ROBERT GORDON UNIVERSITY ABERDEEN

Revision: Evaluation

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.

Contents

- **Predictive model evaluation**
  - **Classification metrics**
  - Experimental design
- Clustering evaluation
- Association rules evaluation

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: 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.

Training and Testing

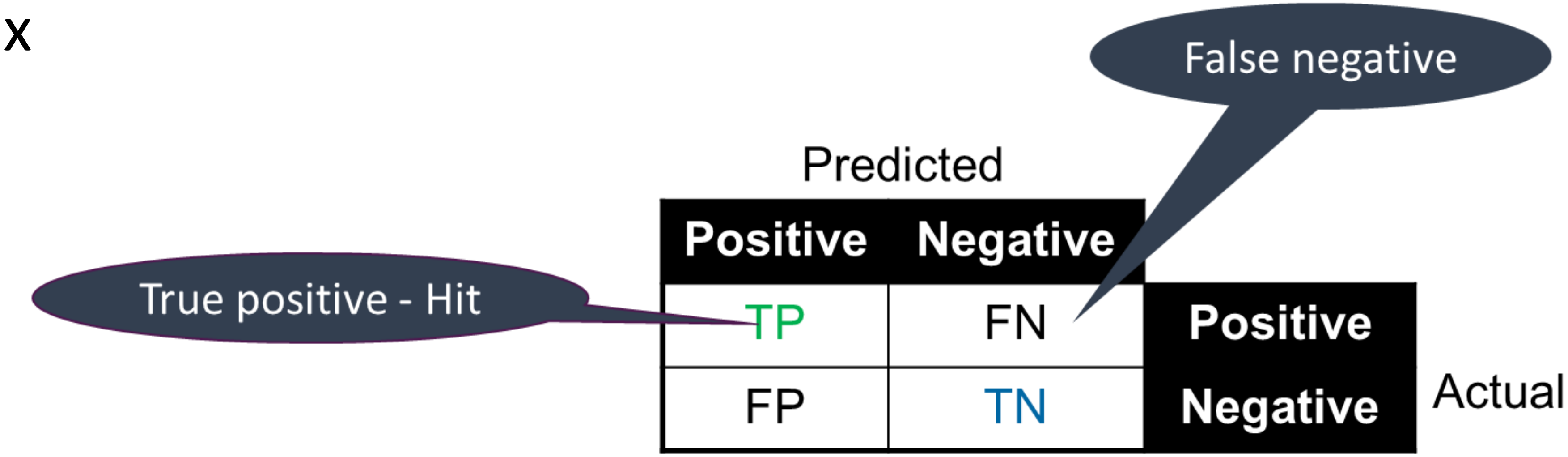
- Resubstitution error
  - Error rate from training data.
  - Hopelessly optimistic!
- Over-fitting of data
- Solution: Split data into training and testing set.
- Test set
  - Independent instances that play no part in learning of classifier.
- Assumption: training and test data are representative samples of the underlying problem.
- What evaluation measures? Depends on whether prediction is a class or a numeric value.

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



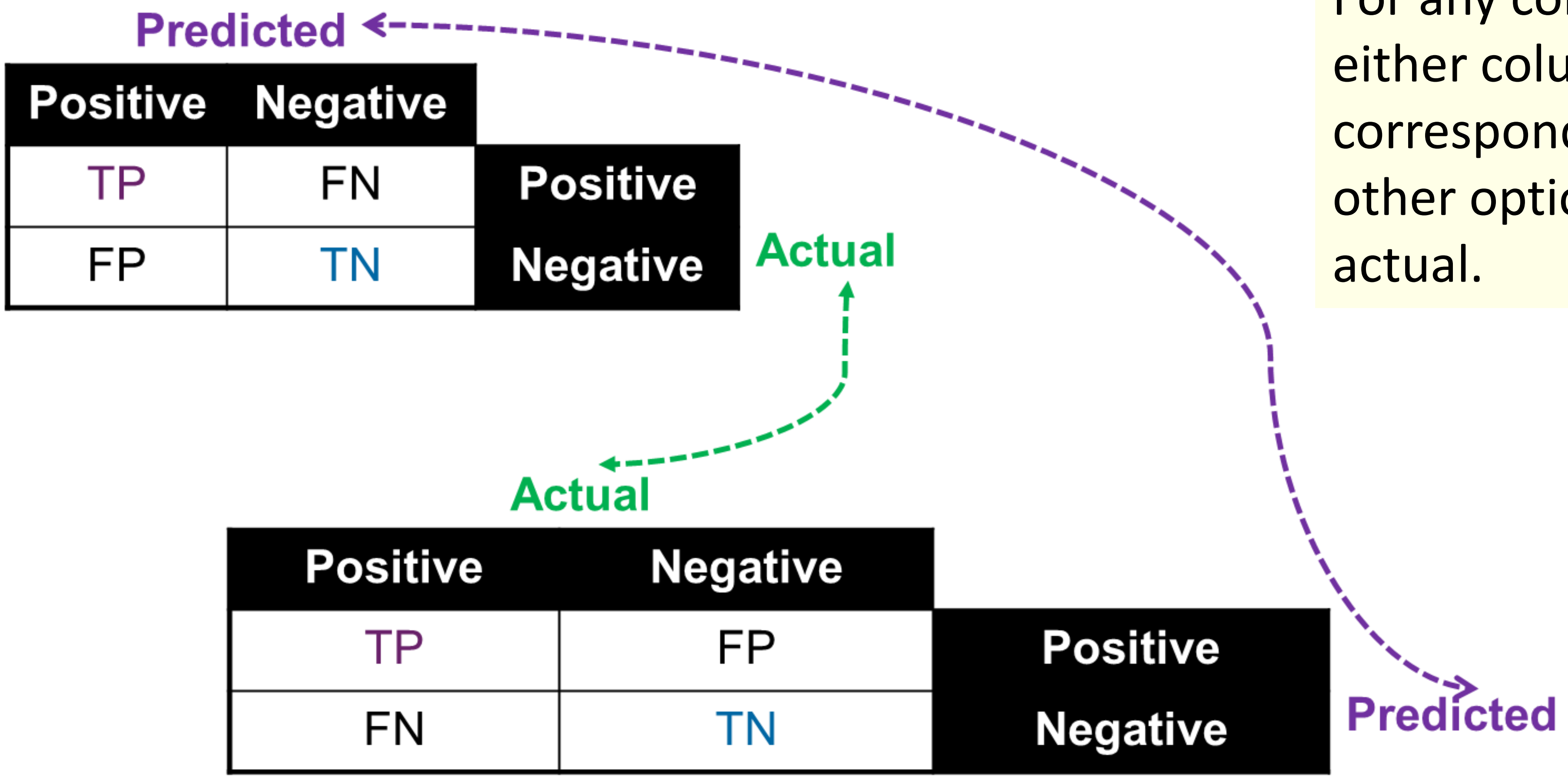
		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

True positive - Hit

False negative

- Predictive Accuracy $= 100 * \dfrac{TP+TN}{TP+TN+FP+FN}$
  - 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.

Accuracy

- Measures the proportion of predictions which are correct.
  - Regardless of the class
- Used widely
- Sometimes error used instead.
  - Error = 1 – accuracy.
- Can be extremely problematic when used with imbalanced datasets, i.e. datasets where one class dominates.

Accuracy example – binary classification

- Test examples contain 500 positives and 500 negatives.

Predicted			
Positive	Negative		
400	100	Positive	Actual
200	300	Negative	

- 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”

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 β = 1.
- β > 1 gives more weight to recall.
- β < 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

- Used in binary classification tests.
- **Sensitivity:** classifier’s ability to correctly classify positive cases – proportion of positives that are correctly classified.
  - $Sensitivity = \frac{TP}{TP+FN}$
- **Specificity:** classifier’s ability to correctly classify negative cases – proportion of negatives that are correctly classified.
  - $Specificity = \frac{TN}{TN+FP}$

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
Total = 14+6+9+28 =

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

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

- Takes into account chance agreement.
- 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: $0.80 \leq k < 1.00$

Contents

- Predictive model evaluation
  - Classification metrics
  - Experimental design
- Clustering evaluation
- Association rules evaluation

Experimental design

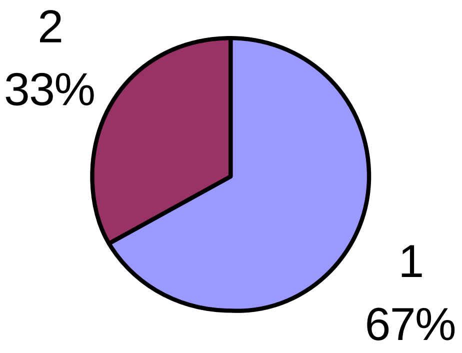
- Design experiments to build and evaluate model.
- Consider bias vs. variance trade-off.
- Consider dataset imbalance – dataset where one type of target scenario is much more represented than another one.
- Lots of data required! Need data to
    - Tune models
        - Optimise parameter values
  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

Making the most of data

- Need data to
    - Tune models
        - Dataset divided into
            - Training set: to build the models with different parameter values
            - Test set: to test the model built with specific parameter settings
            - Once testing is completed model is built with all the data.
        - Methods below refer to this tuning - testing
    - Additional test set required
        - To test the final model, once the tuning has been optimised.

Utilising the Available Data

- Holdout procedure
  - Holdout certain amount of data for testing
    - use remainder for training
    - one third for testing (2), the rest for training (1)
- 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

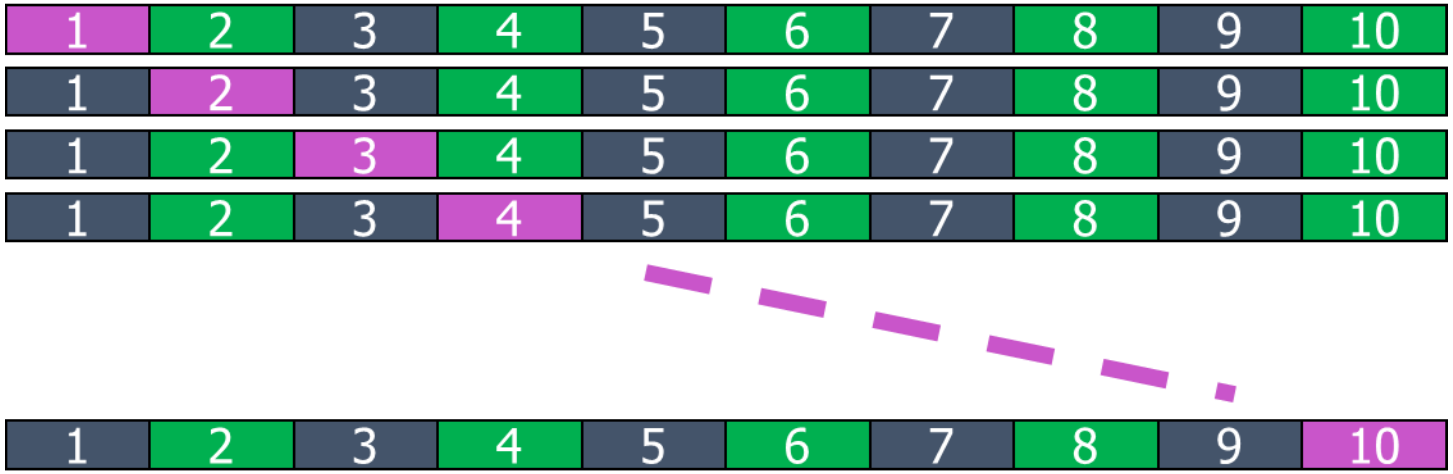
- Samples should be representative
  - Each class is represented with approximately equal proportions in both subsets
  - This is called **stratified hold-out**
- Repeated holdout
  - Randomly select test set each iteration
  - Calculate average metric (e.g. error)
- Can overlapping test sets be avoided?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

k-fold Cross-Validation

- First step
  - dataset is split into k folds/partitions of equal size. Often 10 folds used

1	2	3	4	5	6	7	8	9	10

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



1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

- Calculate metrics over k folds

Standard Cross-Validation

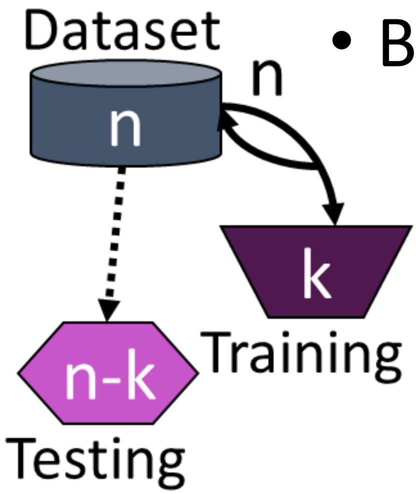
- **Stratified** ten-fold cross-validation
  - Ten-fold is known to give accurate estimate
    - 9/10ths for training and 1/10th for testing x 10 (one per fold)
  - **Stratification** - each class is properly represented in each fold
    - “Properly” = in the same proportion as in the dataset
- Repeated stratified cross-validation
  - E.g. ten-fold cross-validation repeated 3 times
    - Results averaged over 30 experiments (10 folds x 3 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 testing
    - 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 – out of bag sample
  - training set has only ~63.2% of dataset in it

… bootstrap Estimation

- Estimates on test data (holdout) can be 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

Other

- When comparing models
  - The statistical significance of the difference in results needs to be assessed.
    - Calculate/plot confidence intervals:
      - if the intervals overlap: the difference in results is not statistically significant
      - If the intervals do not overlap, the difference in results is statistically significant.
      - E.g. comparing accuracies, the model with the highest accuracy can be said to better in terms of accuracy only if the confidence intervals do NOT overlap.
  - The experimental design (e.g. 10-fold cross validation) should normally be the same for a fair comparison. Also use the same data partitions.
  - Compare various aspects, e.g. :
    - Confusion matrices – where are the errors?
    - Tree sizes (if applicable)
    - Is the model easy to understand?
    - Performance time for new predictions
    - Model building time if model requires frequent re-building with new data.

Contents

- Predictive model evaluation
  - Classification metrics
  - Experimental design
- **Clustering evaluation**
- Association rules evaluation

Evaluating cluster quality

- Good clustering if
  - High-level similarity within each cluster – high **cohesion**
    - Low within cluster sum of squares (WC)
  - Long distance between clusters – high **separation**
    - Sum of distances between clusters (BC) is high
  - Combination of cohesion and separation
    - **BC/WC** - good indicator of overall quality.
  - But also number of clusters – trade-off between cohesion and the number of clusters.

Clustering – how many clusters?

- In labs you have seen two measures which are used to determine the optimal number of clusters:
  - Within clusters sum of squares – uses cohesion.
  - Silhouette – uses separation and cohesion.
- What is the distribution of instances in clusters?
  - Is there any cluster with virtually no instances?
  - Is there any cluster with most instances?

Discussion

- Evaluation of clustering
    - Often by inspection - do the clusters “make sense”?
    - Clusters can be visualised if low dimensionality
    - “Classes to clusters” – known class values vs clusters.
        - Need a class attribute (used only for evaluation) to check if cluster members share the class value
- Interpretation of clusters
    - Supervised learning in a post-processing step
        - Training data is clustered data labelled by cluster id
        - Decision tree or rule-set inferred to predict cluster

Contents

- Predictive model evaluation
  - Classification metrics
  - Experimental design
- Clustering evaluation
- **Association rules evaluation**

Association rule measures

- **Support** (coverage) applies to an itemset - set of items which appear together in a number of instances
  - Proportion of instances where all the items in an itemset appear together
- **Confidence** (accuracy) – applies to a rule
  - It is the accuracy of the rule, i.e. the proportion of instances where the rule is true out of the number of times when the rule is applicable.
- **Lift** - The ratio of the observed support to that expected if rule condition and conclusion were independent.
  - The rise (or decrease) in probability of the conclusion of the rule being true if we have observed that the condition of the rule is true.
    - 1 if condition (LHS) and conclusion (RHS) are independent.
    - > 1 if presence of condition makes conclusion more likely.
    - < 1 if presence of condition makes conclusion less likely.

Association rule interpretation

- Rule overall rating depends on
  - Support, confidence and lift
  - The usefulness of the rule
- Rule usefulness
  - Some rules may have high support, confidence and lift but be of little use
  - Other rules may have lower values for the metrics, but be more useful.

Summary

- Different evaluation methods are used for different types of data mining algorithms.
- Predictive model evaluation
  - Classification: accuracy, kappa, precision, recall, specificity, sensitivity.
  - Regression: mean absolute error, mean squared error, root mean squared error, R-squared, adjusted R-squared.
- Clustering evaluation: cohesion and separation for quality of clusters, but visual inspection or classification algorithm applied to learn what the clusters mean to check that clusters make sense.
- Association rules evaluation; support, confidence, lift.
- In all cases, the usefulness of the findings should be assessed as well as any bias.