Logistic Regression and Classification

Introduction to Statistical Modelling

Prof. Joris Vankerschaver

Overview

1. Introduction: what are classification problems?
2. K-nearest neighbors classification
3. Logistic regression
4. Classification

Introduction

Classification

In many problems, the outcome is a **categorical** variable:

- Figure out whether mutation is deleterious (yes/no), based on DNA sequencing data.
- Predict a person's eye color (blue/brown/green)
- Predict the outcome of surgery (success/failure) for patients with ovarian cancer, based on patient characteristics
- Classify iris (flower) variety given dimensions of leaves

These problems are examples of **classification** problems.

Techniques for classification

- **Logistic regression**
- **K-nearest neighbors**
- Linear discriminant analysis
- Support vector classification (SVC)
- Decision trees
- ...

The techniques in **bold** are discussed in this lecture.

Each technique has its advantages and disadvantages.

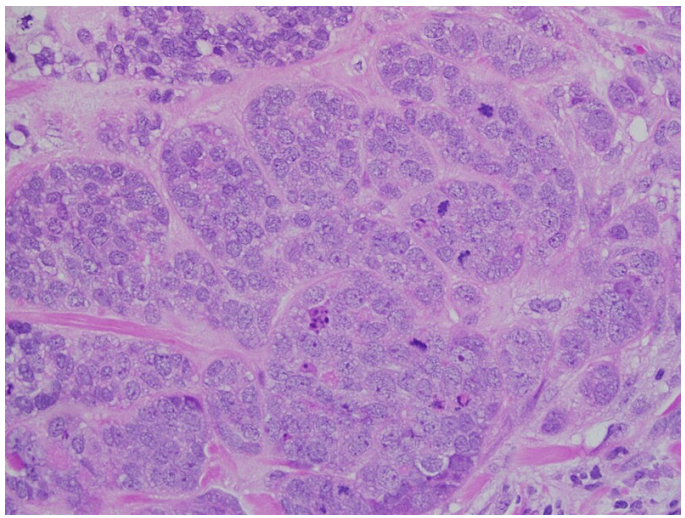
References

- *An Introduction to Statistical Learning*. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. Available for free online at https://www.statlearning.com/.
  - Logistic regression: sections 4.1 - 4.3

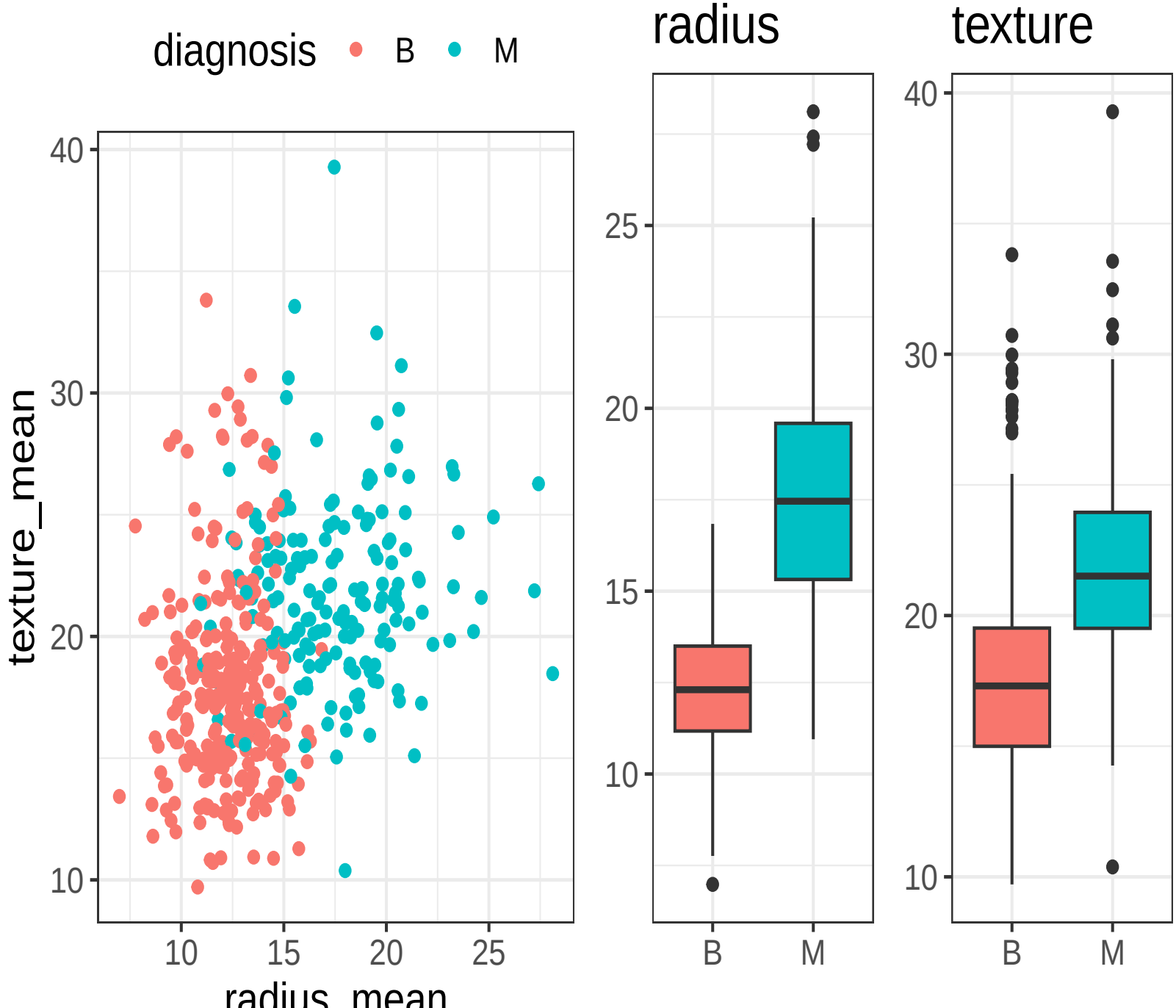
Dataset

`bdiag` – Wisconsin breast cancer diagnostic dataset (*Nuclear feature extraction for breast tumor diagnosis.* W. Street, W. Wolberg, O. Mangasarian. Electronic imaging 29 (1993))

- Cell nuclei from 569 tumor samples
- Classified as malignant or benign
- Features:
  - radius of the cell nucleus
  - texture (variance of gray-scale values)



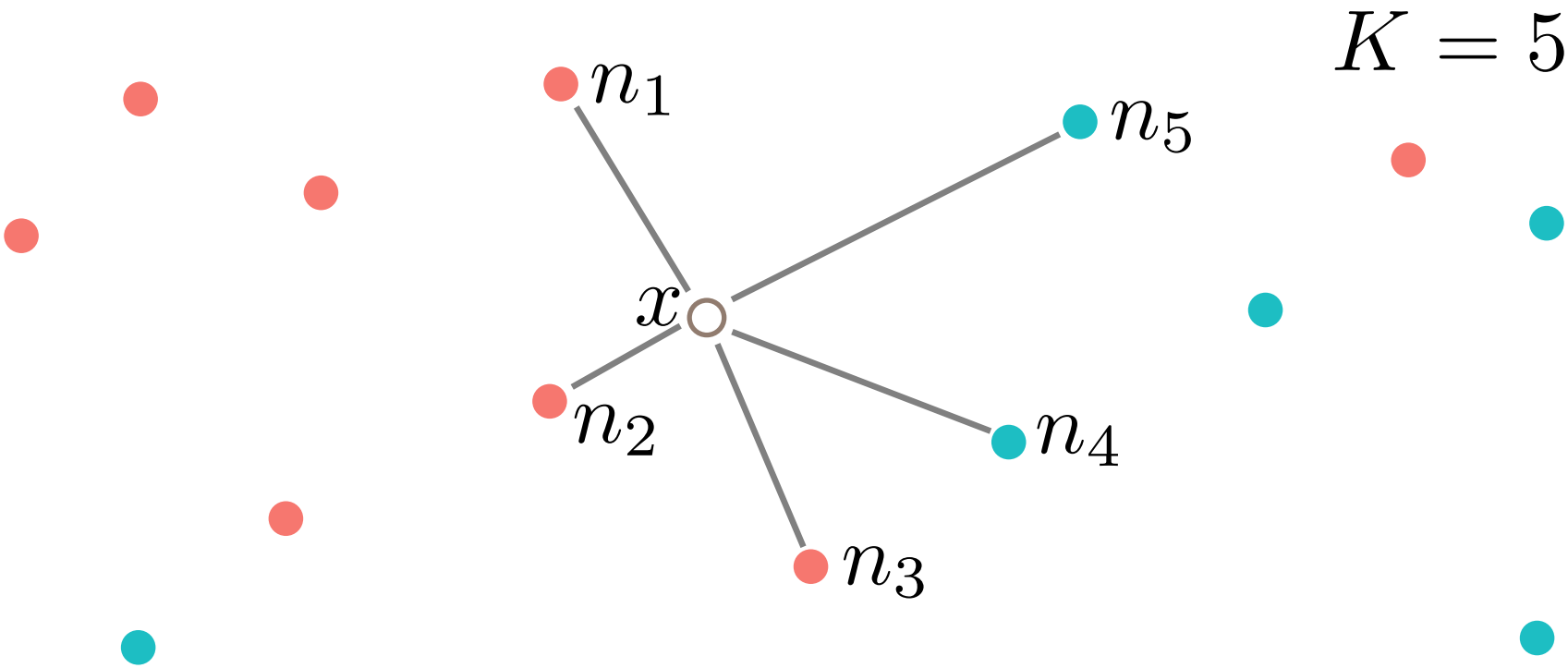
A first look at the data



K-nearest neighbors classification

Principle

- Find K nearest neighbors to x
- Probability of belonging to class i is proportional to number of neighbors in that class



- $P(Y = B|X = x) = \frac{3}{5} = 60\%$
- $P(Y = M|X = x) = \frac{2}{5} = 40\%$

Properties

K-nearest neighbor (KNN) classification estimates probabilities

$$P(Y = j|X = x) = \frac{1}{K} \sum_{i=1}^{K} I(y_i = j)$$

Here:

- The sum is over the K nearest datapoints $y_1, \ldots, y_K$ to x
- $I(y_i = j)$ is equal to 1 if $y_i = j$ and to 0 otherwise

Advantages and disadvantages

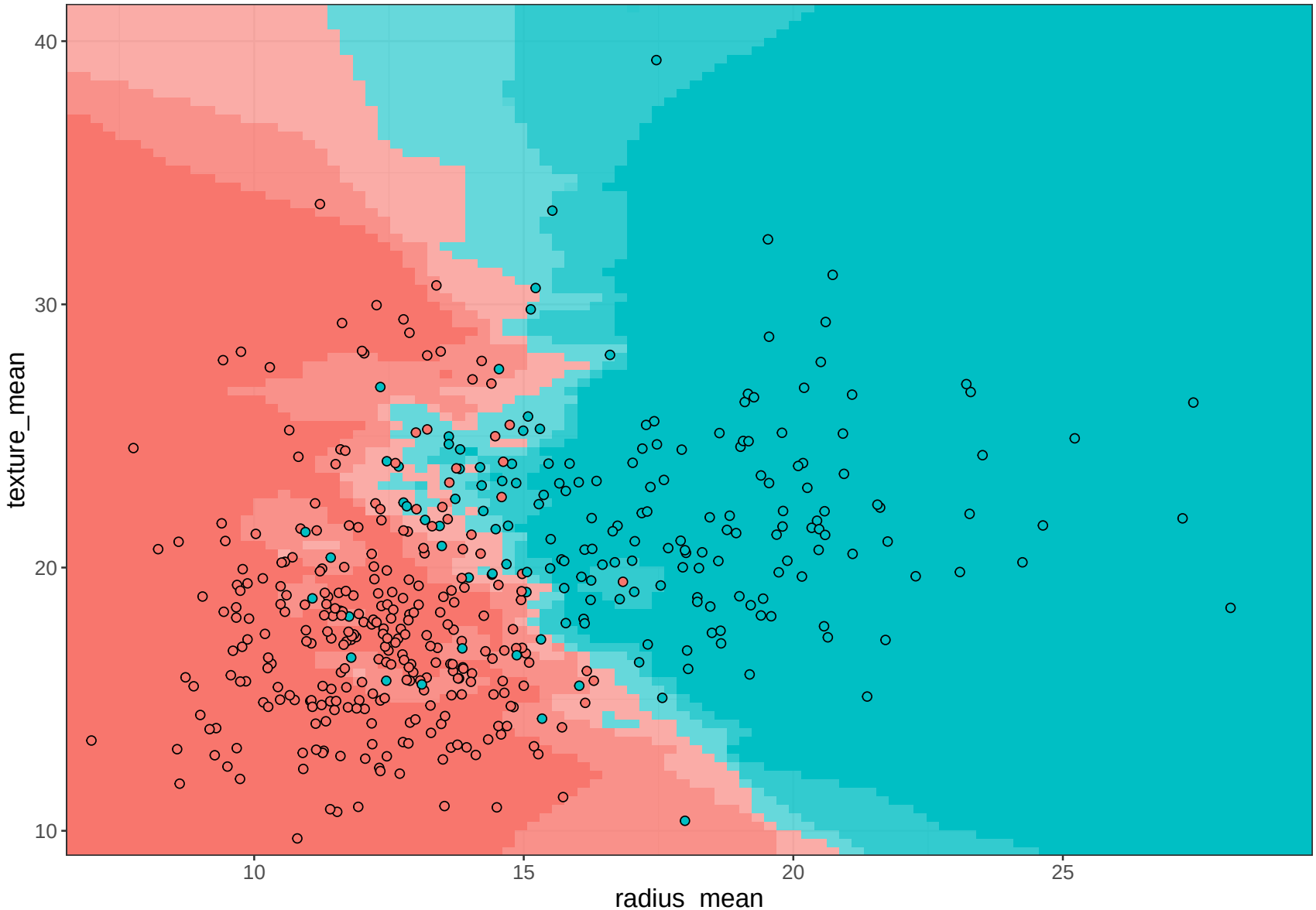
Advantages:

- No "training" necessary
- Robust to outliers
- Can easily deal with more than 2 labels

Disadvantages:

- Not very interpretable – why was class decided?
- Memory-intensive

Decision boundary ($K = 5$)



Logistic regression

Reminder: odds

- If π is the probability of having a malignant tumor, then the **odds** are defined as

$$\text{Odds} = \frac{\pi}{1-\pi}.$$

  For example: if $\pi = 0.8$ then Odds $= 4$, meaning that for every benign tumor there are 4 malignant ones (on average).

- Odds range from 0 (impossible event) to $+\infty$ (almost certain).

Reminder: odds ratio

- **Odds ratio** (OR): indicates by how much the odds change between two treatments. For example: suppose in the treatment group the probability of a malignant tumor drops to $\pi_T = 0.75$ (compared to $\pi_C = 0.8$ in the untreated group). Then

$$\text{OR} = \frac{\text{Odds}(T)}{\text{Odds}(C)} = \frac{3}{4} = 0.75$$

- If OR < 1, then the odds for treatment 1 decrease compared to treatment 2. If OR > 1, the odds increase.

Log-odds (logits)

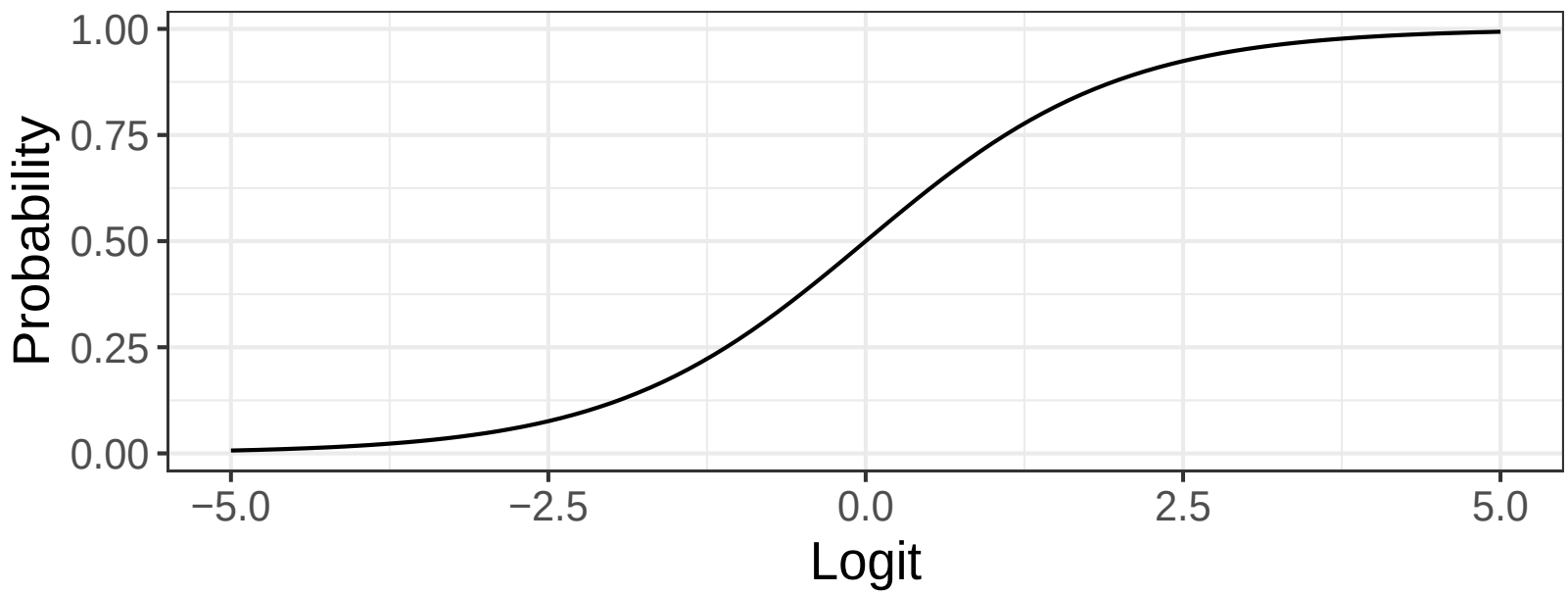
Often it makes sense to work with the logarithm of the odds (**logits**):

$$\text{logit}(\pi) = \ln \text{Odds} = \ln\left(\frac{\pi}{1-\pi}\right).$$

To convert back to probabilities, use the **logistic** function:

$$\pi = \frac{1}{1 + e^{-\text{logit}}}.$$

Logits are unbounded: logit $\to \pm\infty$ for $p \to 0, 1$



Regression for classification

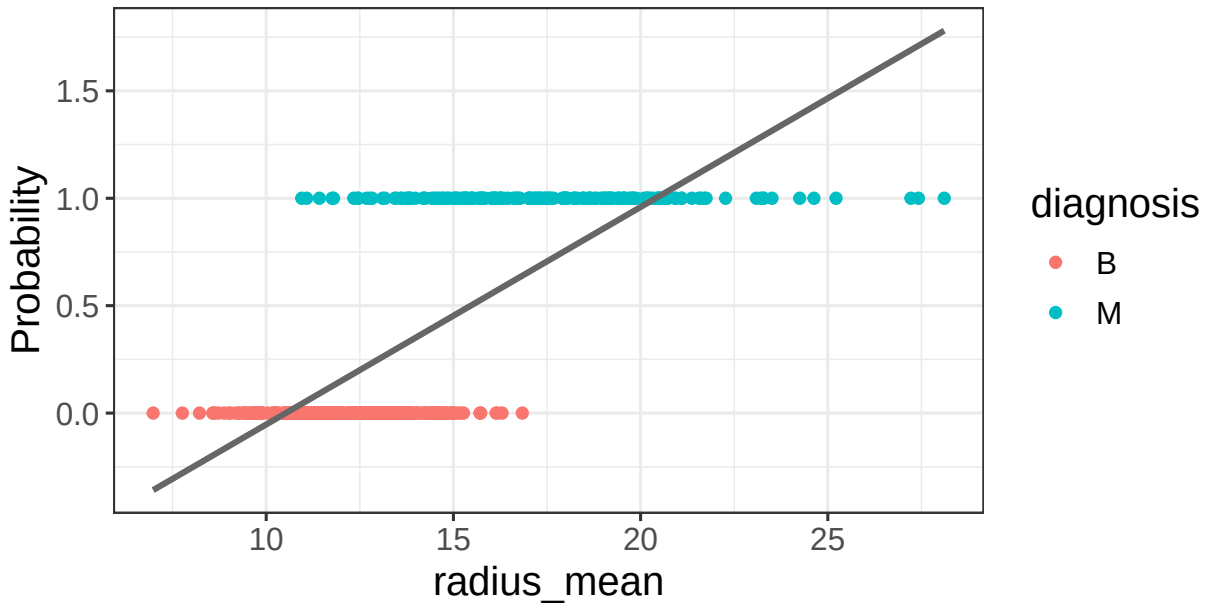
- Given data $(X_1, Y_1), \ldots, (X_n, Y_n)$ where:
  - Outcomes Y_i are categorical (0 or 1)
  - Predictors X_i can be continuous or discrete
- We will model Y_i as a Bernoulli random variable (0 or 1) with probability $\pi(X_i)$:

$$\begin{aligned} Y_i &= 0 \quad \text{with probability } \pi(X_i) \\ Y_i &= 1 \quad \text{with probability } 1 - \pi(X_i) \end{aligned}$$

- Now we need to determine how $\pi(X)$ depends on X.

Idea 1: linear regression (bad)

- One predictor $X = \texttt{radius\_mean}$, outcome $Y = 0$ (benign) or $Y = 1$ (malignant).
- Assume $\pi(X) = \alpha + \beta X$ and determine α, β through linear regression.



Problems:

- Fitted probabilities can take on values outside $[0, 1]$.
- Does not easily generalize to more than two classes.

Idea 2: logistic regression (better)

- Let $\pi(X)$ depend on X through the logistic function

$$\pi(X) = \frac{1}{1 + \exp(-(\alpha + \beta X))}.$$

- **Nonlinear** model in parameters α, β
- Alternatively, apply the logit transformation

$$\text{logit}(\pi) = \alpha + \beta X.$$

- Linear in the logits.

Determining the regression parameters: MLE

- **Likelihood function** $\mathcal{L}$: probability of observing the data given the parameters α, β:

$$\mathcal{L}(\alpha, \beta) = \prod_{i=1}^{n} P(Y = Y_i | X = X_i),$$

where

$$P(Y = Y_i | X = X_i) = \pi(X_i)^{Y_i} (1 - \pi(X_i))^{1-Y_i}.$$

is the probability of observing one data point (X_i, Y_i).

- In practice, often better to use the log of the likelihood:

$$\ell(\alpha, \beta) = \ln \mathcal{L}(\alpha, \beta).$$

Determining the regression parameters: MLE

- **Maximum likelihood estimation** (MLE): find parameters that maximize $\mathcal{L}(\alpha, \beta)$ or $\ell(\alpha, \beta)$
- Finding maximum: set partial derivatives (score functions) equal to zero:

$$\frac{\partial \ell}{\partial \alpha} = 0, \quad \frac{\partial \ell}{\partial \beta} = 0.$$

- Complicated equations, usually maximum cannot be found analytically (unlike least squares)
- Use numerical methods to find maximum (R does this automatically with the `glm` command)

Simplified example: MLE for binomial variable

- Suppose there are *no* predictors. We just have a bunch of categorical outcomes $Y_i = 0, 1$, e.g.

$$Y = (0, 0, 1, 0, 1, 0, \ldots, 1, 1, 0, 0, 1)$$

- In semester 1 we saw that a good estimate for the probability $\pi = P(Y = 1)$ is given by the proportion of 1s in the data:

$$\hat{\pi} = \frac{1}{N} \sum_{i=1}^{N} Y_i = \bar{Y}.$$

- We'll use MLE to re-derive this result.

Simplified example: MLE for binomial variable

- Likelihood

$$\begin{aligned}\mathcal{L}(\pi) &= \Pi_{i=1}^{n} P(Y = Y_i) \\ &= \pi^{n\bar{Y}}(1-\pi)^{n(1-\bar{Y})}\end{aligned}$$

- Log likelihood: $\ell(\pi) = n\bar{Y}\ln\pi + n(1-\bar{Y})\ln(1-\pi)$.
- Maximum occurs when first derivative vanishes:

$$\frac{d\ell}{d\pi} = \frac{n\bar{Y}}{\pi} - \frac{n(1-\bar{Y})}{1-\pi} = 0.$$

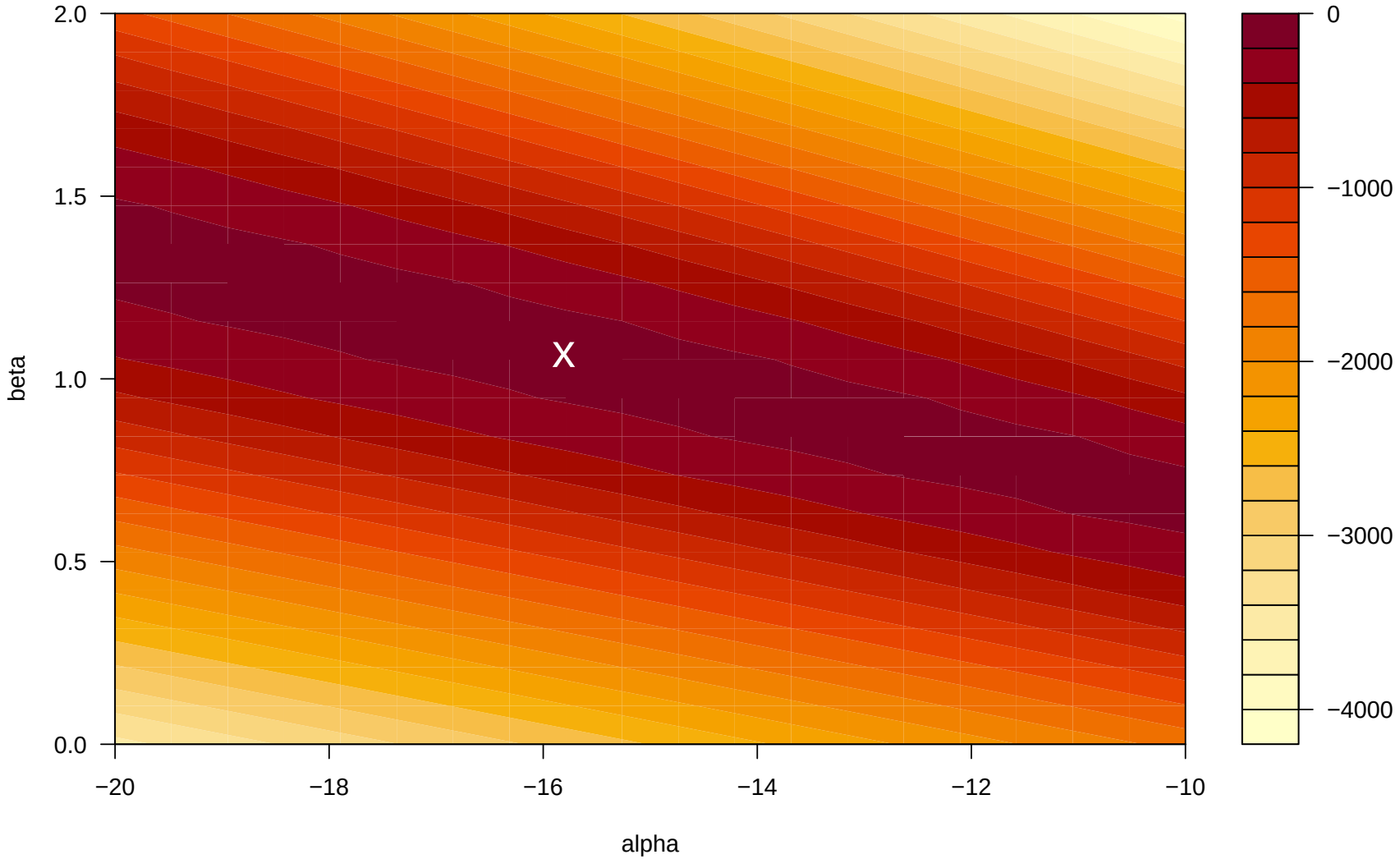
- Simplifies to $\hat{\pi} = \bar{Y}$.

MLE for logistic regression in R

```r
m_simple <- glm(diagnosis ~ radius_mean, data = train, family = "binomial")
summary(m_simple)
```

```
Call:
glm(formula = diagnosis ~ radius_mean, family = "binomial", data = train)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -15.8086     1.5310 -10.326   <2e-16 ***
radius_mean   1.0662     0.1066   9.998   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 604.40  on 454  degrees of freedom
Residual deviance: 256.54  on 453  degrees of freedom
AIC: 260.54

Number of Fisher Scoring iterations: 6
```

The log likelihood



- Value of log likelihood at MLE: $\ell = -128.2701$.
- R reports (residual) deviance: $D = -2 \times \ell = 256.54$

Multiple logistic regression

- Like in linear regression, often the outcome Y is influenced by several predictors $X_1, X_2, \ldots, X_p$.
- For example: `diagnosis` depends on `radius_mean` and `texture_mean`:

$$\text{logit}(\pi) = \alpha + \beta_1 \cdot \texttt{radius\_mean} + \beta_2 \cdot \texttt{texture\_mean}.$$

- Parameters $\alpha, \beta_1, \ldots, \beta_p$ determined through MLE.

In R

```r
m_multi <- glm(diagnosis ~ radius_mean + texture_mean,
               data = train, family = "binomial")
summary(m_multi)
```

```
Call:
glm(formula = diagnosis ~ radius_mean + texture_mean, family = "binomial",
    data = train)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -20.51694    2.04729 -10.021  < 2e-16 ***
radius_mean    1.09536    0.11727   9.341  < 2e-16 ***
texture_mean   0.21749    0.04034   5.391 7.01e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 604.40  on 454  degrees of freedom
Residual deviance: 223.68  on 452  degrees of freedom
AIC: 229.68

Number of Fisher Scoring iterations: 7
```

Interactions between variables

```
m_inter <- glm(diagnosis ~ radius_mean * texture_mean,
               data = train, family = "binomial")
```

Coefficient	Esti-mate	SE	z value	p value
(Intercept)	-8.3046	7.4554	-1.114	0.2653
radius	0.2182	0.5288	0.413	0.6798
texture	-0.4133	0.3855	-1.072	0.2836
radius:texture	0.0455	0.0276	1.647	0.0995

Interaction between radius and texture is not significant

Making predictions (by hand)

What is the probability of a tumor being malignant if the radius is 13 mm?

$$\begin{aligned}\pi(\texttt{radius\_mean} = 13) &= \frac{1}{1 + \exp(15.8086 - 1.0662 \times 13)} \\ &= 0.1247716\end{aligned}$$

No easy formula for confidence interval on the prediction.

Making predictions (using R)

```
predict(m_simple,
        newdata = data.frame(radius_mean = 13),
        type = "response")
```

```
        1
0.1247961
```

Computing a confidence interval for the prediction

Proceeds in three steps:

1. Make a prediction on the **logit** scale (`type = "link"`)
2. Compute CI on logit scale from SE (`se.fit = TRUE`)
3. Map CI back to probabilities

For step 3, use `plogis` to undo the logit transformation:

$$\text{plogis}(x) = \frac{1}{1 + \exp(-x)}.$$

Computing an CI: example

Step 1: Prediction on the logit scale.

```
pred <- predict(m_simple,
                newdata = data.frame(radius_mean = 13),
                type = "link", se.fit = TRUE)
```

Step 2: CI on the logit scale.

```
ci_logits <- c(pred$fit - 1.96 * pred$se.fit,
               pred$fit + 1.96 * pred$se.fit)
ci_logits
```

```
        1         1
-2.358216 -1.537335
```

Step 3: CI on the original scale (probabilities)

```
ci_probs <- c(plogis(ci_logits[1]), plogis(ci_logits[2]))
ci_probs
```

```
         1          1
0.08641491 0.17692307
```

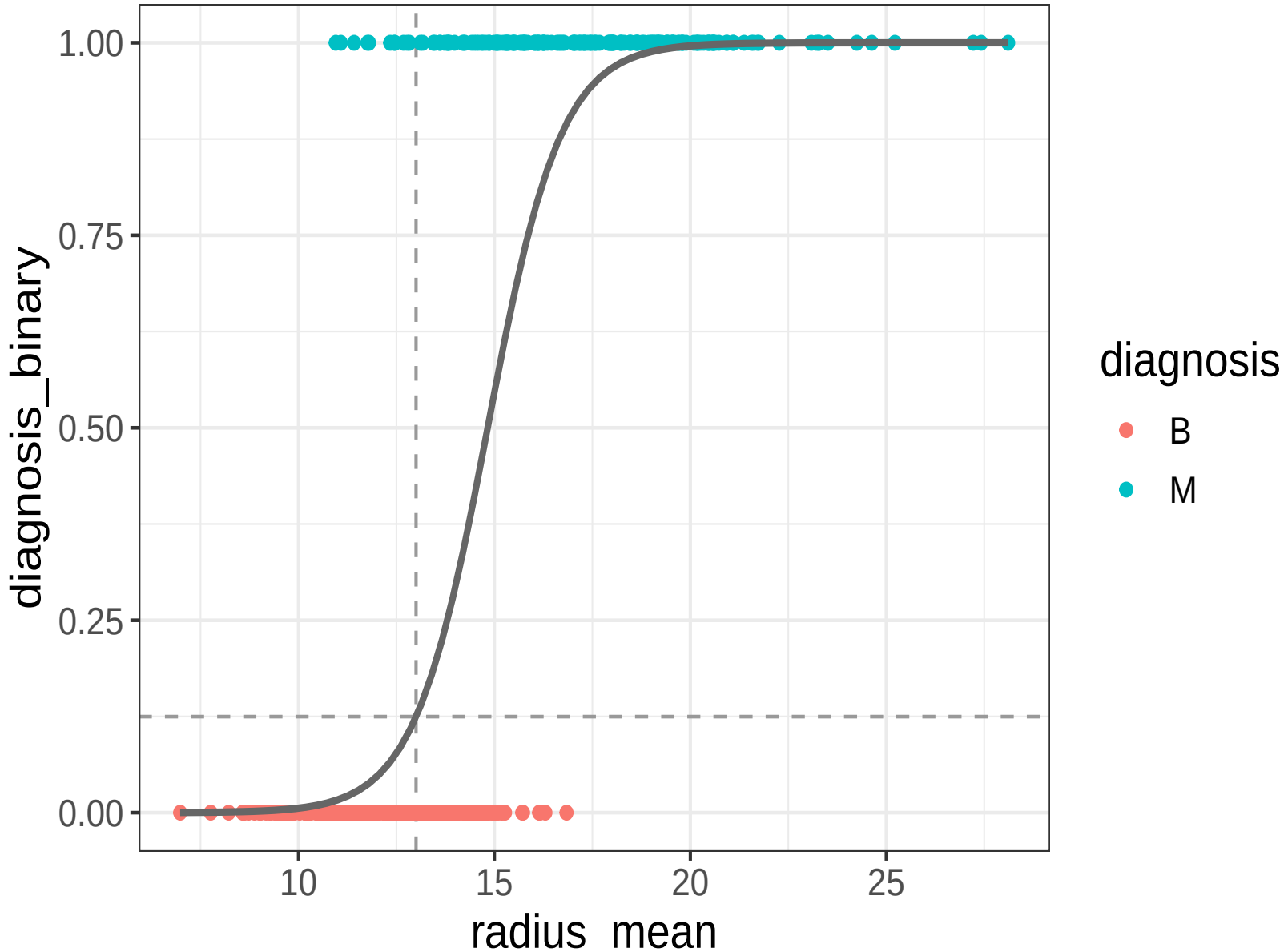
Original prediction:

```
pred_probs <- plogis(pred$fit)
pred_probs
```

```
        1
0.1247961
```

Conclusion: The predicted probability that a tumor of radius 13mm is malignant is 12.5% (95% CI: [8.6%, 17.7%])

Making predictions



Quantifying the strength of an association

Write the logistic regression model in terms of odds as

$$\text{logit}(\pi) = \ln \text{Odds} = \alpha + \beta X.$$

After some algebra:

$$e^{\beta} = \frac{\text{Odds}(X+1)}{\text{Odds(X)}}.$$

In other words: e^{β} is the odds ratio (OR) associated to a 1-unit increase in X.

Breast cancer dataset

Here $\beta = 1.0662$, so OR $= \exp(1.0662) = 2.90$. An increase in 1 mm in tumor radius is associated with odds that are 2.90 times higher (risk increase).

Testing an association

- Often, we want to test whether a model coefficient β is significant.
- Related: check if complex and simple nested models are equivalent (recall F-test from linear regression).

Several ways of testing:

- Wald test (reported in `summary`): can be conservative
- Likelihood ratio test (via `anova` command): more power, preferred
- Score test (not covered)

Testing an association: Wald test

- Null hypothesis $H_0 : \beta = 0$, alternative hypothesis $H_A : \beta \neq 0$
- Test statistic follows $N(0, 1)$ under H_0

$$z = \frac{\hat{\beta}}{SE(\beta)} \sim N(0, 1) \quad \text{under } H_0.$$

- Reported in the R regression output (`summary`):

Coefficient	Estimate	SE	z value	p value
(Intercept)	-20.5169	2.0473	-10.021	$<$ 2e-16
radius_mean	1.0954	0.1173	9.341	$<$ 2e-16
texture_mean	0.2175	0.0403	5.391	7.01e-08

Testing an association: Likelihood ratio test

Useful for:

- Comparing nested models (simple/complex)
- Testing single coefficient

Hypothesis:

- H_0: simple and complex model are equivalent
- H_A: complex model is better

Test statistic: **deviance**

$$\begin{aligned} D &= -2 \ln \frac{\mathcal{L}(\text{simple})}{\mathcal{L}(\text{complex})} \\ &= -2\ell(\text{simple}) + 2\ell(\text{complex}). \end{aligned}$$

Under H_0, D follows a χ^2_k distribution, where k is the number of extra parameters in the complex model.

Worked out example

Nested models:

- Simple: includes `radius_mean` only
- Complex: includes both `radius_mean` and `texture_mean`.

From R summary (listed as residual deviance) or direct calculation:

- $-2\ell(\text{simple}) = 256.54$
- $-2\ell(\text{complex}) = 223.68$

Hence $D = 256.54 - 223.68 = 32.86 > 3.841459 = \chi^2_{1;0.95}$.

Conclusion: reject H_0, significant evidence to decide (at 5% significance level) that complex model is better.

Likelihood ratio test in R (single variable)

```
anova(m_simple, m_multi)
```

```
Analysis of Deviance Table

Model 1: diagnosis ~ radius_mean
Model 2: diagnosis ~ radius_mean + texture_mean
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       453     256.54
2       452     223.68  1   32.864 9.882e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Compare with critical values for χ^2_1 to draw conclusion

Likelihood ratio test in R (groups of variables)

Nested models:

- Simple: includes `radius_mean` and `texture_mean`.
- Complex: adds `concavity_mean` and `symmetry_mean`.

R output:

```
anova(m_multi, m_multi_4)
```

```
Analysis of Deviance Table

Model 1: diagnosis ~ radius_mean + texture_mean
Model 2: diagnosis ~ radius_mean + texture_mean + concavity_mean + symmetry_mea
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       452     223.68
2       450     129.99  2   93.686 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compare with critical value $\chi^2_{2;0.95} = 5.991465$ to conclude that complex model is better.

Confidence interval for regression parameters

Wald-type **approximate** $(1-\alpha)\times 100\%$ confidence interval for β:

$$\hat{\beta} \pm z_{1-\alpha/2} \cdot SE(\beta)$$

Breast cancer dataset

95% confidence interval for $\beta_{\texttt{radius\_mean}}$:

$$1.095 \pm 1.96 \times 0.117 = [0.866, 1.324].$$

Confidence interval for regression parameters in R

```
confint(m_multi)
```

```
                    2.5 %      97.5 %
(Intercept)  -24.8846042 -16.8233485
radius_mean    0.8840034   1.3456187
texture_mean   0.1405734   0.2993915
```

R uses the so-called profile method to compute CI:

- Different from Wald method (narrower CIs, but close)
- Preferred to use this method through R

Confidence interval for odds ratio

- Recall that $\exp(\beta) =$ OR for a 1-unit change in X
- $(1-\alpha) \times 100\%$ confidence interval for the OR:

$$\exp\left(\hat{\beta} \pm z_{1-\alpha/2} \cdot SE(\beta)\right).$$

Breast cancer dataset

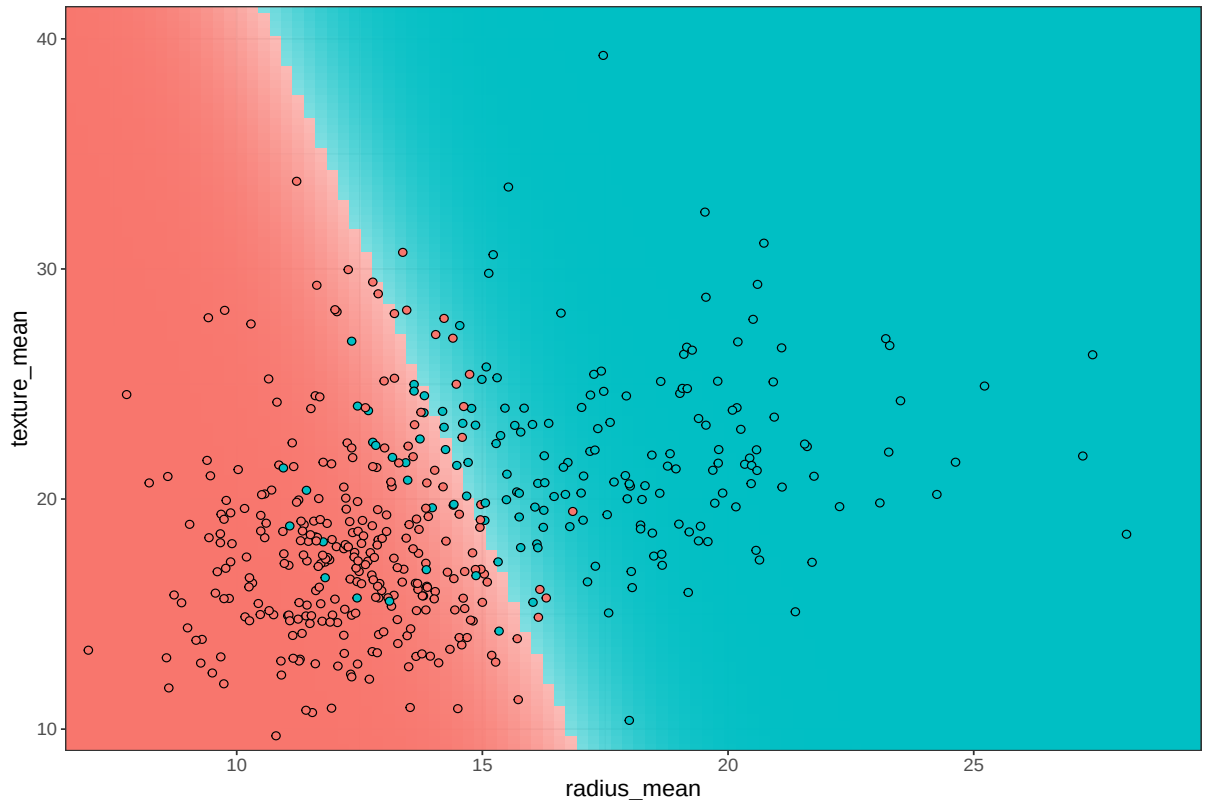
95% confidence interval for $\text{OR}_{\texttt{radius\_mean}}$:

$$\begin{aligned}\exp(1.095 \pm 1.96 \times 0.117) &= [\exp(0.866), \exp(1.324)] \\ &= [2.377, 3.759]\end{aligned}$$

Classification

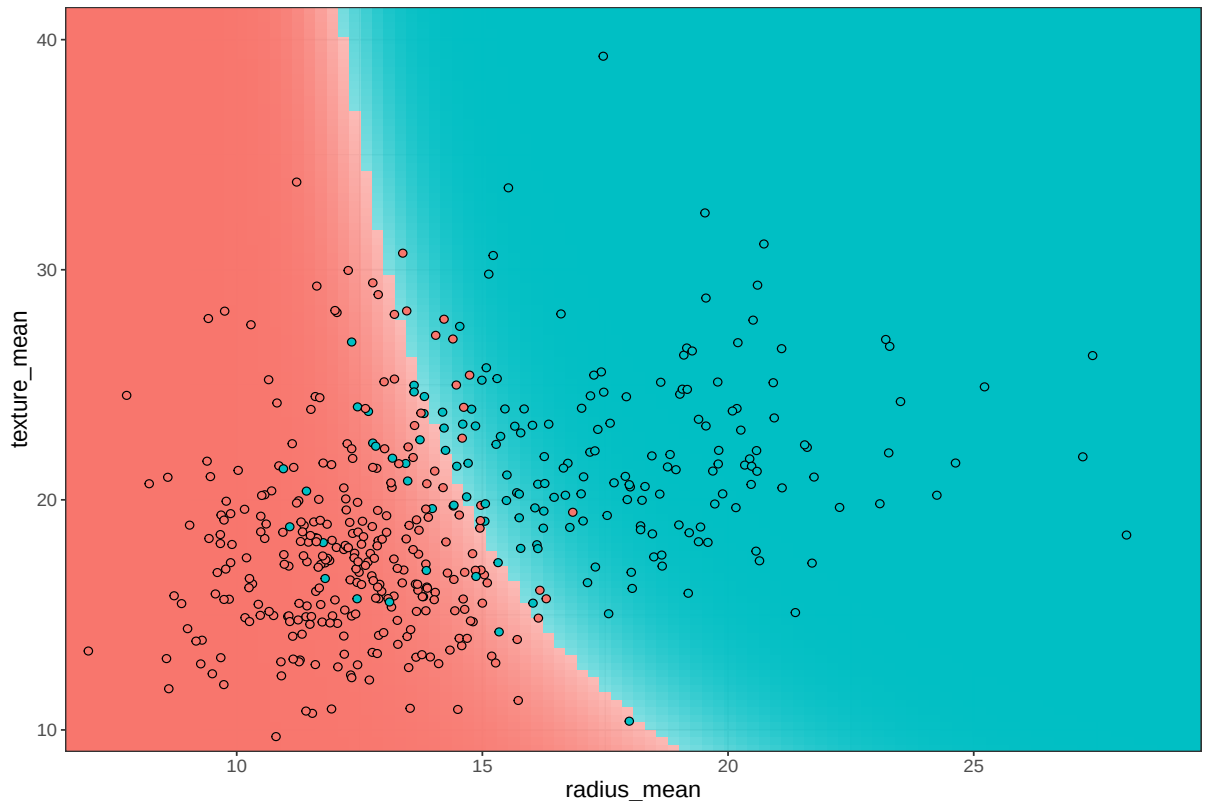
Decision boundary (no interaction terms)

- Model: $\text{logit}(\texttt{diagnosis}) \sim \texttt{radius} + \texttt{texture}$
- Decision boundary is **straight** line



Decision boundary (with interaction terms)

- Model:
  $\text{logit}(\texttt{diagnosis}) \sim \texttt{radius} + \texttt{texture} + \texttt{radius} : \texttt{texture}$
- Decision boundary is **curved** line



Classification

- Once we have a (logistic) model for $\pi(X)$, we can use it to classify new data X as negative $(Y = 0)$ or positive $(Y = 1)$, by comparing $\pi(X)$ with a fixed threshold C:

$$Y = 1 \quad \text{if } \pi(X) > C, \text{ otherwise } Y = 0.$$

- Performance **depends on choice of** C

i Breast cancer dataset

We computed earlier that $\pi(\texttt{radius\_mean} = 13) = 0.12$. Assuming that the threshold for malignant samples is $C = 0.5$, this sample would be classified as **benign**.

Confusion matrix

By comparing labels given by our model with "actual" labels, we can get an idea of the performance of our classifier.

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Figure source: https://en.wikipedia.org/wiki/Confusion_matrix

Performance metrics

Name	Definition	Value for example
Accuracy	(TP + TN)/(P + N)	0.84
Sensitivity (recall)	TP / P	0.93
Specificity	TN / N	0.67
PPV (precision)	TP / PP	0.84
NPV	TN / PN	0.84

- Many other metrics exist
- Which one is important depends on the problem
- Metrics can give surprising results in case of unbalanced data

In R (via caret package)

```
Confusion Matrix and Statistics

          Reference
Prediction  B  M
         B 64  6
         M 11 33

               Accuracy : 0.8509
                 95% CI : (0.772, 0.9107)
    No Information Rate : 0.6579
    P-Value [Acc > NIR] : 3.039e-06

                  Kappa : 0.6786

 Mcnemar's Test P-Value : 0.332

            Sensitivity : 0.8462
            Specificity : 0.8533
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.9143
             Prevalence : 0.3421
         Detection Rate : 0.2895
   Detection Prevalence : 0.3860
      Balanced Accuracy : 0.8497

       'Positive' Class : M
```

Trading sensitivity and specificity

What is important?

- Diagnostic test: **sensitivity** (don't tell people with tumor that they are healthy). Choose low threshold.
- Classifying email as spam: **specificity** (don't put regular email in the spam folder). Choose high threshold.

By changing the threshold, sensitivity and specificity can be traded against one another.

- Lowering threshold: Sensitivity $\uparrow$, Specificity $\downarrow$.
- Increasing threshold: Sensitivity $\downarrow$, Specificity $\uparrow$.

Breast cancer dataset

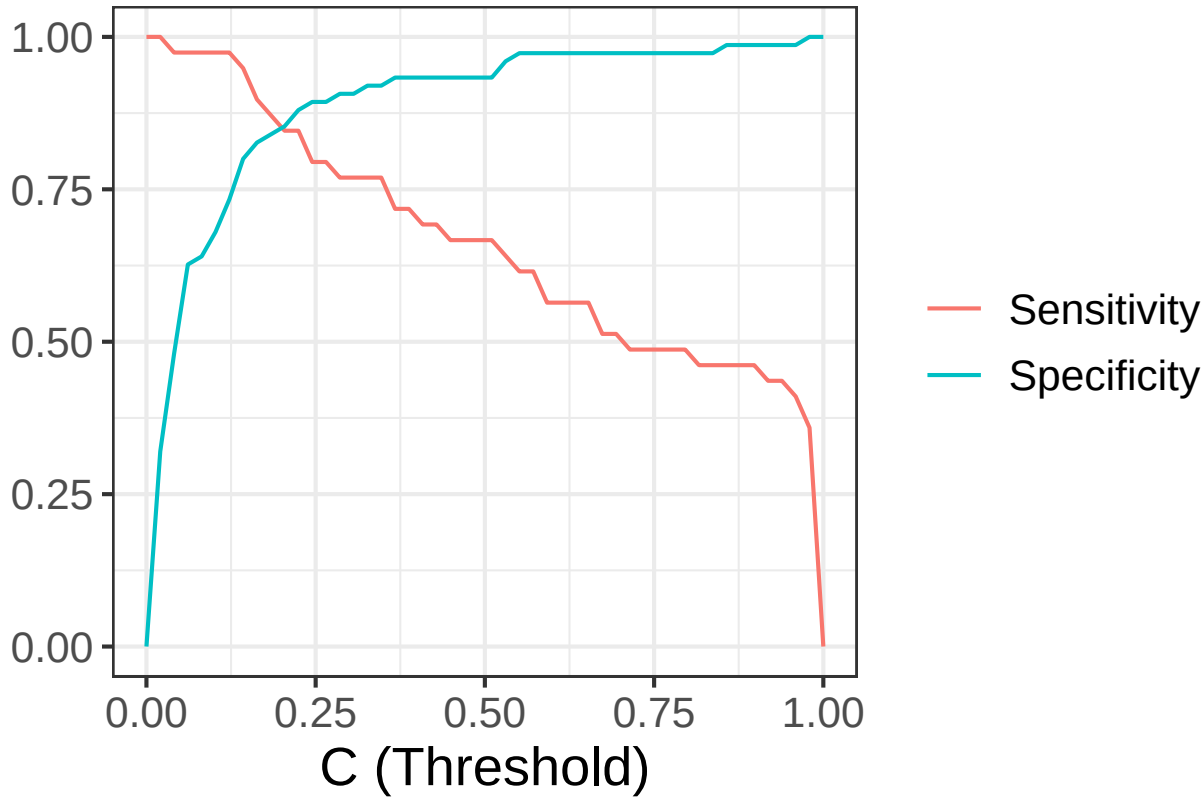
- For $C = 0.5$: sensitivity 0.84

Prediction	Reference	B	M
B		70	**13**
M		5	26

- For $C = 0.2$: sensitivity **0.85**

Prediction	Reference	B	M
B		64	**6**
M		11	33

Sensitivity and specificity as a function of threshold

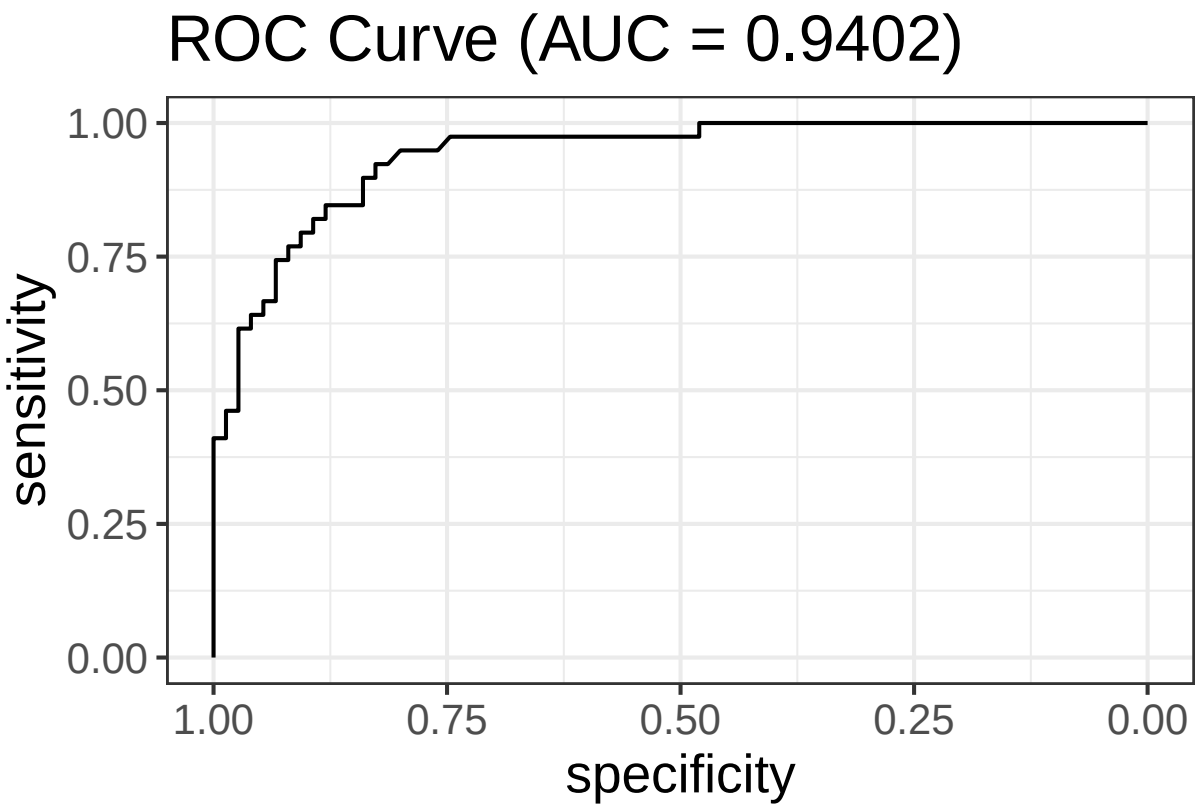


As threshold increases:

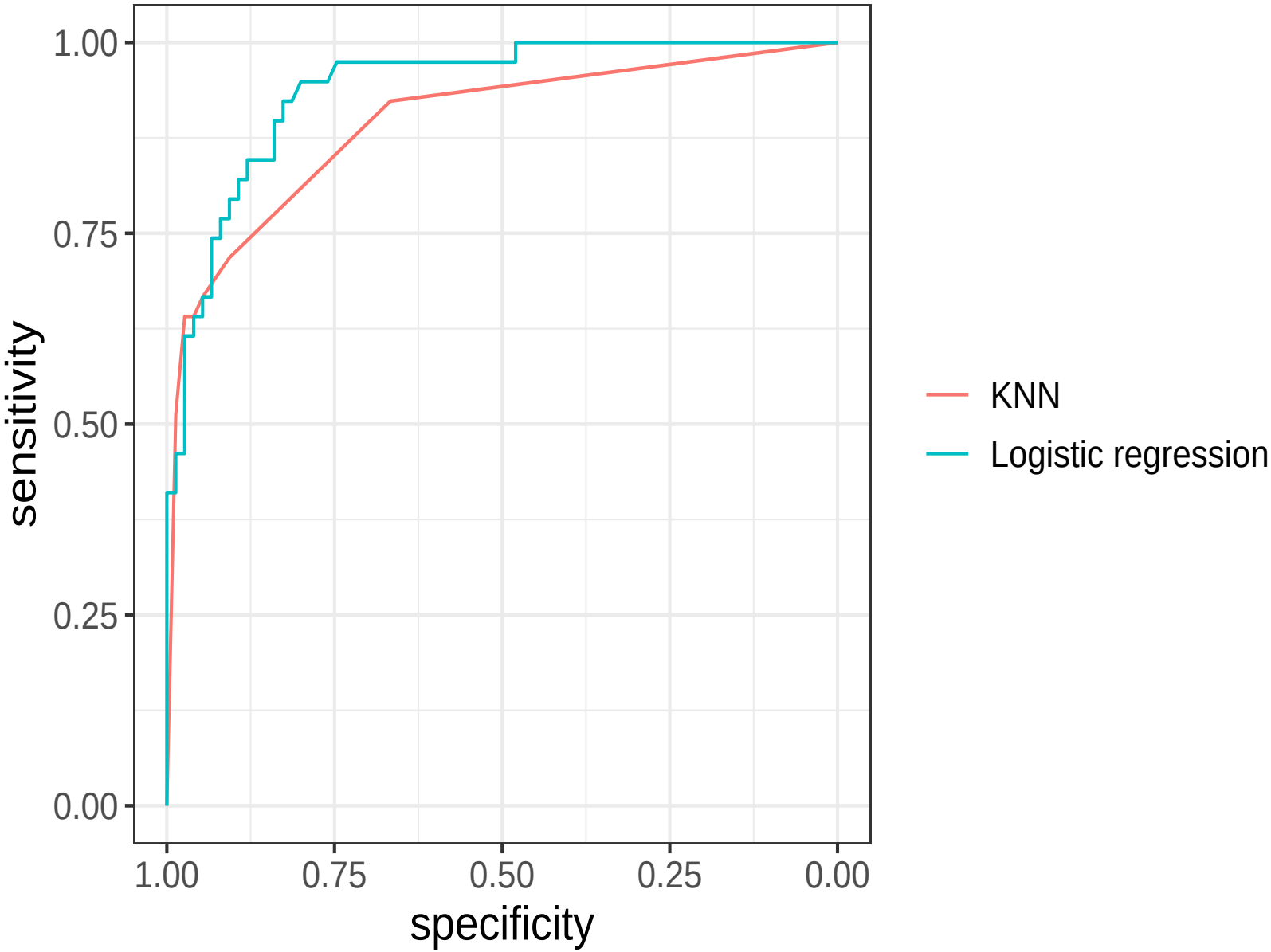
- Sensitivity **decreases** (less true positives)
- Specificity **increases** (less false positives)

ROC curve

- By varying C from 0 to 1, sensitivity and specificity change continuously and trace out the **Receiver Operator Curve (ROC)**.
- The closer the curve sticks to the upper left corner, the better
- Can be used to compare classifiers



ROC: KNN versus logistic regression



AUC: Area under the ROC

Single number to quantify performance of classifier:

- AUC $= 1.0$: distinguishes perfectly between two classes
- AUC $= 0.5$: classifier no better than guessing randomly
- $0.5 <$ AUC < 1.0: varying degrees of performance.

AUC: Link with concordance probability

Concordance probability: probability that classifier will give a negative sample a lower probability than a positive sample.

The AUC is equal to the concordance probability

$$\text{AUC} = P(\pi(x_{\text{neg}}) \leq \pi(x_{\text{pos}}))$$

Important for model calibration:

- Often, we don't care much about probability π to belong to the positive class
- But, want negative samples to have lower probability than positive samples

Example: clinical research

The ROC and AUC are often reported in clinical research.

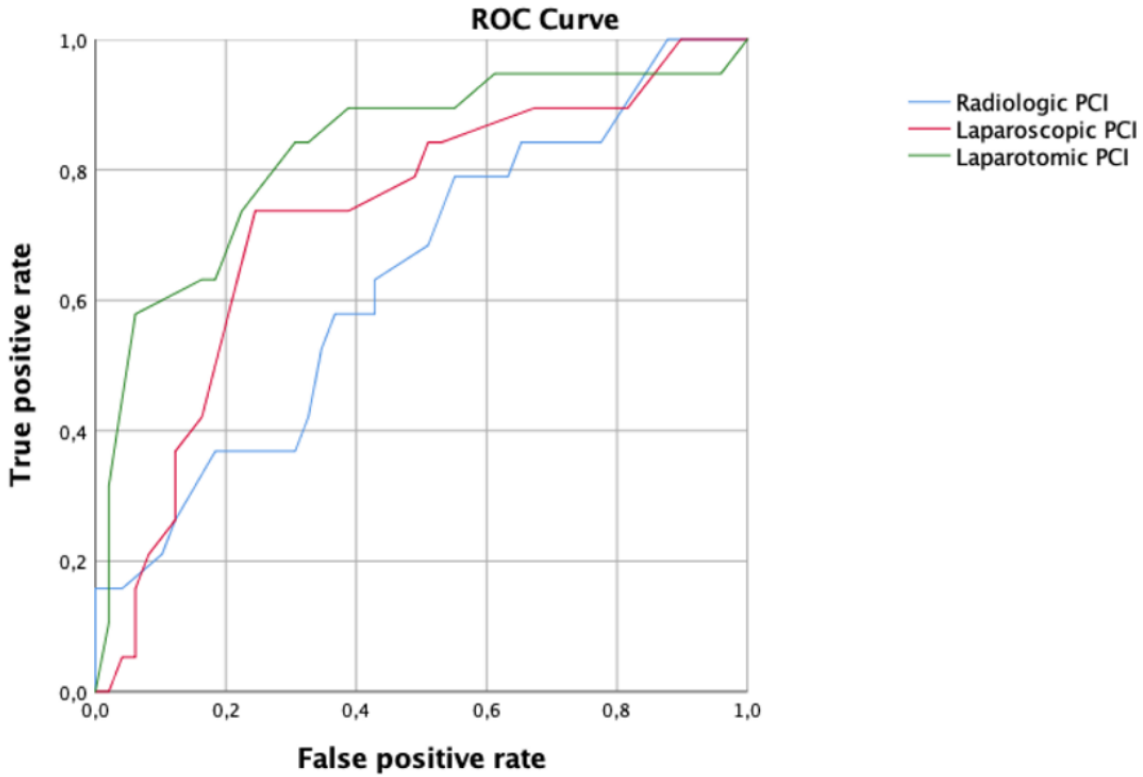


Figure 4. Receiver operating characteristic (ROC) curve comparing radiologic (blue line), laparoscopic (red line) and laparotomic (green line) peritoneal cancer index (PCI) in women who underwent primary debulking surgery.

Example: clinical research

Regarding the PCI score (Figure 4), the best performance to predict residual disease, with an AUC 0.83, CI 95% 0.71–0.95 was observed applying the laparotomic PCI, while the accuracy of the radiological PCI and laparoscopic PCI was AUC 0.64, CI 95% 0.49–0.78 and AUC 0.73, CI 95% 0.59–0.86, respectively. The cut-off value associated with the best performance of the laparotomic PCI score was 18.

Figure and text from Di Donna *et al.*, Concordance of Radiological, Laparoscopic and Laparotomic Scoring to Predict Complete Cytoreduction in Women with Advanced Ovarian Cancer. Cancers (2023)