

# How to evaluate output?

資料科學 Data Science

張家銘 Jia-Ming Chang

政治大學資訊科學系

```
224  
225 #wpstats { display: none; }  
226  
227 .sticky {  
228     margin-bottom: 50px;  
229 }  
230  
231 .sticky .content-inner {  
232     margin-bottom: 0px!important;  
233     padding-bottom: 0px!important;  
234     border-bottom: 0px!important;  
235     -o-box-shadow: 0 1px 2px rgba(0,0,0,.2);  
236     -moz-box-shadow: 0 1px 2px rgba(0,0,0,.2);  
237     -webkit-box-shadow: 0 1px 2px rgba(0,0,0,.2);  
238     box-shadow: 0 1px 2px rgba(0,0,0,.2);  
239     background-color: #fff;  
240     padding: 25px!important;  
241     position: relative;  
242 }  
243  
244 .side-box {  
245     padding: 10px 0;  
246     margin-bottom: 10px;  
247     border: 1px solid #CCC;  
248     background-color: #E6E6E6;  
249     text-align: center;  
250 }  
251  
252 .side-box a:link,  
253 .side-box a:visited {  
254     font-weight: normal;  
255     color: #06c55b;  
256     font-size: 12px;
```

# Copyright declaration 版權說明

- Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.
- [The web site of the book](#)
- The credit of individual is indicated in the bottom part of the slide.
  - ie.,

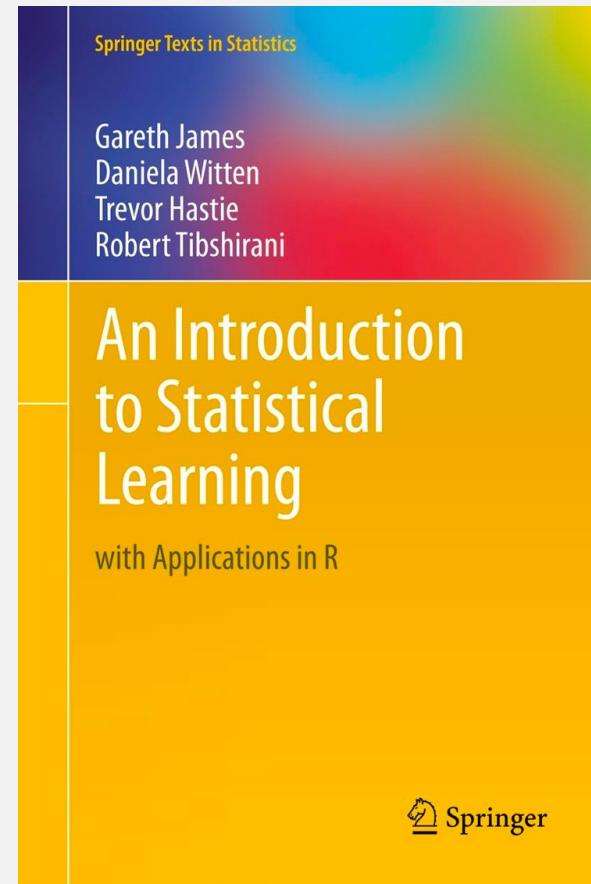
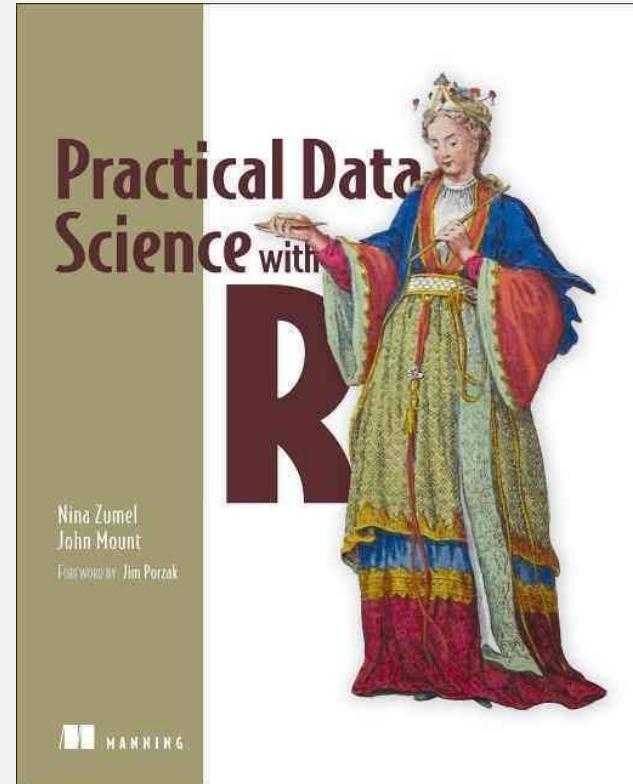


Figure 3.18, *An Introduction to Statistical Learning with Applications in R*, by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani

# Copyright declaration 版權說明

- Some of the figures in this presentation are taken from "Practical Data Science with R (Manning, 2019)"
- [The web site of the book](#)
- The credit of individual is indicated in the bottom part of the slide.
  - ie.,

Figure 7.6, *Practical Data Science with R* by Nina Zumel and John Mount



# Very nice e-book, interactive!!!

- <https://www.manning.com/books/practical-data-science-with-r-second-edition>

The screenshot shows the product page for 'Practical Data Science with R, Second Edition'. On the left, there's a thumbnail of the book cover featuring a woman in traditional Korean clothing. Below the cover are links to 'Look Inside', 'resources', 'Source code', 'Book Forum', and a 'FREE' section where users can see the entire first chapter.

The main title is 'Practical Data Science with R, Second Edition' with a heart icon. It has a 4.5-star rating from 10 reviews. The authors are Nina Zumel and John Mount, with a foreword by Jeremy Howard and Rachel Thomas. The publication date is November 2019, ISBN 9781617295874, and it contains 568 pages, printed in black & white.

A green box highlights a 'free previous edition eBook included' offer, stating that an eBook copy of the previous edition is included at no additional cost and will be automatically added to the Manning Bookshelf within 24 hours of purchase.

A quote box says: "Full of useful shared experience and practical advice. Highly recommended." - From the Foreword by Jeremy Howard and Rachel Thomas.

The right side shows two purchasing options:

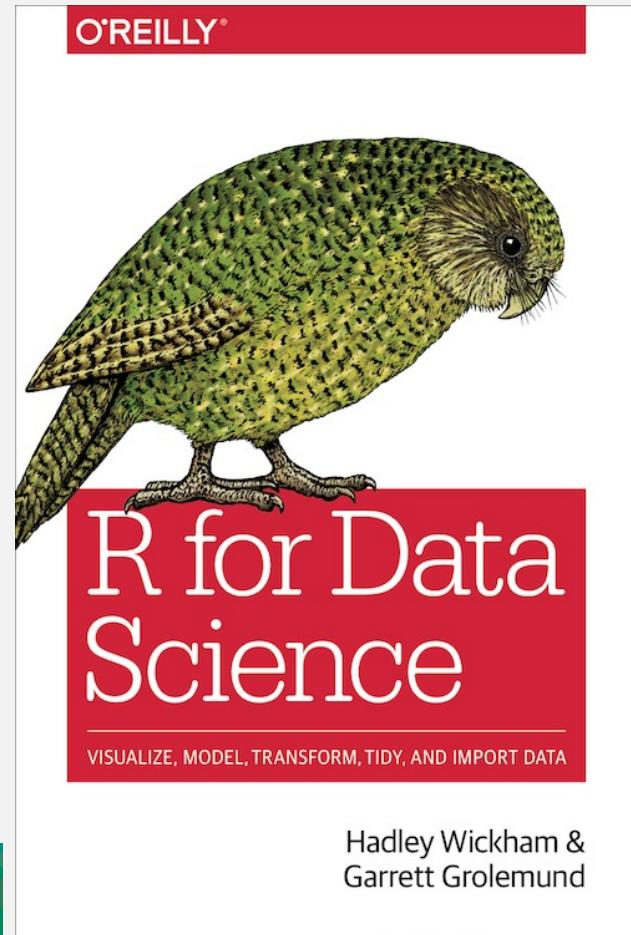
- print book**: \$29.99 (includes previous edition eBook). Buttons: 'add to cart' and '2-click buy'.
- eBook**: \$24.99 (3 formats + liveBook, includes previous edition eBook). Buttons: 'add to cart' and '2-click buy'.

Below these options is a 'custo' section containing a snippet of R code for building and applying a logistic regression model to spam data. A callout box labeled '5' points to the final line of the code: 'Makes predictions on the training and test sets'.

```
Listing 6.1. Building and applying a logistic regression spam model
1  spamD <- read.table('spamD.tsv',header=T,sep='\t')
2
3  spamTrain <- subset(spamD,spamD$rgroup >= 10)
4  spamTest <- subset(spamD,spamD$rgroup < 10)
5
6  spamVars <- setdiff(colnames(spamD), list('rgroup','spam'))
7  spamFormula <- as.formula(paste('spam == "spam"', 
8    paste(spamVars, collapse = ' + '),sep = ' ~ '))
9
10 spamModel <- glm(spamFormula,family = binomial(link = 'logit'),
11                      data = spamTrain)
12
13 spamTrain$pred <- predict(spamModel,newdata = spamTrain,
14                           type = 'response')
15 spamTest$pred <- predict(spamModel,newdata = spamTest,
16                           type = 'response')
```

# Copyright declaration 版權說明

- Some of the figures in this presentation are taken from "R for Data Science" under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License.
- [The web site of the book](#)
- The credit of individual is indicated in the bottom part.
  - ie.,



# Data Science this week

- Data Science Books
  - <https://www.datascienceweekly.org/data-science-resources/data-science-books>
- A list of books covering Data Analysis, Data Science, Machine Learning, Data Visualization, Statistics & Associated Programming Languages

# Recap from the last week



# R language

- Assignment : <- not =
- &, | vs &&, ||
- identical, all.equal
- [[]] vs []
- read\_table()
- summary()

# Primary R data types

- Numbers
  - Number Sequences
  - Vectors
  - Lists
  - Matrices
  - Data Frames
  - Factors
  - Null and NA
- Which one is central data structure?

# Source code management system



- a distributed version control system
- The development of Git began on 3 April 2005
- Torvalds quipped about the name *git*, which is British English slang meaning "unpleasant person".
- <http://git-scm.com/video/what-is-git>
- <https://git-scm.com/downloads>
- <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

# Github



- **GitHub** is a web-based Git repository hosting service.
- Development of the GitHub platform began on 1 October 2007.
- The site was launched in April 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett after it had been made available for a few months prior as a beta period.

Owner  Repository name

Great repository names are short and memorable. Need inspiration? How about [mustached-octo-batman](#).

Description (optional)

**Public**  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add `.gitignore`: **None** | Add a license: **None**

**Create repository**

# Example Code

- Readthedocs
  - code03.zip
- Run R script from command line
  - cd code03
  - Rscript <example>.R

# Today

- Practical Data Science with R by Nina Zumel and John Mount, (Manning, 2019)
  - Cha 05. Choosing and evaluating models
- An Introduction to Statistical Learning with Applications in R by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani
  - Cha 02. Statistical Learning

# Schematic model construction and evaluation

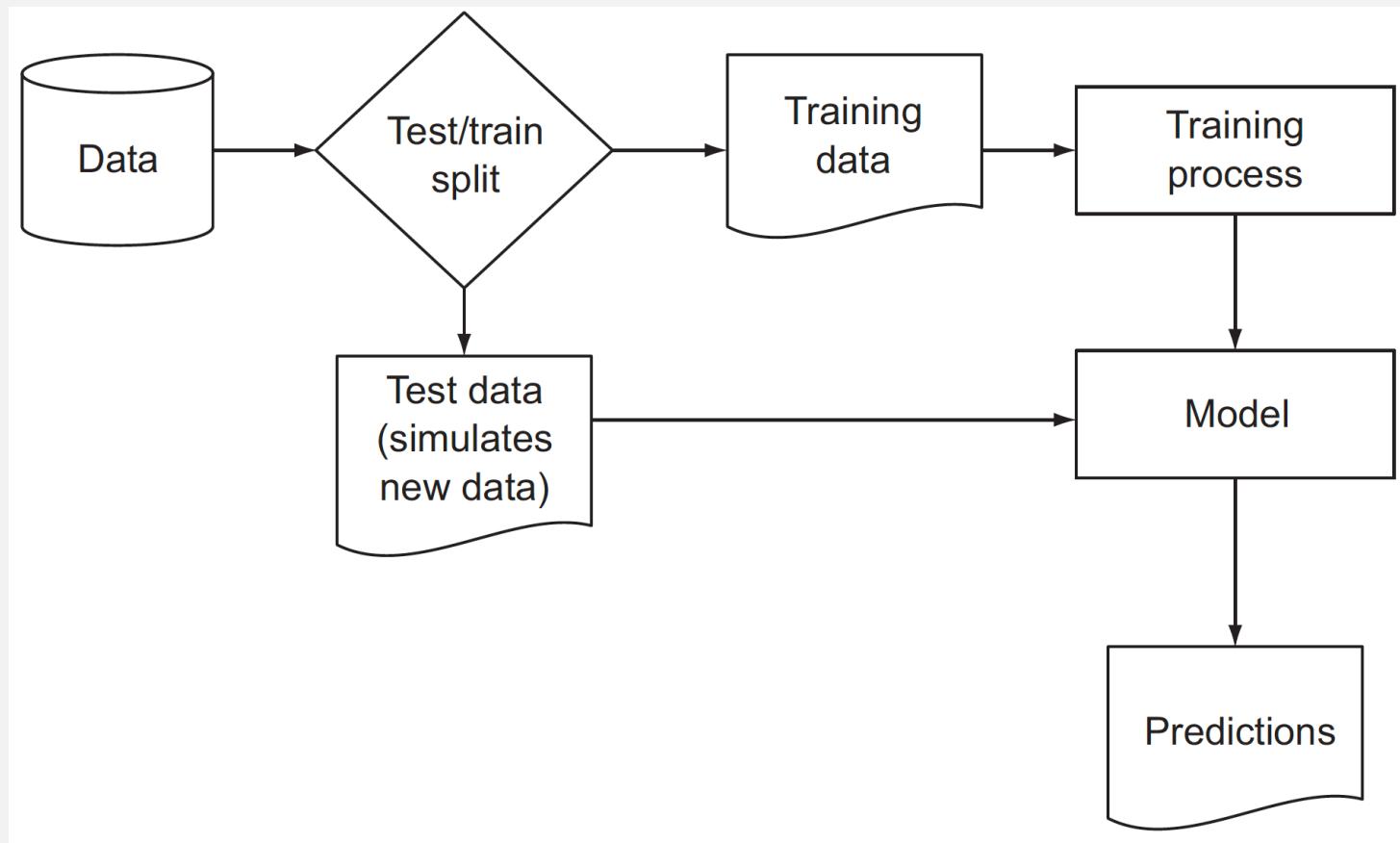


Figure 5.1, *Practical Data Science with R* by Nina Zumel, John Mount

# Schematic model construction and evaluation

- model evaluation = quantifying the performance of a model that's appropriate to both the original business goal and the chosen modeling technique.
  1. a classification task (predicting who would default on loans)
    - Precision
    - recall
  2. a probability task
  3. a scoring task (predicting revenue lost to defaulting loans)
    - root mean square error (RMSE)

# Model evaluation and critique

- Is it accurate enough for your needs? Does it generalize well?
- Does it perform better than “the obvious guess”? Better than whatever estimate you currently use?
- Do the results of the model (coefficients, clusters, rules) make sense in the context of the problem domain?

# Outline

- The accuracy of the Model
- Evaluating models
  - Evaluating classification models
    - Covid-19 general sieve?
    - Multi-class classification
  - Evaluating scoring models
  - Evaluating probability models
  - Evaluating ranking models
  - Evaluating clustering models
    - Internal
    - External

# The accuracy of the Model



# Why Estimate $f$ ?

- Prediction
  - $\hat{Y} = \hat{f}(X)$
  - $\hat{f}$ : our estimate for  $f$
  - $\hat{Y}$  : resulting prediction for  $Y$

$$\hat{Y} = \hat{f}(X)$$

- The accuracy of  $\hat{Y}$  as a prediction for  $Y$  depends on two quantities
  1. the reducible error: we can potentially improve the accuracy of  $\hat{f}$  by using the most appropriate statistical learning technique to estimate  $f$
  2. the irreducible error

$$\hat{Y} = \hat{f}(X)$$

- The accuracy of  $\hat{Y}$  as a prediction for  $Y$  depends on two quantities
  1. the reducible error
  2. the irreducible error
    1. By definition:  $Y$  is also a function of  $\epsilon$ , which, by definition, cannot be predicted using  $X$
    2. The quantity  $\epsilon$  may contain unmeasured variables that are useful in predicting  $Y$
    3. The quantity  $\epsilon$  may also contain unmeasurable variation.

- $Y = f(X) + \epsilon$
  - $\hat{Y} = \hat{f}(X)$
  - $E(Y - \hat{Y})^2 = E[f(X) + \epsilon - \hat{f}(X)]^2$   
 $= [f(X) - \hat{f}(X)]^2 + Var(\epsilon)$ 
    - $E(Y - \hat{Y})^2$  : the average, or expected value, of the squared expected difference between the predicted and actual value of  $Y$
    - variance associated with the error term  $\epsilon$
    - Which is Reducible? Which is Irreducible?
- $[f(x) - \hat{f}(x)]^2$        $Var(\epsilon)$
- the part of the error that  
 can be reduced by using  
 a better model or learning  
 algorithm.

1. Start with the definition of the prediction error:

$$Y - \hat{Y} = (f(X) + \epsilon) - \hat{f}(X)$$

2. Square the prediction error to find the squared error:

$$(Y - \hat{Y})^2 = [f(X) + \epsilon - \hat{f}(X)]^2$$

3. Expand the squared term:

$$(Y - \hat{Y})^2 = [f(X) - \hat{f}(X)]^2 + 2\epsilon[f(X) - \hat{f}(X)] + \epsilon^2$$

4. Take the expected value of the squared error:

$$E[(Y - \hat{Y})^2] = E\{[f(X) - \hat{f}(X)]^2\} + E\{2\epsilon[f(X) - \hat{f}(X)]\} + E(\epsilon^2)$$

5. Evaluate each term:

- $E\{[f(X) - \hat{f}(X)]^2\}$  is the variance or the reducible error. It's the part of the error we can potentially reduce by improving our model.
- $E\{2\epsilon[f(X) - \hat{f}(X)]\}$  is zero, because the error  $\epsilon$  has an expected value of zero and is independent of  $X$ . (Var(x) = E[x^2] - (Ex)^2)
- $E(\epsilon^2)$  is just  $Var(\epsilon)$  because the error has a mean of zero, so the expected value of the square of an error is its variance.

6. The final equation for the expected prediction error becomes:

$$E[(Y - \hat{Y})^2] = [f(X) - \hat{f}(X)]^2 + Var(\epsilon)$$

So, you have two components:

- Reducible error:  $[f(X) - \hat{f}(X)]^2$
- Irreducible error:  $Var(\epsilon)$

# Inference v.s. Prediction

- interested in understanding the way that  $Y$  is affected as  $X_1, \dots, X_p$  change.
  - wish to estimate  $f$ , not necessarily to make predictions for  $Y$
  - understand the relationship between  $X$  and  $Y$
  - understand how  $Y$  changes as a function of  $X_1, \dots, X_p$
  - $\hat{f}$  cannot be treated as a black box : *as opposed to prediction*

# Inference

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?
- Can the relationship between  $Y$  and each predictor be adequately summarized
  - using a linear equation
  - the relationship more complicated?

# Inference

- the Advertising data
  - Which media contribute to sales?
  - Which media generate the biggest boost in sales?
  - How much increase in sales is associated with a given increase in TV advertising?

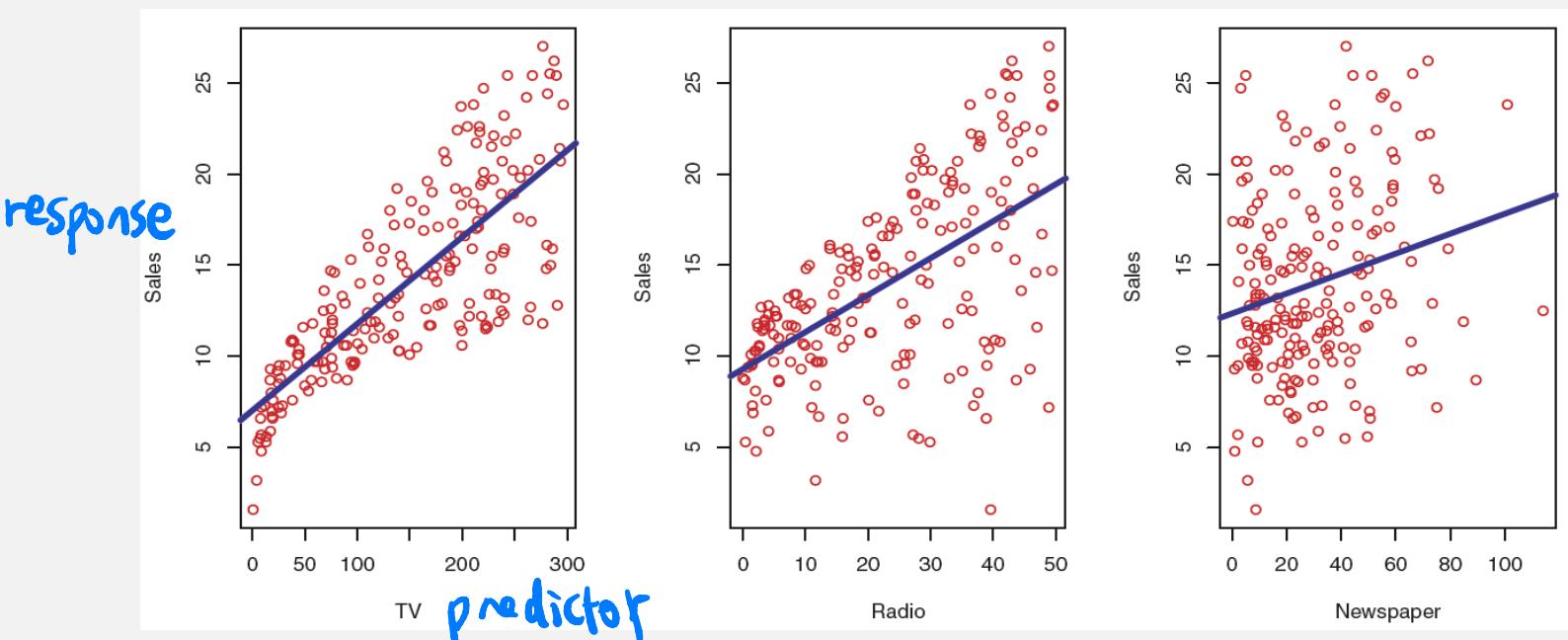


Figure 2.1, *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

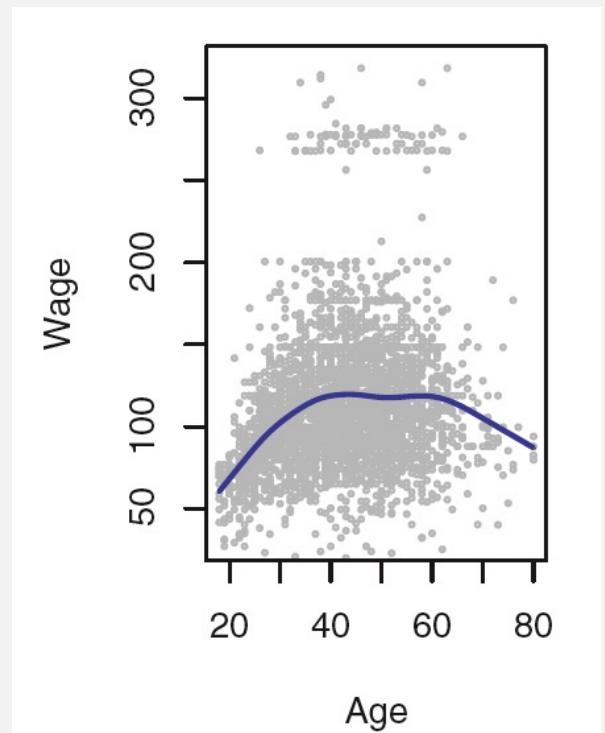
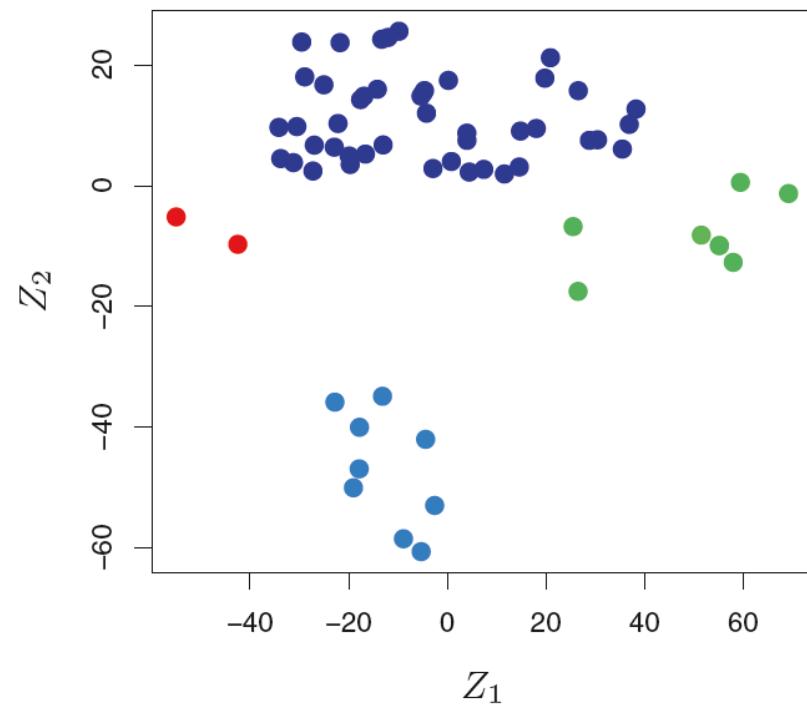
# Evaluating models

# Evaluating models

- Classification
- Scoring
- Probability estimation
- Ranking
- Clustering

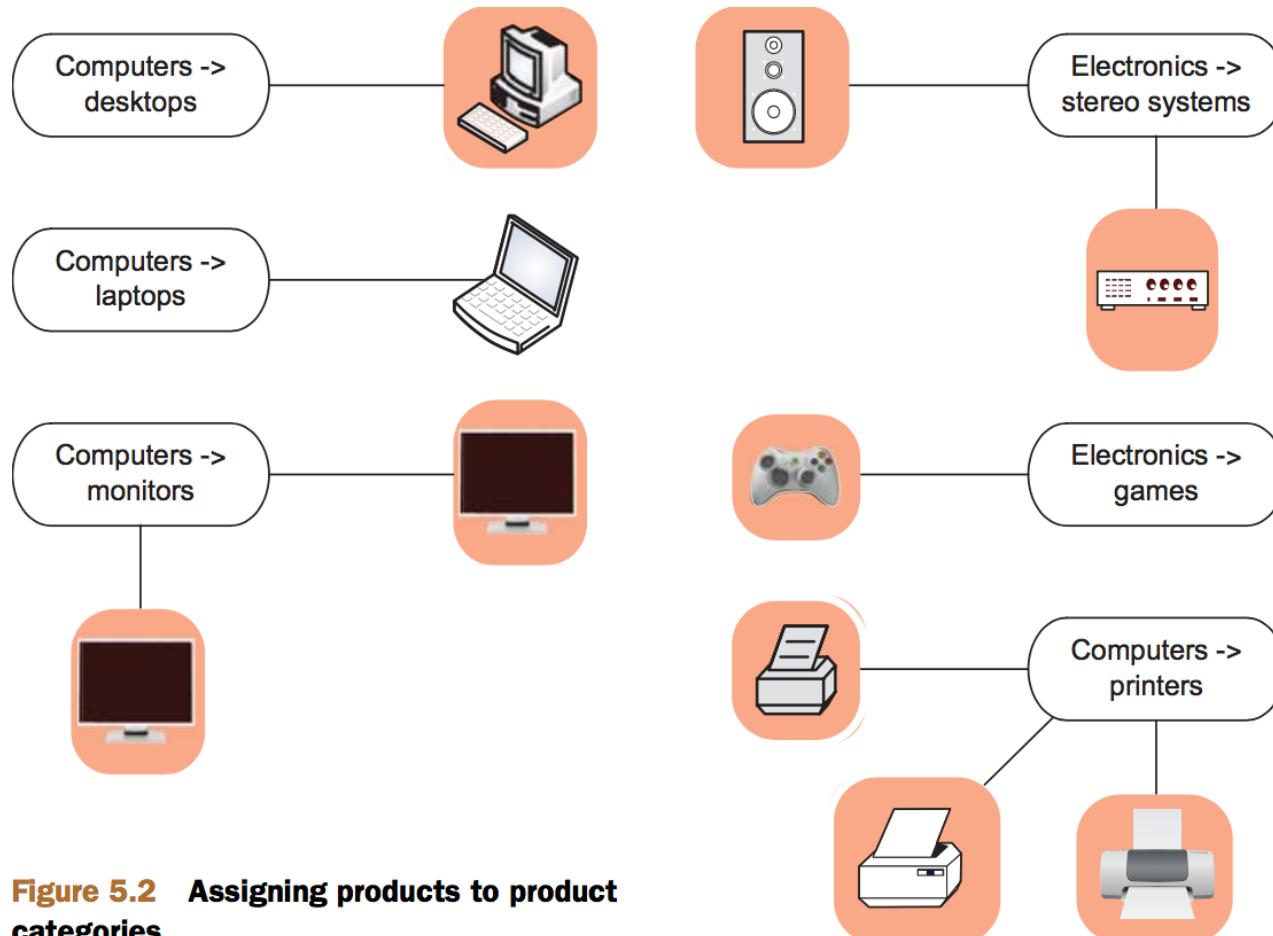
# Regression v.s. Classification

- quantitative : regression
- qualitative : classification



# Evaluating classification models

# Assigning products to product categories



**Figure 5.2** Assigning products to product categories

# Multicategory vs. two-category classification

- using binary classifiers to solve multicategory problems
  - building one classifier for each category, called a **one-versus-rest classifier** (one-versus-all)
- find a suitable multiple-category implementation
  - logistic regression : use the package *mlogit* instead of the base method *glm*

# Confusion matrix

- A good summary of classifier accuracy is the *confusion matrix*
  - which tabulates actual classifications against predicted ones

# Confusion matrix

- confusionMatrix.R
- Step1: A decision tree model for finding bad loan applications

```
library('rpart')  
load('GCDData.RData')  
model <- rpart(Good.Loan ~ Duration.in.month +  
    Installment.rate.in.percentage.of.disposable.income +  
    Credit.amount + Other.installment.plans,  
    data=d, control=rpart.control(maxdepth=4),  
    method="class")
```

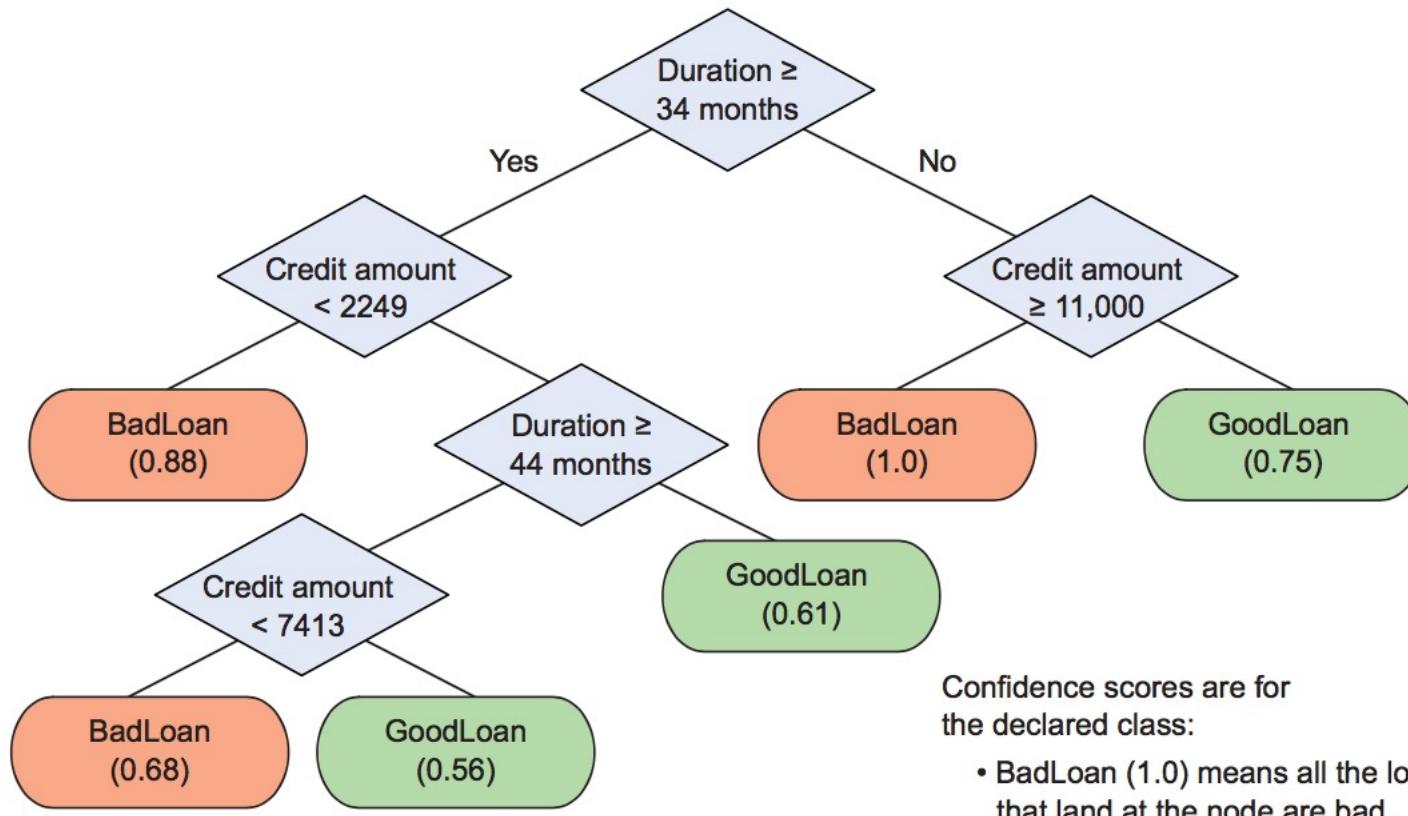
*training set*

*classification model*

*decision tree model*

*predict based on*

# A decision tree



Confidence scores are for  
the declared class:

- BadLoan (1.0) means all the loans that land at the node are bad.
- GoodLoan (0.75) means 75% of the loans that land at the node are good.

# Confusion matrix

result frame	Good. Loan	pred
Y-test		
Y-pred		

- Step2: plotting the confusion matrix

```
resultframe <- data.frame(Good.Loan=creditdata$Good.Loan,  
                           pred=predict(model, type="class"))  
(rtab <- table(resultframe))
```

- Confusion matrix  
*generates a confusion matrix*

- Rows represent actual loan status
- columns represent predicted loan status.
- The diagonal entries represent correct predictions.

↑  
the decision  
tree model

↑  
predicted values  
should be class  
labels rather  
than probabilities

# Confusion matrix

```
> resultframe <- data.frame(Good.Loan=creditdata$Good.Loan,  
                           pred=predict(model, type="class"))  
> rtab <- table(resultframe)  
> rtab  
      pred  
Good.Loan BadLoan GoodLoan  
BadLoan      41     259  
GoodLoan     13     687
```



Create the confusion matrix. Rows represent actual loan status; columns represent predicted loan status. The diagonal entries represent correct predictions.

# Evaluating classification models

- Definition (`BadLoan` => positive case)

- True positive
- False positive
- True negative
- False negative

IS THE prediction correct?

		pred	
		Good.Loan	BadLoan
	BadLoan	TP	41
	GoodLoan	FP	13
		GoodLoan	
		FN	259
		TN	687

# Evaluation

		pred	
		GoodLoan	BadLoan
		BadLoan	TP 41 FN 259
		GoodLoan	FP 13 TN 687

- Accuracy *correctly predicted*
  - # of items categorized correctly divided by #of items
  - $(TP+TN)/(TP+TN+FP+FN)$

<https://stats.stackexchange.com/questions/336455/fpr-false-positive-rate-vs-fdr-false-discovery-rate>

- Precision *\* Precision + False Positive Rate = 1*
  - fraction of the items the classifier flags as being in the class actually are in the class = how often a positive indication turns out to be correct *predicted positive*
  - $TP/(TP+FP)$

- Recall, *True Positive Rate (TPR), Sensitivity*

- what fraction of the things that are in the class are detected by the classifier
- $TP/(TP+FN)$

- False positive rate =  $FP/(FP+TN)$  *(FPR)*

*comparison*

$$TNR = \frac{TN}{TN+FP}$$

*(Specificity)*

$$\text{False Discovery Rate (FDR)} = \frac{FP}{FP+TP}$$

# Building and applying a logistic regression spam model

- Spambase dataset (<http://mng.bz/e8Rh>)
  - <https://raw.githubusercontent.com/WinVector/zmPDSwR/master/Spambase/spamD.tsv>
- [spamExam.R](#)

```
# load data
spamD <- read.table('spamD.tsv', header=T, sep='\t')
spamTrain <- subset(spamD, spamD$rgroup>=10)
spamTest <- subset(spamD, spamD$rgroup<10)
spamVars <- setdiff(colnames(spamD), list('rgroup', 'spam'))
```

select all the column names except  
for 'rgroup' and 'spam'

# Building and applying a logistic regression spam model

```
# build model  
  
spamFormula <- as.formula(paste('spam=="spam"',  
paste(spamVars,collapse=' + '),sep=' ~ '))  
  
spamModel <-  
glm(spamFormula,family=binomial(link='logit'),  
data=spamTrain)
```

logistic regression model

# Building and applying a logistic regression spam model

```
# print predicted result  
print(with(spamTest, table(y=spam, glmPred=pred>0.5)))  
sample <- spamTest[c(7,35,224,327),c('spam','pred')]  
print(sample)
```

*select rows*      *select columns*

equivalent to :

table (y = spamTest\$spam , glmPred = spamTest\$pred >0.5)

# Spam confusion matrix

```
cM <- table(truth=spamTest$spam, prediction=spamTest$pred>0.5)
print(cM)

##           prediction
##   truth      FALSE  TRUE
##   non-spam    264    14
##   spam        22    158
```

**Table 5.4 Standard two-by-two confusion matrix**

	<b>Prediction=NEGATIVE</b>	<b>Prediction=POSITIVE</b>
<b>Truth mark=NOT IN CATEGORY</b>	True negatives (TN) $cM[1, 1] = 264$	False positives (FP) $cM[1, 2] = 14$
<b>Truth mark=IN CATEGORY</b>	False negatives (FN) $cM[2, 1] = 22$	True positives (TP) $cM[2, 2] = 158$

# Changing a score to a classification

- Picking thresholds other than 0.5 can allow the data scientist to trade *precision* for *recall*

```
(table(truth=spamTest$spam, prediction=spamTest$pred>0.9))  
(table(truth=spamTest$spam, prediction=spamTest$pred>0.5))  
(table(truth=spamTest$spam, prediction=spamTest$pred>0.1))
```

# Akismet filter

TP	FN
287	1
FP	TN
17	13865

```
t <- as.table(matrix(data=c(288-1,17,1,13882-  
17),nrow=2,ncol=2))  
  
rownames(t) <- rownames(cM) confusion matrix  
colnames(t) <- colnames(cM)  
  
print(t)
```

# Precision and Recall

- Precision & Recall
  - Spam data?
  - Akismet filter? A spam filtering service
- Precision is a measure of confirmation
  - when the classifier indicates positive, how often it is in fact correct.
- Recall is a measure of utility
  - how much the classifier finds of what there actually is to find.
- spam filtering application, which one is important?
  - precision is emphasized over recall

$$\frac{TP}{TP+FP}$$

$$\frac{TP}{TP+FN}$$

# The F1 score

- Sørensen-Dice coefficient, Sørensen–Dice index, Sørensen index, Dice's coefficient
- Harmonic mean (調和平均數) of precision and recall
  - a useful combination of precision and recall.
- $$\frac{\frac{2}{precision} + \frac{1}{recall}}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Now, substitute the formulas for precision and recall:

$$F1 = 2 \times \frac{\left(\frac{TP}{TP+FP}\right) \times \left(\frac{TP}{TP+FN}\right)}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}}$$

To combine the fractions under a common denominator, we multiply the top and bottom of the F1 score formula by  $(TP + FP)(TP + FN)$ :

$$F1 = 2 \times \frac{TP \times TP}{TP(TP+FN) + TP(TP+FP)}$$

Simplify the denominator by factoring out TP:

$$F1 = 2 \times \frac{TP \times TP}{TP(2TP+FP+FN)}$$

Cancel one TP in the numerator and denominator:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

And that gives us the final F1 score formula in terms of the number of true positives (TP), false positives (FP), and false negatives (FN). The F1 score thus calculated is the harmonic mean of precision and recall and ranges between 0 and 1, where 1 is the best possible F1 score (perfect precision and recall), and 0 is the worst.

# Accuracy

- Accuracy
  - Spam data?
  - Akismet filter?
- Accuracy is an inappropriate measure for unbalanced classes
  - a situation where we have a rare event (say, severe complications during childbirth)
  - the null model—the rare event never happens—is very accurate
  - different costs of “type I” and “type II” errors



You

What is a null hypothesis?



ChatGPT

The null hypothesis is a statement used in statistics that proposes there is no significant effect or association between variables, or no difference among groups, within a population. It's a default position that indicates that any kind of prediction or claim is not present until proven otherwise.

In hypothesis testing, the null hypothesis, denoted as  $H_0$ , is tested against an alternative hypothesis ( $H_1$  or  $H_a$ ), which states that there is a significant effect, association, or difference.

For example:

- In a drug efficacy study, the null hypothesis might state that the new drug has no effect on patients, while the alternative hypothesis would contend that the drug does have an effect.
- In an A/B testing scenario, the null hypothesis could claim that there is no difference in conversion rate between Version A and Version B of a webpage, whereas the alternative hypothesis would suggest there is a difference.

The goal of hypothesis testing is to determine whether there is enough evidence from the sample data to reject the null hypothesis for the entire population. Rejecting the null hypothesis suggests that there is significant evidence for the alternative hypothesis.



# Type I and Type II errors

$H_0$

- null hypothesis = the message is not a spam
- a type I error : the rejection of a true null hypothesis (*a false alarm*)
- a type II error : the non-rejection of a false null hypothesis (*a miss*)
  - Type I error : Spam filtering or spam blocking techniques wrongly classify a legitimate email message as spam.
  - Type II error : Spam email is not detected as spam, but is classified as non-spam.

# Different costs of type I and type II errors?

- null hypothesis = the message is not a spam
- the null model: the rare event never happens => predict all mails are spam mails !
- Bad in Type I or Type II error?

	null model	
truth	FALSE	TRUE
non-spam	0	288
spam	0	13882

# spam filtering application, which one is important?

- Precision vs Recall
  - precision is emphasized over recall
- Type I vs Type II
  - Type I error is more critical than Type II error

# Sensitivity and Specificity

Reflects the test's ability to detect (sense)

the condition when it is present.

$$\frac{TP}{TP + FN}$$

- Sensitivity = the true positive rate = recall
  - Spam data?
  - Akismet filter?
- Specificity = the true negative rate
  - $TN / (TN + FP)$
  - Spam data?
  - Akismet filter?
- Why use those two terms?
  - null classifiers always return a zero score on either sensitivity or specificity.

$$\frac{TN}{TN + FP}$$

		P	N
truth	P	0	$\alpha$
	N	0	$\beta$

Null classifiers, sometimes referred to as baseline classifiers, make predictions based solely on the most frequent class without considering any of the input features. The term "null" reflects the fact that these classifiers do not use any information from the data other than class distribution.

Now, depending on how the null classifier is designed, it will always predict either the majority class or a constant value. If it always predicts the majority class, the resulting scores for sensitivity or specificity can be as follows:

assumption 1

### 1. Sensitivity (True Positive Rate): assumption 2

- If the majority class is the negative class (e.g., 'not spam'), the classifier will never predict a positive class (e.g., 'spam'). Therefore, there will be no true positives or false negatives, which makes the sensitivity zero.
- If the majority class is the positive class and the null classifier predicts always positive, then the sensitivity could be high because it catches all true positives but at the expense of also having many false positives.

### 2. Specificity (True Negative Rate):

- Conversely, if the null classifier always predicts the negative class, it will have no true negatives or false positives if the positive class is the majority, making specificity zero.
- If the negative class is the majority and the null classifier predicts always negative, then the specificity could be high because it correctly identifies all true negatives but it will miss all true positives, making the sensitivity zero.

$$\begin{aligned} * \text{sensitivity} &= \frac{TP}{TP+PN} \\ &= \frac{0}{0+\alpha} = 0 \end{aligned}$$
  

$$\begin{aligned} * \text{specificity} &= \frac{TN}{TN+FP} \\ &= \frac{\beta}{\beta+0} = 1 \end{aligned}$$

So, when a null classifier always predicts the majority class, it will have a sensitivity of zero if the majority class is negative (since it never predicts positives), and a specificity of zero if the majority class is positive (since it never predicts negatives).

# Common classification performance measures

**Table 5.5 Example classifier performance measures**

Measure	Formula	Email spam example	Akismet spam example
Accuracy	$(TP+TN) / (TP+FP+TN+FN)$	0.9214	0.9987
Precision	$TP / (TP+FP)$	0.9187	0.9999
Recall	$TP / (TP+FN)$	0.8778	0.9988
Sensitivity	$TP / (TP+FN)$	0.8778	0.9988
Specificity	$TN / (TN+FP)$	0.9496	0.9965

# Classifier performance measures business stories

Measure	Typical business need	Follow-up question
Accuracy	"We need most of our decisions to be correct."	"Can we tolerate being wrong 5% of the time? And do users see mistakes like spam marked as non-spam or non-spam marked as spam as being equivalent?" <b>type I error</b>
Precision	"Most of what we marked as spam had darn well better be spam."	"That would guarantee that most of what is in the spam folder is in fact spam, but it isn't the best way to measure what fraction of the user's legitimate email is lost. We could cheat on this goal by sending all our users a bunch of easy-to-identify spam that we correctly identify. Maybe we really want good specificity."
Recall	"We want to cut down on the amount of spam a user sees by a factor of 10 (eliminate 90% of the spam)."	"If 10% of the spam gets through, will the user see mostly non-spam mail or mostly spam? Will this result in a good user experience?"
Sensitivity	"We have to cut a lot of spam, otherwise the user won't see a benefit."	"If we cut spam down to 1% of what it is now, would that be a good user experience?"
Specificity	"We must be at least <i>three nines</i> on legitimate email; the user must see at least 99.9% of their non-spam email."	"Will the user tolerate missing 0.1% of their legitimate email, and should we keep a spam folder the user can look at?"

# Common classification performance measures

Sensitivity

“We have to cut a lot of spam, otherwise the user won’t see a benefit.”

“If we cut spam down to 1% of what it is now, would that be a good user experience?”

Specificity

“We must be at least *three nines* on legitimate email; the user must see at least 99.9% of their non-spam email.”

“Will the user tolerate missing 0.1% of their legitimate email, and should we keep a spam folder the user can look at?”

Recommend writing the business goals as maximizing sensitivity while maintaining a specificity of at least 0.999

 recommend writing the business goals as maximizing specificity while maintaining a sensitivity at least 0.999

# Covid-19 general sieve?

Based on 2020/04/28

# Prevalence $\pi$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

- Prevalence = positive / all
- Sensitivity & specificity
  - prevalence-independent test characteristics, as their values are intrinsic to the test and do not depend on the disease prevalence in the population of interest.
- Positive & negative predictive value
  - values influenced by the prevalence of disease in the population that is being tested.

$$PPV = \frac{TP}{TP + FP}, \quad NPV = \frac{TN}{TN + FN}$$

= Precision

A new test has been developed to diagnose HIV

{ +ve: positive  
-ve: negative

- Calculation of predictive value at 50% prevalence, gold standard
  - Test+ve : 1000
  - Test-ve : 1000
- Sensitivity =  $900/1000 = 90.0\%$   $TP / (TP + FN)$
- Specificity =  $950/1000 = 95.0\%$   $TN / (TN + FP)$
- Positive Predictive Value (PPV, Precision) =  $900/950 = 94.7\%$   $TP / (TP + FP)$
- Negative Predictive Value =  $950/1050 = 90.5\%$   $TN / (TN + FN)$

$$\frac{\text{positive}}{\text{all}} = \frac{900+100}{2000} = 50\%$$

true

New test	Gold standard	
	Test +ve	Test -ve
Test +ve	900 <b>TP</b>	50 <b>FP</b>
Test -ve	100 <b>FN</b>	950 <b>TN</b>
Total	1000	1000

pred

positive                      all

Prevalence is 1% (=10,000/10,00,000)

- Sensitivity =  $9000/10000 = 90.0\%$
  - Specificity =  $940500/990000 = 95.0\%$
  - PPV =  $9000/58500 = 15.4\%$
  - Negative Predictive Value =  $940500/941500 = 99.9\%$
- π-independent*                      *π-dependent*

New test	Gold standard	
	Test +ve	Test -ve
Test +ve	9000	49500
Test -ve	1000	940500
Total	10,000	9,90,000

# Type I ( $\alpha$ ), II ( $\beta$ ) error

- Null hypothesis: a patient is non-carrier
- Type I error,  $\alpha$ : the rejection of a true null hypothesis (false alarm)
  - non-carrier => positive
- Type II error,  $\beta$ : the non-rejection of a false null hypothesis (miss)
  - Carrier => negative

New test	Gold standard	
	Test +ve	Test -ve
Test +ve	9000	49500
Test -ve	1000	940500
Total	10,000	9,90,000

		受檢者		列的邊際機率
		不帶原 (non-carrier)	帶原 (true-carrier)	
採檢結果	陽性 (positive)	偽陽性 $\alpha(1 - \pi)$	真陽性 $(1 - \beta)\pi$	$\alpha + (1 - \alpha - \beta)\pi$
	陰性 (negative)	真陰性 $(1 - \alpha)(1 - \pi)$	偽陰性 $\beta\pi$	$(1 - \alpha) + (1 - \alpha - \beta)\pi$
行的邊際機率 (隨機採檢人士帶原的先驗 機率)		$1 - \pi$	$\pi$	1

# $\pi$ vs Sensitivity, Specificity vs PPV

- prevalence-independent

- Sensitivity =  $\frac{(1-\beta)\pi}{(1-\beta)\pi+\beta\pi} = (1 - \beta) \frac{TP}{TP+FN}$
- Specificity =  $\frac{(1-\alpha)(1-\pi)}{(1-\alpha)(1-\pi)+\alpha(1-\pi)} = (1 - \alpha) \frac{TN}{TN+FP}$

- prevalence-dependent

- PPV =  $\frac{(1-\beta)\pi}{\alpha+(1-\alpha-\beta)\pi} \frac{TP}{TP+FP}$

		受檢者		列的邊際機率
採檢結果	陽性 (positive)	偽陽性 $FP$ $\alpha(1 - \pi)$	真陽性 $TP$ $(1 - \beta)\pi$	$\alpha + (1 - \alpha - \beta)\pi$
	陰性 (negative)	真陰性 $TN$ $(1 - \alpha)(1 - \pi)$	偽陰性 $FN$ $\beta\pi$	$(1 - \alpha) + (1 - \alpha - \beta)\pi$
行的邊際機率 (隨機採檢人士帶原的先驗 機率)		$1 - \pi$	$\pi$	1

- Specificity = 真陰性的機率 = 信心水平 =  $1 - \alpha$
- Sensitivity = 真陽性的機率 = 檢定強度 =  $1 - \beta$

		虛無假設( $H_0$ ) v. 研究假設( $H_A$ )	
		虛無假設為真 ( $H_0$ True)	研究假設為真 ( $H_A$ True)
採檢結果	拒絕虛無假設 (positive)	型一錯誤 (size of test= $\alpha$ )	檢定強度 (power of test= $1 - \beta$ )
	無法拒絕虛無假設 (negative)	信心水平 ( $1 - \alpha$ )	型二錯誤 ( $\beta$ )

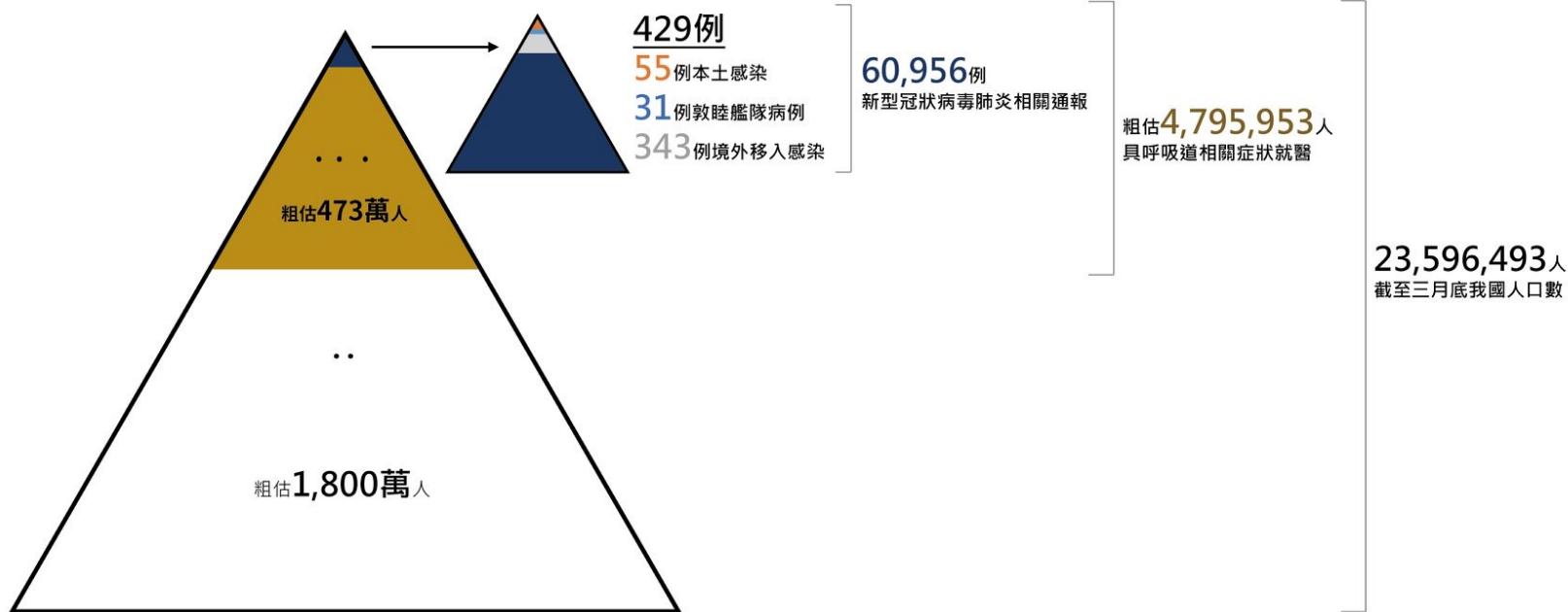
# 從 $\pi$ 看各種檢測工具的精密性

$$\frac{TP}{TP+FP} = PPV = Precision$$

- $PPV=0.4865$  是什麼概念？
  - 抗體檢測為陽性者，48.65% 是真帶原者，偽陽性 > 50%
- 如果抗體施測單位所獲得的試劑品質不高怎麼辦？

	specificity 特異性 (1- $\alpha$ )	sensitivity 敏感性 (1- $\beta$ )	精密性 (PPV)	
			$\pi=0.05$	$\pi=0.52$
台灣病毒PCR	0.9999	0.95	0.9980	0.9999
台灣病毒快篩	0.99	0.75	0.7979	0.9878
CDC抗體檢測	0.95	0.90	0.4865	0.9512
南加大抗體檢測	0.995	0.827	0.8970	0.9945
CDC抗體二採陽	0.9975	0.81	0.9446	0.9972

# 國內疫情現況



# 名詞解釋

Sensitivity

**敏感性**：有病的人被檢出陽性的比例

Specificity

**特異性**：沒病的人被檢出陰性的比例

		確診	
		+	-
檢驗	+	a TP 陽性個案	c FP 偽陽性
	-	b FN 偽陰性	d TN 真正陰性

$$\text{敏感性} = \frac{\text{檢驗陽性}(a)}{\text{有病的人}(a+b)} = \frac{TP}{TP+FN}$$

$$\text{特異性} = \frac{\text{檢驗陰性}(d)}{\text{沒病的人}(c+d)} = \frac{TN}{TN+FP}$$

# 名詞解釋

1 盛行率：全國人口中感染者的比例。

快篩：1小時內完成的檢驗方法。

普篩：不依據臨床症狀判斷，全面檢驗。

檢驗方法	檢驗時間	時機	Sensitivity	Specificity
Real-time RT-PCR	2-4hrs	發病早期 防疫圍堵與阻斷傳播	> 95%	> 95%
抗原(快篩)	15min	發病早期 高盛行區	70%	>95%
抗體(快篩)	15min	發病7天後 了解是否感染	75%	>95%

# 普篩迷思解析-以快篩為例

假設快篩檢驗敏感性75%、特異性99%，在台灣的盛行率下，每檢驗1萬人，預計會有114人出現陽性，其中100人是偽陽性。



# 無症狀人數為例

## 以我國無症狀人數為例(極大值)

PCR		+ 武漢肺炎		- 武漢肺炎		武漢肺炎 盛行率 $= \frac{\text{確診個案}}{\text{總檢驗數}} = \frac{55}{31,156} = \frac{18}{10,000}$ <small>資料來源：2020/4/27</small>
敏感性	特異性	陽性個案	偽陽性	真正陰性	總數	
95%	99.99%	30,780	1,797	17,965,803	32,577	
		1,620	17,967,423			
32,400	17,967,600	18,000,000	(單位：人)			

快篩		+ 武漢肺炎		- 武漢肺炎		
敏感性	特異性	陽性個案	偽陽性	真正陰性	總數	
75%	99%	24,300	179,676	17,787,924	203,976	
		8,100	17,796,024			
32,400	17,967,600	18,000,000	(單位：人)			

## 以我國無症狀人數為例(合理值)

PCR		+ 武漢肺炎		- 武漢肺炎		武漢肺炎 盛行率 $= \frac{\text{確診個案}}{\text{總檢驗數}} = \frac{10}{18,000,000} = \frac{56}{100,000,000}$ <small>資料來源：2020/4/27</small>
敏感性	特異性	陽性個案	偽陽性	真正陰性	總數	
95%	99.99%	9	1,800	17,998,190	1,809	
		1	17,998,191	17,998,191		
10	17,999,990	18,000,000	(單位：人)			

快篩		+ 武漢肺炎		- 武漢肺炎		
敏感性	特異性	陽性個案	偽陽性	真正陰性	總數	
75%	99%	8	180,000	17,819,990	180,008	
		2	17,819,992	17,819,992		
10	17,999,990	18,000,000	(單位：人)			

9

10

# 呼吸道就醫人數

## 以呼吸道症狀就醫人數為例(極大值)

**PCR**

敏感性 95%  
特異性 99.99%

+ 武漢肺炎 -		武漢肺炎 盛行率 $= \frac{\text{確診個案}}{\text{總檢驗數}} = \frac{55}{31,156} = \frac{18}{10,000}$
陽性個案	偽陽性	
8,208	479	
432	4,790,881	
	真正陰性	
8,640	4,791,360	8,687
		4,791,313
		4,800,000 (單位：人)

**快篩**

敏感性 75%  
特異性 99%

+ 武漢肺炎 -		武漢肺炎 盛行率 $= \frac{\text{確診個案}}{\text{總檢驗數}} = \frac{55}{31,156} = \frac{18}{10,000}$
陽性個案	偽陽性	
6,480	47,914	
2,160	4,743,446	
	真正陰性	
8,640	4,791,360	54,394
		4,745,606
		4,800,000 (單位：人)

## 以呼吸道症狀就醫人數為例(合理值)

**PCR**

敏感性 95%  
特異性 99.99%

+ 武漢肺炎 -

$$\text{武漢肺炎}\text{盛行率} = \frac{\text{確診個案}}{\text{總檢驗數}} = \frac{75}{4,800,000} = \frac{16}{1,000,000}$$

+ 陽性個案	480	551
- 偽陰性	4,799,445	4,799,449
	75	4,799,925
		4,800,000 (單位：人)

**快篩**

敏感性 75%  
特異性 99%

+ 武漢肺炎 -

+ 陽性個案	47,999	48,056
- 偽陰性	4,751,926	4,751,945
	75	4,799,925
		4,800,000 (單位：人)

# PCR 及快篩的精密性

$$\frac{TP}{TP+FP} = PPV = \text{Precision}$$

(π-prevalent)

檢測工具 \ 普篩對象	呼吸道症狀就醫人口 ( 4800000 )		無症狀人口 ( 18000000 )	
	盛行率極大值： ( π=0.0018 )	盛行率合理值： ( π=0.000016 )	盛行率極大值： ( π=0.0018 )	盛行率合理值： ( π=0.00000056 )
PCR 之精密性 ( 真陽性/採檢陽性 )	0.9448 ( 8208/8687 )	0.1319 ( 71/551 )	0.9448 ( 30780/32577 )	0.0050 ( 9/1809 )
快篩之精密性 ( 真陽性/採檢陽性 )	0.1191 ( 6480/54394 )	0.0012 ( 56/48056 )	0.1191 ( 24300/203976 )	0.0000 ( 8/180008 )

# Multi-class classification

# Measurement by *sklearn*

- `sklearn.metrics.classification_report`

```
>>> y_pred = [1, 1, 0]
>>> y_true = [1, 1, 1]
>>> print(classification_report(y_true, y_pred, labels=[1, 2, 3]))
          precision    recall  f1-score   support

           1       1.00      0.67      0.80       3
           2       0.00      0.00      0.00       0
           3       0.00      0.00      0.00       0

     micro avg       1.00      0.67      0.80       3
     macro avg       0.33      0.22      0.27       3
  weighted avg       1.00      0.67      0.80       3
```

# Convert confusion matrix to binary confusion matrix by class

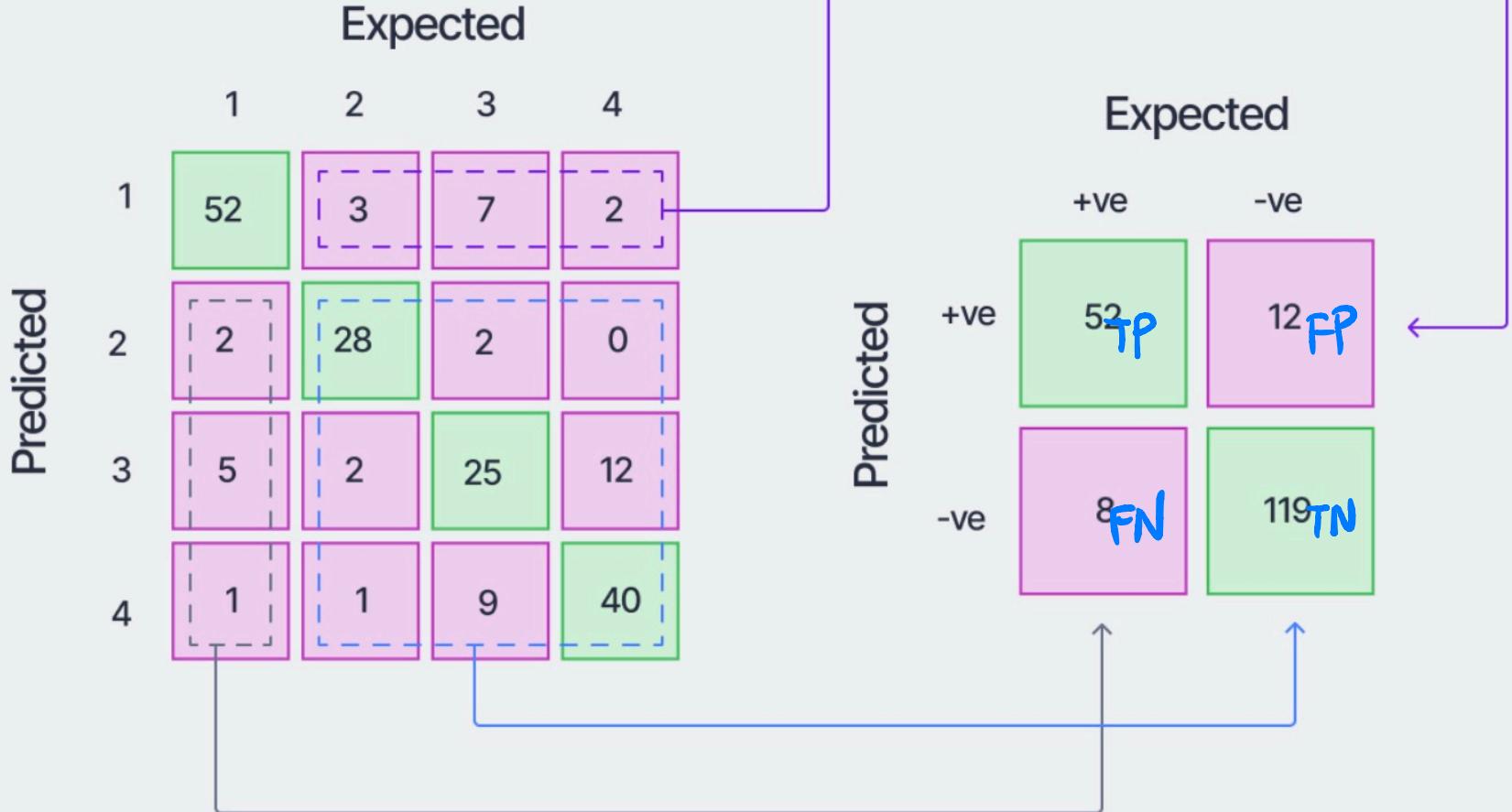
ex. class 1 *(class of interest)*  
*(real)*

	1	2	3	4	5
1	TP				
2		FN			
3			TN		
4	FP			FN	
5					TP

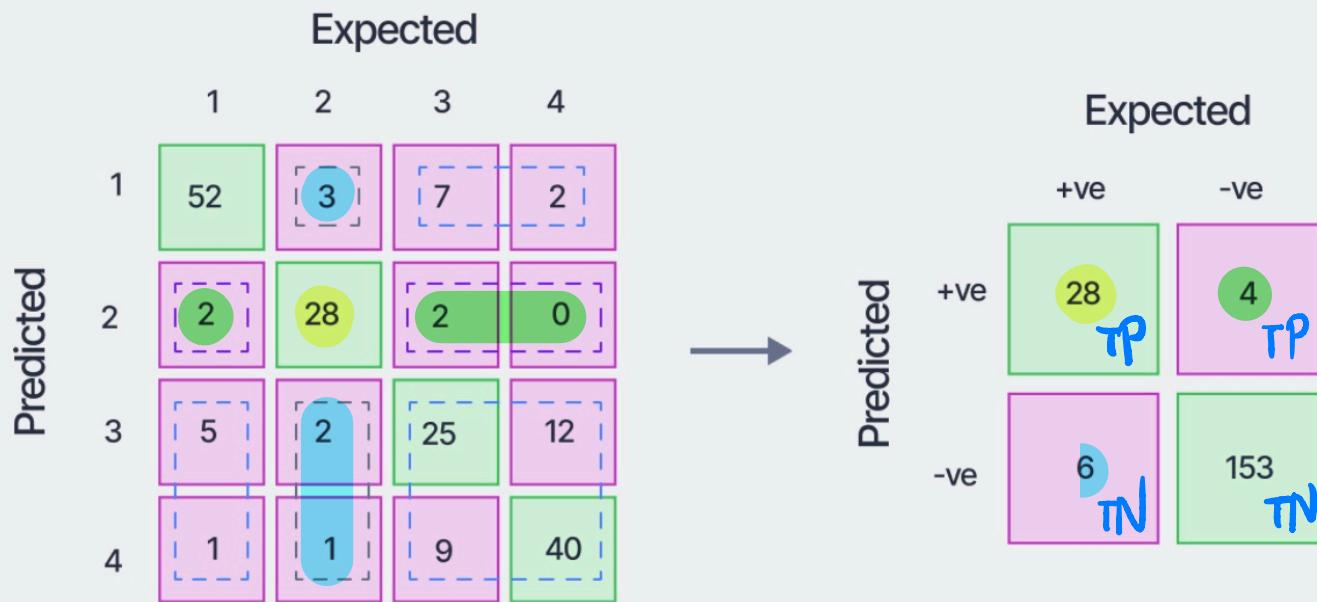
*(pred)*



	1	0
1	TP	FN
0	FP	TN



Similarly, for class-2, the converted one-vs-all confusion matrix will look like the following:



Then calculate *Sen*, *Spe*, *Acc* by class

ex. class 1

	1	0
1	TP	FN
0	FP	TN

$$\text{Sensitivity(Sen)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity(Spe)} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{Accuracy(Acc)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

# Specificity becomes inflated

- there are 10 classes with 10 observations each
- Sensitive =  $8/18 = 44.4\%$   $TP / (TP + FN)$
- Specificity =  $80/90 = 88.9\%$   $TN / (TN + FP)$

Prediction/Reference	Class 1	Other Class
Class 1	8 $TP$	10 $FN$
Other Class	2 $FP$	80 $TN$

# The micro average

- it pools the performance over the smallest possible unit
- The micro-average is not sensitive to the predictive performance for individual classes. => The micro-average can be particularly misleading when the class distribution is imbalanced.

$$Precision_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i}$$

$$Recall_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FN_i}$$

$$F1_{micro} = 2 \frac{P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}}$$

# The macro average

- over the performance for individual classes rather than observations
- If  $F1_{macro}$  has a large value, this indicates that a classifier performs well for each individual class. The macro-average is therefore more suitable for data with an imbalanced class distribution.

$$P_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{TP_i}{TP_i + FP_i} = \frac{\sum_{i=1}^{|G|} P_i}{|G|}$$

$$R_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} \frac{TP_i}{TP_i + FN_i} = \frac{\sum_{i=1}^{|G|} R_i}{|G|}$$

$$F1_{macro} = 2 \frac{P_{macro} \cdot R_{macro}}{P_{macro} + R_{macro}}$$

macro average + weight

## Weighted F1-Score

The weighted-average scores take a sample-weighted mean of the class-wise scores obtained. So, the weighted scores obtained are:

$$\text{Weighted Precision} = \frac{81.25 \times 60 + 87.50 \times 34 + 56.82 \times 43 + 78.43 \times 54}{64 + 32 + 44 + 51} \% = 76.07\%$$

$$\text{Weighted Recall} = \frac{86.67 \times 60 + 82.35 \times 34 + 58.14 \times 43 + 74.07 \times 54}{64 + 32 + 44 + 51} \% = 75.92\%$$

$$\text{Weighted F1 - Score} = \frac{83.87 \times 60 + 84.85 \times 34 + 57.47 \times 43 + 76.19 \times 54}{64 + 32 + 44 + 51} \% = 75.93\%$$

# Evaluating scoring models

# Evaluating scoring models

- *residuals* or *difference* between our predictions and actual outcomes
- residual.R

```
d <- data.frame(y=(1:10)^2,x=1:10) y=x2
model <- lm(y~x,data=d) fit a linear regression model
d$prediction <- predict(model,newdata=d)
library('ggplot2')
ggplot(data=d) + geom_point(aes(x=x,y=y)) +
  geom_line(aes(x=x,y=prediction),color='blue') +
  geom_segment(aes(x=x,y=prediction,yend=y,xend=x)) +
  scale_y_continuous('')
```

# Scoring residuals

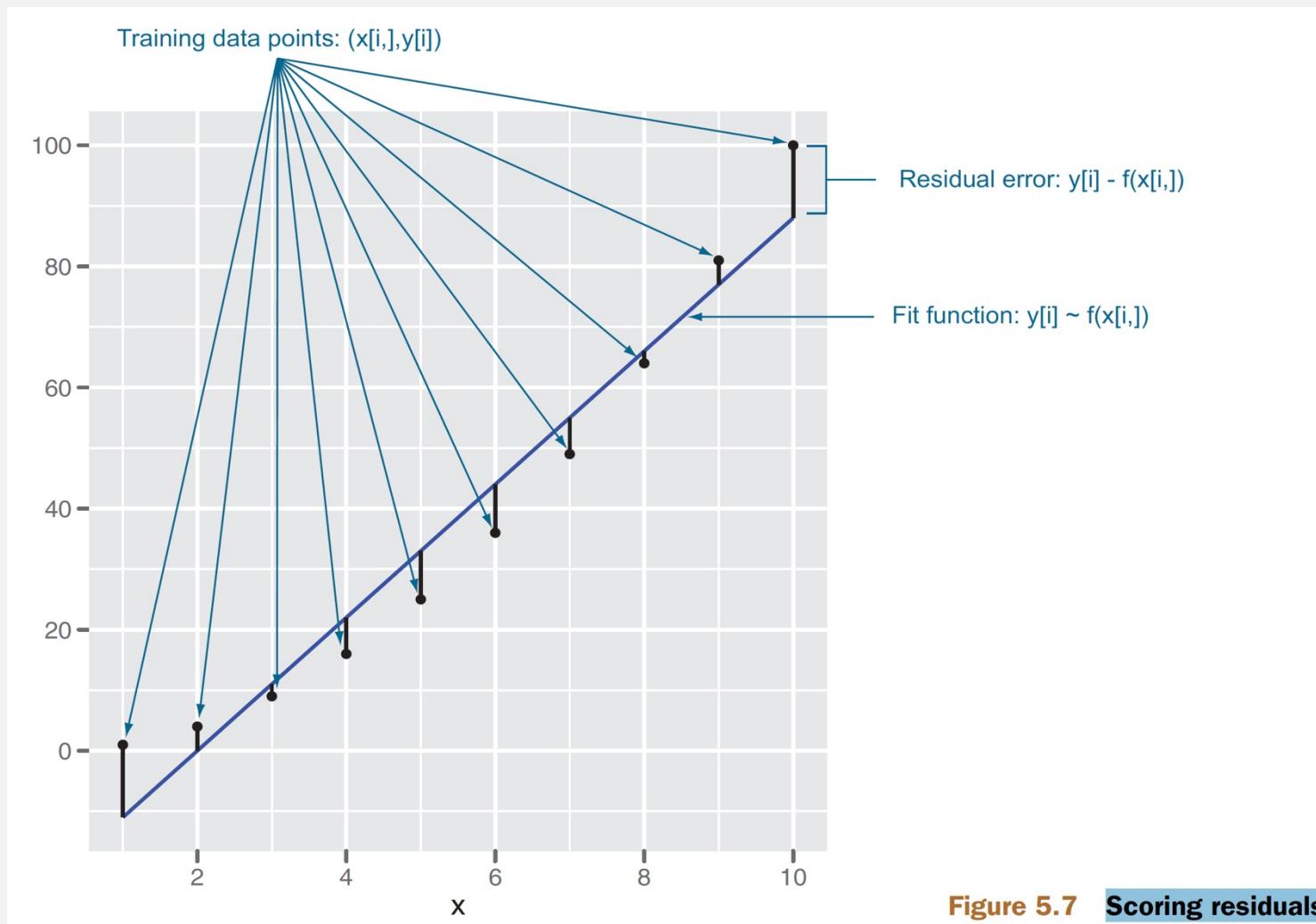


Figure 5.7 Scoring residuals

# Absolute error

- residual.R

- absolute error

```
sum(abs(d$prediction-d$y))
```

- mean absolute error

```
sum(abs(d$prediction-d$y))/length(d$y)
```

- relative absolute error

```
sum(abs(d$prediction-d$y))/sum(abs(d$y))
```

# Absolute error

- three advertisement purchases returning \$0, \$0, and \$25 respectively =>picking a single summary value not too far from the original three prices
  - 0 or 8.33 is better?
- Not advisable to make these measures the project goal

# Mean Squared Error

- $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$

# Root Mean Square Error (RMSE)

- the square root of the average square of the difference between our prediction and actual values

```
sqrt (mean ( (prediction-actualValues) ^2) )
```

# R-Squared, $R^2$ , the coefficient of determination

- measured relative to a null model of just using the average  $y$  as a prediction
- $1 - \text{“how much unexplained variance your model leaves”}$

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y}_i)^2}$$

 Model error  
 Variance in the dependent variable

- can be thought of as a normalized version of  $RMSE$

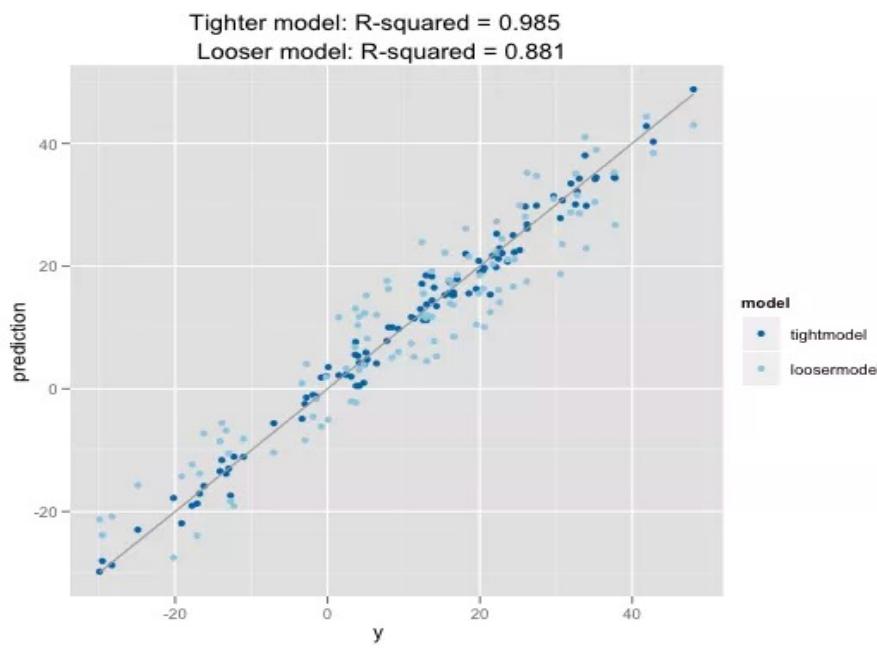
$$R^2 = \frac{\text{Var(mean)} - \text{Var(line)}}{\text{Var(mean)}} = 1 - \frac{\text{Var(line)}}{\text{Var(mean)}} = 1 - \frac{\sum(y_i - f_i)^2 / (N-1)}{\sum(y_i - \bar{y}_i)^2 / (N-1)}$$

# How to check if a model fit is good?

- The  $R^2$  statistic has become the almost universally standard measure for model fit in linear models.
- It is the ratio of error in a model over the total variance in the dependent variable.
- 1 or 0 better?      **1**
  - Hence the lower the error, the higher the  $R^2$  value.
- A good statement of a  $R^2$  business goal = We want the model to explain at least 70% of variation in account value.

$R^2$

- close to 1: then the model's predictions mirror true outcome
- is low: then either the model does not mirror true outcome, or it only mirrors it loosely



# Correlation

- at least three calculations
  - the Pearson coefficient : checks for linear relations
  - the Spearman coefficient : checks for rank or ordered relations
  - the Kendall coefficient: checks for degree of voting agreement.

# Correlation

- Each of these coefficients performs a progressively more drastic transform than the one before and has well-known direct significance tests
- **!!!DON'T USE CORRELATION TO EVALUATE MODEL QUALITY IN PRODUCTION**
  - correlation ignores shifts and scaling factors

# Correlation ignores shifts and scaling factors

```
a <- c(0.1, 0.2)
```

- ignores shifts
  - `b <- c(0.3, 0.4)`
  - `cor(a, a)`
  - `cor(a, b)`
- ignores scaling
  - `c <- c(0.2, 0.4)`
  - `d <- c(0.4, 0.8)`
  - `cor(a, c)`
  - `cor(a, d)`

## *R-squared = correlation<sup>2</sup>*

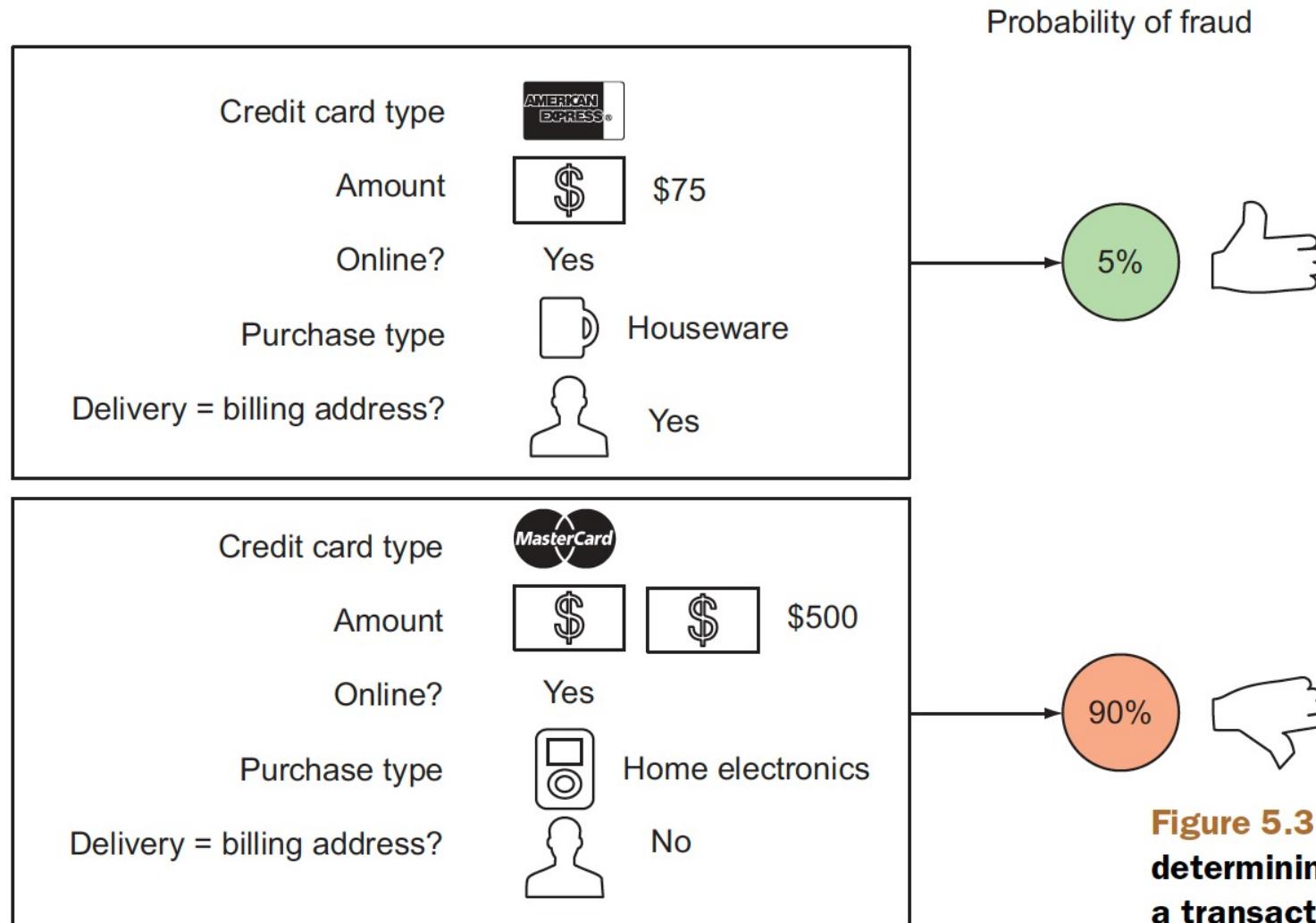
- Under general conditions, as Wikipedia says,  $R^2$  is also the square of the correlation between the actual and predicted outcomes.
- How to prove?  $\Rightarrow$  Note 3

# Correlation = 0.8, is it good?

- A good  $R$ -squared business goal would be “We want the model to explain 70% of account value.”
- Correlation = 0.8
  - $R^2 = 0.8 * 0.8 = 0.64 < 70\%$
  - Unexplained variance =  $1 - 0.64 = 0.36$

# Evaluating probability models





**Figure 5.3 Notional example of determining the probability that a transaction is fraudulent**

# Evaluating probability models

- Making a double density plot

```
ggplot (data=spamTest) +  
  geom_density(aes(x=pred, c  
olor=spam, linetype=spam))
```

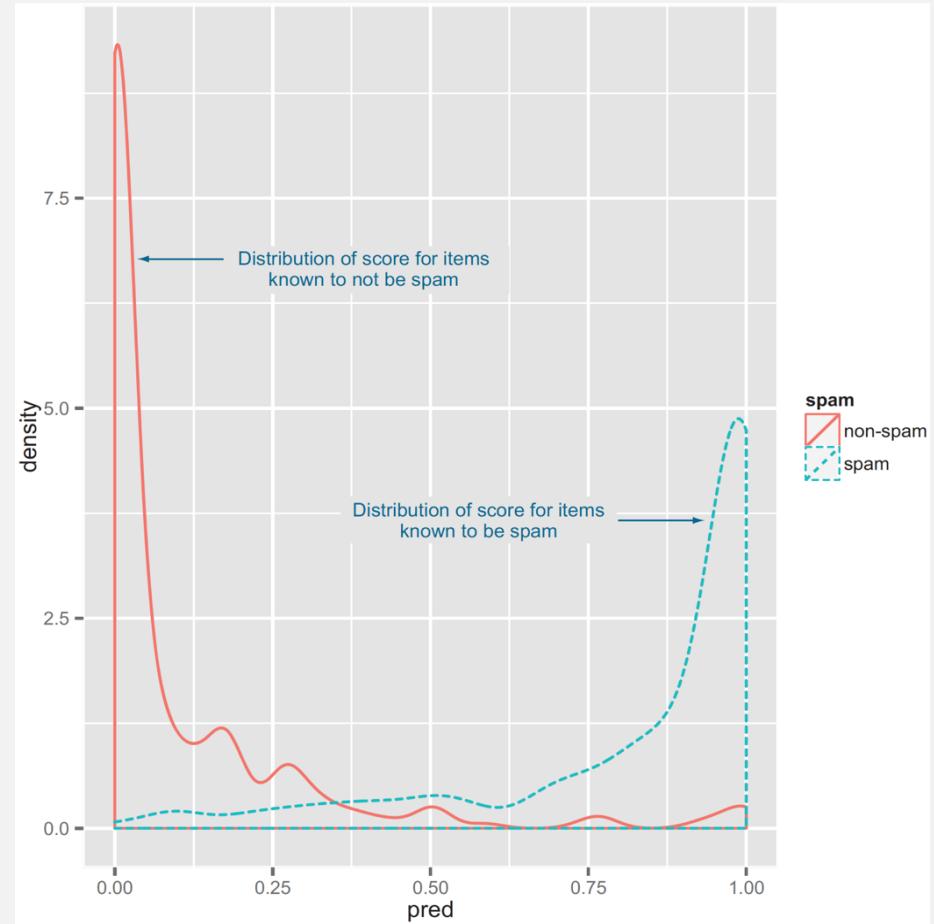


Figure 5.8 Distribution of score broken up by known classes

# The Receiver Operating Characteristic Curve

- ROC curve
  - by picking a different score threshold between positive and negative determination
  - represents every possible trade-off between sensitivity and specificity
- AUC or area under the curve
  - 1.0 = perfect classifiers
  - 0.5 = do no better than random guesses

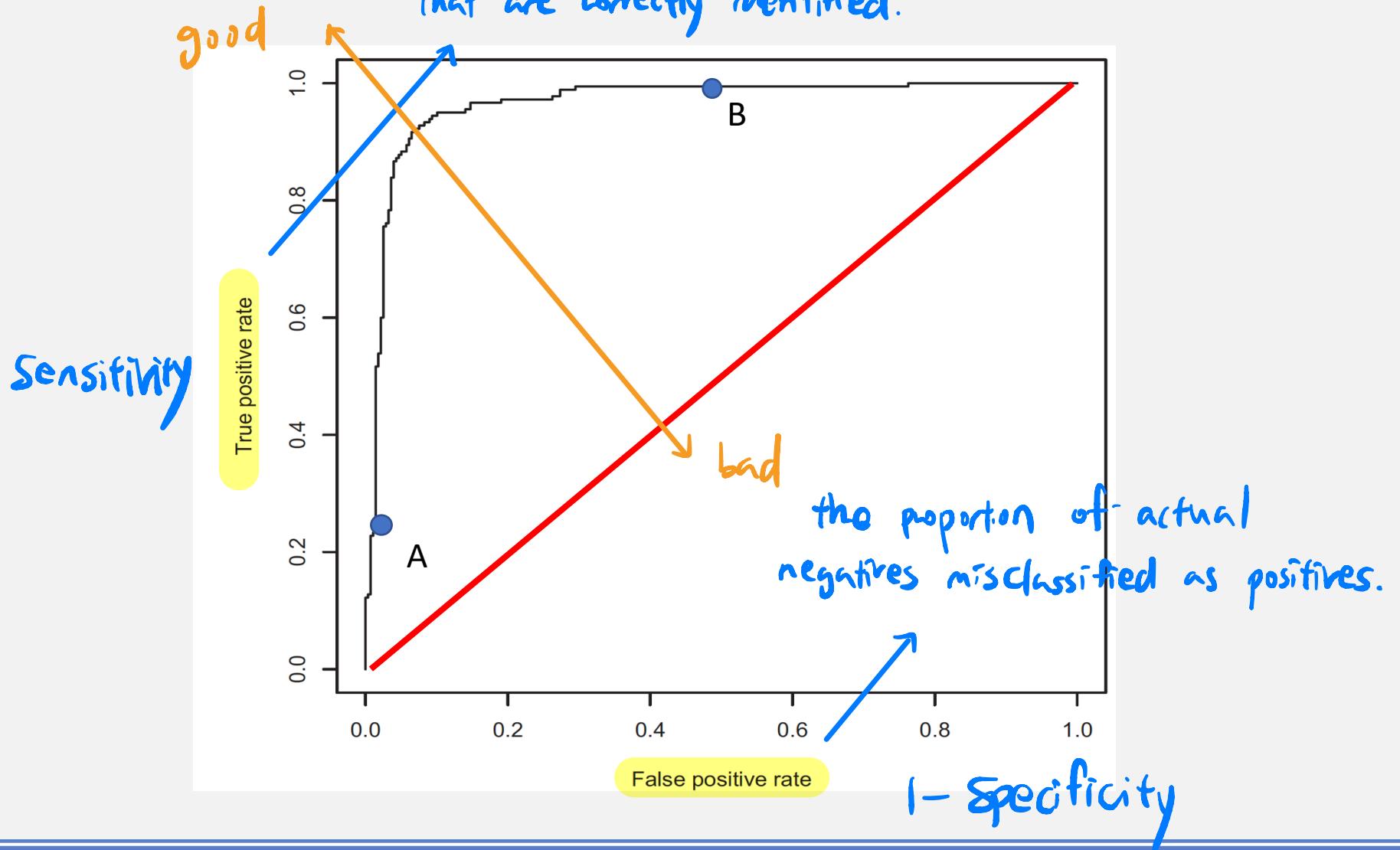
The term "ROC curve" stands for "Receiver Operating Characteristic" curve. The name originates from signal detection theory developed during World War II, when operators used radar to detect enemy objects among various signals, distinguishing between true signals (targets) and false signals (noise). The ROC curve was a tool used to assess how well a radar receiver could detect these signals under various conditions. The technique was later adopted in various fields for measuring the performance of diagnostic systems and classifiers, particularly in medical diagnostics and machine learning.

# The Receiver Operating Characteristic Curve

- spamExam.R

```
library('ROCR')
eval <- prediction(spamTest$pred, spamTest$spam)
plot(performance(eval, "tpr", "fpr"))
print(attributes(performance(eval, 'auc'))$y.values[[1]])
```

# ROC curve



# More on ROC/AUC

- The ROC curve is a useful tool, but you have to use it for appropriate tasks.
- The ROC curve is useful tool designing a classifier from a scoring function (though I prefer the “double hump graph”).
- Hand, D.J. [Measuring classifier performance: a coherent alternative to the area under the ROC curve](#). Mach Learn 77, 103–123 (2009).

# Log Likelihood

likelihood :  $\prod p(x_i)$



log likelihood :  $\sum \log(p(x_i))$

- the logarithm of the product of the probability the model assigned to each example
  - a spam with an estimated spam 0.9 =>  $\log(0.9)$
  - a non-spam email an estimated spam 0.1 =>  $\log(0.1)$

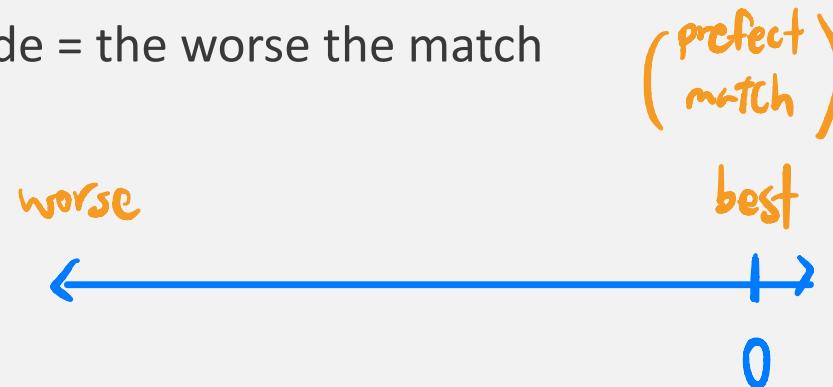
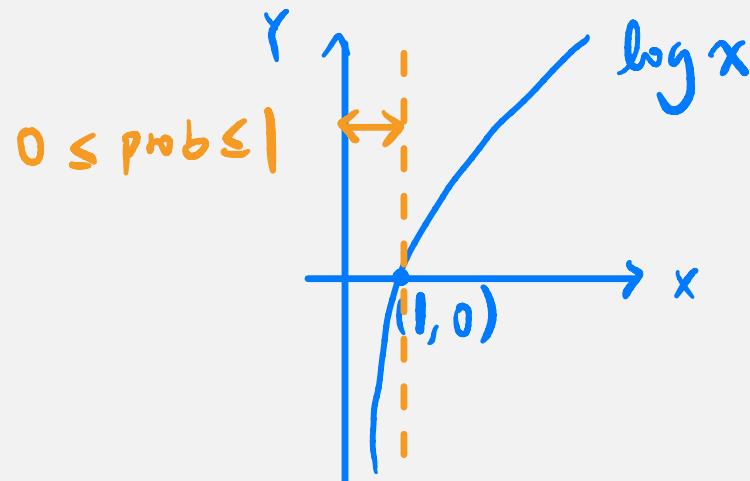
$y = \begin{cases} 1 & \text{if spam} \\ 0 & \text{if not spam} \end{cases}$			
match	spam	and $P(\text{spam}) = 0.98$	contribution : $1 * \log(0.98) = -0.02$
match	not spam	and $P(\text{spam}) = 0.02$	contribution : $(1-0) * \log(1-0.02) = -0.02$
mismatch	spam	and $P(\text{spam}) = 0.02$	contribution : $1 * \log(0.02) = -3.9$
mismatch	not spam	and $P(\text{spam}) = 0.98$	contribution : $(1-0) * \log(1-0.98) = -3.9$

spam :  $\log(P(\text{spam}))$   
 $(y=1)$

not spam :  $\log(1 - P(\text{spam}))$   
 $P(\text{not spam})$   
 $(y=0)$

# Log Likelihood

- a non-positive number
  - 0 = a perfect match
    - scores all the spam as being spam with a probability of 1
    - all the nonspam as having a probability 0 of being spam
  - The larger the magnitude = the worse the match



# Log Likelihood

- spamExam.R

```
sum(ifelse(spamTest$spam=='spam',  
          log(spamTest$pred),  
          log(1-spamTest$pred)))  
sum(ifelse(spamTest$spam=='spam',  
          log(spamTest$pred),  
          log(1-spamTest$pred)))/dim(spamTest) [[1]]
```

average

# Computing the null model's log likelihood

spamExam.R

```
pNull <-  
sum(ifelse(spamTest$spam=='spam',1,0))/dim(spamTes  
t) [[1]] // pNull is the proportion of the emails that are spam  
sum(ifelse(spamTest$spam=='spam',1,0))*log(pNull)  
+sum(ifelse(spamTest$spam=='spam',0,1))*log(1-  
pNull)
```

- -134.9478 better than -306.8952.

# Deviance

(fits the model perfectly)  
the most complex model  
for a given dataset



- $-2*(\text{logLikelihood} - S)$ 
  - where  $S$  is a technical constant called “the log likelihood of the saturated model”. (the max likelihood that could be achieved given the data)
- Deviance is unnormalized, so you should only compare deviances that are computed over the same dataset.
- The lower the deviance, the better the model.

It's important to note that while a lower deviance indicates a better fit to the data, it does not automatically mean the model is the best choice. Overfitting can occur when the model becomes too complex, capturing the noise along with the signal in the data. This is why additional criteria, such as Akaike's Information Criterion (AIC) or Bayesian Information Criterion (BIC), which penalize model complexity, are often used in conjunction with deviance for model selection.

# pseudo R-squared

→ the saturated model is a perfect match

- With  $S = 0$ , the deviance can be used to calculate a *pseudo R-squared* (see <http://mng.bz/j338>)
  - $1 - \text{deviance(model)}/\text{deviance(null model)}$
  - What is the psedudo R-square of the spam model?
- The pseudo  $R$ -squared is normalized, so in principle you can compare pseudo  $R$ -squareds even when they were computed over different test sets.
- prefer models with larger pseudo  $R$ -squareds

$$\begin{aligned} & \text{maximize} \left[ 1 - \frac{\text{deviance(model)}}{\text{deviance(null model)}} \right] \Rightarrow \text{minimize} \left[ \frac{\text{deviance(model)}}{\text{deviance(null model)}} \right] \\ & \text{constant} \\ & \Rightarrow \text{minimize} \left[ \text{deviance (model)} \right] \end{aligned}$$

# Akaike information criterion (AIC)

- deviance +  $2 * \text{numberOfParameters}$
- deviance penalized for model complexity (more parameters)
- useful for
  - comparing models with different measures of complexity
  - modeling variables with differing numbers of levels
- prefer the model with the smaller AIC

# Evaluating ranking models

# Evaluating ranking models

- Ranking models are models that, given a set of examples, sort the rows or assign ranks to the rows.
- measured by Spearman's rank correlation coefficient (treating assigned rank as a numeric score)

# Evaluating clustering models

# Notional example of clustering your customers by purchase pattern and purchase amount

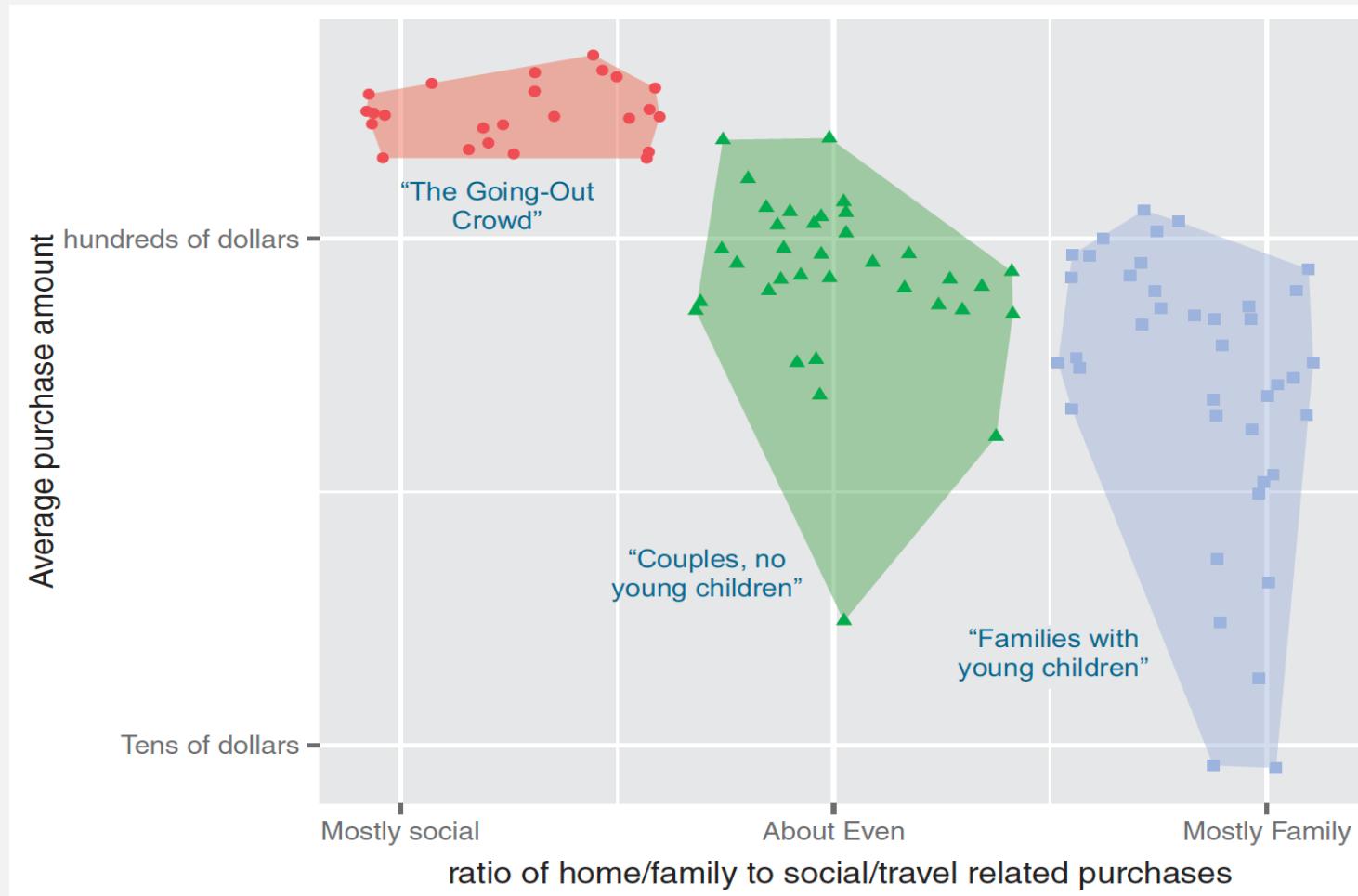


Figure 5.4, Practical Data Science with R by Nina Zumel, John Mount

# Evaluating clustering models

- clusterModel.R
- Clustering random data in the plane

```
set.seed(32297)
d <- data.frame(x=runif(100), y=runif(100)) // uniform
clus <- kmeans(d, centers=5) // 5 clusters distribution
d$cluster <- clus$cluster
```

- Calculating the size of each cluster

```
table(d$cluster)
```

# Evaluating clustering models

- clusterModel.R
- Plotting our clusters

```
library('ggplot2'); library('grDevices')
h <- do.call(rbind,
  lapply(unique(clus$cluster),
    function(c) { f <- subset(d, cluster==c); f[chull(f),]}))
```

apply a function to each unique cluster identifier

```
ggplot() +
  geom_text(data=d, aes(label=cluster, x=x, y=y,
    color=cluster), size=3) +
  geom_polygon(data=h, aes(x=x, y=y, group=cluster, fill=as.factor(cluster)),
    alpha=0.4, linetype=0) +
  theme(legend.position = "none")
```

f contains only the data belonging to this cluster

chull(f)  
use (the points that form a convex hull)  
to (subset f) f[,]

plot clusters

# Clustering example

- hair clusters : clusters with very few points
- waste clusters : clusters with a very large number of points

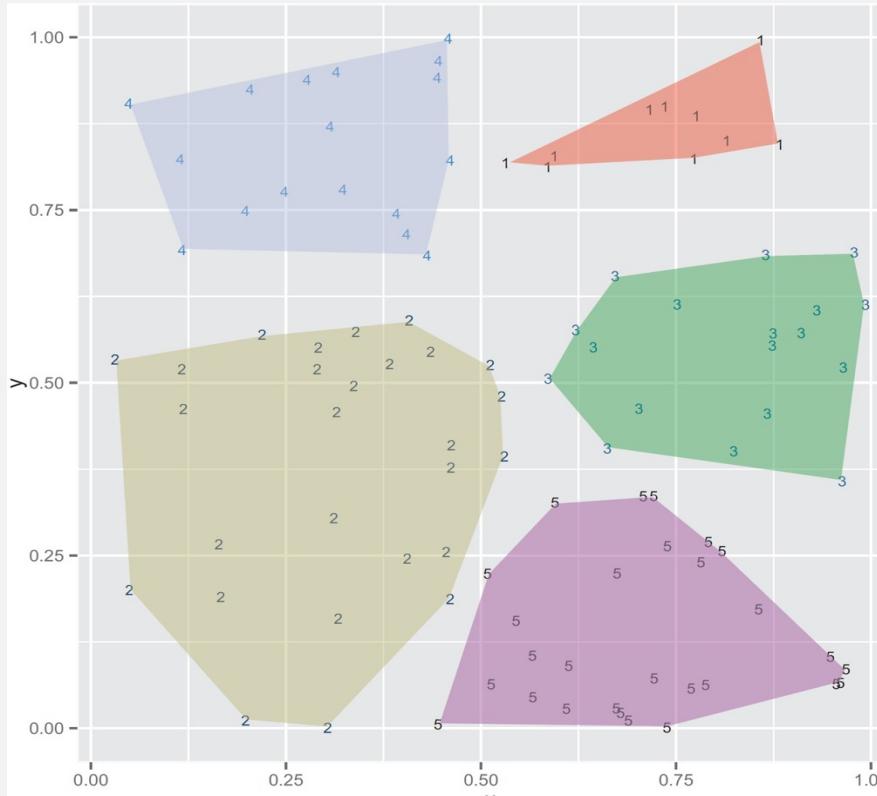


Figure 5.10, *Practical Data Science with R* by Nina Zumel, John Mount

# Internal / External validations

- Internal
  - use only the data and the clustering results to measure the goodness of fit of the clusters
- External
  - compare the clustering results with a predefined of the data (i.e., class labels)

Internal  
Without labels

# Intra-cluster distances v.s. Cross-cluster distances

- Calculating the typical distance between items in every pair of clusters

```
library('reshape2')
n <- dim(d) [[1]]
pairs <- data.frame(
  ca = as.vector(outer(1:n,1:n,function(a,b) d[a,'cluster'])),
  cb = as.vector(outer(1:n,1:n,function(a,b) d[b,'cluster'])),
  dist = as.vector(outer(1:n,1:n,function(a,b)
    sqrt((d[a,'x']-d[b,'x'])^2 + (d[a,'y']-d[b,'y'])^2)))
)
dcast(pairs,ca~cb,value.var='dist',mean)
```

- intra-cluster distances (the diagonal elements of the table) < inter-cluster distances (the off-diagonal elements of the table)

	1	2	3
1	0.71	6.01	10.54
2	6.01	1.12	4.38
3	10.54	4.38	1.41

# Internal metrics

- maximize the inter-cluster distance
- minimize the intra-cluster distance
- metrics
  - Calinski-Harabasz index, 1974
    - details @ topic08\_unsupervised.pptx
  - Silhouette, 1987

# Silhouette

- Rousseeuw,P.J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**, 53–65.

- $s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$ 
  - $a(i) = \frac{1}{|C_I|-1} \sum_{j \in C_I, i \neq j} d(i, j)$
  - $b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j)$

$a(i)$ : the average distance from point  $i$  to all other points in the same cluster

$b(i)$ : the lowest average distance from point  $i$  to points in a different cluster, minimized over all other clusters

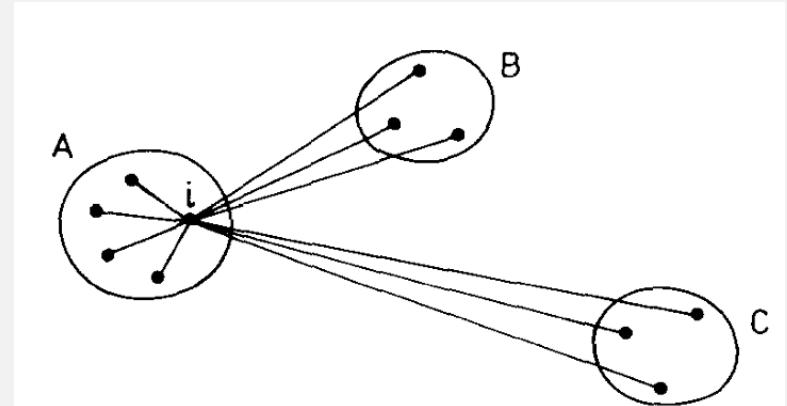


Fig 1 from Rousseeuw,P.J. (1987) *J. Comput. Appl. Math.*

# The range of Silhouette

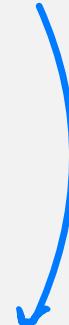
- 1: appropriately clustered(good)  $a(i) \ll b(i)$
- 0: overlapping clusters  $a(i) = b(i)$
- -1:  $i$  would be more appropriate if it was clustered in its neighbouring cluster(bad)  $b(i) \ll a(i)$

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) \nless b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } b(i) \nless a(i) \end{cases}$$

# External With labels

# External metrics

- Measure the similarity or agreement between the clustering and the **external partition**
- $0 \sim 1: 1 = \text{a perfect match}$
- Normalized Mutual Information



Assume we have some prior knowledge  
about the correct clustering structure

# Normalized Mutual Information (NMI)

- comparison of two partitions even when a different number of clusters
- NMI depends on the
  - Mutual Information  $I$ 
    - accounts to the “amount of information” one can extract from a distribution regarding a second one
  - the entropy of the labeled  $H(Y)$  and clustered set  $H(C)$ .

# Normalized Mutual Information (NMI)

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]} \quad \text{normalization}$$

$Y$  = class labels (ground truth)

$C$  = cluster labels

$H(.)$  = entropy

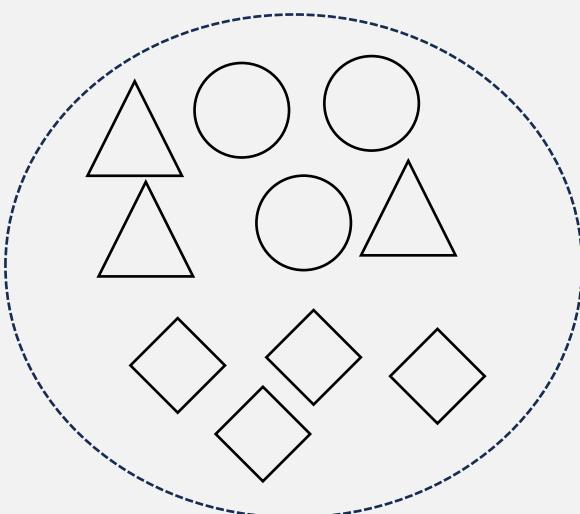
$$H(X) = - \sum_{i=1}^{|X|} p(i) \log_2 p(i)$$

$I(Y; C)$  = mutual information b/w  $Y$  &  $C$

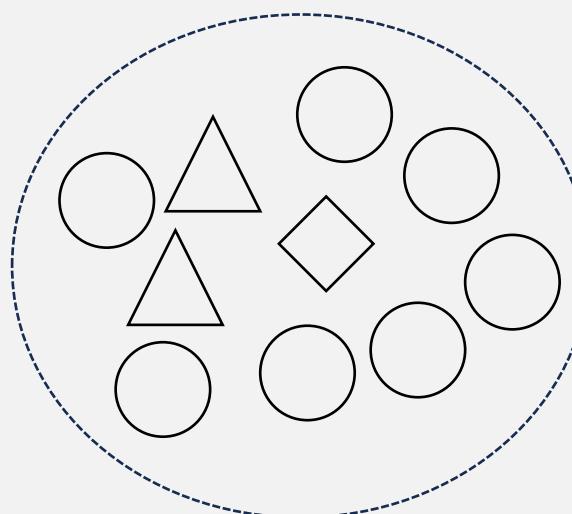
$$I(Y; C) = H(Y) - H(Y|C)$$

# Example1

- How many class labels? Ans: 3
- How many cluster? Ans: 2



cluster1



cluster2



# $H(Y)$

- $H(Y) = -\frac{1}{4} \log\left(\frac{1}{4}\right) - \frac{1}{4} \log\left(\frac{1}{4}\right) - \frac{1}{2} \log\left(\frac{1}{2}\right) = 1.5$

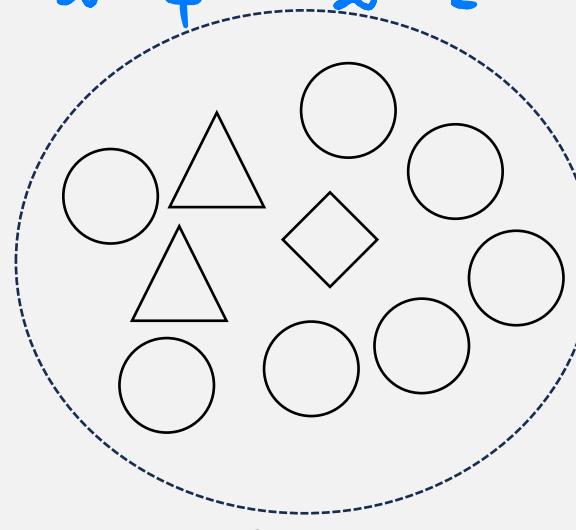
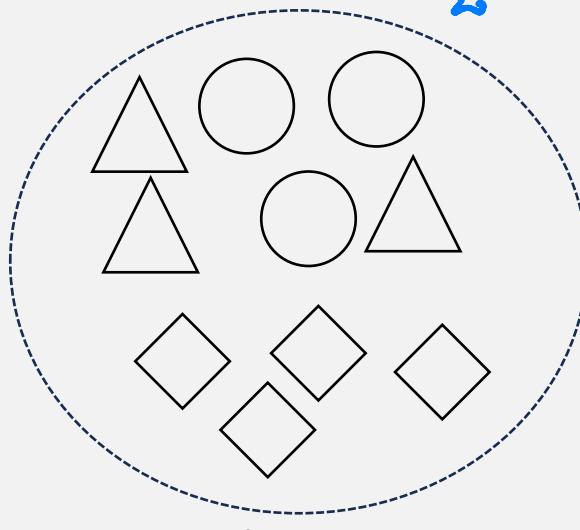
(5 triangles, 5 diamonds, 10 circles)

$$5 + 5 + 10 = 20$$

$$\frac{5}{20} = \frac{1}{4}$$

$$\frac{5}{20} = \frac{1}{4}$$

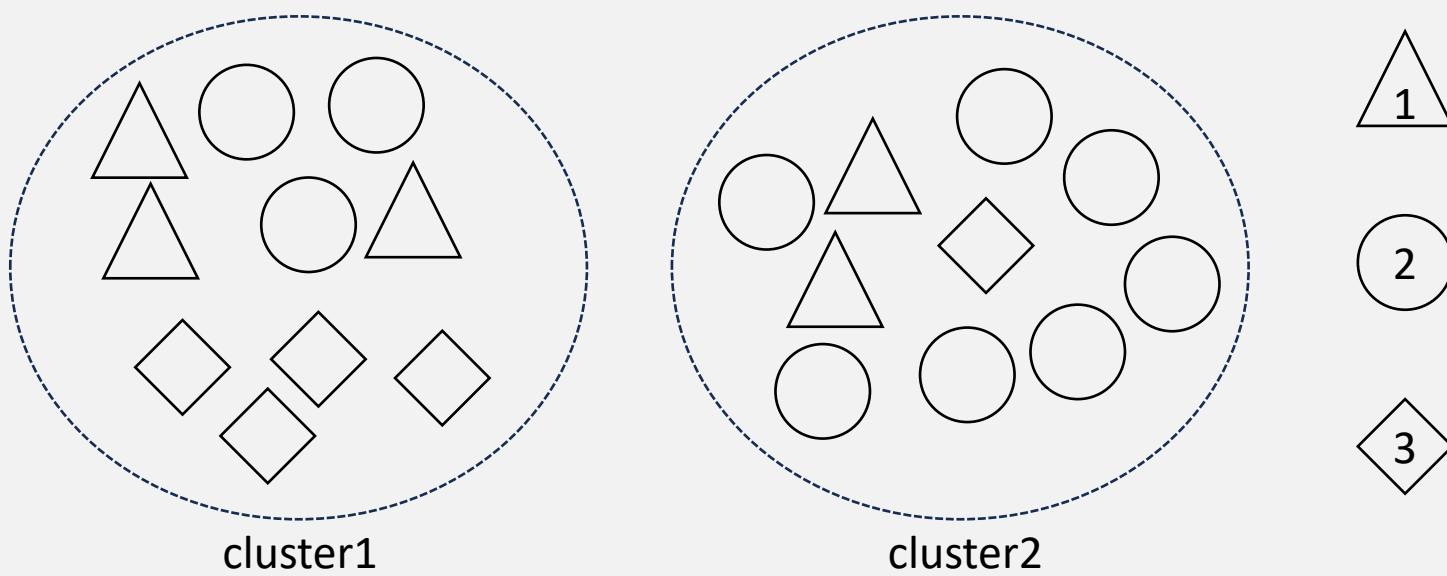
$$\frac{10}{20} = \frac{1}{2}$$



- 1
- 2
- 3

# $H(C)$

- $H(C) = -\frac{1}{2} \log\left(\frac{1}{2}\right) - \frac{1}{2} \log\left(\frac{1}{2}\right) = 1$



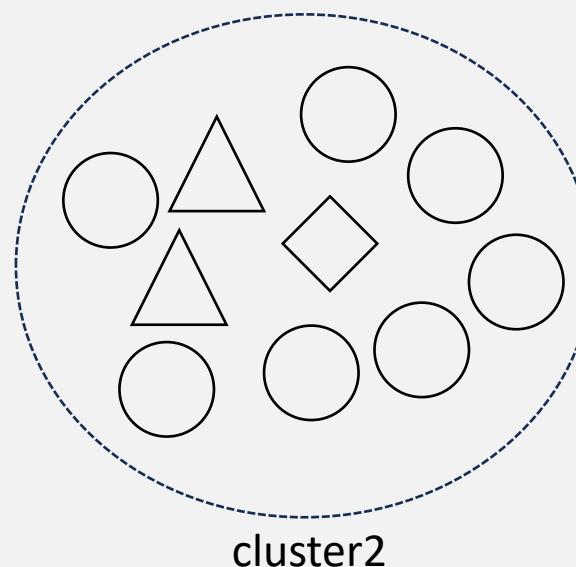
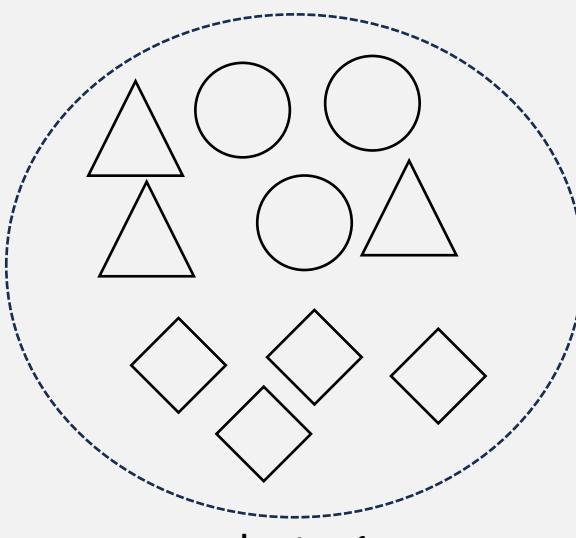
$I(Y; C)$

the amount of uncertainty remaining about  
the class labels after observing which  
cluster each point belongs to.

$$I(Y; C) = H(Y) - H(Y|C)$$

$I(Y; C)$

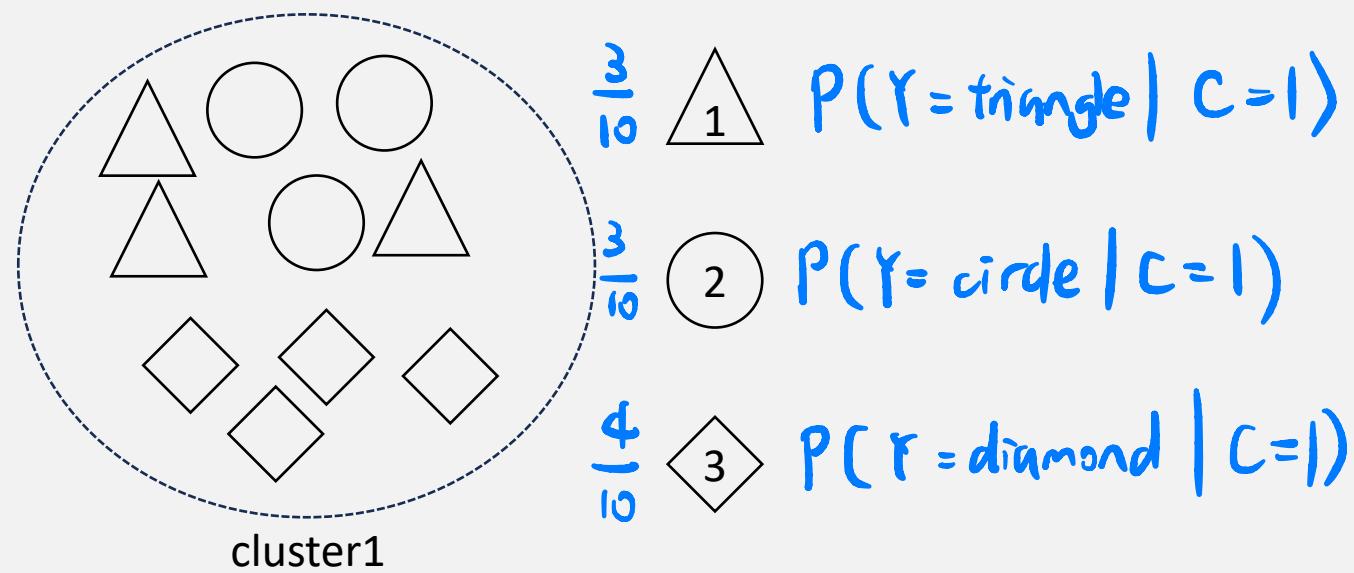
$H(Y|C)$  : the reduction in the entropy of class labels that we get if we know the cluster labels



$H(Y|C)$ : conditional entropy of class labels for clustering  $C$

$$P(Y = 1|C = 1) = 0.3 \cdot -\frac{1}{2} \left[ \frac{3}{10} \log \left( \frac{3}{10} \right) + \frac{4}{10} \log \left( \frac{4}{10} \right) + \frac{4}{10} \log \left( \frac{4}{10} \right) \right]$$

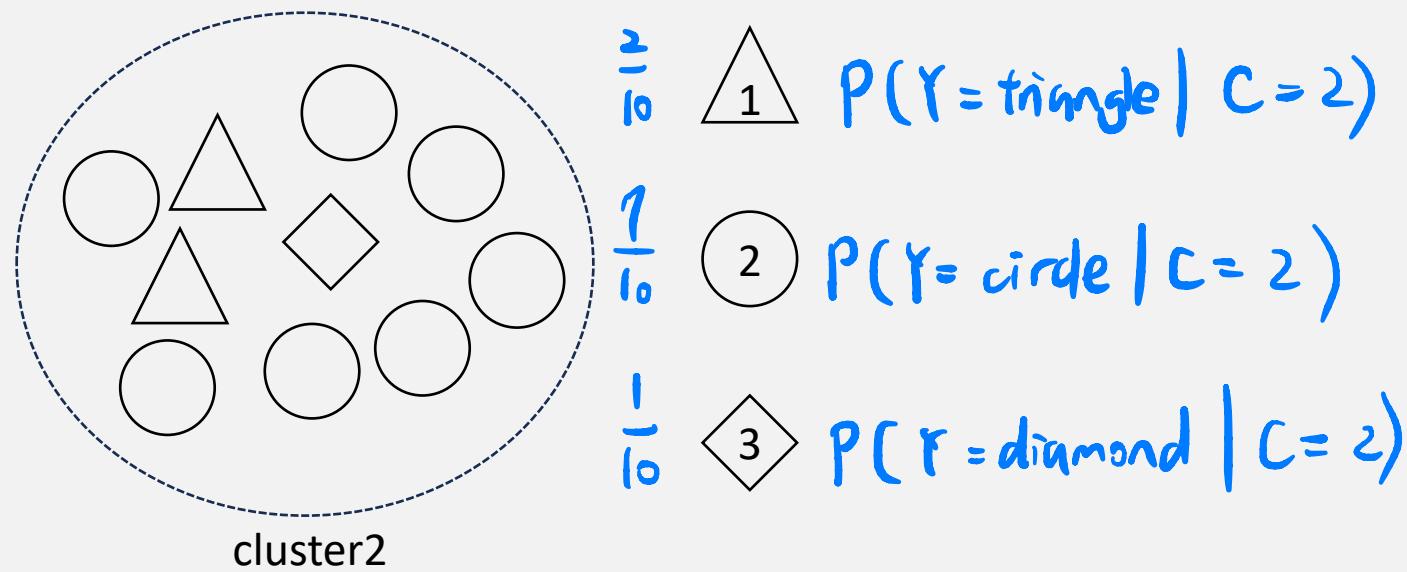
$$H(Y|C = 1) = -P(C = 1) \sum_{y \in \{1, 2, 3\}} P(Y = y|C = 1) \log(P(Y = y|C = 1)) = 0.7855$$



$H(Y|C)$ : conditional entropy of class labels for clustering  $C=2$

$$-\frac{1}{2} \left[ \frac{2}{10} \log\left(\frac{2}{10}\right) + \frac{1}{10} \log\left(\frac{1}{10}\right) + \frac{1}{10} \log\left(\frac{1}{10}\right) \right]$$

$$H(Y|C = 2) = -P(C = 2) \sum_{y \in \{1,2,3\}} P(Y = y|C = 2) \log(P(Y = y|C = 1)) = 0.5784$$



## NMI

$$I(Y; C) = H(Y) - H(Y|C)$$

$$I(Y; C) = 1.5 - (0.7855 + 0.5784) = 0.1361$$

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]} = \frac{2 \times 0.1361}{1.5 + 1}$$

$$= 0.1089$$

↳ There is little agreement between the cluster assignments & the actual class labels

⇒ There is still significant room for improvement

# NMI property

- This metric is dependent or independent?
  - of the absolute values of the labels
  - a permutation of the class or cluster label values won't change the score value in any way
- If we have to compare two clustering that have different number of clusters we can still use NMI. => it's normalized



Thank You  
Any Question?



AITC

教育部人工智慧技術及應用人才培育計畫  
Artificial Intelligence Talent Cultivation Program