ROBERT GORDON UNIVERSITY ABERDEEN

Evaluation of Classification Learning

Statement for Audio and Video Learning Resources

Video and audio content at the University uses closed captions generated by automatic speech recognition (ASR). The ASR process is based on machine learning algorithms which automatically transcribe voice to text. According to our technology providers, this process is approximately 70-90% accurate depending on the quality of the audio, and consequently video and audio closed captions may include some transcription errors. It is therefore important to recognise that the original recording is the most accurate reflection of the content, and not the captions.

If you require accurate captions as part of your reasonable adjustments, please contact the Inclusion Centre to discuss your requirements.

Lab Postmortem (Classifiers and missing data)

Weird errors happen when missing a library:

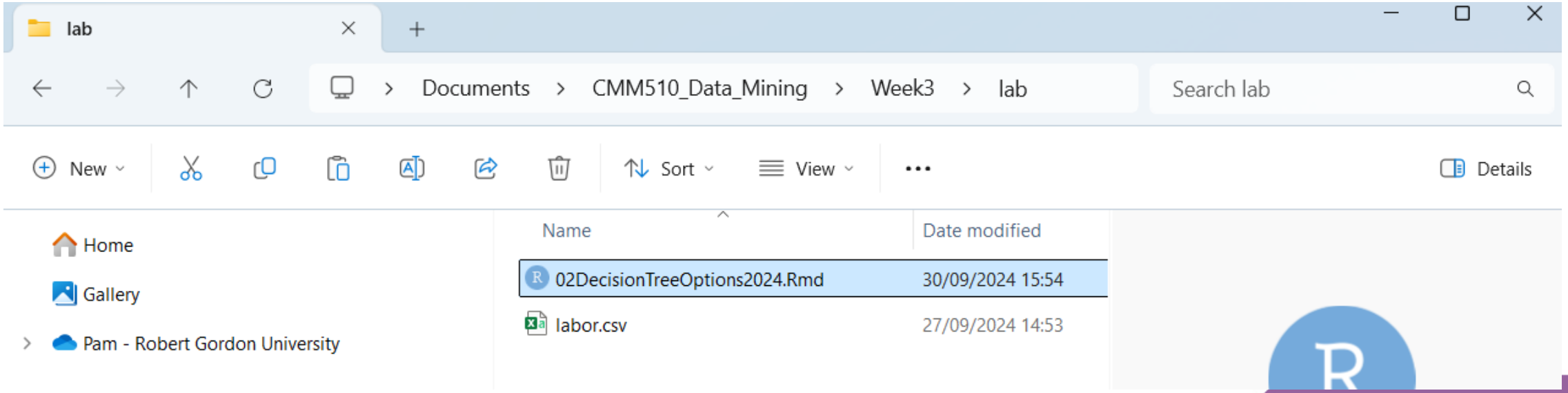
rm(list=ls()) # Cleans the working environment

library() # Imports a library – you might have more than one.

Know where your .rmd file and your dataset

Easiest: keep them both in the same folder and don't bother with paths.

labor <- read.csv("labor.csv", header = T, stringsAsFactors=T)



Know the “train() function

```
# Running the algorithm
set.seed(123)
c5model <-  train(play ~ .,
                  data = WeatherPlay,
                  method = "C5.0Tree",
                  trControl = control1)
summary(c5model$finalModel)
```

Know the “train() function

```
# Running the algorithm
set.seed(123)
c5model <- train(play ~ .,
                 data = WeatherPlay,
                 method = "C5.0Tree",
                 trControl = control1)
summary(c5model$finalModel)
```

Your model object

What is label? What is data?

Dataset object

A model architecture

A control object

Know what we added to the train function…

```
control1 <- trainControl(method = "cv", number = 5)
set.seed(123)
c5Tree <- train(      Class ~ .,
                      data = labor,
                      method = "C5.0Tree",
                      na.action= na.pass,
                      trControl = control1,
                      control = C5.0Control(CF = 0.35 ))
```

What to do with missing data

Confidence factor (smaller is more accurate tree but maybe overfitting?)

minCases, winnow – other models had different control options!

What is this hot mess?

```
Decision tree:
WageIncY1 <= 2.5: bad (15.3/2.3)
WageIncY1 > 2.5:
:...LTDIyes <= 0:
    :...Duration > 2: good (2.2)
    : Duration <= 2:
    : :...Hours <= 39: good (2.5/0.3)
        : Hours > 39: bad (5.4/0.7)
    LTDIyes > 0:
    :...WageIncY1 > 3: good (26.3)
        WageIncY1 <= 3:
        :...Holidays <= 10: bad (2)
            Holidays > 10: good (3.4)

Evaluation on training data (57 cases):
Decision Tree
----------------
Size Errors
7     2( 3.5%) <<

           (a) (b) <-classified as
           ---- ----
           20         (a): class bad
           2     35   (b): class good
```

It's the model summary (but specific to model)

```
Decision tree:
WageIncY1 <= 2.5: bad (15.3/2.3)
WageIncY1 > 2.5:
:...LTDIyes <= 0:
    :...Duration > 2: good (2.2)
    : Duration <= 2:
    : :...Hours <= 39: good (2.5/0.3)
        : Hours > 39: bad (5.4/0.7)
    LTDIyes > 0:
    :...WageIncY1 > 3: good (26.3)
        WageIncY1 <= 3:
        :...Holidays <= 10: bad (2)
            Holidays > 10: good (3.4)
```

Your model.
A series of rules that can classify an instance of the dataset.

```
Evaluation on training data (57 cases):
Decision Tree
----------------
Size Errors
7     2( 3.5%) <<
```

```
 (a) (b) <-classified as
 ---- ----
 20          (a): class bad
 2     35    (b): class good
```

The confusion matrix.
You can do loads with this.

Contents

Model evaluation

- Modelling: use a ML algorithm on a dataset to produce a (predictive) model.
- A model needs to be evaluated.
- The results of using a predictive model can be:
  - A class prediction: may be
    - Class output: can be converted to probabilities (controversial!)
    - Probability output: logistic regression, random forest , gradient boosting. A class is selected according to the probability outputs.
  - A numeric prediction.
- This lecture concentrates on the evaluation of models for class prediction.

All models are wrong

Some models are useful

Why Evaluate classification models?

Quality of model: how good the classification model is solving unseen problems

- Error on training data is not good indicator of performance on unseen data

Simple solution if lots of labelled (classified) data available

- Split data into disjoint training and test sets
- Learn model from training set; evaluate model on test set

But labelled data is usually limited

- Need more sophisticated evaluation techniques

Answer questions

- How “right” is the model?
- Should we trust it?
- We deployed it. Should we still trust it?

But first…

- How do we synthesise unseen data?
- Should we “hide” some data from the model?
- But I don’t have that much data!

Training and Testing

Training set used for testing	Test set used for testing	Assumptions
• Resubstitution error is error rate from training data • Hopelessly optimistic as it is also used for training, i.e. creating the model! • Overfitting of data.	• Independent instances that play no part in learning of classifier.	• Training and testing data sets are representative samples of the underlying problem. • Training and testing sets contain different instances.

Data sets (all 3 are different)

Training set

The set of data used to build a model.

Validation set

The set of data used to evaluate a model fit when tuning the model.

Test set

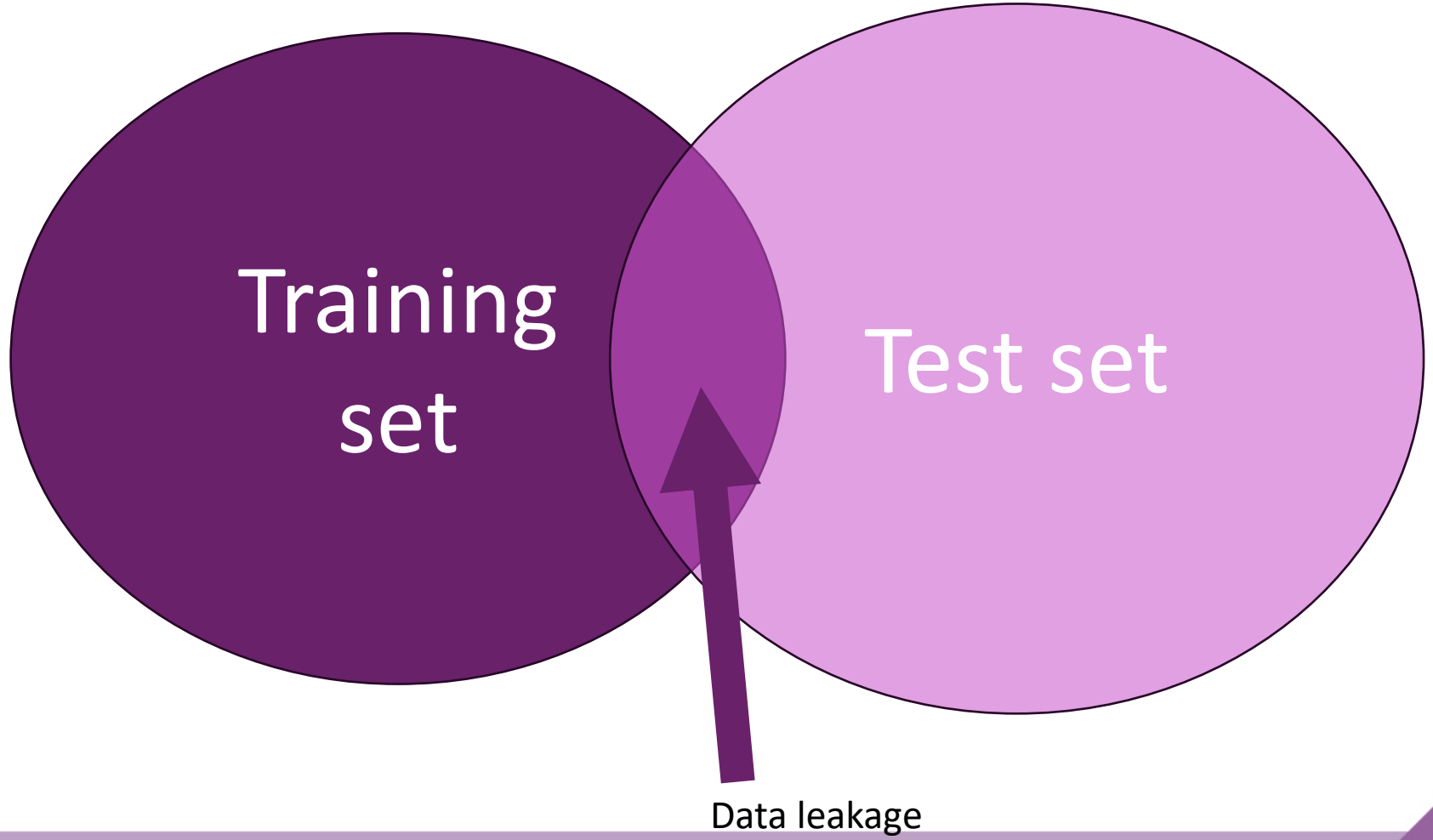
The set of data used in the final evaluation.

But if you don’t have enough data…

Training Set

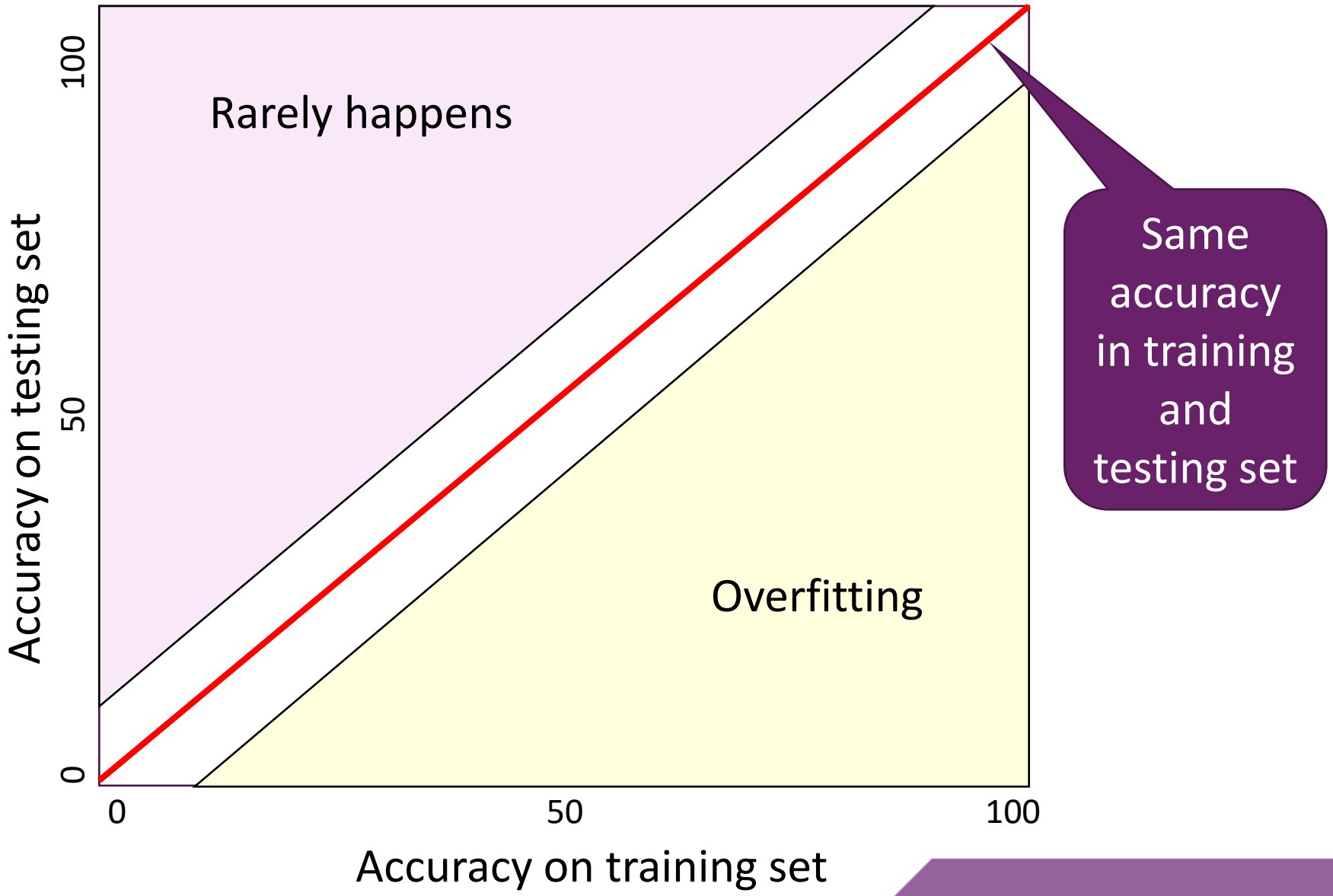
Test Set

Data leakage – a novice's error



Over-fitting

- Overfitting: model biased towards training set.
  - Accuracy on test set is not as good.

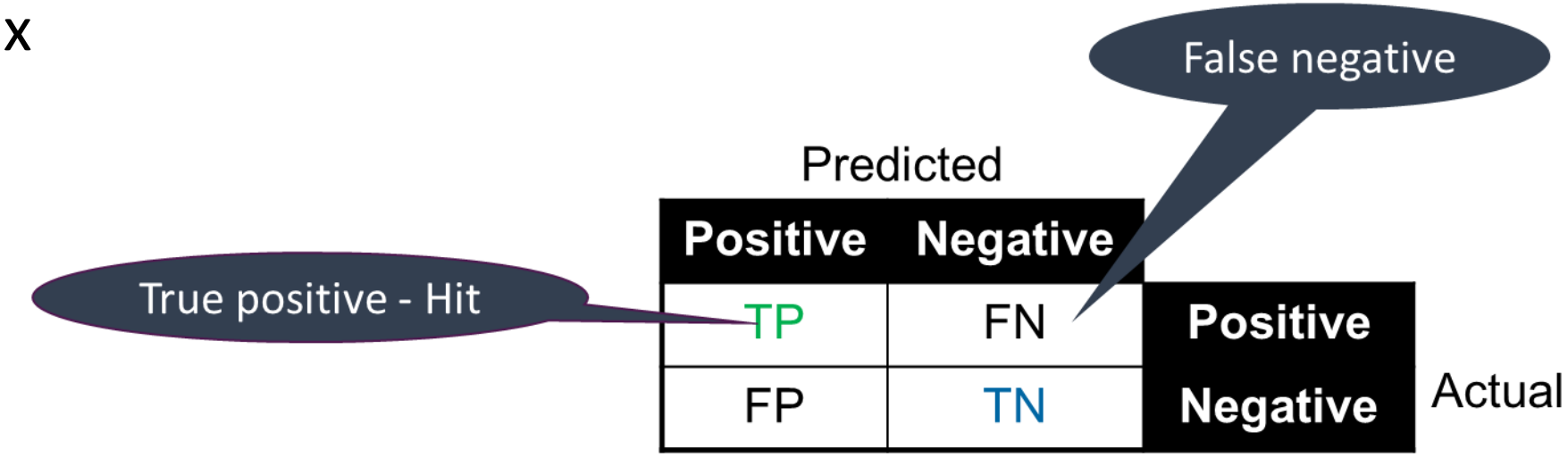


Evaluation measures: class prediction

- Confusion matrix
- Evaluation metric
  - Accuracy / Error
  - Precision and Recall
    - F1 measure
  - Sensitivity
  - Specificity
  - Kappa statistics
  - ROC curves / AUC

Confusion matrix - binary classification

- Confusion Matrix



- Predictive Accuracy = $100 * \frac{TP+TN}{TP+TN+FP+FN}$
  - TP = Number of positives correctly classified as positive – also called Hit.
  - FP = Number of negatives falsely classified as positive – also called Type I error.
  - TN = Number of negatives correctly classified negative.
  - FN = Number of positives falsely classified as negative – also called miss or Type II error.

Confusion matrix – 2 options

Predicted

Positive	Negative	
TP	FN	Positive
FP	TN	Negative

Actual

Actual

Positive	Negative	
TP	FP	Positive
FN	TN	Negative

Predicted

For any confusion matrix, either columns or rows correspond to predicted. The other option corresponds to actual.

Confusion matrix

- Correct predictions fall on the diagonal of the matrix
- Off the diagonal are instances that has been misclassified
- Performance is based on the counts of the predictions on and off the diagonal of the confusion matrix

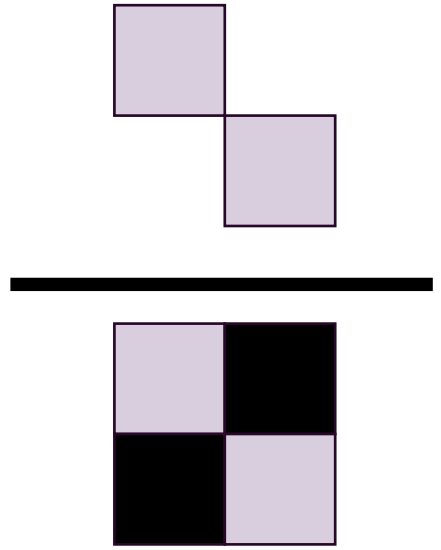
Actually Positive	**Actually Negative**	
True Positive (TP)	False Positive (FP)	Predicted Positive
False Negative (FP)	True Negative (TN)	Predicted Negative

Evaluation maths: Accuracy

Number right out of the total number of samples

Percentage the model got right overall

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



Good metric if the dataset is quite balanced (i.e. same number of all classes). Good for multiclass.

Predictive Accuracy

- Predictive Accuracy
  - proportion of test examples correctly classified
    - Number between 0 and 1
    - Often expressed as percentage
- Error Rate
  - Proportion of test examples incorrectly classified
    - often expressed as percentage
- Error rate (%) = 100 – Predictive Accuracy (%)

… predictive accuracy

Correctly Classified Instances	42	73.6842 %
Incorrectly Classified Instances	15	26.3158 %
Total Number of Instances	57	

Accuracy (42/57)

Error 15/57

Accuracy example – binary classification

- Test examples contain 500 positives and 500 negatives.

	Predicted: Positive	Predicted: Negative
Actual: Positive	400	100
Actual: Negative	200	300

- Predictive Accuracy =

$$= 100 * \frac{TP+TN}{TP+TN+FP+FN} =$$

$$= 100 * \frac{400+300}{400+300+200+100} =$$

$$= 100 * \frac{700}{1000} = 70\%$$

Accuracy – non-binary classification

Predicted

a	b	c	d	e	
100	60	0	40	0	a
30	50	30	30	60	b
20	0	150	0	30	c
0	0	0	200	0	d
0	50	0	50	100	e

Actual

- Predictive Accuracy =

  = 100 * (Σ diagonal) / (Σ table)

  = $100 * \frac{TP+TN}{TP+TN+FP+FN} =$

  = $100 * \frac{100+50+150+200+100}{1000} =$

  = $100 * \frac{600}{1000} = 60\%$

Σ means "sum"

What is “Good” accuracy?

- Binary classification >50%
- Multiclass classification?
  - 10 classes >10%
  - 3 classes >33%

- Any of these suggest the model has learned something. BUT only if the dataset is approximately balanced.

Pam's sweetie classifier

- Jar has 100 sweeties
- 1 is "poisoned"
- 99 are sweet
- Model is 99% accurate

The classifier

Pam's sweetie classifier confusion matrix

- Jar has 100 sweeties
- 1 is "poisoned"
- 99 are sweet
- Model is 99% accurate

Actually Poisonend	Actually Nice	
0	0	Predicted Poisoned
1	99	Predicted Nice

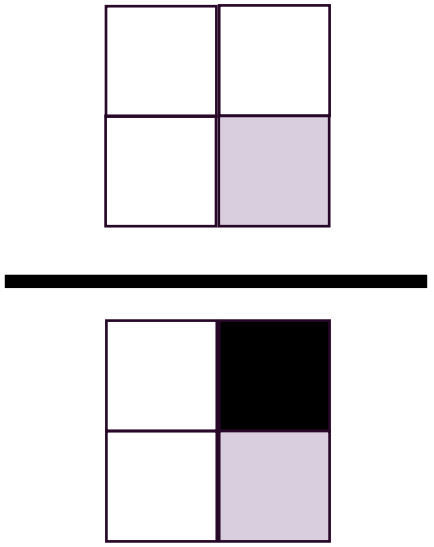
Evaluation maths: Specificity

Proportion of true negatives out of all negatives.

True negative rate.

Goodness of the model at determining a negative sample.

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



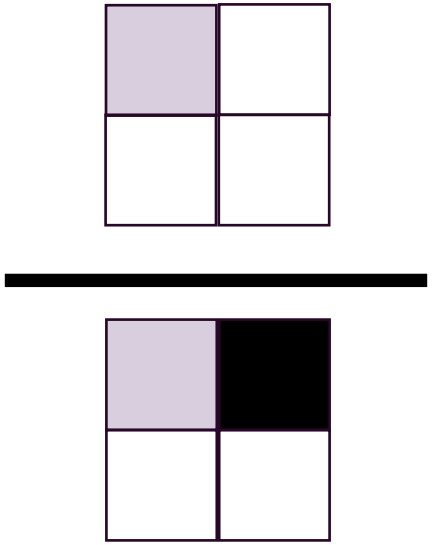
Sweetie Classifier has 100% Specificity

Evaluation maths: Precision

Proportion of true positives out of all predicted positives.

Proportion of the positives predicted by the model that are correct.

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



Sweetie Classifier has 0% Precision

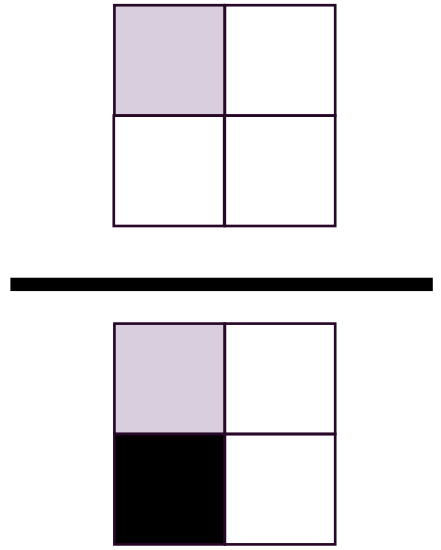
Evaluation maths: Sensitivity (or Recall)

Proportion of true positives out of all actual positives.

True positive rate.

Goodness of the model at determining a positive sample.

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



Sweetie Classifier has 0% Recall

Precision and Recall

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.700	0.243	0.609	0.700	0.651	0.444	0.695	0.559	bad
	0.757	0.300	0.824	0.757	0.789	0.444	0.695	0.738	good
WA.	0.737	0.280	0.748	0.737	0.740	0.444	0.695	0.675	

Weighted average

a	b	<-- classified as
14	6	a = bad
9	28	b = good

Measures are per class value

- **Precision:** Fraction of returned as class X that are really class X
  - E.g precision("bad") =14/23 = 0.609
- **Recall:** fraction of all class X which are returned as class X.
  - E.g recall("bad") = 14/20 = 0.7

F-score or F-measure

- Combines precision and recall – used with uneven class distribution.
- $F_\beta\_score = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$
- β is the trade-off between precision and recall.
  - Recall is considered β times as important as precision.
  - Common values are **1**, 0.5 and 2.
  - F_1 is the F score with $\beta = 1$.
- $\beta > 1$ gives more weight to recall.
- $\beta < 1$ gives more weight to precision.
- F_1 is 1 if all true positives and all true negatives are correctly identified (worst value would be zero).

Sensitivity and Specificity (binary classification)

```
  a     b   <-- classified as
 14     6 |  a = positive
  9    28 |  b = negative
```

$$Sensitivity = \frac{14}{14+6} = 0.7$$

$$Specificity = \frac{28}{28+9} = 0.7568$$

Kappa (inter-related agreement – nominal class)

- $k = \frac{p_o - p_e}{1 - p_e}$
- P_o probability of observation
- P_e probability of chance agreement

```
 a   b    <-- classified as
14   6 |   a = bad
 9  28 |   b = good
```

Actual:
14 + 6 = 20 bad
9+28= 37 good

Classified as:
14+9 = 23 bad
6 + 28 = 34 good

Total = 14+6+9+28 = 57

P_e = 20/57 * 23/57 + 37/57 * 34/57 = 0.529 (agree by chance)

P_o = (14+28)/57 = 0.737 (accuracy in testing)

$k = \frac{p_o - p_e}{1 - p_e} = \frac{0.737 - 0.529}{1 - 0.529} = 0.442$

Kappa interpretation

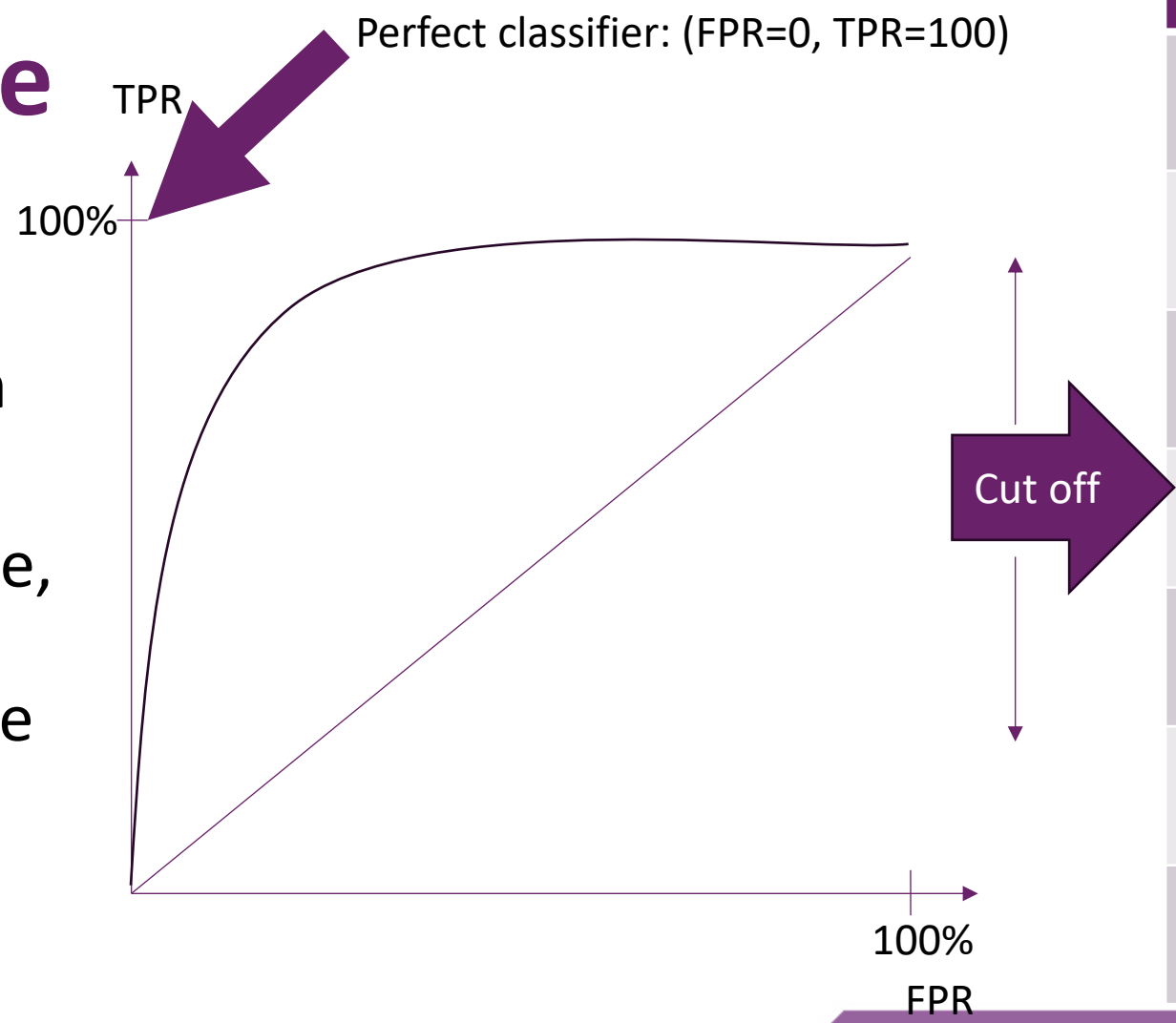
- Little agreement: $k < 0.20$
- Some agreement: $0.20 \leq k < 0.40$
- Moderate agreement: $0.40 \leq k < 0.60$
- Good agreement: $0.60 \leq k < 0.80$
- Very good agreement: $0.80 \leq k < 1.00$

ROC Curves

- ROC curves– used for binary classification
- Receiver Operating Characteristic
  - Hit rate/false alarm trade-off in noisy communication
- ROC plot
  - y axis: % true positives in sample
  - X axis: % false positives in sample
- A classifier that has no skill (e.g. predicts the majority class) will be represented by a diagonal line from the bottom left to the top right.
- Plot sensitivity and (1-specificity), i.e. the difference between the true positive rate and the false positive rate.
- Measure Area Under Curve (AUC). Whole area is 1, so AUC will be < 1.

Plotting a ROC Curve

- Binary classifier outputs *probability*
- We select the cut off between classes
- To get points on the ROC Curve, select different cut offs and measure the True Positive Rate (TPR) and False Positive Rate (FPR)



P	A
0.99	T
0.95	T
0.89	F
0.88	T
0.79	T
0.66	F
0.65	T

Curve may not match data.

Generating a ROC Curve

P_{100}=20/100
N_{100}=80/100

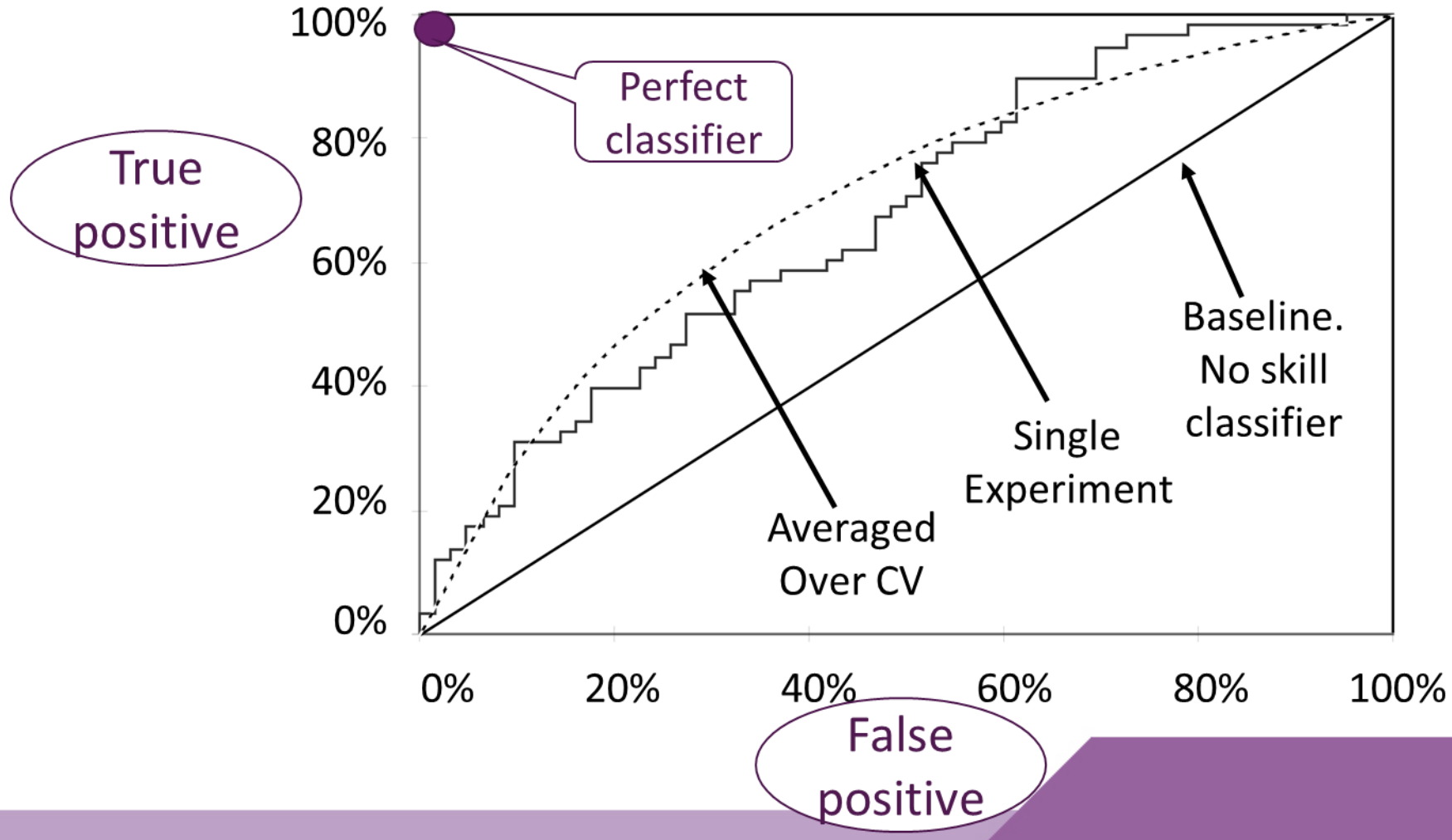
Rank	Predicted Probability	Actual Class
1	0.95	Yes
2	0.93	Yes
3	0.93	No
4	0.88	Yes
5	0.86	Yes
6	0.85	Yes
7	0.83	Yes
8	0.80	Yes
9	0.80	No
10	0.79	Yes

P_{10}=8/10
N_{10}=2/10

Rank	Predicted Probability	Actual Class
11	0.77	No
12	0.76	Yes
13	0.73	Yes
14	0.65	No
15	0.63	Yes
16	0.58	No
17	0.56	Yes
18	0.49	No
19	0.48	Yes
...	...	...

- TP_{10}rate=8/20, FP_{10}rate=2/80
- ROC_{100} = (2/80, 8/20) = (2.5%, 40%)

Example ROC Chart



… ROC curve interpretation

- Area under curve is
  - Excellent 0.9-1
  - Good: 0.8-0.9
  - Fair: 0.7-0.8
  - Poor: 0.6-0.7
  - Fail: 0.5-0.6

Experimental design

- Design experiments to build and evaluate model.
- Consider bias vs. variance trade-off.
- Consider dataset imbalance – dataset where one type of target scenario is much more represented than another one.
- Lots of data required! Need data to
  - Tune models
    - Optimise parameter values
    - Need training and testing sets
  - Test final model
    - Need additional testing set.

Model bias vs variance

- Bias – incorrect assumptions made
  - Can introduce systematic prejudice.
  - Often the result of a cost function (but data may also be biased) which favours some distribution.
  - While bias may not be intentional, the consequences can be serious (illegal/dangerous).
- Common bias problems:
  - Algorithm .
  - Sample data – not representative.
  - Prejudice bias in data – the data collection process made assumptions which contained bias. E.g. all presidents are male.
  - Measurement bias in data collected: e.g. if a patient knows they are given a test drug, the may feel better because of what they know, not because of what they take.

Examples of bias

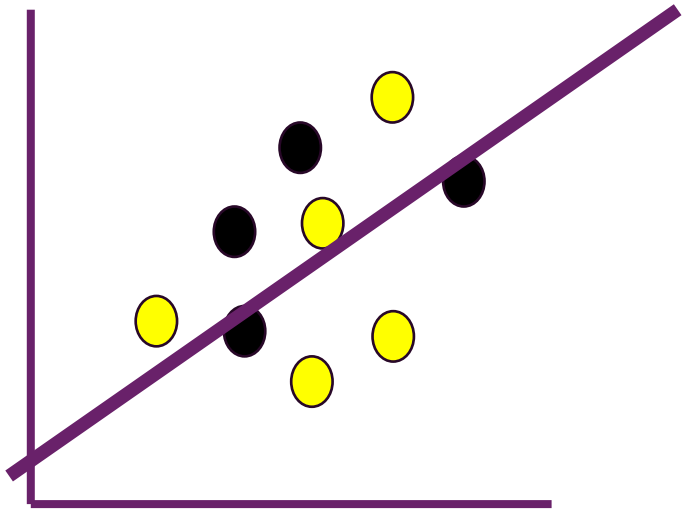
- Amazon HR recruitment algorithm was biased.
  - AI algorithm scored candidates from CVs.
  - BUT was not gender-neutral, discriminating against women.
  - Had learnt patterns from CVs, most of which had come from men.
  - Check https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G [accessed 1/10/2024]
- COMPAS Software to assess the likelihood of a defendant from recurring was found to be biased according to skin colour.
  - https://en.wikipedia.org/wiki/COMPAS_(software) [accessed 1/10/2024]

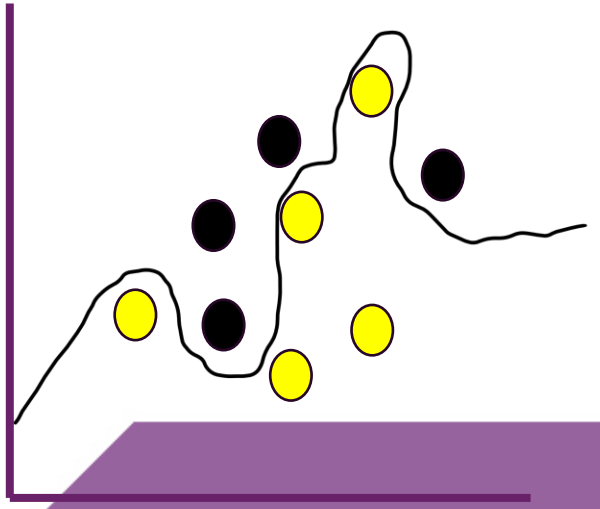
Variance

- Variance: incorrect assumptions learnt from noise in the data.
  - But the data is correct, not biased.
  - Algorithm too sensitive to fluctuations in data.
- High variance helps reduce bias.
- High bias helps reduce variance.

Bias vs variance

- High bias → the model does not match the dataset closely (or incorrect dataset).
  - Low variance. Generalised/simplified model.
  - Models will be similar for different training sets.
  - Data trends may not be adequately captured.
  - **Underfitting.**
- Low bias → the model matches the dataset closely
  - High variance. More complex model.
  - Models will be different for different training sets.
  - **Overfitting.**





Bias vs variance trade-off

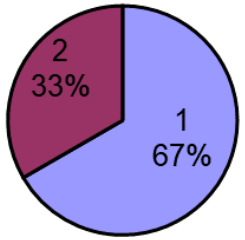
- Select appropriate data
  - Representative
  - Large enough
  - No sample bias or prejudice bias
- **Test** models to ensure there is no algorithm or dataset bias.
- Monitor system
  - Bias may crop up over time as reinforcement learning.
- Combat overfitting
  - Model dependent. E.g. simplify models

Contents

Utilising all the available data

Holdout procedure

- Two thirds of data for training (1)
- The remaining data, i.e. one third, for testing (2)
- The split could be different, e.g. 80% training, 20% testing



Dilemma

- The larger the training data the better the classifier
- The larger the test data the more accurate the error estimate

After evaluation

- All data may be used to build final classifier

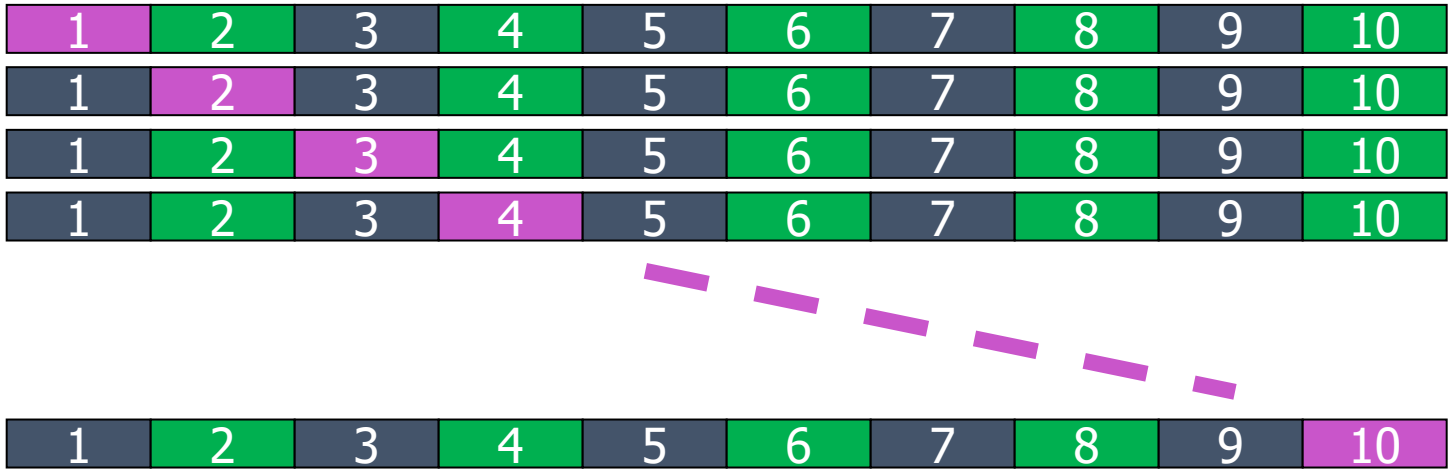
Holdout evaluation - stratification

- Samples should be representative
  - Each class is represented with approximately equal proportions in both subsets
  - This is called stratified hold-out
- Repeated holdout
  - Randomly select test set each iteration
  - Calculate average error rate
- Can overlapping test sets be avoided?
  - Exploit test-train splits, but…

k-fold Cross-Validation

- First step
  - dataset is split into k folds/partitions of equal size

- Second step
  - each fold used as **test set**; **remainder for training**



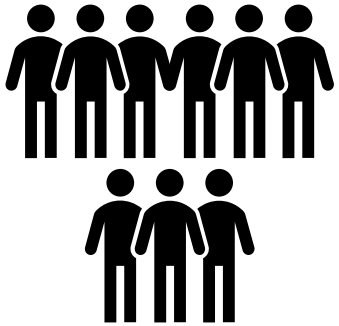
- Calculate errors over k folds

Standard Cross-Validation

- **Stratified** ten-fold cross-validation
  - Ten-fold is known to give accurate estimate
    - 9/10ths for training and 1/10th for testing x 10 (one per fold)
  - **Stratification** - each class is properly represented in each fold
    - "Properly" = in the same proportion as in the dataset
    - Stratification reduces the estimate's deviation
- Repeated stratified cross-validation
  - E.g. ten-fold cross-validation repeated ten times
    - Results averaged over 100 experiments (10 folds x 10 repetitions)
    - At each repetition, fold membership will change.
  - Can be computationally expensive
    - Reduces the variance

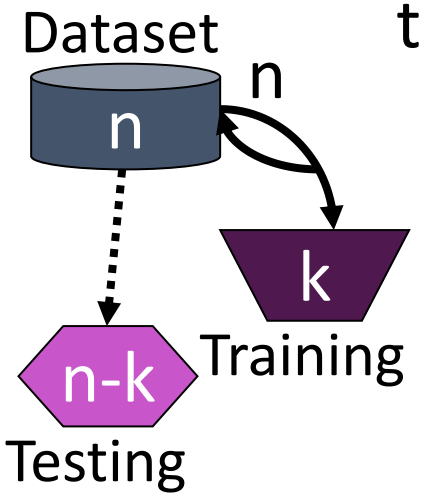
Leave-One-Out Testing

- A particular form of cross-validation
  - number of folds = number of training instances
    - no random sampling involved
    - stratification is not possible
      - only one instance in the test set!
- Makes maximum use of the data
  - test sets contain single instances
  - classifier is built n times when n training instances
  - n-1 instances in each training set
- Very computationally expensive
  - Particularly for model builders
  - So best for small datasets



Bootstrap Estimation

- Both Holdout and Cross-validation creates sample without replacement
  - instance cannot be selected again for a particular fold
    - i.e. duplicates not allowed in training set
- Bootstrap uses sampling with replacement to form training set
  - n instance dataset is sampled n times with replacement for new dataset of n instances
    - new dataset is training set
    - instances from old dataset not in new training set are used for testing
  - training set has only ~63.2% of dataset in it



Bootstrap Estimation

- Estimates on test data (holdout) are very pessimistic
  - trained on only ~63.2% of instances
  - compared to 90% with 10-fold CV
- Combine with resubstitution (training set) error
  - Error = 0.632 * $error_{test}$ + 0.368 * $error_{training}$
    - resubstitution error weight < test data error weight
- Repeat experiments several times with different selections for training data
  - average results
- Good way to estimate performance for very small datasets - but some disadvantages

Bootstrap why ~63.2%?

- Given n instances
  - chance of picking an instance 1/n
  - so chance of not picking is (1- 1/n)
- Multiply these probabilities for n picking opportunities
  - chance of an instance not being picked
    - $(1 - 1/n)^n \approx 0.368 = 36.8\%$
  - chance of being picked = (100 – 36.8)% = 63.2%

Contents

Issues

Statistical significance	Performance measures	Costs assigned to different types of errors
• How likely it is that the result is not due to chance? • Coin tossing does not maintain 50:50 balance	• Effectiveness (error rate, mean squared error) • Efficiency (CPU cycles)	• Medical false alarms are often less costly than a missed diagnosis • False security intrusion alert less costly than missed intrusion

Other issues - classification

- % Accuracy is a measure of “goodness”
  - But there are other measures
- Where the errors are may be key
  - E.g. Data about cows in heat. 97% of the data is about cows which are not in heat
    - So if classifier answers “no”, it will achieve a 97% accuracy
    - But it will be useless to a farmer!
  - E.g. Errors missing a true illness may be worse than errors where illness is diagnosed when patient is healthy
- Distribution of errors may also be important.
- Is the classifier’s performance better than the expert’s?
  - If current technology allows correct diagnosis of disease X 60% of the time, a classifier which diagnoses it correctly 68% of the time is good!

Example (Confusion Matrices)

Classed as

	a	b	c
a	20	2	2
b	1	23	2
c	1	2	22

10/75 % errors, evenly distributed

10/75 % errors, all in the classification of “b”

Classed as

	a	b	c
a	24	0	0
b	4	16	6
c	0	0	25

…example

Classed as

	a	b	c
a	80	0	15
b	0	90	0
c	15	0	80

30/280 errors, none for class “b”

Provided we have a big enough test set, if classifier answers “b”, it is a “b” as no instances of other classes are classifier as “b”

Classed as

	a	b	c
a	80	15	0
b	0	90	0
c	0	15	80

30/280 errors, none in the classification of “b”

If classifier answers “b”, it may, or may not be a “b” as 30/120 of “b” answers are errors

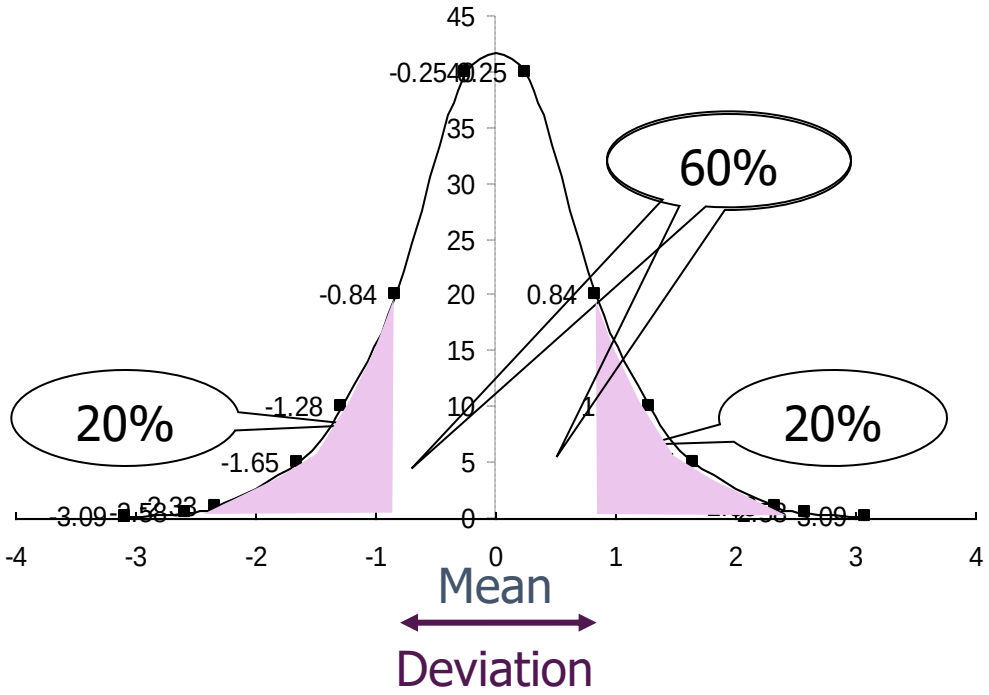
Contents

- Why Evaluate classification models?
  - Experimental criteria
  - Training and test sets
- Measures
- Bias-variance trade-off
- Experimental Design
  - Holdout
  - Cross-validation
  - Leave-one-out
  - Bootstrap
- Other considerations
- Statistical significance

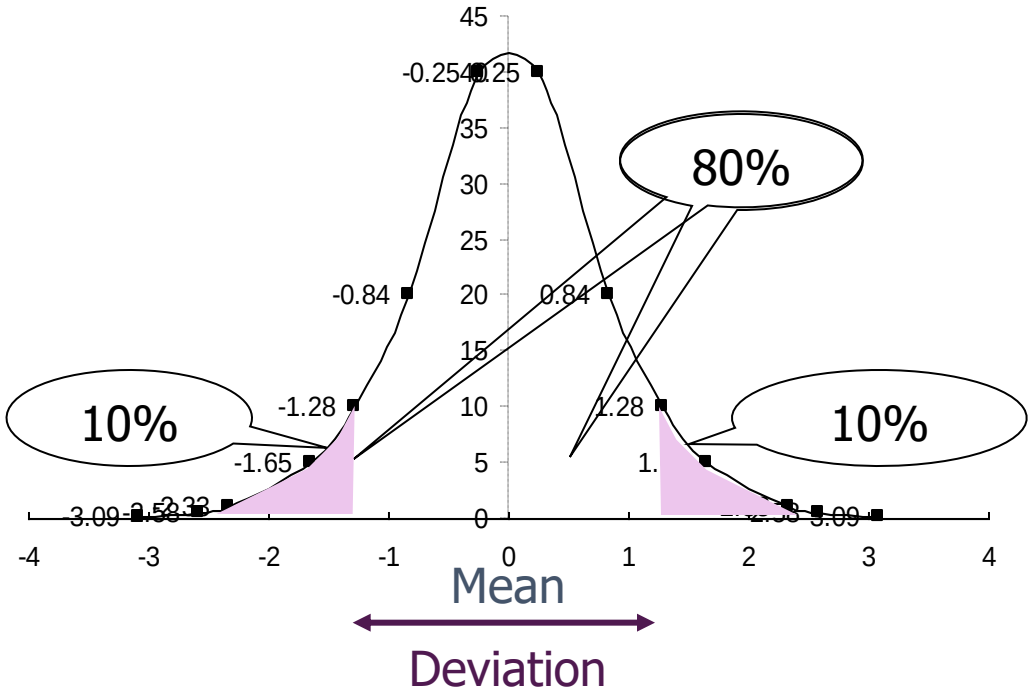
Statistical significance

- Suppose estimated error rate is 25%
  - Accuracy on test set 75%
- How close to true error rate?
  - Depends on the amount of test data
    - You will be more confident if your estimate is based on a test set of 10,000 instances
    - rather than a test set of 10 instances
- Prediction is just like tossing a biased coin
  - “heads” = success; “tails” = failure
  - a succession of independent events
- Statistics provides confidence intervals for the true underlying proportion!

Confidence Intervals



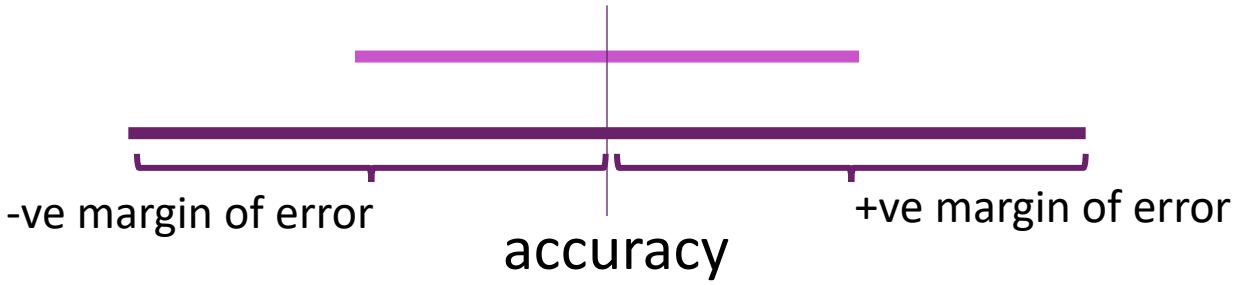
Confidence Intervals



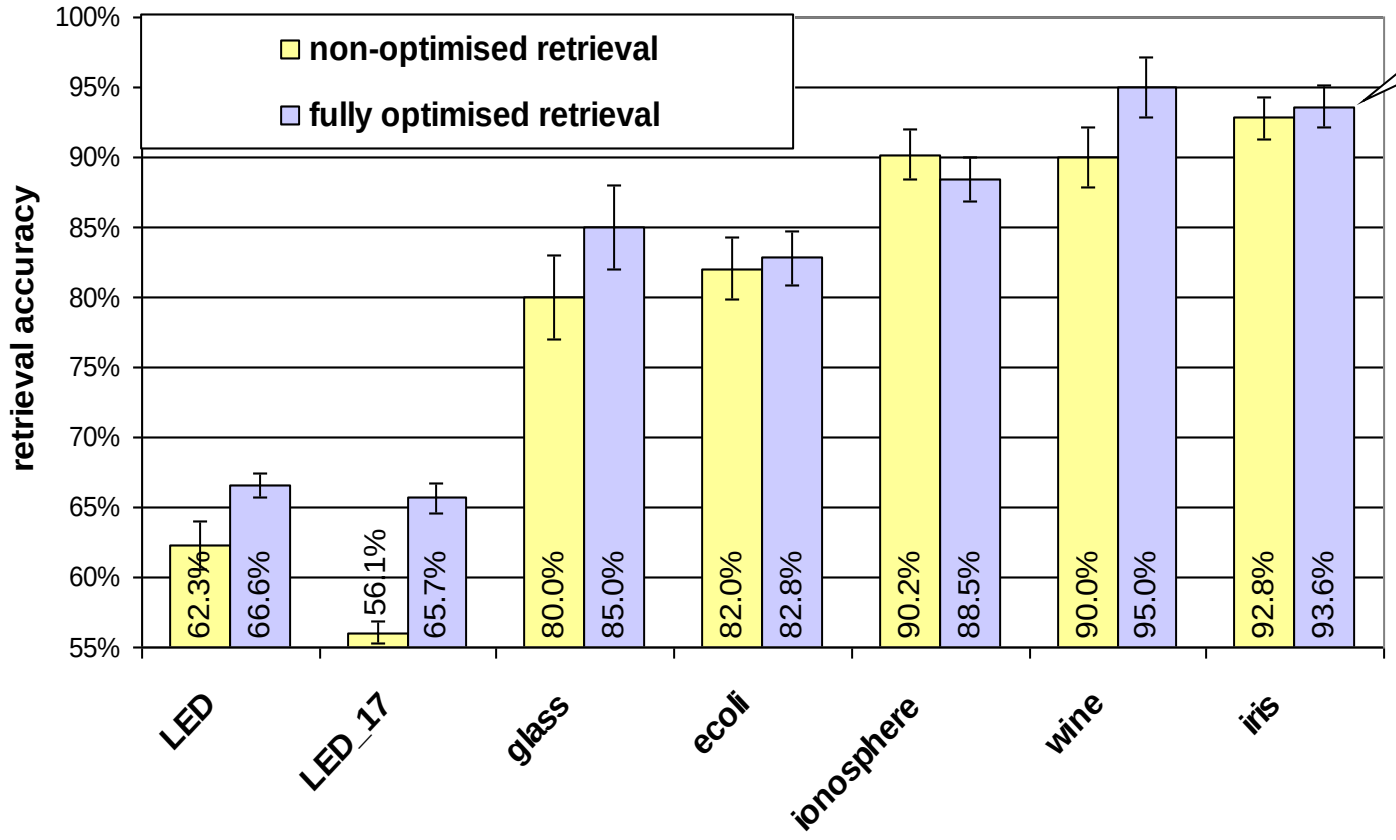
Confidence intervals

- Give an indication of how close the true value may be to our estimate.
- 95% confidence often used.
- An algorithm has an estimated (test) accuracy ± margin of error
  - E.g. 75% ± 5.5, i.e. an accuracy ∈ [69.5%, 80.5%]

- Higher confidence level ➔ bigger margin of error ➔ wider interval
- Interval for 80% confidence ⊂ Interval for 90% confidence



Confidence Intervals



Error bars show confidence interval

Performance of 2 algorithms on 7 different datasets.
Is the purple algorithm better than the yellow algorithm?

Statistical significance

- Calculate confidence intervals for the performance of algorithms to be compared – use desired confidence level (often 95%).
- If 2 algorithms' confidence intervals don't overlap
  - The difference in performance is **statistically significant** at that level
  - The algorithm with the higher confidence interval values is said to perform better.
- If confidence intervals overlap
  - The performance of the algorithms cannot be said to be different as the difference in performance is not statistically significant.

Confidence Intervals

- Suppose two models A and B generate accuracies of 75% and 70%. Is A better than B?
  - What confidence do you wish?
    - Say 80%
  - Depends on size of test set!
- If 100 text examples
  - A has true accuracy [69.5%,80.5%]
  - B has true accuracy [64.5%,75.5%]
  - No – at 80% confidence

Overlapping intervals

- If 1000 text examples
  - A has true accuracy [73.2%, 76.7%]
  - B has true accuracy [68.1%, 71.8%]
  - Yes – at 80% confidence

Disjoint intervals

Summary

- Evaluation on training data is over-optimistic
  - various alternative experimental designs
- After any evaluation
  - all data may be used to build final classifier
- There are a number of measures which can be used to evaluate models.
- Metrics used for class prediction include: accuracy, error, precision, recall, F1-measure, sensitivity, specificity, ROC.
- Experimental design requires careful consideration of assumptions made, outliers and the balance of the data.
- Two stage evaluation: model tuning, final evaluation.
- Results comparison should consider statistical significance