Utrecht University

Classification

Applied Data Analysis and Visualization I

Department of Methodology and Statistics
Anastasia Giachanou, Ayoub Bagheri, Emmeke Aarts

Today

What	When
Linear regression for data science	Week 4
Classification	Week 5
Interactive visualizations with R shiny	Week 6
Tree-based methods	Week 7
Introduction to text mining	Week 8

Classification

Image from https://intellipaat.com/

Supervised learning: regression and classification
Classification: predict to which category an observation belongs (qualitative outcomes)

Classification

Many supervised learning problems concern categorical outcomes:

- Cancer: yes / no
- Weather: sunny / cloudy / windy / rainy / stormy
- Banking data: default on payment of debt
- Images: cat / no cat, or gazelle/tank/pirate/sea lion/tandem bicycle/. . .

Classification: predict to which category an observation belongs (qualitative outcomes)

Which one is a classification task?

Copy participation link

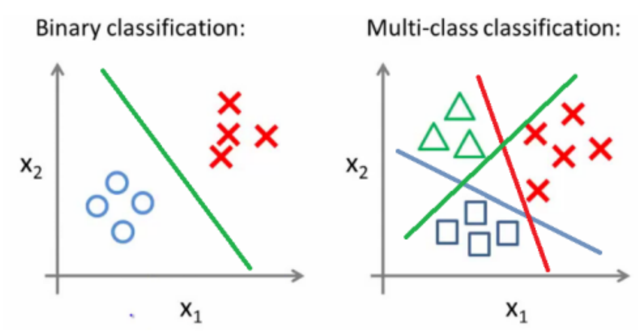
1 Go to **wooclap.com**

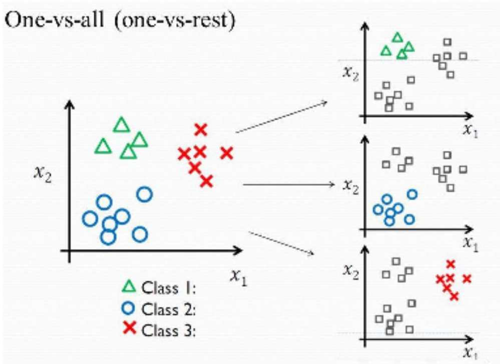
2 Enter the event code in the top banner

Event code
UKFZMG

Enable answers by SMS

Types of classifications





- Binary classification -> y: {0, 1}
- Multi-class classification: {0, 1, 2, 3, …, N}

Classification algorithms

- k-nearest neighbors (kNN)
- Logistic regression
- Naive Bayes (NB)
- Neural networks (deep learning)
- Support vector machine (SVM)
- Decision tree
- Random forest (RF)

Which algorithm to choose: Generalization

Training set (labels known)

Test set (labels unknown)

Generalization: How well does a learned model generalize from the data it was trained on to a new test set?

No Free Lunch Theorem

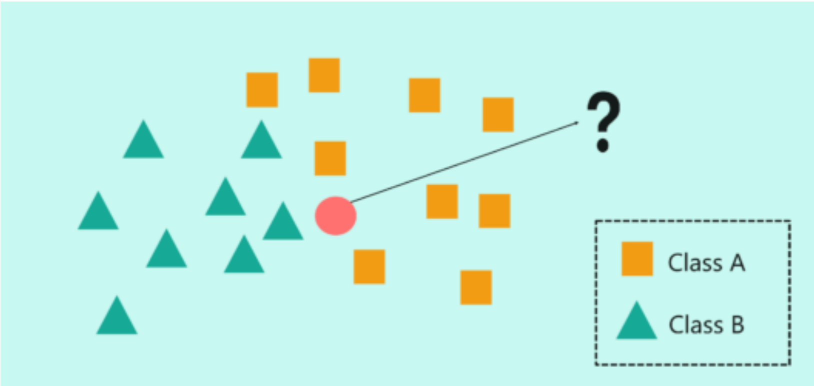
- No universally best classification algorithm

Classification Algorithms

K-nearest neighbors (kNN)

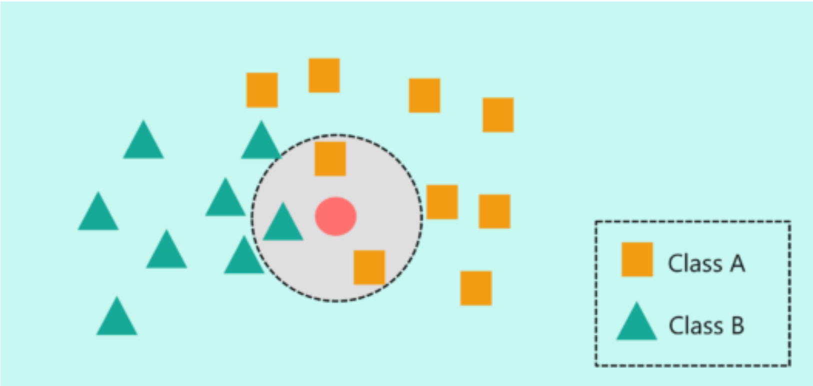
- One of the simplest (supervised) machine learning methods;
- Based on feature similarity: how similar is a data point to its neighbor(s) and classifies the data point into the class it is most similar to;

kNN



How does kNN Algorithm work? – kNN Algorithm In R – Edureka

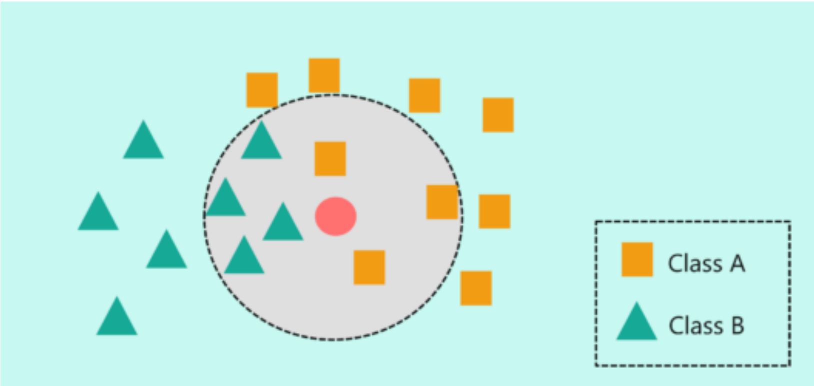
kNN



How does kNN Algorithm work? – kNN Algorithm In R – Edureka

$$K = 3$$

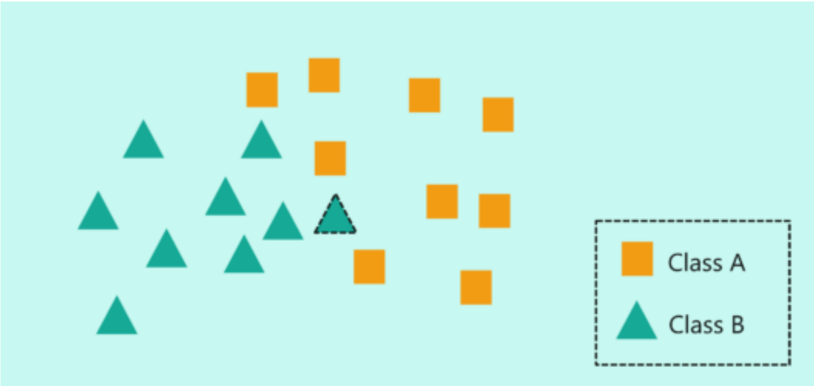
kNN



How does kNN Algorithm work? – kNN Algorithm In R – Edureka

$$K = 7$$

kNN



How does kNN Algorithm work? – kNN Algorithm In R – Edureka

kNN

Given the memorized training data, and a new data point (test observation):

- Identify the K closests points in the training data to the new data point x_0. This set of 'nearest neighbors' we call N_0

kNN

Given the memorized training data, and a new data point (test observation):

- Identify the K closests points in the training data to the new data point x_0. This set of 'nearest neighbors' we call N_0
- Estimate the probability that the new data point belongs to categroy j by $Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$ (so, the fraction of points in N_0 whose response equal j)

kNN

Given the memorized training data, and a new data point (test observation):

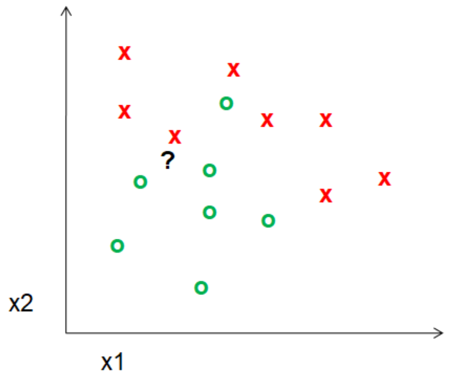
- Identify the K closests points in the training data to the new data point x_0. This set of 'nearest neighbors' we call N_0
- Estimate the probability that the new data point belongs to categroy j by $Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$
  (so, the fraction of points in N_0 whose response equal j)
- Majority vote: classify the test observation x_0 to the categroy with the largest probability

kNN points

- Non-parameteric model: does not make assumptions about the dataset, no fixed number of parameters;
- Lazy algorithm: memorizes the training dataset itself instead of learning a function from it;
- Can be used for both classification and regression (but more commonly used for classification);
- Although a very simple approach, kNN can often produce classifiers that are surprisingly good!

Quiz

Apply kNN methods with k = 1, 3 and 5 to the data points below and find the category of the test observation represented by (?) for each classifier.



Copy participation link

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code

UKFZMG

Enable answers by SMS

kNN

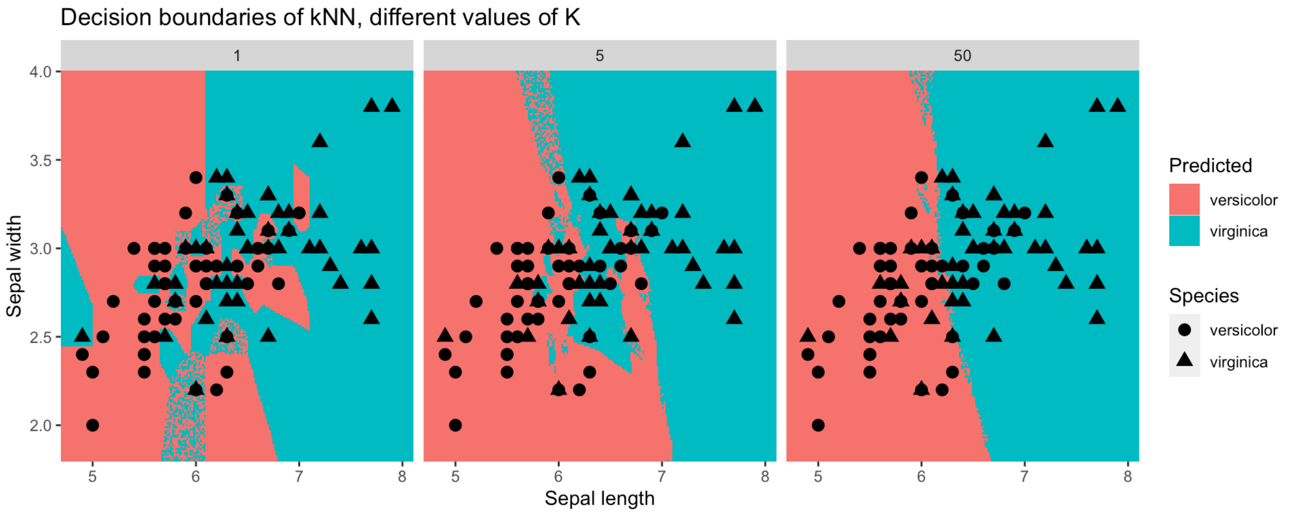
- Results obtained with kNN highly depend on chosen value for K, the number of neighbors used
- Small K (e.g., K = 1): low bias but very high variance, 'overly flexible decision boundary' (see next slides)
- Large K: low-variance but high-bias, 'decision boundary' that is close to linear
- The optimal value for K needs to be determined using a (cross-)validation approach

Example: Iris dataset

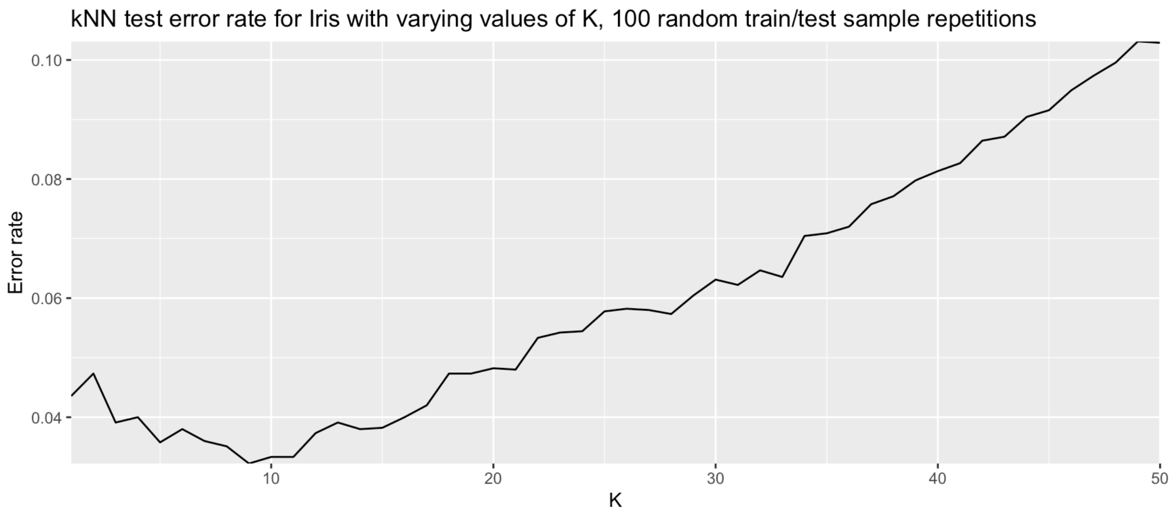
Iris is a (famous) dataset that contains species of flowers and various features of the flower such as Sepal length and Sepal width

Example: Iris dataset

And, for two species that are less well separated:



Example: Iris dataset

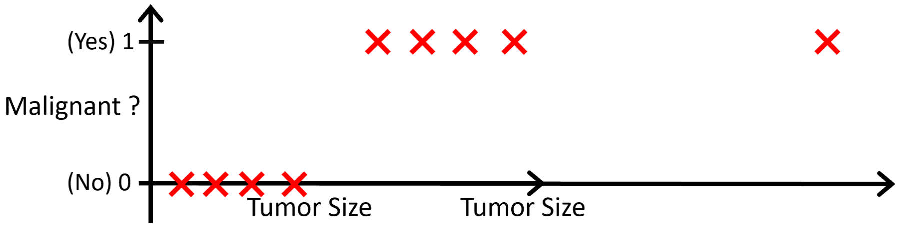


10 minute break

Logistic regression

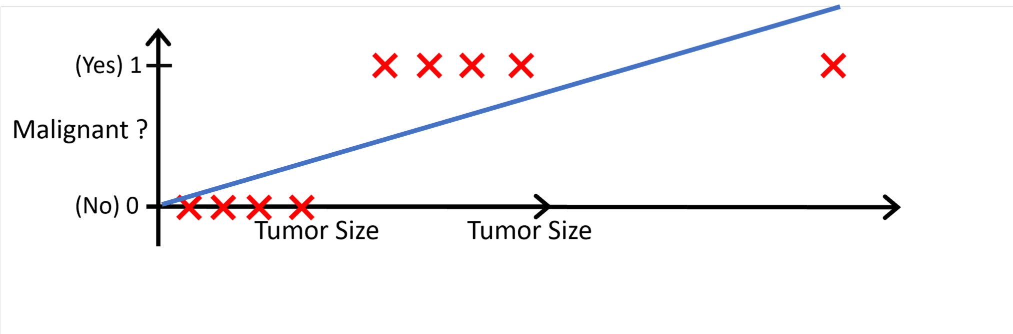
- Models the *probability* that y belongs to one of two categories (i.e., a binary outcome), for example:
  - Smoking / non smoking
  - Pass / fail an exam
  - Survival / Nonsurvival
  - Default yes / no
- Can be extended to model > 2 outcome categories: multinomial logistic regression (not treated in this course)
- Other option to model > 2 outcome categories: Neural networks, naive Bayes, linear discriminant analysis (not treated in this course, but treated in ISLR)

Logistic regression



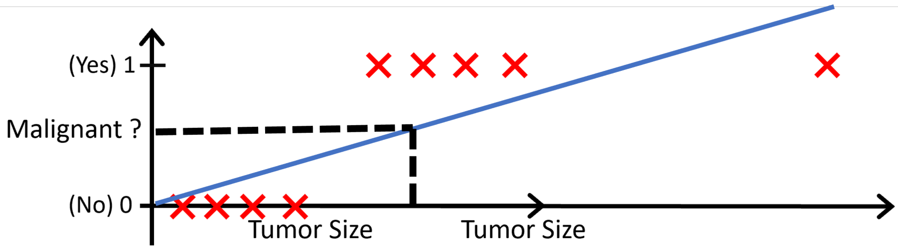
(Example by Andrew Ng)

Logistic regression



(Example by Andrew Ng)

Logistic regression



Threshold classifier output $h_\theta(x)$ at 0.5:

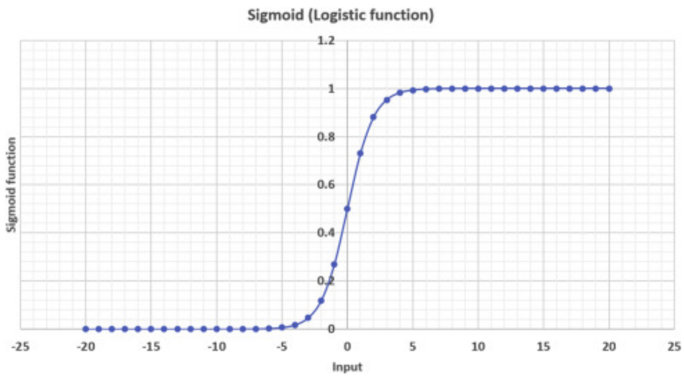
If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Andrew Ng

Logistic regression: logit

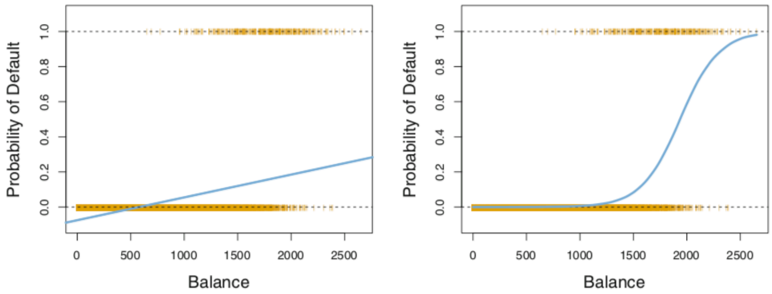
- Classification: y= 0 or 1
- Linear regression: can be <0 or >1
- Logistic regression: the prediction is between 0 and 1
- Solution: Use the logistic function



Logistic regression

Why can linear regression not be used on this type of data?

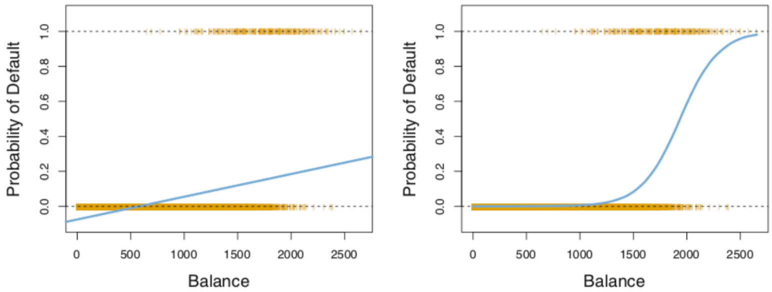
- Linear regression would predict impossible outcomes ($Pr(x)$ < 0 and > 1)
- To avoid this problem, we use a 'trick': we use a logistic 'link function (logit)'



Logistic regression

This results in the following logistic function: $Pr(Y = 1|X) = \frac{e^{\beta_0+\beta_1 X_1+...}}{1+e^{\beta_0+\beta_1 X_1+...}}$

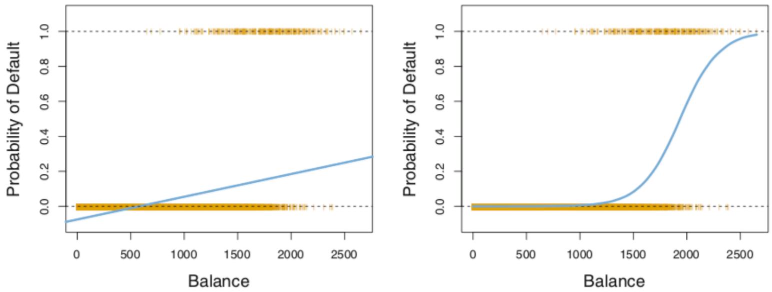
- Advantage: all predicted probabilities are above 0 and below 1
- Note: the linear predictor is contained in the exponent (i.e., $e^{\cdots}$)
- For the example below: $Pr(Default = yes|balance) = \frac{e^{\beta_0+\beta_1 balance}}{1+e^{\beta_0+\beta_1 balance}}$



Logistic regression

The logistic function: $Pr(Y=1|X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots}}{1+e^{\beta_0+\beta_1 X_1+\ldots}}$ continued..

- odds = $\frac{Pr(Y=1)}{Pr(Y=0)} = \frac{p_i}{1-p_i} = e^{\beta_0+\beta_1 X_1+\ldots}$
- ln(odds) $= \beta_0 + \beta_1 X_1 + \ldots$
- So the linear part of the function models the *log of the odds*.



Intermezzo: odds

Hence, when using logistic regression, we are modelling the log of the odds. Odds are a way of quantifying the probability of an event E.

- The odds for an event E are: $odds(E) = \frac{Pr(E)}{Pr(E^c)} = \frac{Pr(E)}{1-Pr(E)}$
- The odds of getting heads in a coin toss is: $odds(heads) = \frac{Pr(heads)}{Pr(tails)} = \frac{Pr(heads)}{1-Pr(heads)}$
- For a fair coin: $odds(heads) = \frac{0.5}{1-0.5} = 1$

Intermezzo: odds

Another example: The game Lingo has 44 balls: 36 blue, 6 red and 2 green balls

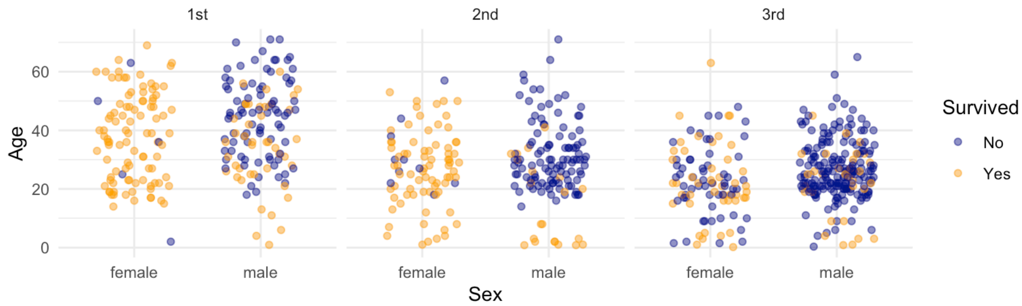
- The odds of a player choosing a blue ball are $odds(blue) = \frac{36}{8} = \frac{36/44}{8/44} = \frac{0.8182}{0.1818} = 4.5$
- The odds of a player choosing a green ball are $odds(green) = \frac{2}{42} = \frac{2/44}{42/44} = \frac{0.0455}{0.9545} \approx 0.05$
- Hence,
  - Odds of 1 indicate an equal likelihood of the event occurring or not occurring
  - Odds < 1 indicate a lower likelihood of the event occurring vs. not occurring
  - Odds > 1 indicate a higher likelihood of the event occurring.

Logistic regression

- Interpretation regression coefficients $\beta_1, \beta_2, \ldots$
  - Qualitatively: positive or negative effect of the predictor on the log of the odds (logit)
  - Quantitatively: effect on the odds is $exp(\beta)$
  - Is the effect statistically significant?
- Making predictions:
  - by filling in the equation $Pr(Y = 1|X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots}}{1+e^{\beta_0+\beta_1 X_1+\ldots}}$, we can predict the probability of the event to occur for a (hypothetical) case in our data

An example: Titanic dataset

```
##                                         Name PClass   Age    Sex Survived
## 1                 Allen, Miss Elisabeth Walton    1st 29.00 female        1
## 2                  Allison, Miss Helen Loraine    1st  2.00 female        0
## 3          Allison, Mr Hudson Joshua Creighton    1st 30.00   male        0
## 4 Allison, Mrs Hudson JC (Bessie Waldo Daniels)    1st 25.00 female        0
## 5                Allison, Master Hudson Trevor    1st  0.92   male        1
## 6                           Anderson, Mr Harry    1st 47.00   male        1
```



An example: Titanic dataset

```
log_mod_titanic <- glm(Survived ~ PClass + Sex + Age, data = titanic, family="binomial")
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.760	0.398	9.457	0
PClass2nd	-1.292	0.260	-4.968	0
PClass3rd	-2.521	0.277	-9.114	0
Sexmale	-2.631	0.202	-13.058	0
Age	-0.039	0.008	-5.144	0

- Compared to being in 1st class (reference category)
  - being in 2nd class decreases your probability of survival
  - being in 3rd class decreases your probability of survival
- Being male instead of female decreases your probability of survival
- Being older also decreases your probability of survival

An example: Titanic dataset

```
log_mod_titanic <- glm(Survived ~ PClass + Sex + Age, data = titanic, family="binomial")
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.760	0.398	9.457	0
PClass2nd	-1.292	0.260	-4.968	0
PClass3rd	-2.521	0.277	-9.114	0
Sexmale	-2.631	0.202	-13.058	0
Age	-0.039	0.008	-5.144	0

- odds ratio = $\frac{odds_{2ndclass}}{odds_{1stclass}} = e^{-1.292} = 0.275$. The odds of survival in 2nd class are 0.275 times the odds compared to first class
- odds ratio = $\frac{odds_{3rdclass}}{odds_{1stclass}} = e^{-2.521} = 0.080$. The odds of survival in 3rd class are 0.080 times the odds compared to first class

An example: Titanic dataset

```
log_mod_titanic <- glm(Survived ~ PClass + Sex + Age, data = titanic, family="binomial")
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.760	0.398	9.457	0
PClass2nd	-1.292	0.260	-4.968	0
PClass3rd	-2.521	0.277	-9.114	0
Sexmale	-2.631	0.202	-13.058	0
Age	-0.039	0.008	-5.144	0

Quiz: Predictions

The probability to survive for a:

- 30 year old female from 1st class?
- 45 year old male from 3rd class?

Copy participation link

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
UKFZMG

Enable answers by SMS

Making predictions (function `predict()` in `R`):

- The probability for a 30 year old female from 1st class to survive is: $Pr(Survival = yes|1^{st} class, female, 30 years) = \frac{e^{3.760-0.039*30}}{1+e^{3.760-0.039*30}} = 0.93$
- The probability for a 45 year old male from 3rd class to survive is only:

  $Pr(Survival = yes|3^{rd} class, male, 45 years) = \frac{e^{3.760-2.521*1-2.631*1-0.039*45}}{1+e^{3.760-2.521*1-2.631*1-0.039*45}} = 0.04$

Evaluating Classifiers

Evaluating classifiers

When applying classifiers, we have new options to evaluate how well a classifier is doing besides model fit:

- Confusion matrix (used to obtain most measures below)
- Sensitivity (‘*Recall*’)
- Specificity
- Positive predictive value (‘*Precision*’)
- Negative predictive value
- Accuracy (and error rate)
- ROC and area under the curve
- For even more: https://en.wikipedia.org/wiki/Sensitivity_and_specificity

Confusion matrix: Counts

You have trained a model on your training data and you now want to check the performance of the model on the validation set.

In case of a binary outcome (e.g., survival yes or no), we either correctly classify, or make two kind of mistakes:

- Label an item that belongs to the positive class as negative (False negative)
- Label an item that belongs to the negative class as positive (False positive)

Confusion matrix: Counts

In case of a binary outcome (e.g., survival yes or no), we either correctly classify,

- Label a survivor as someone who survived → True positive (TP)
- Label someone who did not survive as non-survived → True negative (TN)

or make two kind of mistakes:

- Label a survivor as someone who did not survive → False negative (FN)
- Label someone who did not survive as a survivor → False positive (FP)

Confusion matrix: Counts

- Label a survivor as someone who survived → True positive (TP)
- Label someone who did not survive as non-survived → True negative (TN)
- Label a survivor as someone who did not survive → False negative (FN)
- Label someone who did not survive as a survivor → False positive (FP)
- Total errors: FN + FP

	Not survived	Survivor
Survived (predicted)		
No	372 (TN)	91 (FN)
Yes	71 (FP)	222 (TP)

Confusion matrix: Specificity

- Measures the percentage of actual negatives which are correctly identified
- Of the people who did not survive, which proportion did the model 'find'
- Specificity: $\frac{TN}{TN+FP} = 372/(372 + 71) \approx 0.84$

	Not survived	Survivor
Survived (predicted)		
No	372 (TN)	91 (FN)
Yes	71 (FP)	222 (TP)

Confusion matrix: Sensitivity

- Measures the percentage of actual positives which are correctly identified (or recall or True Positive Rate)
- Sensitivity: Of the people who survived, which proportion did the model 'find'
- Sensitivity: $\frac{TP}{TP+FN} = 222/(222+91) \approx 0.71$

	Not survived	Survivor
Survived (predicted)		
No	372 (TN)	91 (FN)
Yes	71 (FP)	222 (TP)

Confusion matrix: Specificiy and Sensitivity

	Not survived	Survivor
Survived (predicted)		
No	0.84 (Specificity)	0.29 (1 - Sensitivity)
Yes	0.16 (1 - Specificity)	0.71 (Sensitivity)

Confusion matrix: Accuracy

- Measures the percentage of overall correct predictions
- Accuracy (ACC): $\frac{TP+TN}{TP+FP+TN+FN} \approx 0.79$, Error rate: 1 - accuracy ≈ 0.21

	Not survived	Survivor
Survived (predicted)		
No	372 (TN)	91 (FN)
Yes	71 (FP)	222 (TP)

Confusion matrix: Pos and Neg predicted value

- Negative predicted value (NPV): $\frac{TN}{TN+FN} = 372/(372 + 91) \approx 0.80$
  Of the people we predicted to not survive, which proportion actually did die
- Positive predicted value (*'precision'*): $\frac{TP}{TP+FP} = 222/(222 + 71) \approx 0.76$
- Of the people we predicted to survive, which proportion actually survive

	Not survived	Survivor
Survived (predicted)		
No	0.80 (NPV)	0.20 (1 - NPV)
Yes	0.20 (1 - PPV)	0.80 (PPV)

Thresholds

- Moving around the threshold affects the sensitivity and specificity!
- Moving the threshold especially makes sense when the predicted categories are unbalanced. For example, many more non survivors compared to survivors in the dataset.

```
with(titanic,
     table(p_ped > 0.4, Survived))
```

```
##        Survived
##           0   1
##   FALSE 346  63
##   TRUE   97 250
```

```
with(titanic,
     table(p_ped > 0.6, Survived))
```

```
##        Survived
##           0   1
##   FALSE 401 114
##   TRUE   42 199
```

ROC curve

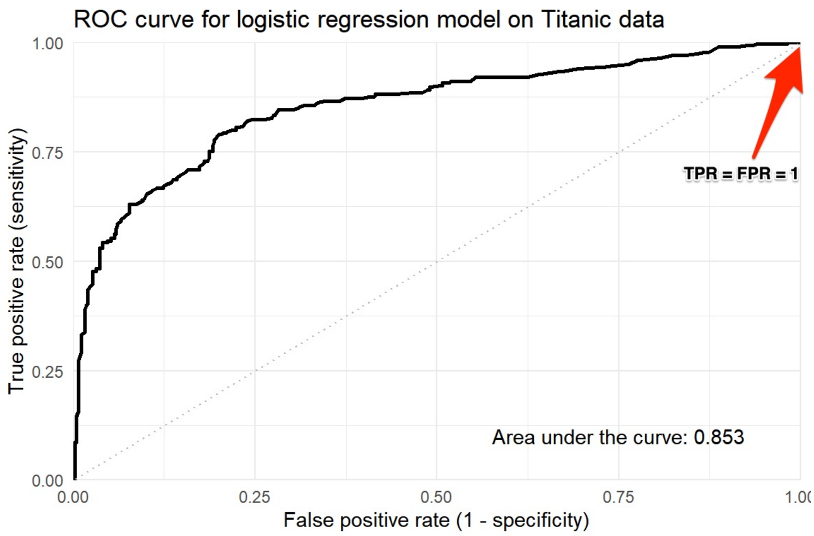
- ROC curve is a popular graphic for simultaneously displaying the true and false positive rate *for all possible thresholds*
- TPR (sensitivity), percentage of actual positives (survived) which are correctly predicted as survived
- FPR (1 - specificity): proportion of actual negatives (non survivors) that were incorreclty classified as survived and which are the FP
- The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the curve (AUC)

ROC curve - Titanic data

- Assume a very low threshold such as 0.01
- TPR = Sensitivity: 313 / (313 + 0) = 1 (every survivor was correctly classified)
- FPR = 1 - Specificity = 443 / (443 + 0) = 1 (every single passenger that did not survive was classified as survived)

	Not survived	Survivor
Survived (predicted)		
No	0	0
Yes	443	313

ROC curve - Titanic data



- The higher the curve and the larger the area under the curve (AUC), the better the classifier is

Conclusion

Classification: predict to which category an observation belongs (qualitative outcomes)

When predicting categorical outcomes (= classification)

- We can use a completely non-parametric approach with kNN.
- As no assumptions are made about the decision boundary, kNN will outperform logistic regression when the decision boundary is highly non-linear.
- kNN does not give any information on the prediction process, e.g., which variable is most important in providing an accurate prediction.

Conclusion

When predicting categorical outcomes (= classification)

- We can use a parametric approach such as logistic regression, modeling the log of the odds with a linear function.
- Provides both information on the prediction process (i.e., regression coefficients) and predicted class probabilities for each observation.
- To classify observations based on their probabilities, it can make sense to use a different threshold than 0.50 (in case of binary data).
- We can use various metrics based on the confusion matrix to assess performance of classifiers.
- More classification methods will be discussed in week 7!

Final note

Lab session on **Thursday**.

Next week: Interactive visualizations with R shiny

Have a nice day!

Extra

Parametric vs non-parametric classifiers

Parametric model	Non-parametric model
It uses a fixed number of parameters to build the model.	It uses flexible number of parameters to build the model.
Considers strong assumptions about the data.	Considers fewer assumptions about the data.
Computationally faster	Computationally slower
Require lesser data	Require more data
Example – Logistic Regression & Naïve Bayes models	Example – KNN & Decision Tree models

Generative classifiers try to model the data. Discriminative classifiers try to predict the label.