 2-class classification (red/blue)
- 2 features
- 3 new samples (stars)

- Prediction for new ***x***
  k = 3



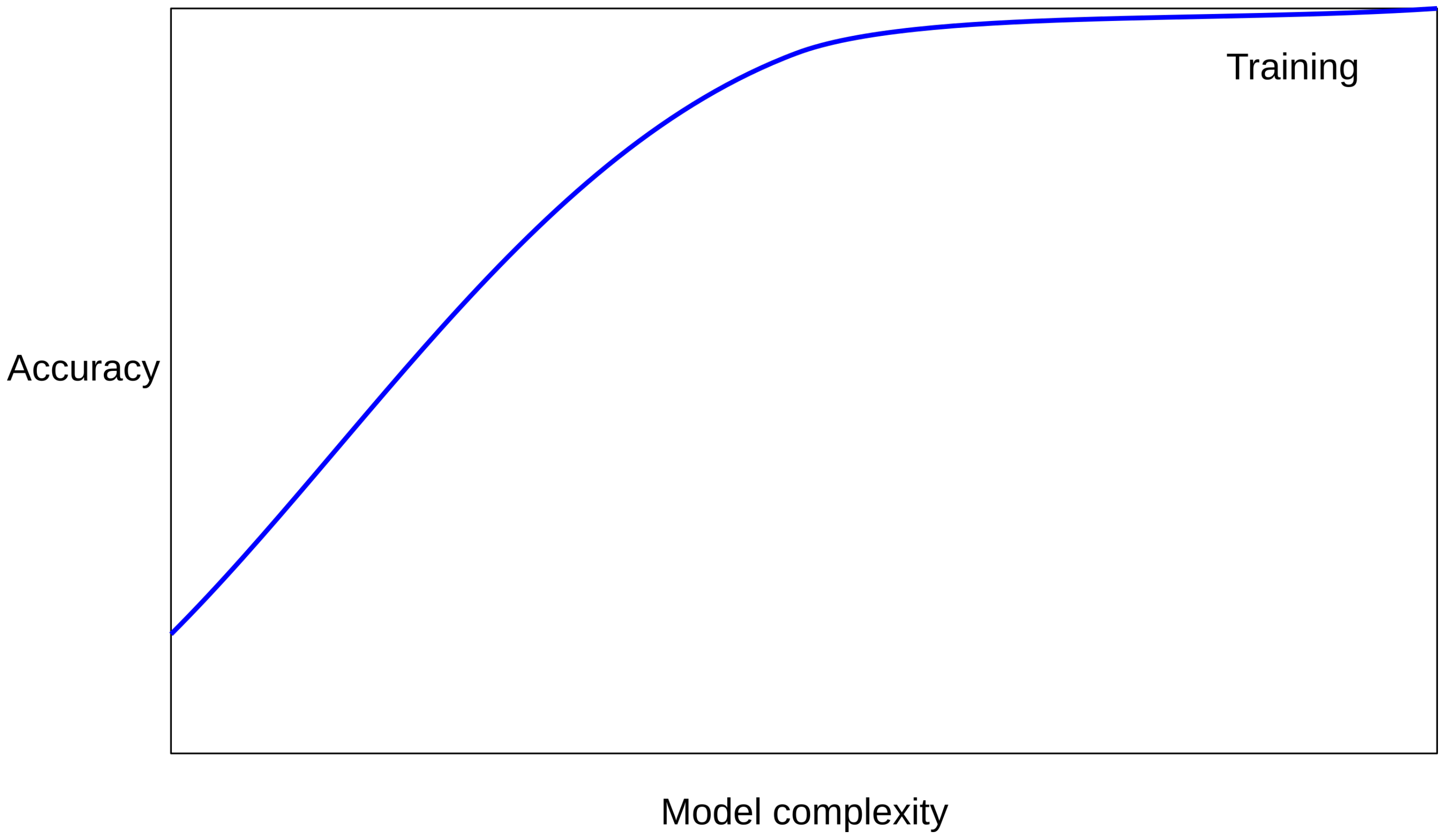
How to select k?



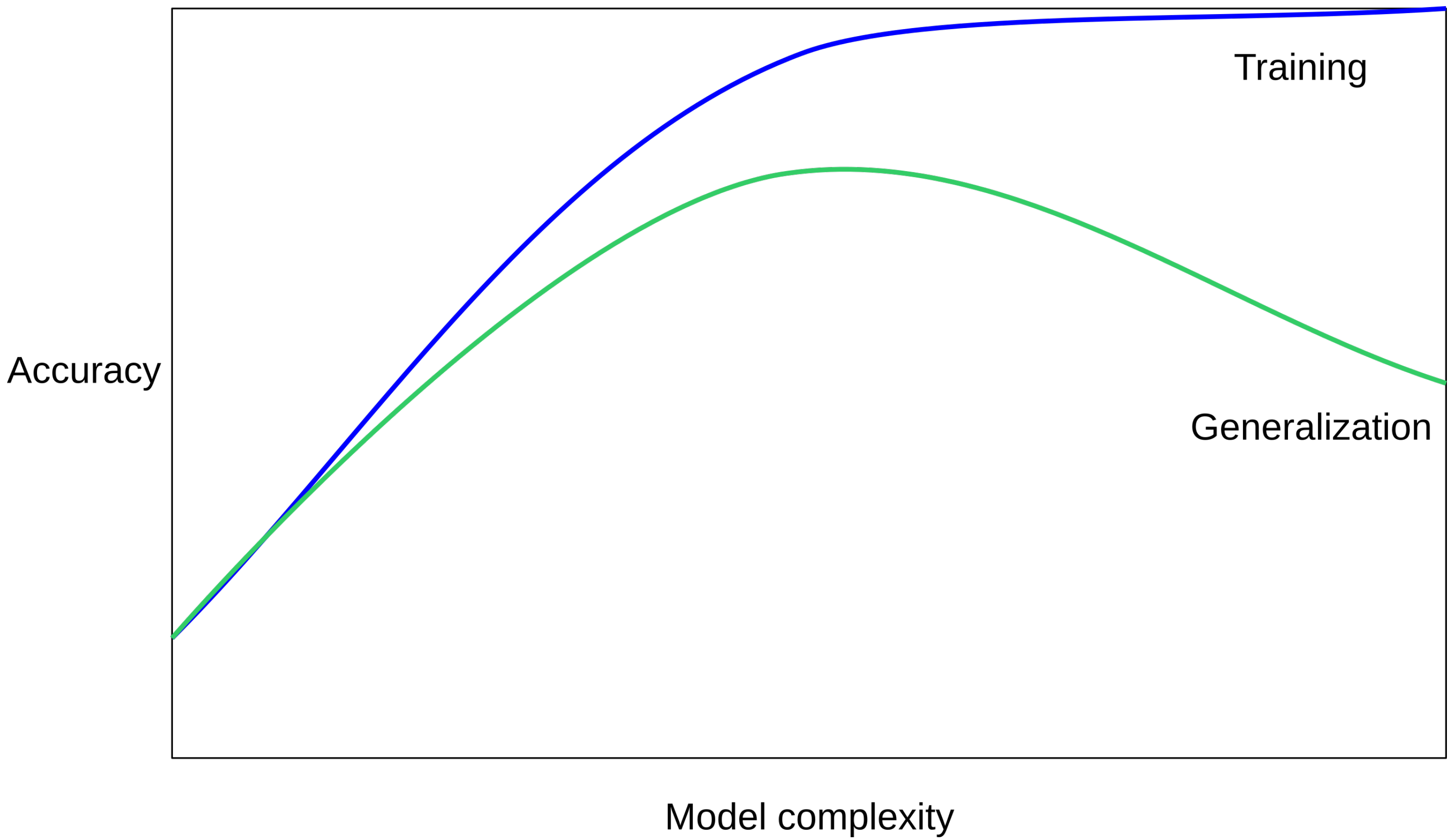


- K acts as a smoother

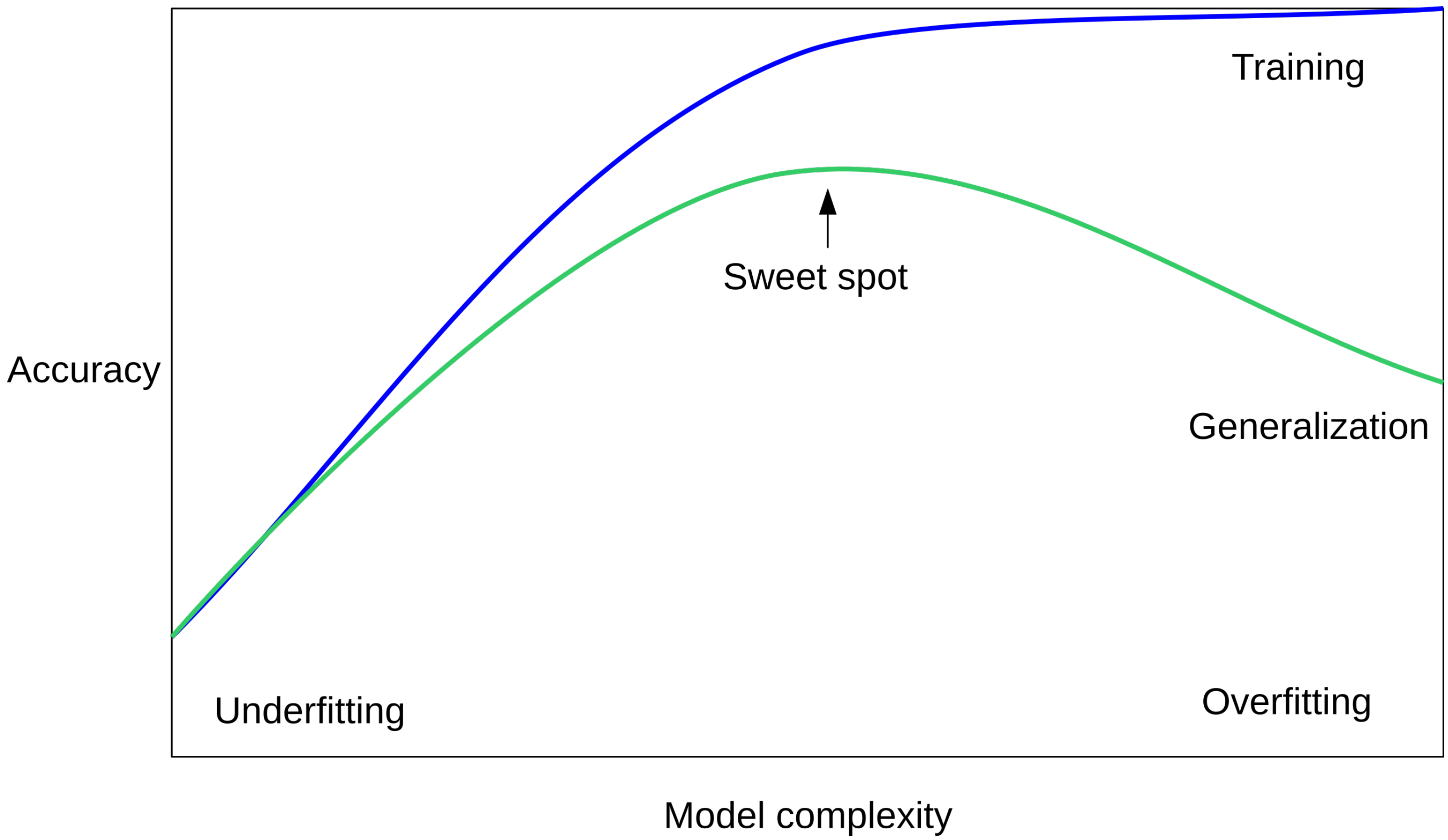
Overfitting and Underfitting



Overfitting and Underfitting



Overfitting and Underfitting



Nearest Neighbor Summary

- Prediction is slow as training data grows
  - Fit: no time
  - Memory: O(*n***p*)
  - Predict: O(*n***p*)
- Memory over time
  - Keeps all training data in memory
  - Solution: Can store data in clever data structures

  - What if the training data grows (you continue to see points and keep adding them to memory), may run out
  - Solution - delete points far away from boundaries