CDS

Cornell Data Science

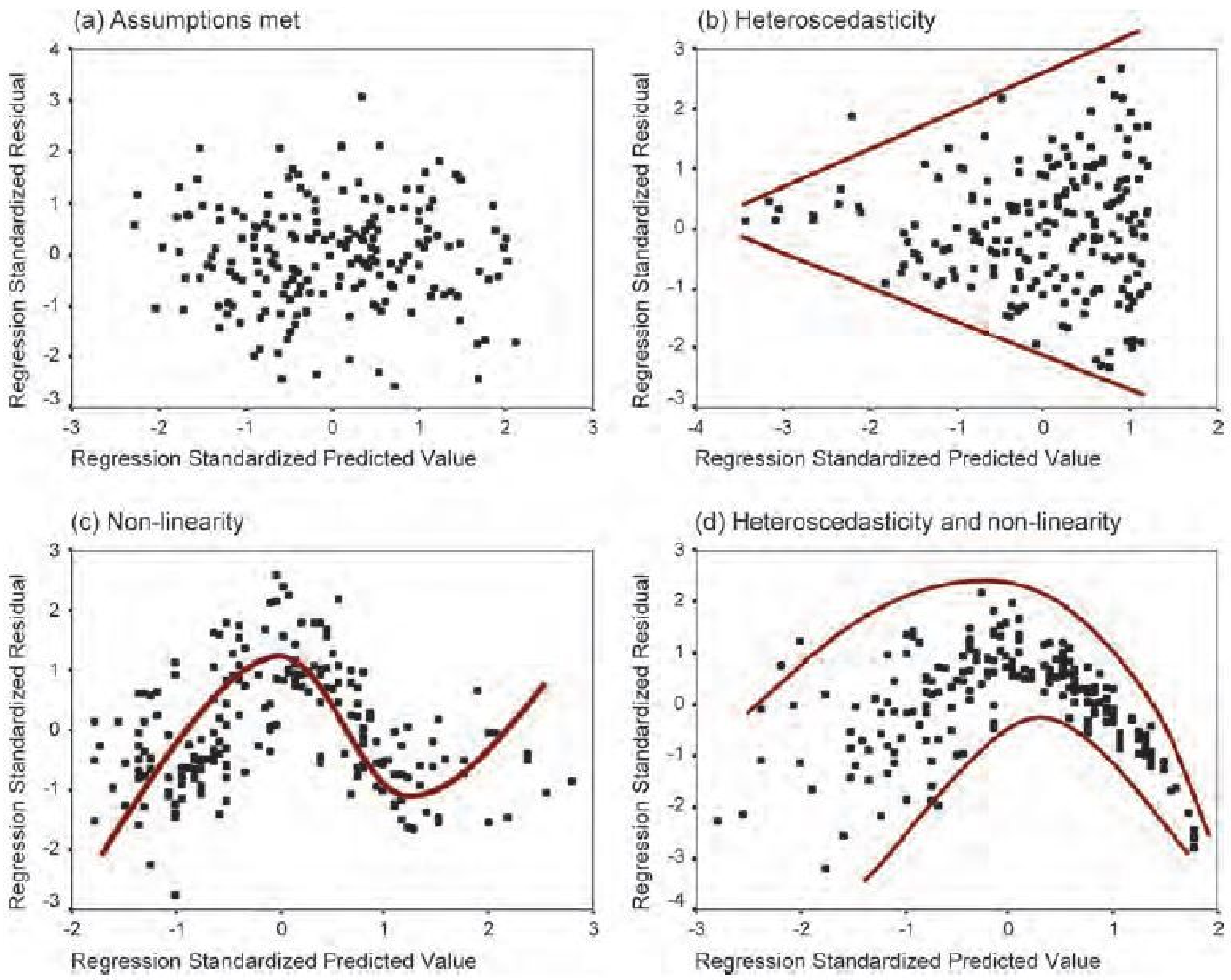
Intro to Classification

Question:

Last week we talked about regression. What is supervised learning? What is regression?

Heteroscedasticity

- Variability of the data
- Check whether assumption of linearity is valid



Source

Review:Least Squares Error

We define our error as follows:

theoretical

$$\sum_{i=0}^{n}(y_i - (B_0 + B_1 x_1 + ... + B_p x_p))^2$$

observed

We call this **Least Squares Error**. Sum of squared *vertical* distance between observed and theoretical values.

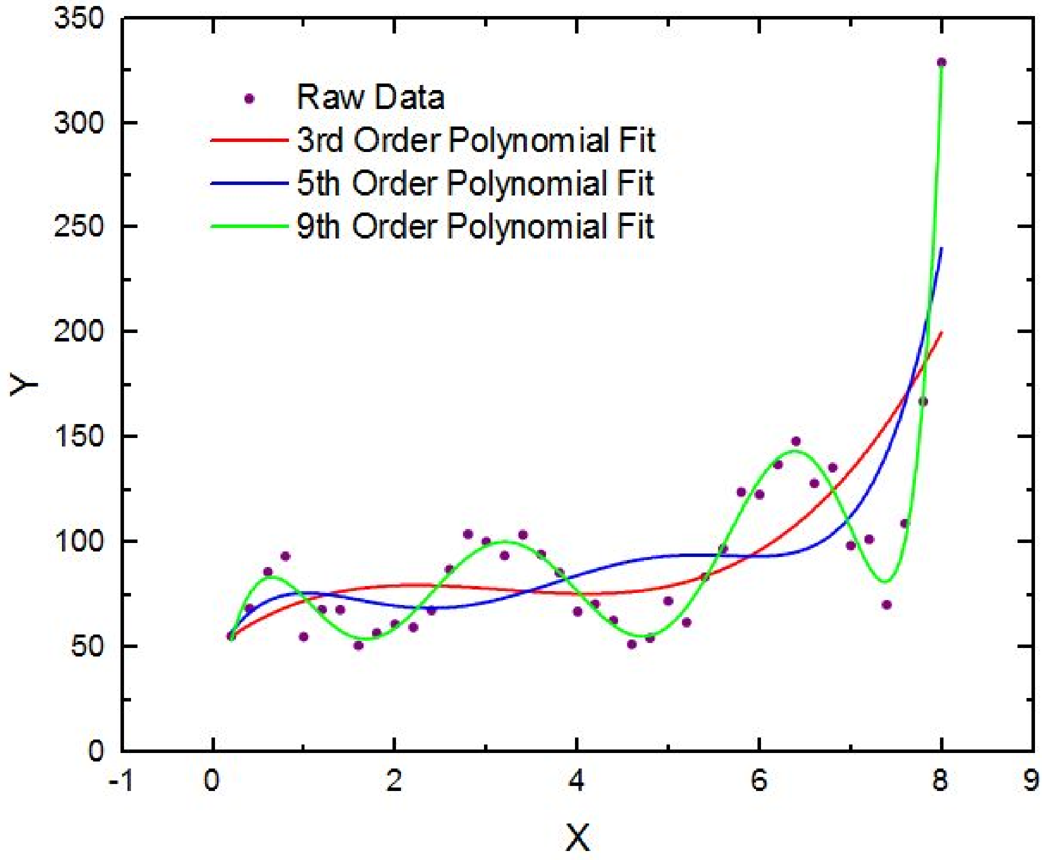
Model "Goodness of Fit"

Common metric is called $\mathbf{R^2}$.

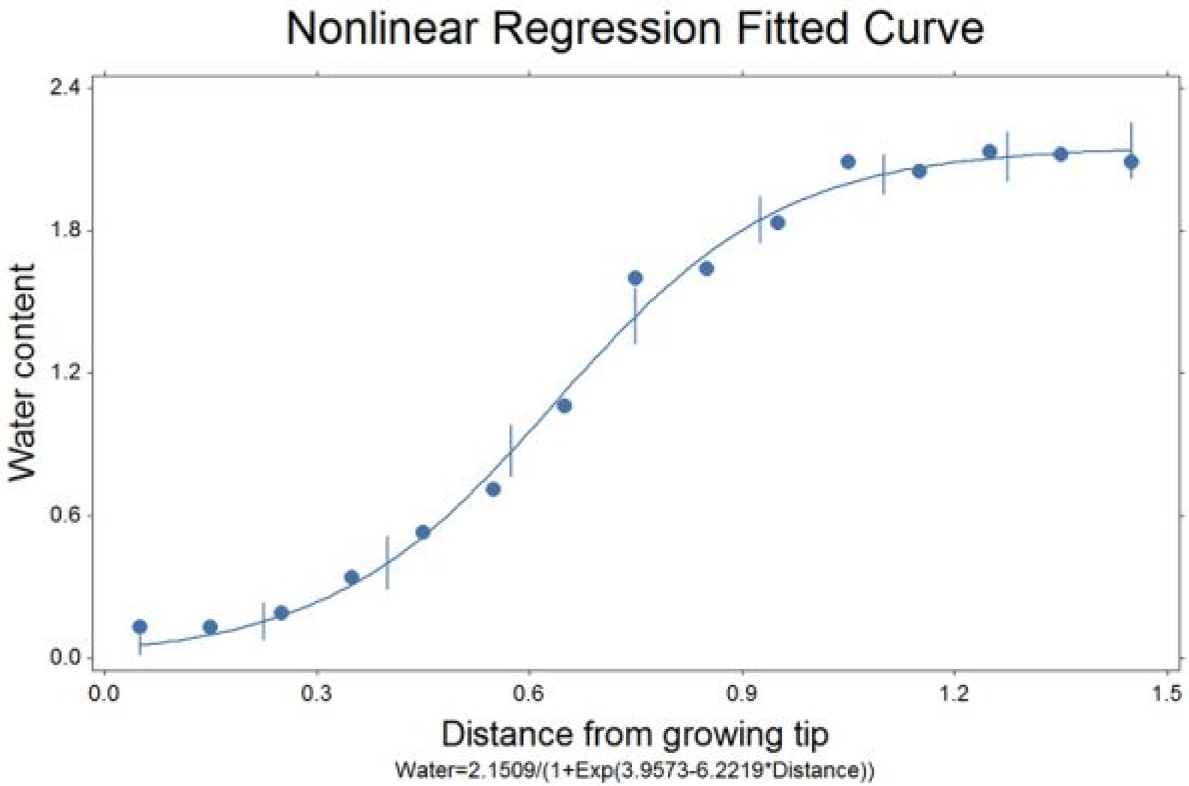
- We compare our model to a **benchmark model**
  - Predict the mean y value, no matter what the x_i's are
- SST = least-squares for benchmark
- SSE = least-squares error for our model
- $R^2 = 1 - SSE/SST$

Source

Non-linear Regression



Source



Source

Intro to Classification

- “What species is this?”
- “How would consumers rate this restaurant?”
- “Which Hogwarts House do I belong to?”
- “Am I going to pass this class?”

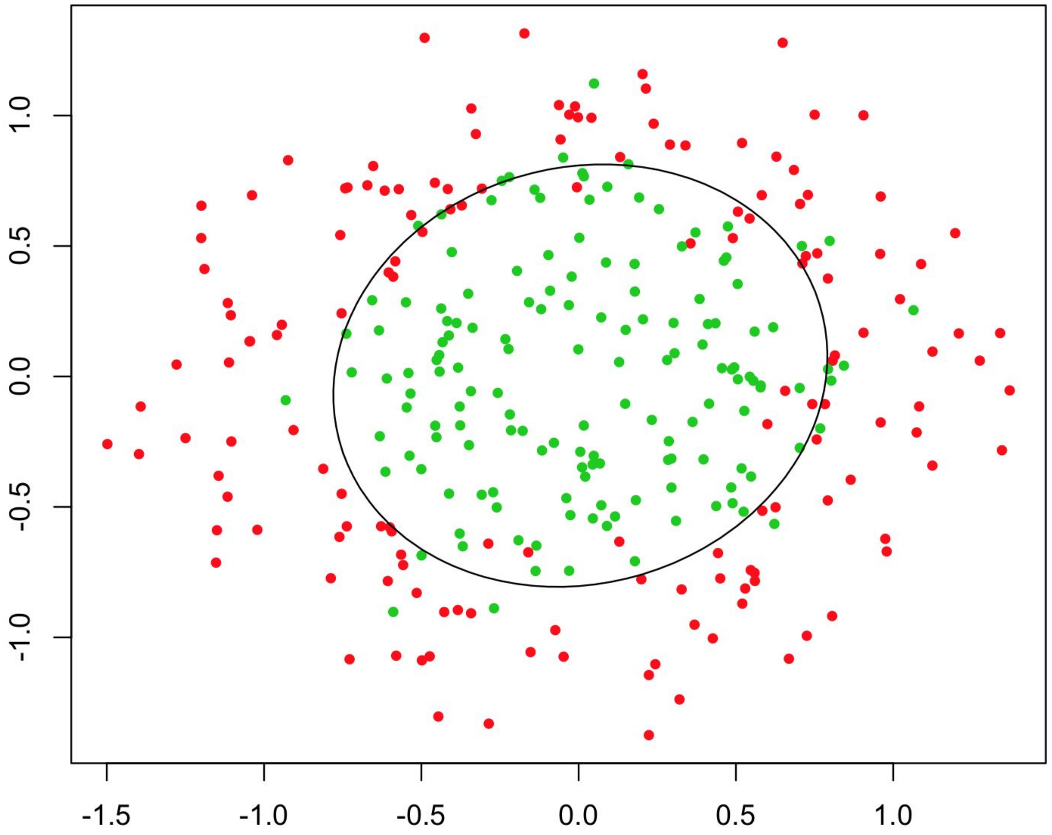
Source

The Bayesian Classifier

- The ideal classifier: a theoretical classifier with the highest accuracy
- Picks the class with the highest conditional probability for each point
- Assumes conditional distribution is known
- Exists only in theory!
- A conceptual **Golden Standard**

Decision Boundary

- The **decision boundary** partitions the outcome space
- Classification algorithm you should use differs depending on whether the data is or is not linearly separable



Source

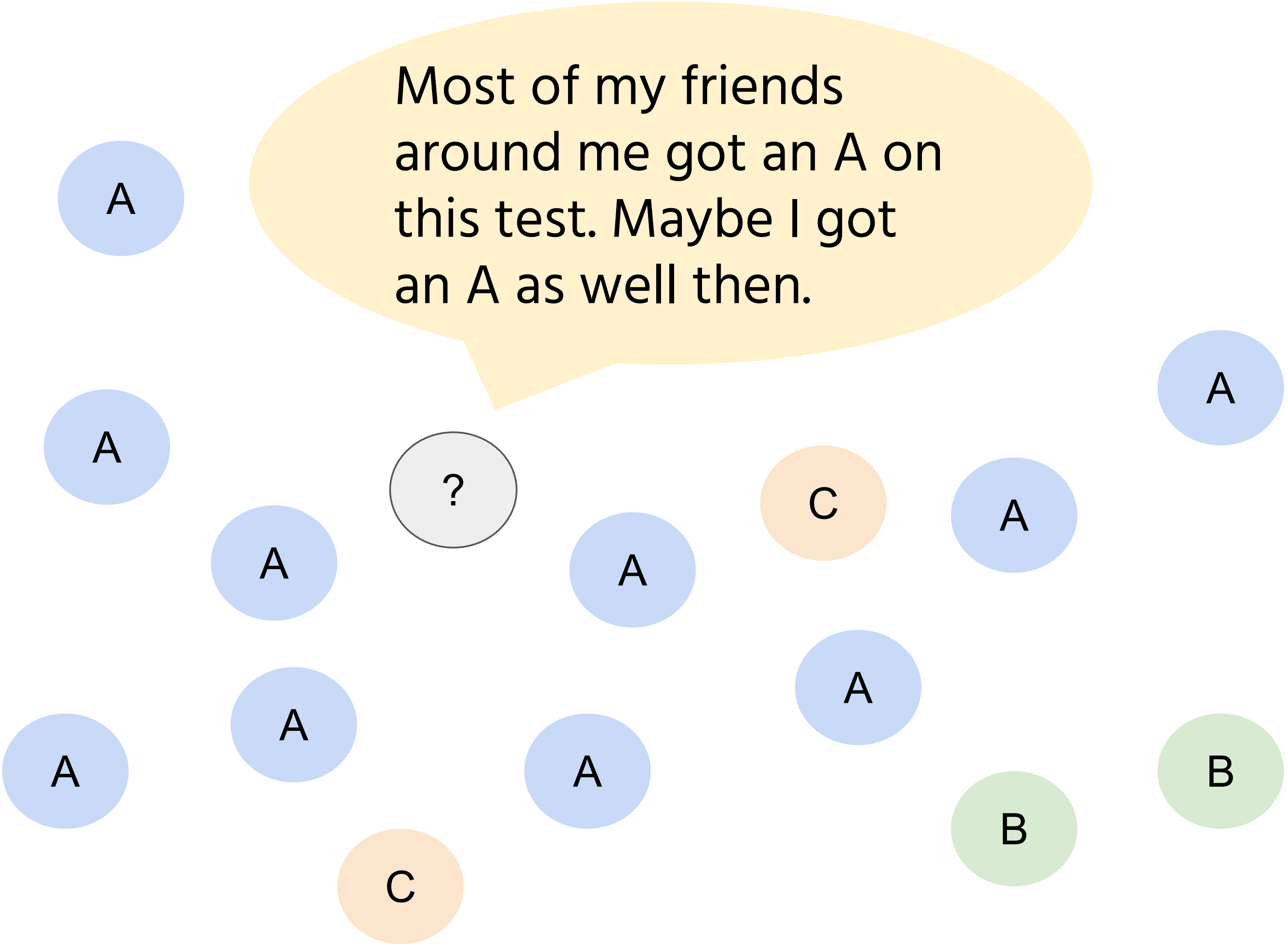
k-Nearest Neighbors (KNN)

Easy to interpret

Fast calculation

No prior assumptions

Good for coarse analysis



KNN

How does it work?

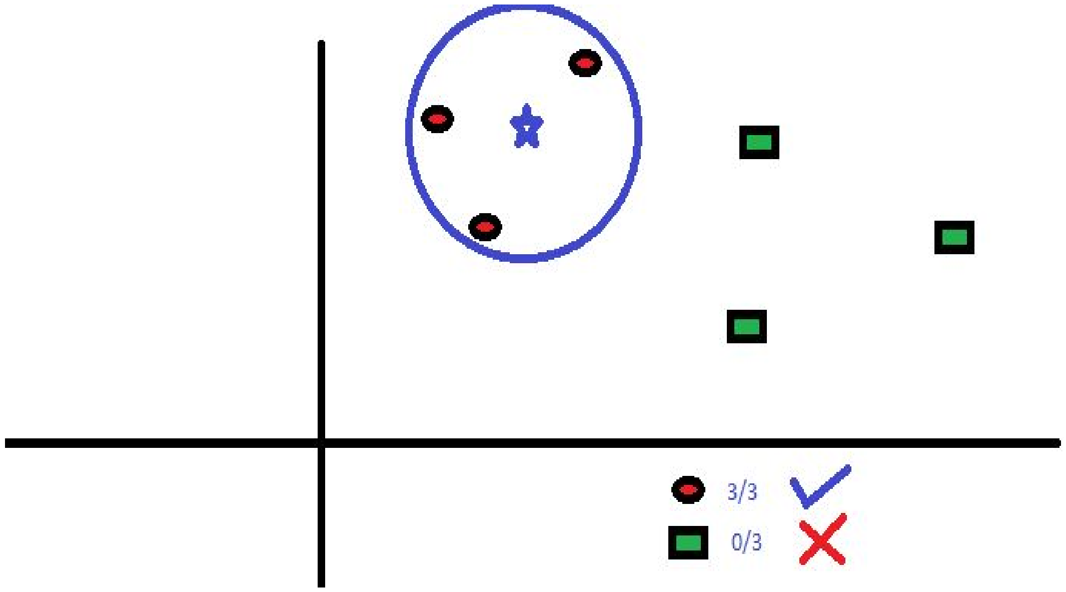
Define a *k* value (in this case k = 3)

Pick a point to predict (blue star)

Count the number of closest points

Increase the radius until the number of points within the radius adds up to 3

Predict the blue star to be a red circle!



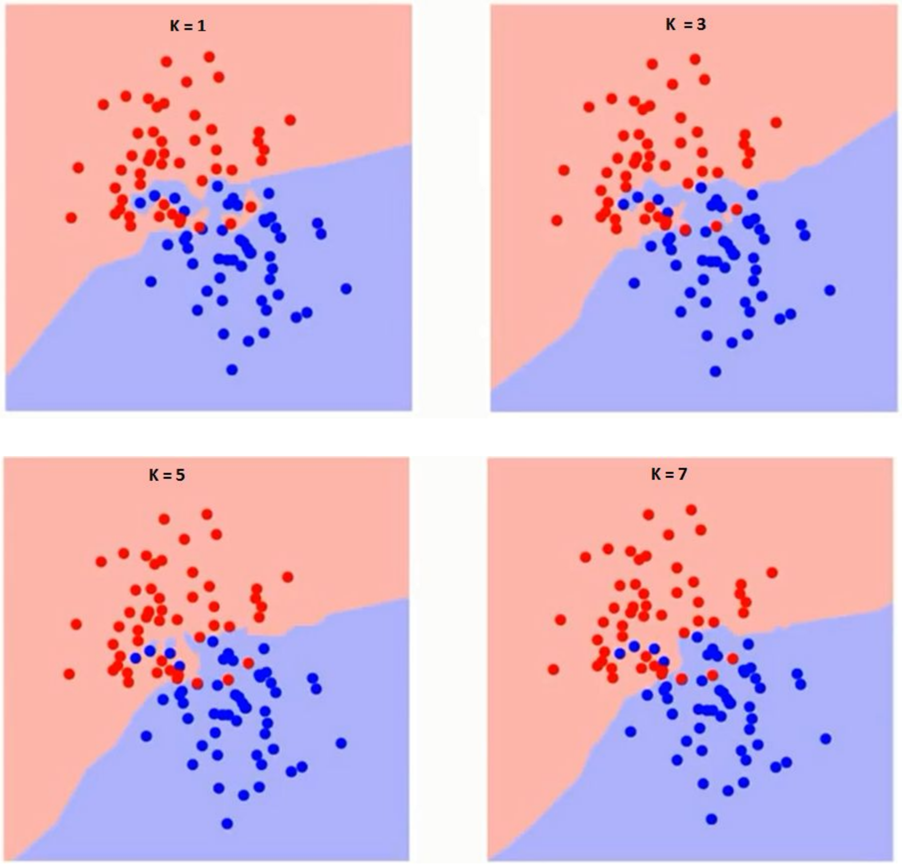
Source

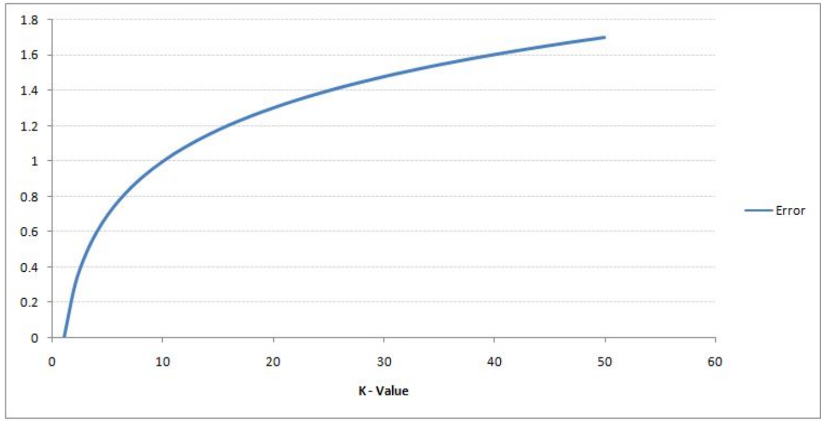
Question:

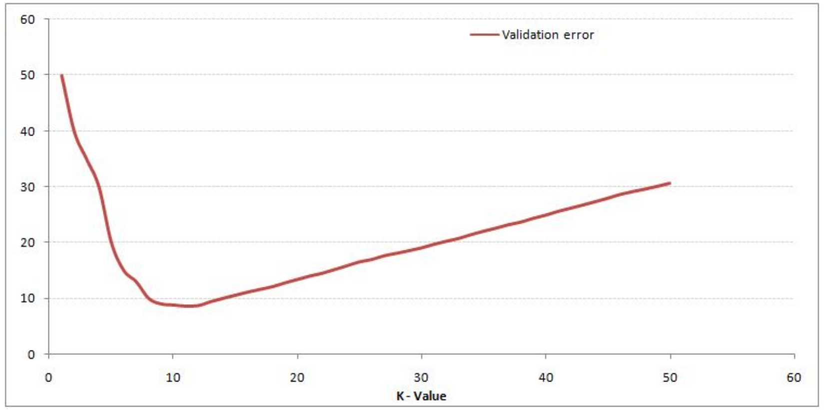
What defines a good k value?

KNN

The *k* value you use has a relationship to the fit of the model.

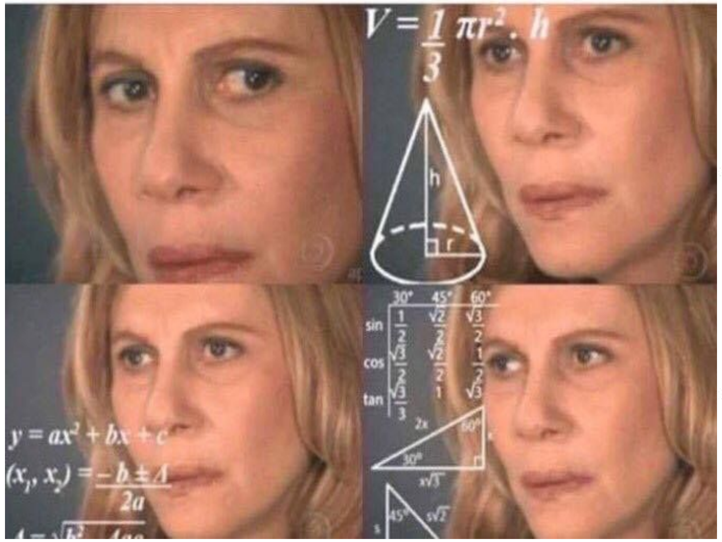




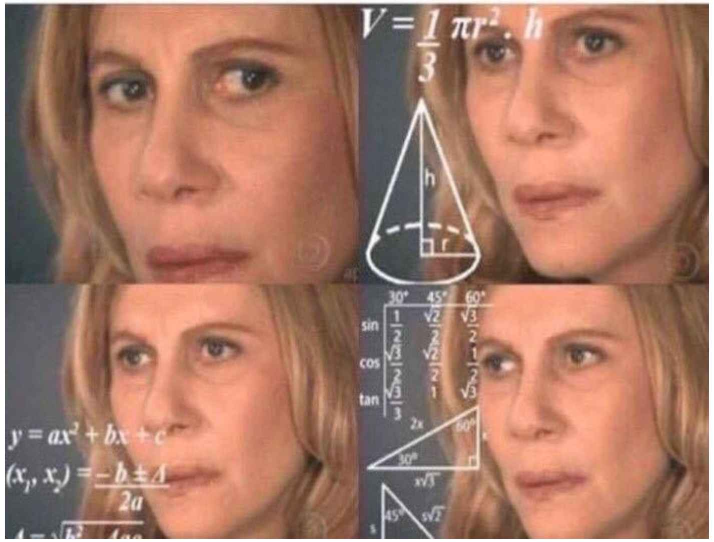


Source

Confusion Matrix



	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative



Sensitivity

Sensitivity = True Positive/ (True Positive + False Negative)

Also called **True Positive Rate**.

How many positives are correctly identified as positives?

Useful for:

- Airport security
- Initial diagnosis of fatal disease

Source

Specificity

Specificity = True Negative/
(True Negative + False Positive)

Also called **True Negative Rate**.

How many negatives are correctly identified as negative?

Question:

Name some examples of situations where you'd want to have a high specificity.

Other Important measures

- **Overall accuracy** - proportion of correct predictions
- **Overall error rate** - proportion of incorrect predictions
- **Precision** - proportion of correct positive predictions among all positive predictions
- **Recall** = sensitivity

Accuracy =
(True Positive + True Negative)/Total
Error Rate =
(False Positive + False Negative)
/Total

Precision =
True Positive
/(True Positive + False Positive)
Recall =
True Positive
/(True Positive + False Negative)

Example

Given this confusion matrix, what is the:

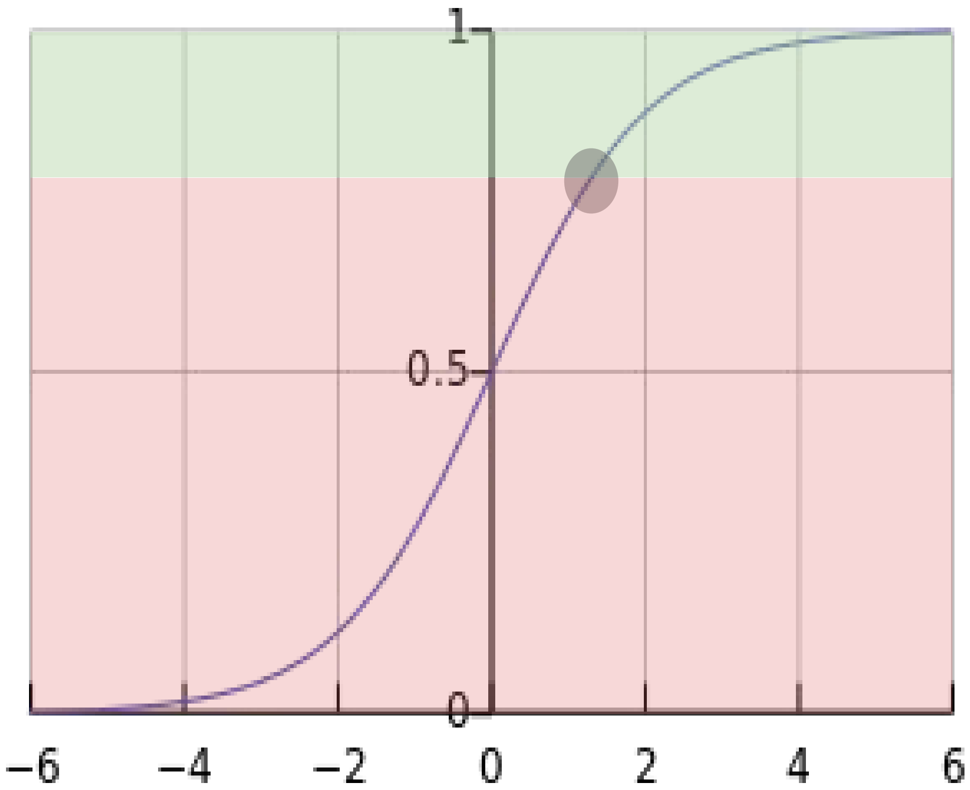
- Specificity?
- Sensitivity?
- Overall error rate?
- Overall accuracy?
- Precision?

	p' (Predicted)	n' (Predicted)
p (Actual)	146	32
n (Actual)	21	590

Threshold

Where between 0 and 1 do we draw the line?

- $P(x)$ below threshold: predict 0
- $P(x)$ above threshold: predict 1

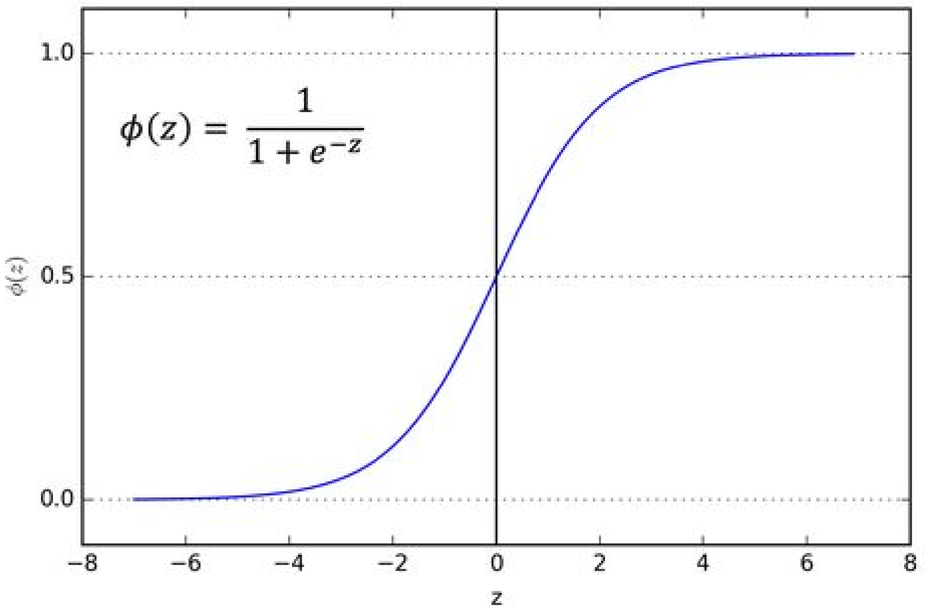


Source

Thresholds matter (a lot!)

What happens to the specificity when you have a

- Low threshold?
- High threshold?

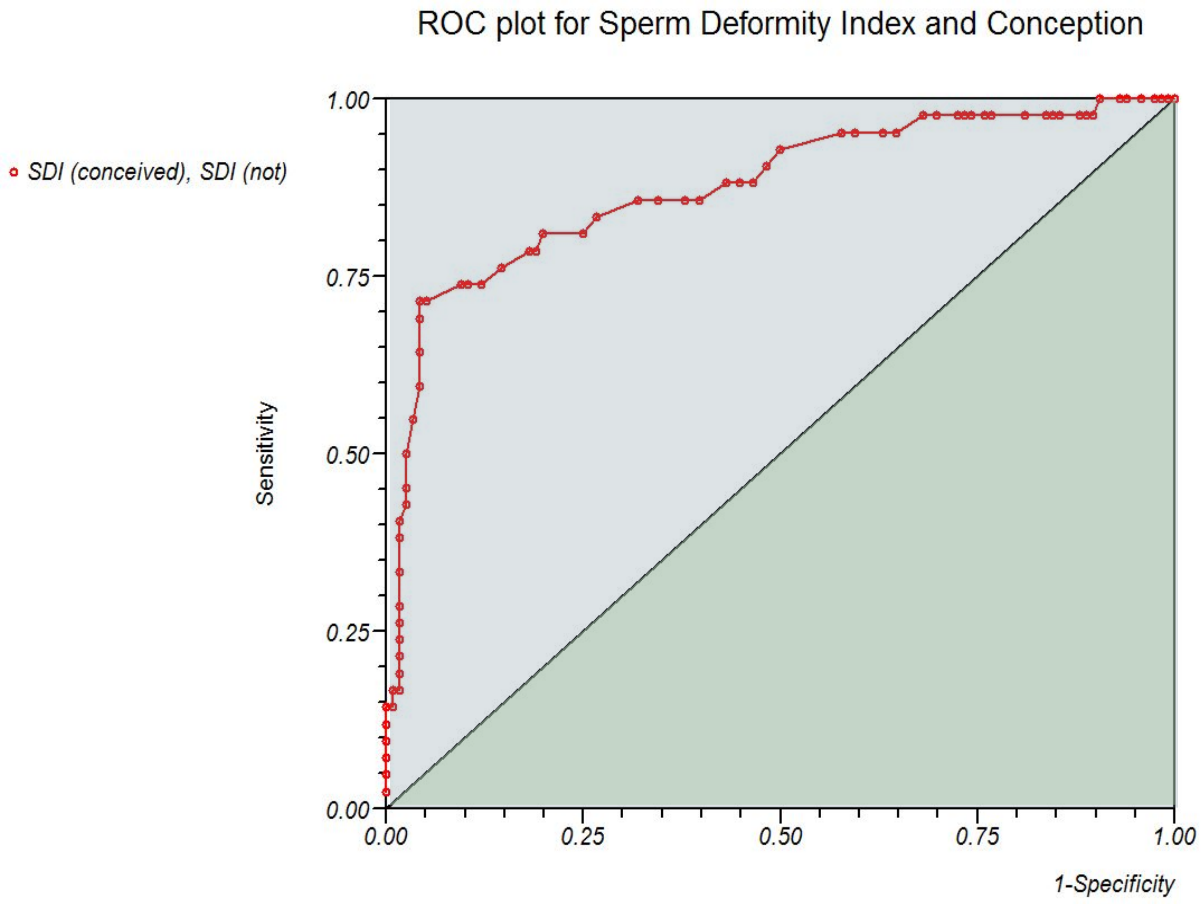


Source

ROC Curve

Receiver **O**perating **C**haracteristic

- Visualization of trade-off
- Each point corresponds to a specific threshold value



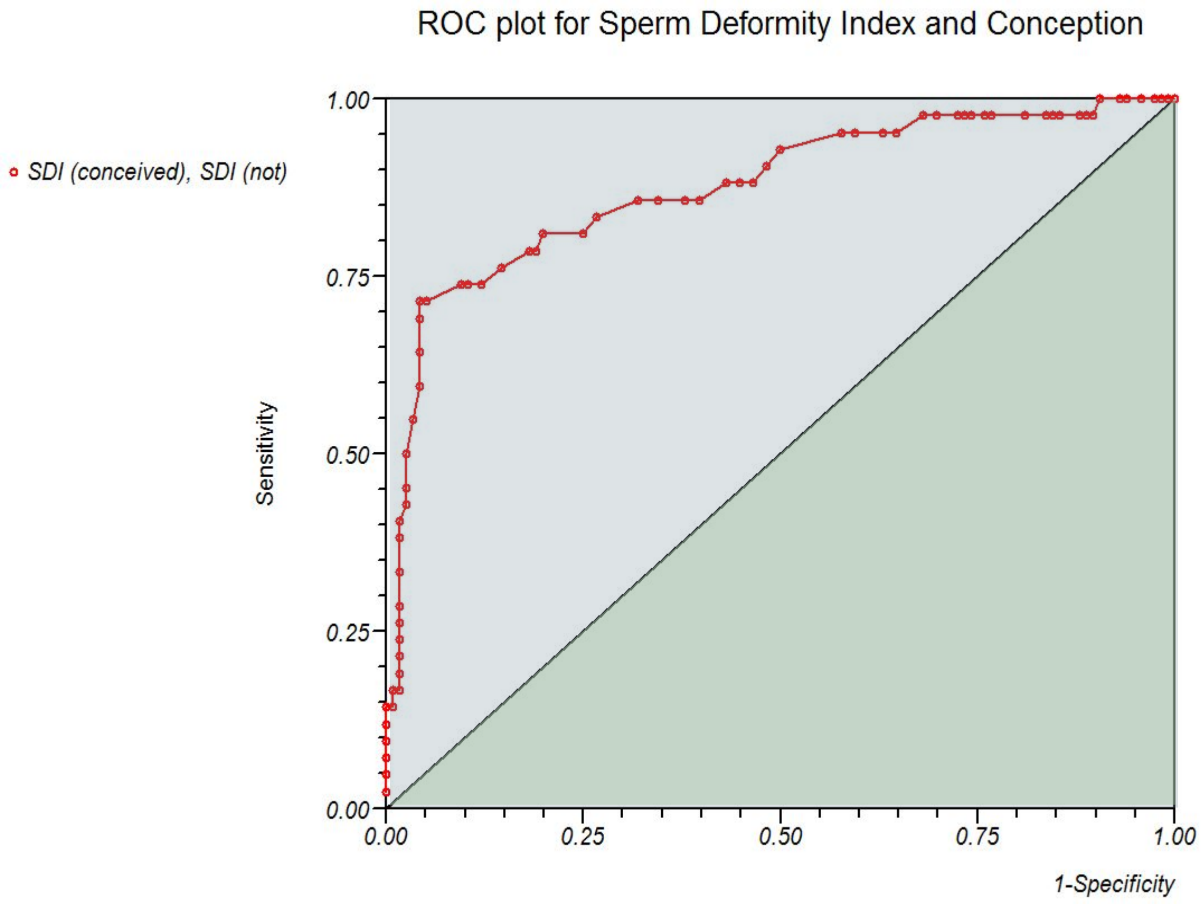
Area Under Curve

$$\mathrm{AUC} = \int ROC\text{-}curve$$

Always between 0.5 and 1.

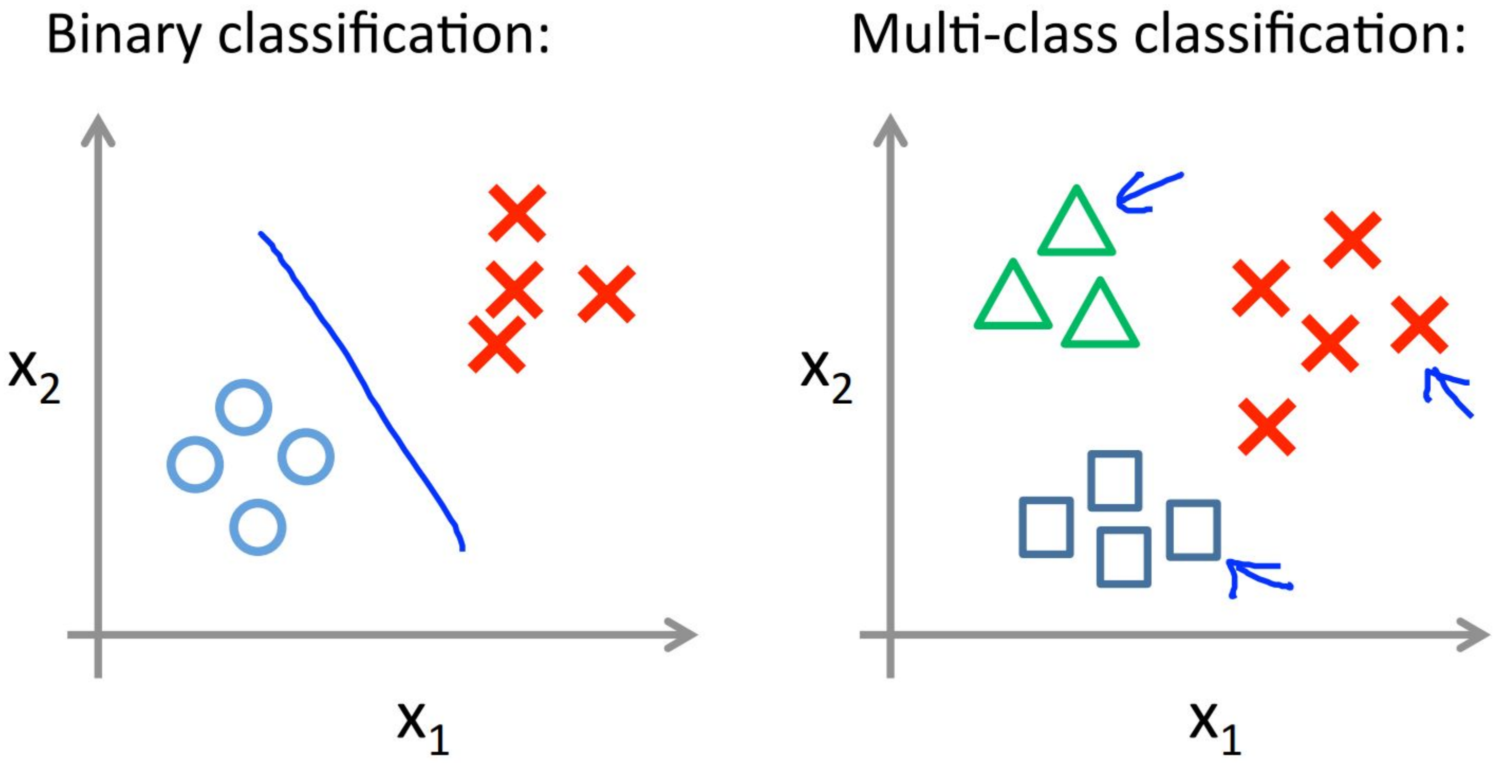
Interpretation:

- 0.5: Worst possible model
- 1: Perfect model



Multi-class Classification

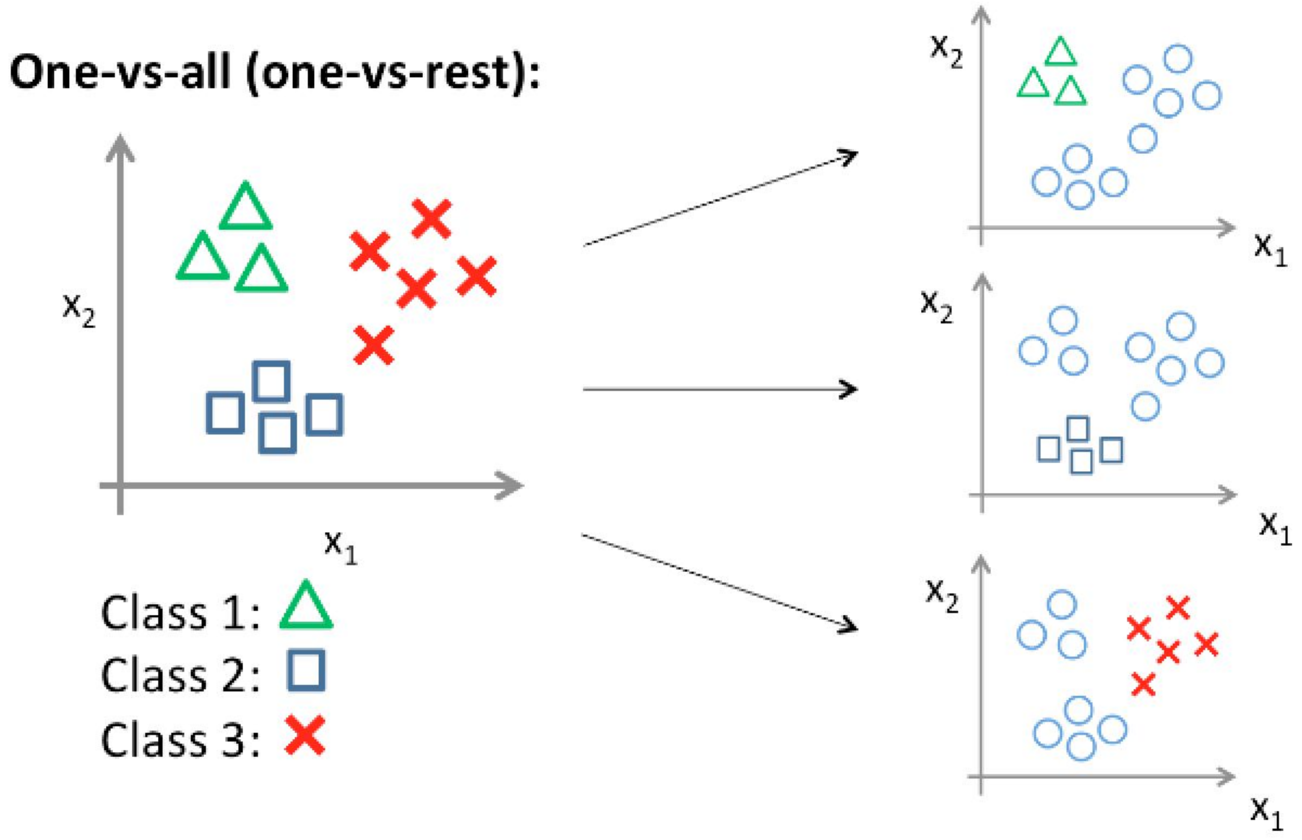
Classifying instances into three classes or more



Source

One-vs-All

- Train a single classifier per class
- All samples of that class classified as positive, all other samples as negative



Coming Up

Your problem set: Start working on Project Part B

Next week: More classifiers

See you then!