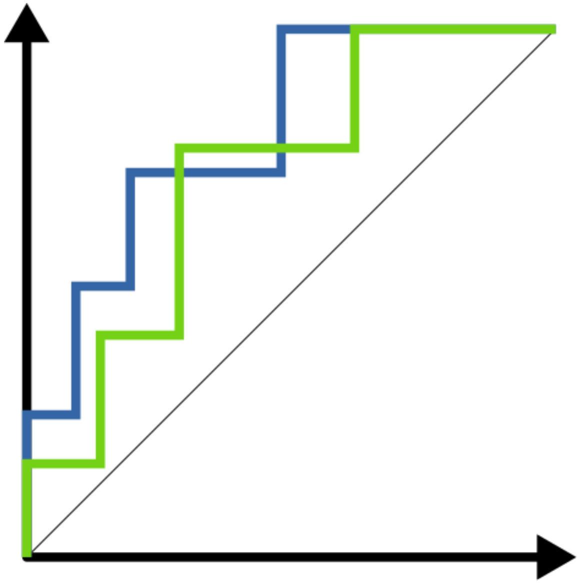
Lecture 19.3
Evaluating classification performance



Nicola Romanò
nicola.romano@ed.ac.uk

浙江大学爱丁堡大学联合学院
ZJU-UoE Institute

Learning objectives

At the end of this lecture you will be able to:

- Explain the concepts of
  - confusion matrices
  - ROC and AUC
  - Cross-validation
- Choose an appropriate classification threshold
- Compare different classifiers

R workshop #6 on classification will show you how to practically apply these concepts and perform these analyses in R (will be published on Learn on 27th March).

Classification algorithms

Problem

- We are interested in a discrete outcome (e.g. yes/no, mild/medium/severe)
- Collect data from several subjects (outcome + predictors)
- We want to **predict** the outcome for a new set of predictors

Many different classification models exist; classification trees, RandomForest, SVM, logistic regression, Naïve Bayes …

How do we choose one?

Need a measure of how good is the model prediction

In the next lecture we will talk about specific methods!

Predicting heart disease

303 patients with chest pain syndrome undergoing angiography at Cleveland Hospital
96 women, 207 men
Data collected between May 1981 and September 1984.

Outcome variable: heart disease → binary yes/no

Descriptors*:

- Age
- Gender
- Resting heart rate (bpm)
- Cholesterol level (mg/dl)
- Exercise-induced angina (yes/no)
- Chest pain type (typical anginal, atypical anginal, nonanginal, asymptomatic)

* the full "Cleveland dataset" has more descriptors

Given a new patient of a certain age/gender/etc. can we predict whether they are at risk of heart disease?

Fitting a model

For this example we can use logistic regression

```
model <- glm(Disease ~ ., data = heart, family = "binomial")

summary(model)

[…]

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  7.710271   1.608494   4.793 1.64e-06 ***
Age         -0.057784   0.018142  -3.185  0.00145 **
SexM        -1.844392   0.368588  -5.004 5.62e-07 ***
RestBP      -0.018822   0.008869  -2.122  0.03382 *
Cholesterol -0.003652   0.002974  -1.228  0.21942
ExAng       -1.423848   0.334959  -4.251 2.13e-05 ***
ChestPain    0.839769   0.155814   5.390 7.06e-08 ***
```

The dot means to use all the other columns

Using the model for prediction

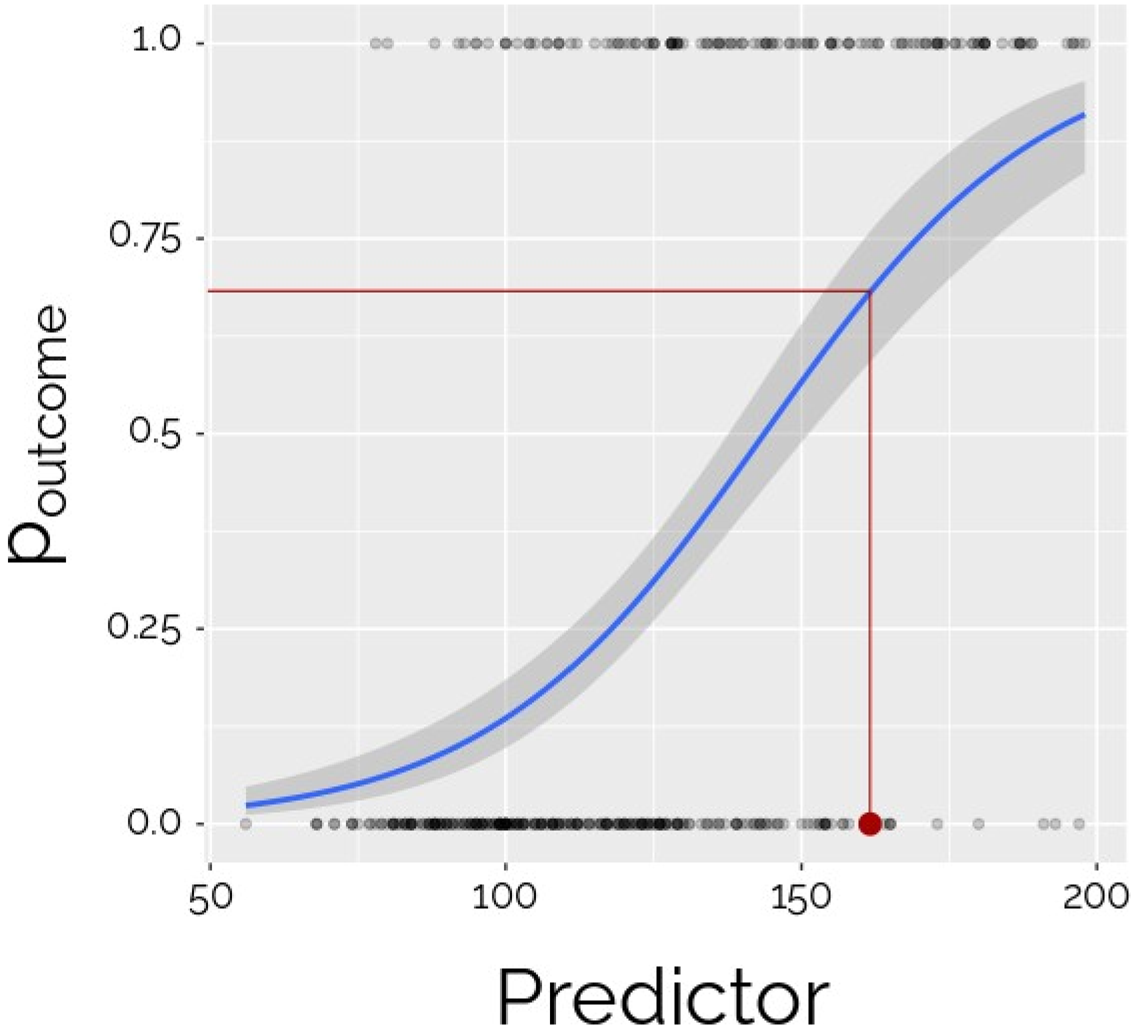
The *predict* function can be used to predict new outcomes from a model

```
new.patient <- data.frame(Age = 45,
                          Sex = "M",
                          RestBP = 125,
                          Cholesterol = 350,
                          ExAng = 0,
                          ChestPain = 2)

predict(model, new.patient, type = "response")

        1
0.7882252
```

There is ~79% probability that the patient will have heart disease.

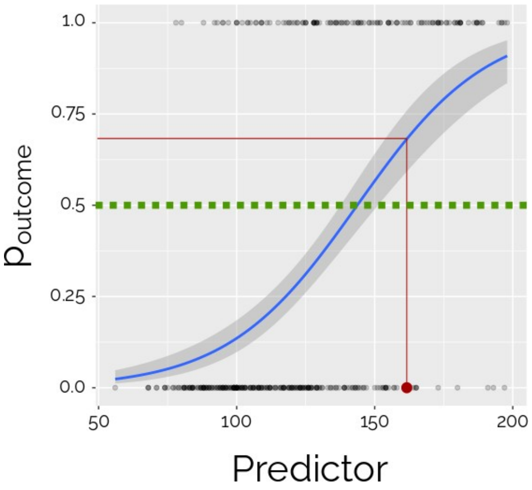
See Workshop 4 for another example!

Generating binary outcomes

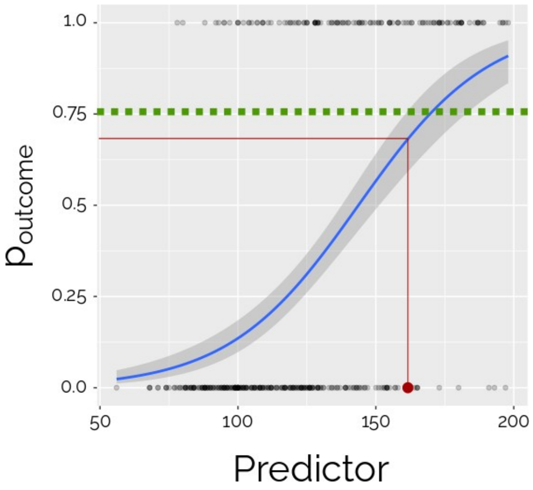


Do we classify this as positive or negative?

Setting a threshold



Positive



Negative

```
ifelse(predict(model, new.patient, type = "response") > threshold, 1, 0)
```

Condititon

If true If false

In what instances would you use a threshold different from 0.5?

NOTE: although most classification methods return a probability, some only return a label (positive/negative). The concept of threshold does not apply to those methods.

OK but… how good is our model?

We are still unsure…

What model do I choose?

What parameters and threshold do I choose?

How do I know if the model prediction was right?

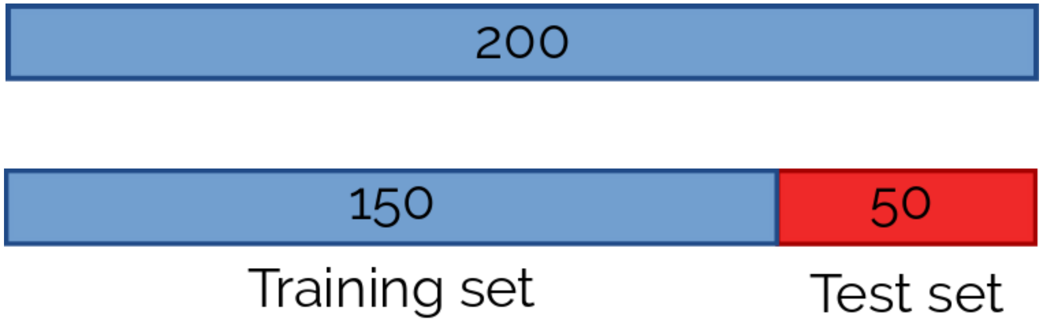
Test and training sets

1) Randomly divide the data for which you have labels into a **training set** and a **test set**

2) The training set is used to generate (in ML this is called training) the model

3) The test set is used to test how good the predictions of the model are!

Test and training sets



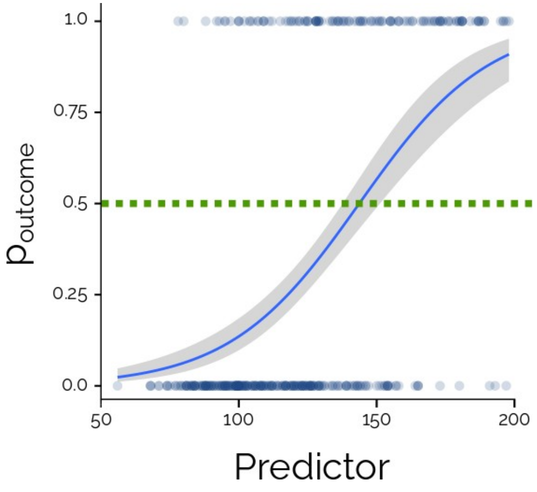
200

150

50

Training set

Test set



Run the model on the training set

Apply the prediction to the test set

Prediction 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0

Check how good the model did!
(we know the labels!)

Apply the model to a new unlabelled dataset!

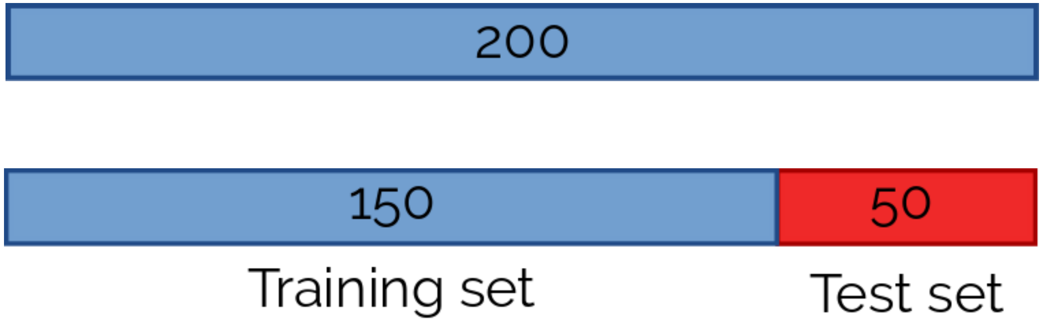
The confusion matrix

A confusion matrix tells us how good our model is at classifying

		Actual	
		Disease	No disease
Predicted	Disease	24 (true positives)	1 (false positive)
	No disease	5 (false negative)	20 (true negatives)

Sensitivity = TP / (TP + FN) = 24 / (24 + 5) = 0.83 = **83%** (% true positives)
Specificity = TN / (FP + TN) = 20 / (20 + 1) = 0.95 = **95%** (% true negative)

Test and training sets

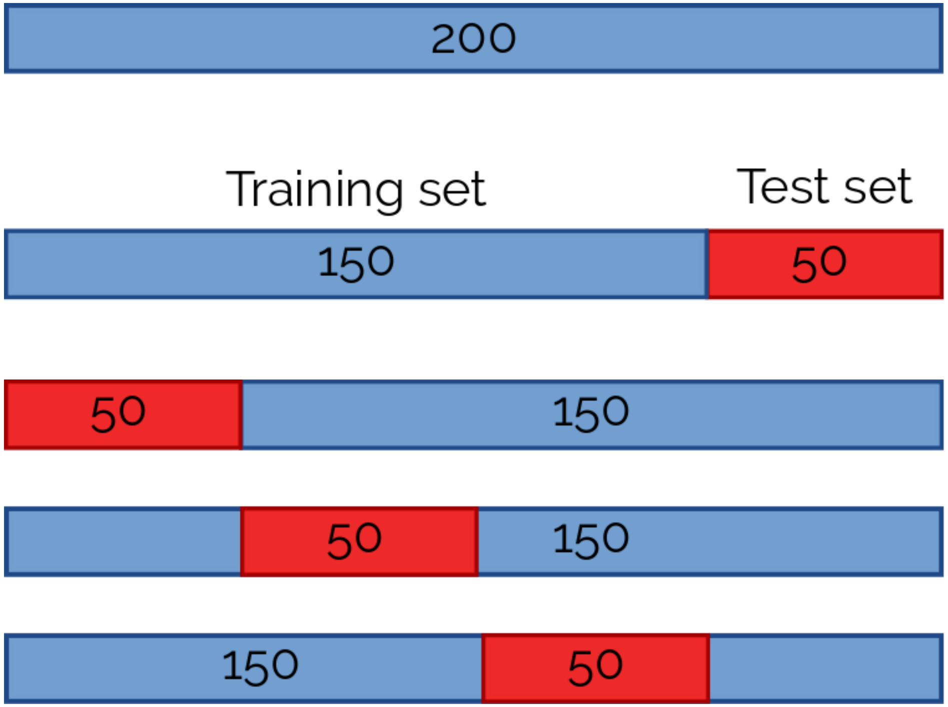


Problem: why choose the last 50 observation (or the first 50, for that matter)?

We risk **overfitting** the model to the training set!

This means that the model is very good at classifying the training set, but may perform poorly on other data. This is also called **variance.**

Cross-validation



Generate model on training set, apply to test

Generate model on training set, apply to test

Generate model on training set, apply to test

Generate model on training set, apply to test

Generate confusion matrix!

We call this 4-fold cross-validation.
If we divided the data into 10 we would call it 10-fold cross-validation, etc.

CV of heart disease dataset

We call perform CV in R using the caret package (code will be in the workshop)

10-fold CV on the heart disease dataset

```
Cross-Validated (10 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)

          Reference
Prediction    0    1
         0 32.3  8.9
         1 13.2 45.5

 Accuracy (average) : 0.7789
```

CV of heart disease dataset

So… now we have a way of telling how good our model is!

We can:

- Compare different models
- Compare the same model with different parameters (e.g. different logistic regressions classifying using different thresholds)

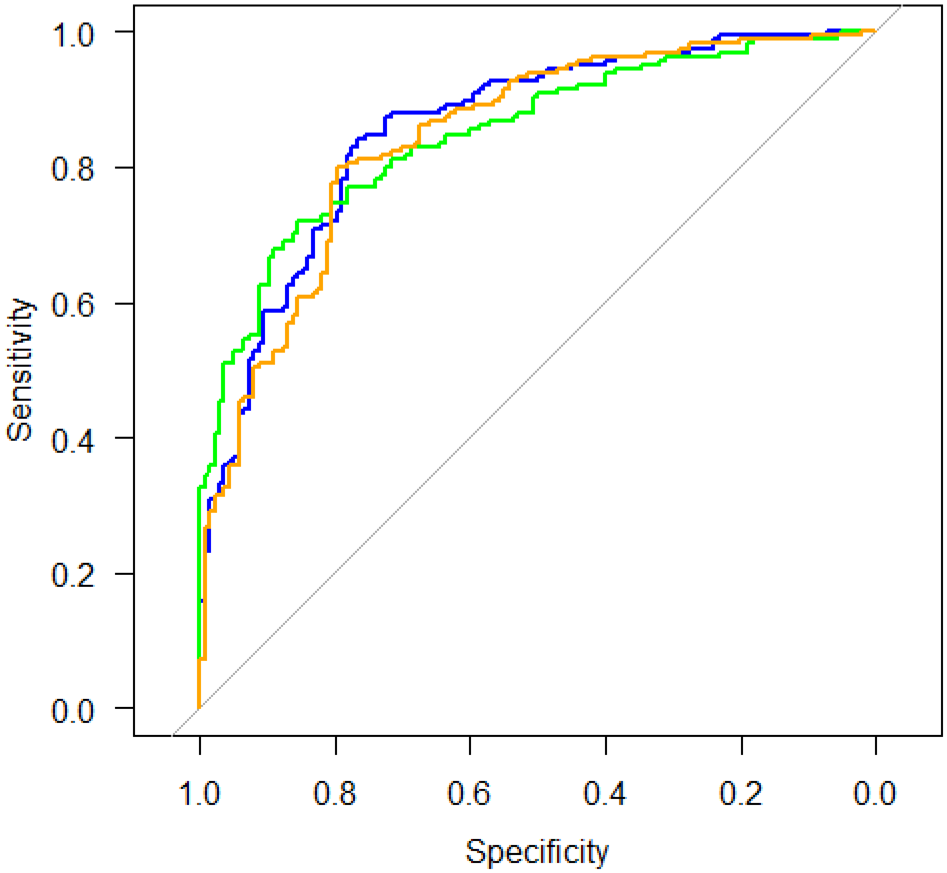
ROC curves

A **Receiver Operating Characteristic** (ROC) curve shows the classification ability of a binary classifier at different thresholds.

We plot Specificity and Sensitivity while varying the threshold

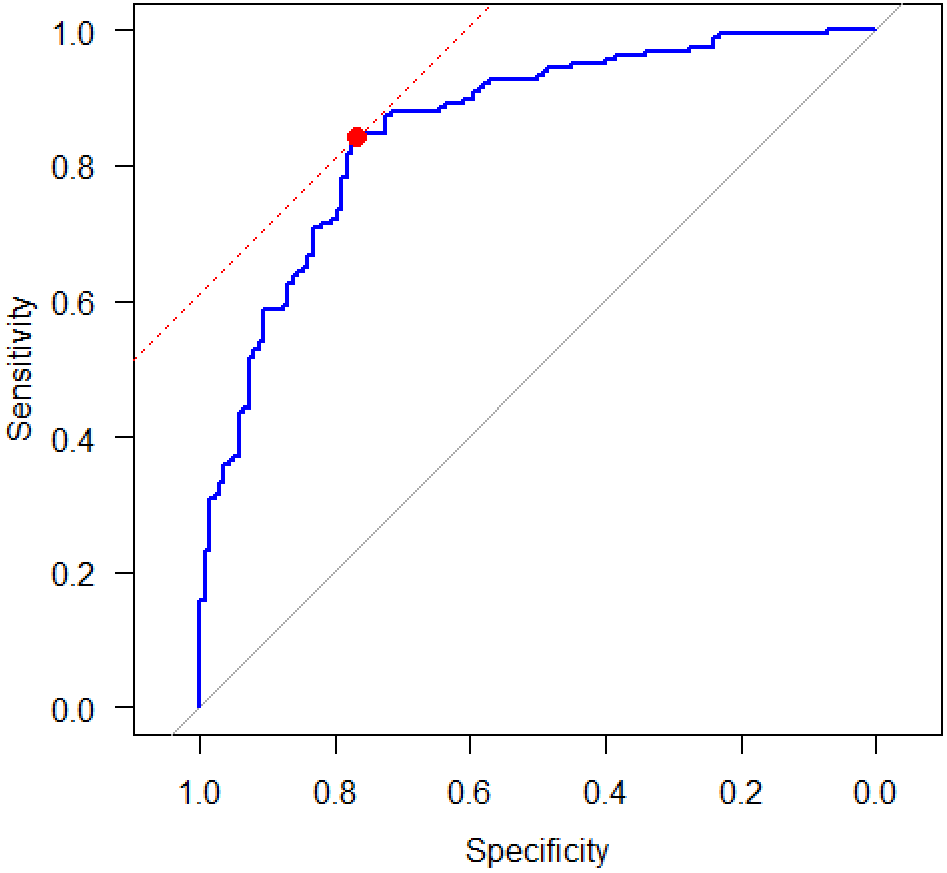
The area under the curve (AUC) is a good measure of how well the classifier predicts the outcome

Comparing methods



ROC curve for the heart disease dataset using **logistic regression** – AUC = 0.856
ROC curve for the same dataset using **randomForest** – AUC = 0.848 or
Naïve Bayes – AUC = 0.842

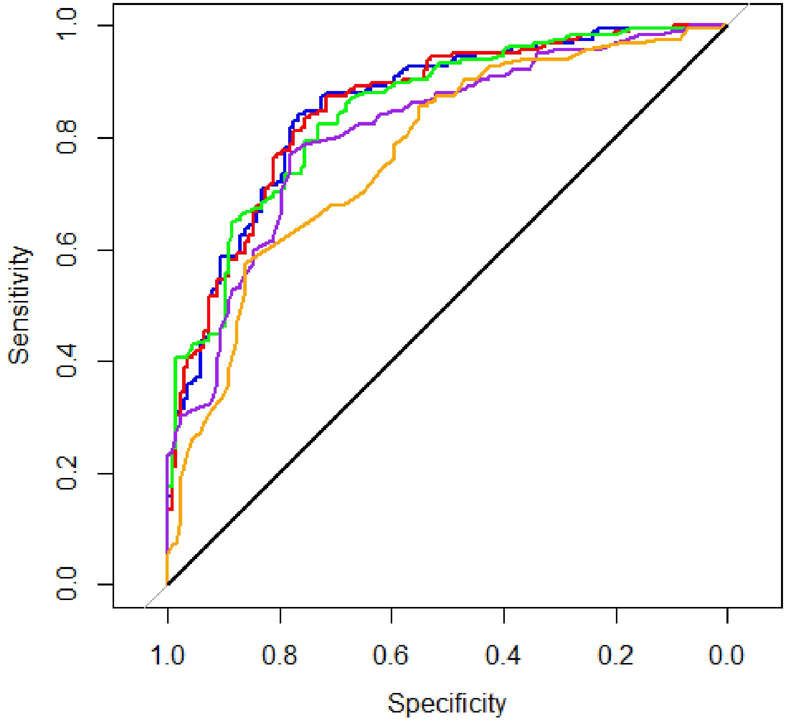
Choosing threshold



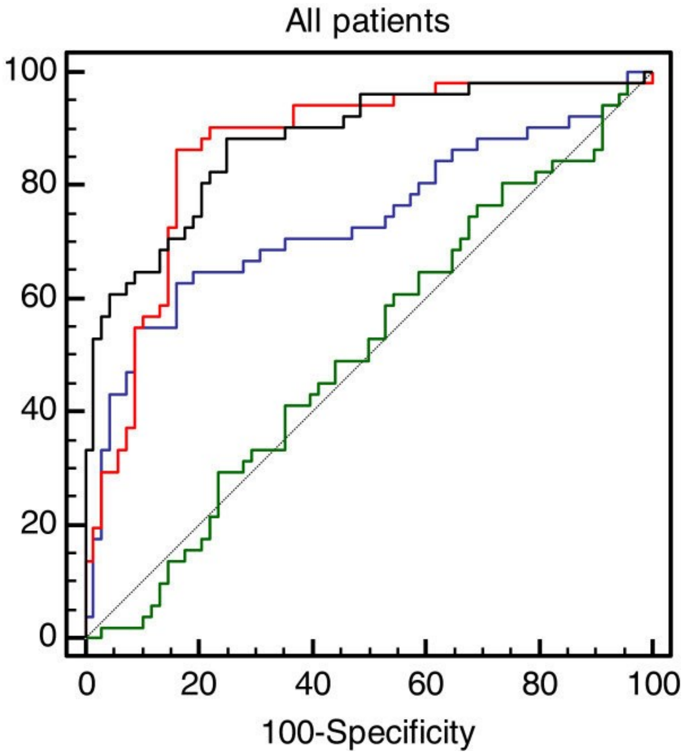
In most situations, the best threshold is the one giving highest specificity and sensitivity

But… sometimes you may want to compromise on one of the two!

Choosing predictors



Logistic regression on the heart disease dataset
Blue curve is ROC for full model, black curve for intercept-only model
Other curves have different numbers of predictors



ROC curves for models using different predictors.
From Yates et al. 2013

Summary

- Classification models allow prediction of discrete outcomes
- Cross-validation allows to avoid overfitting
- Confusion matrices and ROC curves allow to compare models

- Next week… some practical examples of classification models!