



Data Mining with Weka

Class 2 – Lesson 1

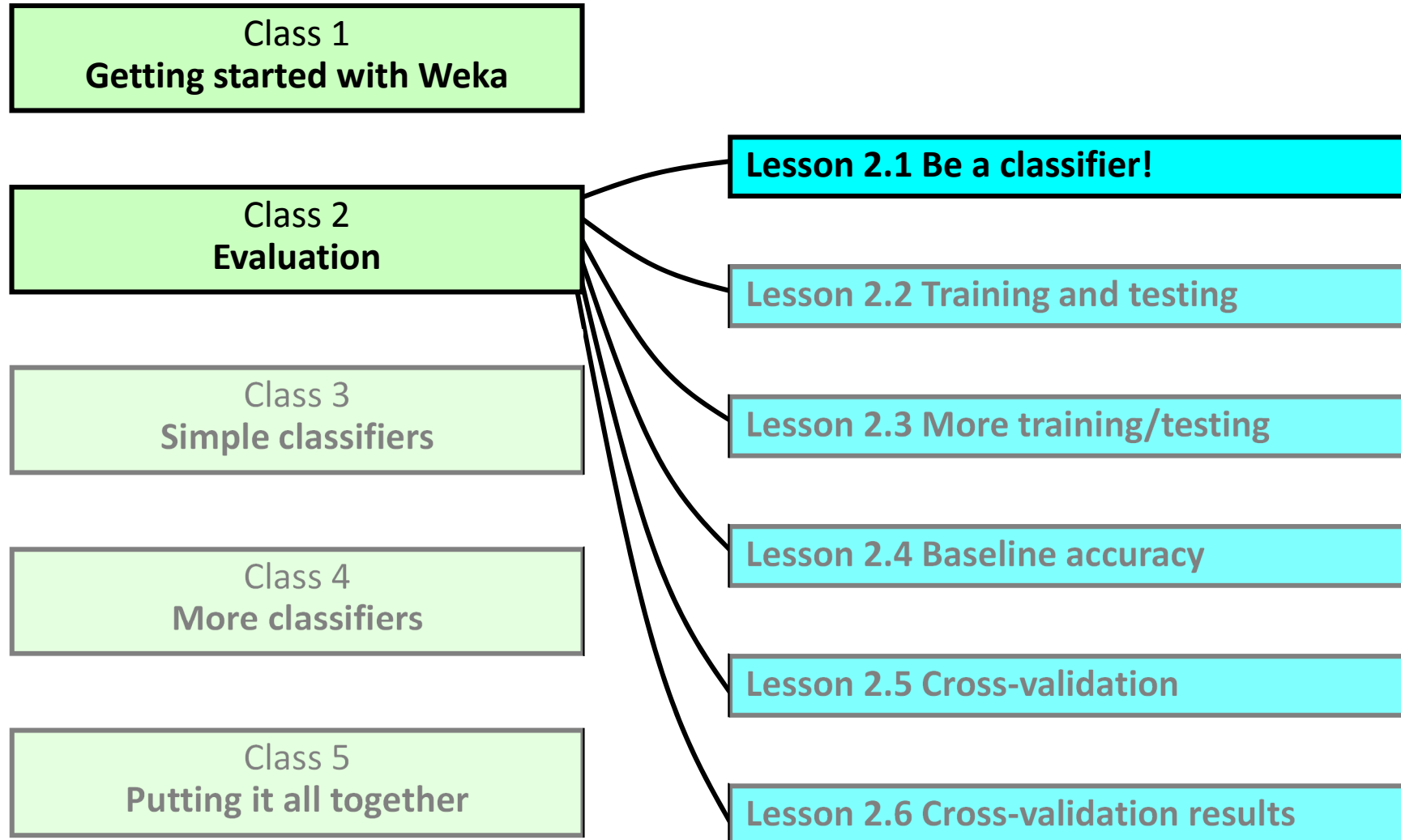
Be a classifier!

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 2.1: Be a classifier!



Lesson 2.1: Be a classifier!

Interactive decision tree construction

- ❖ Load `segmentchallenge.arff`; look at dataset
- ❖ Select `UserClassifier` (tree classifier)
- ❖ Use the test set `segmenttest.arff`
- ❖ Examine data visualizer and tree visualizer
- ❖ Plot `regioncentroidrow` vs `intensitymean`
- ❖ Rectangle, Polygon and Polyline selection tools
- ❖ ... several selections ...
- ❖ Rightclick in `Tree visualizer` and `Accept the tree`

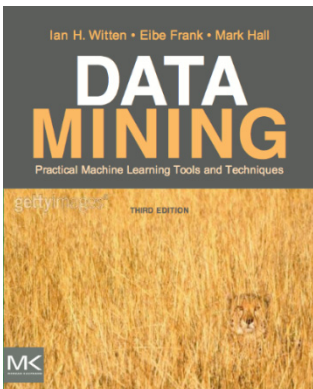
Over to you: how well can you do?

Lesson 2.1: Be a classifier!

- ❖ Build a tree: what strategy did you use?
- ❖ Given enough time, you could produce a “perfect” tree for the dataset
 - but would it perform well on the test data?

Course text

- ❖ Section 11.2 *Do it yourself: the User Classifier*





Data Mining with Weka

Class 2 – Lesson 2

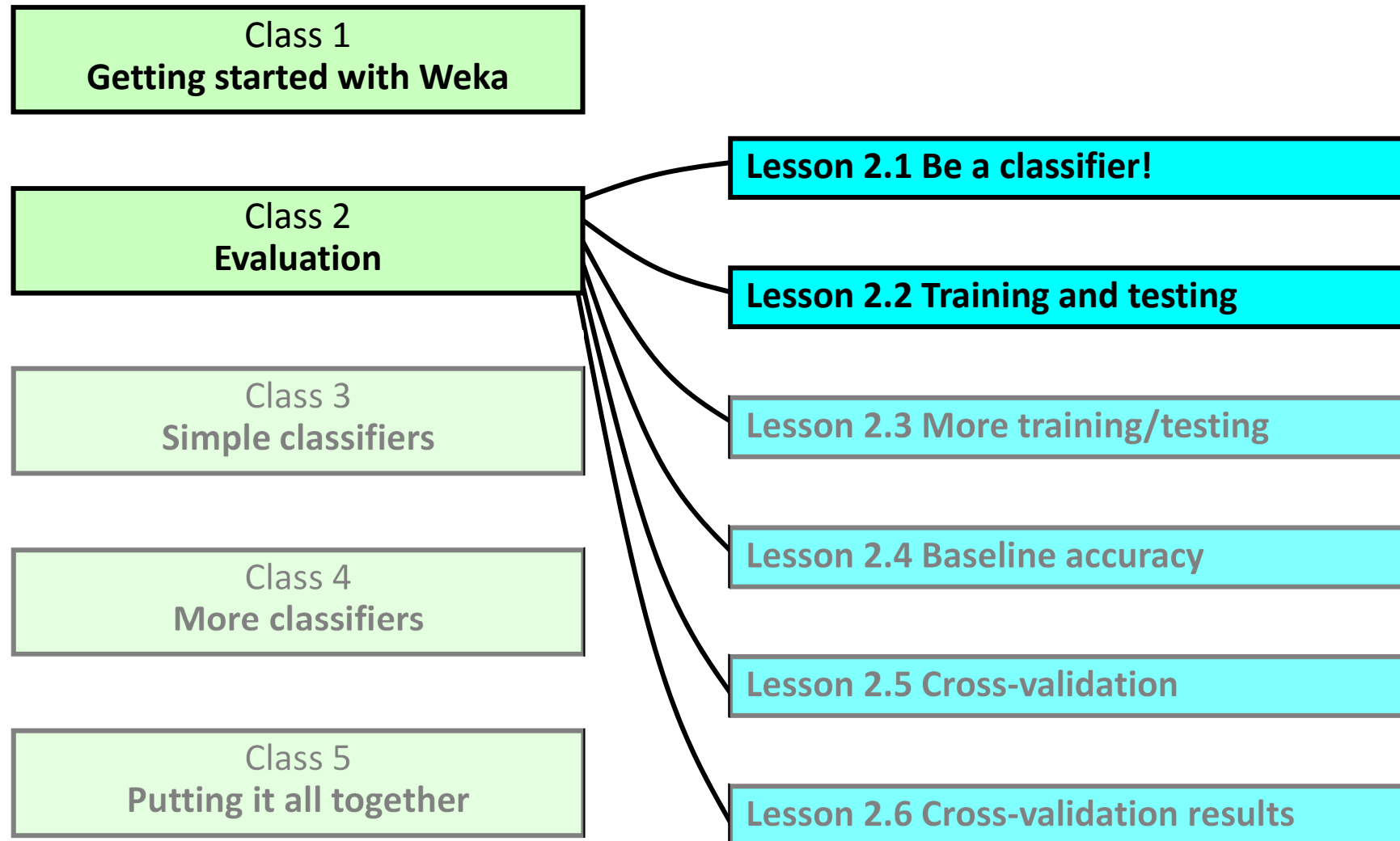
Training and testing

Ian H. Witten

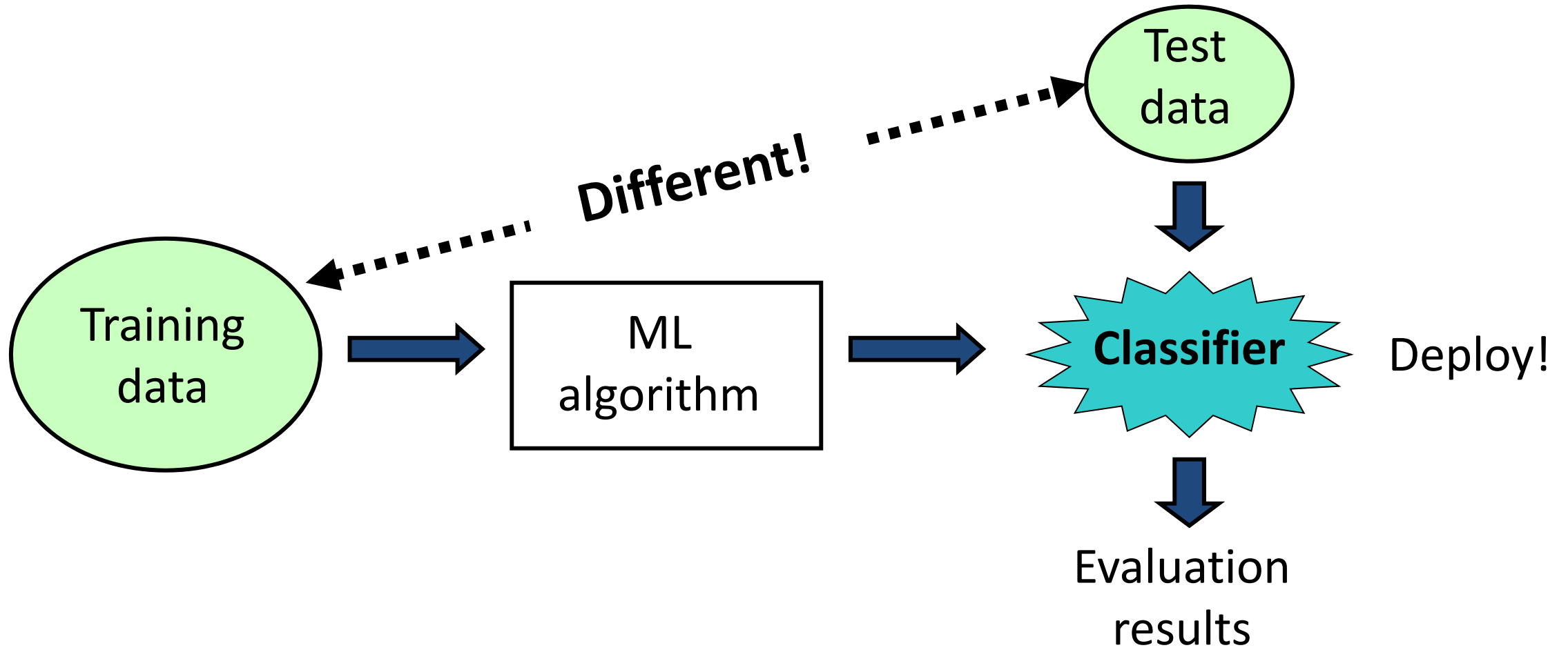
Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

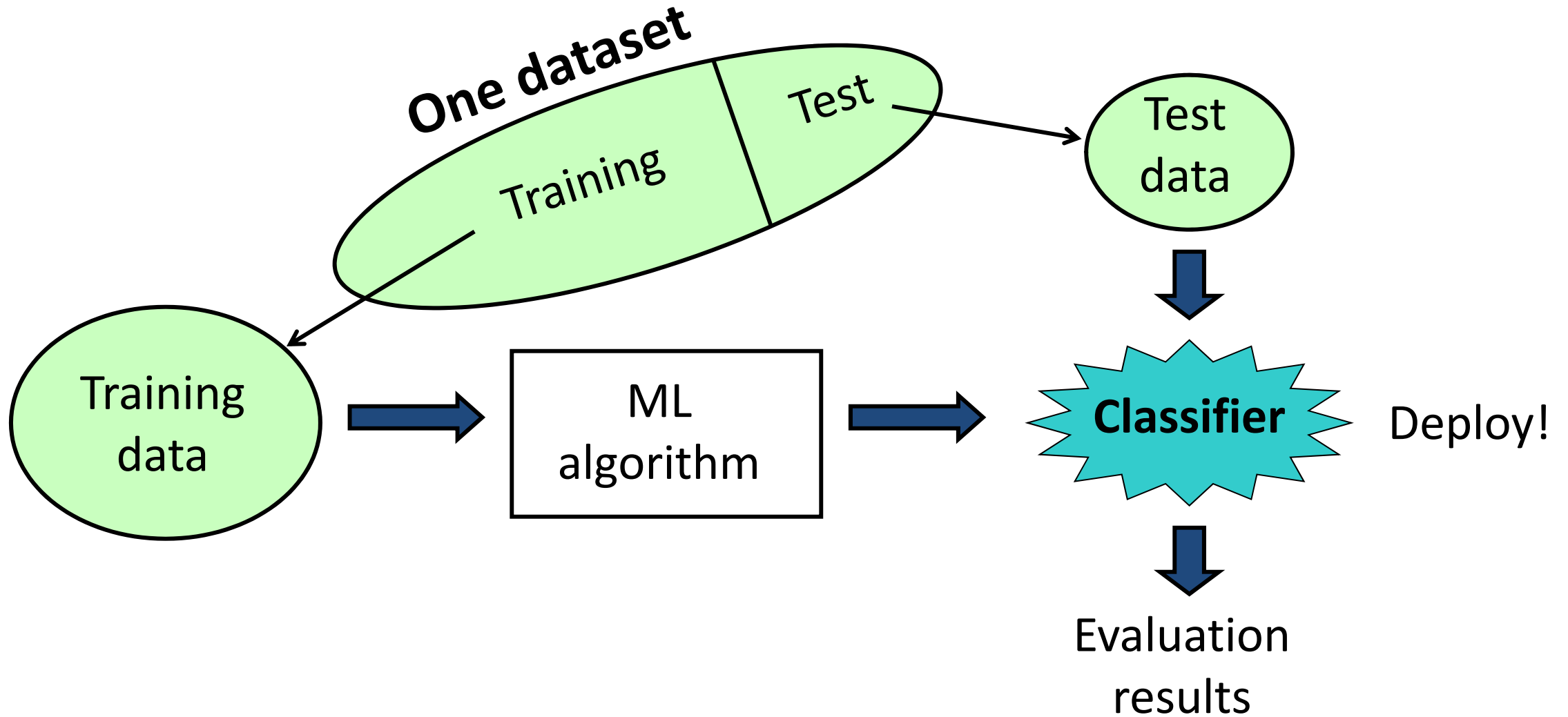
Lesson 2.2: Training and testing



Lesson 2.2: Training and testing



Lesson 2.2: Training and testing



Basic assumption: training and test sets produced by independent sampling from an infinite population

Lesson 2.2: Training and testing

Use J48 to analyze the segment dataset

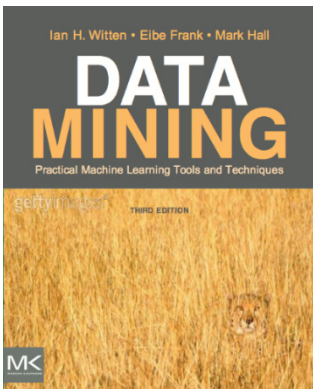
- ❖ Open file `segment-challenge.arff`
- ❖ Choose J48 decision tree learner (`trees>J48`)
- ❖ Supplied test set `segment-test.arff`
- ❖ Run it: 96% accuracy
- ❖ Evaluate on training set: 99% accuracy
- ❖ Evaluate on percentage split: 95% accuracy
- ❖ Do it again: get exactly the same result!

Lesson 2.2: Training and testing

- ❖ Basic assumption:
training and test sets sampled independently from an infinite population
- ❖ Just one dataset? — hold some out for testing
- ❖ Expect slight variation in results
- ❖ ... but Weka produces same results each time
- ❖ J48 on segment-challenge dataset

Course text

- ❖ Section 5.1 *Training and testing*





Data Mining with Weka

Class 2 – Lesson 3

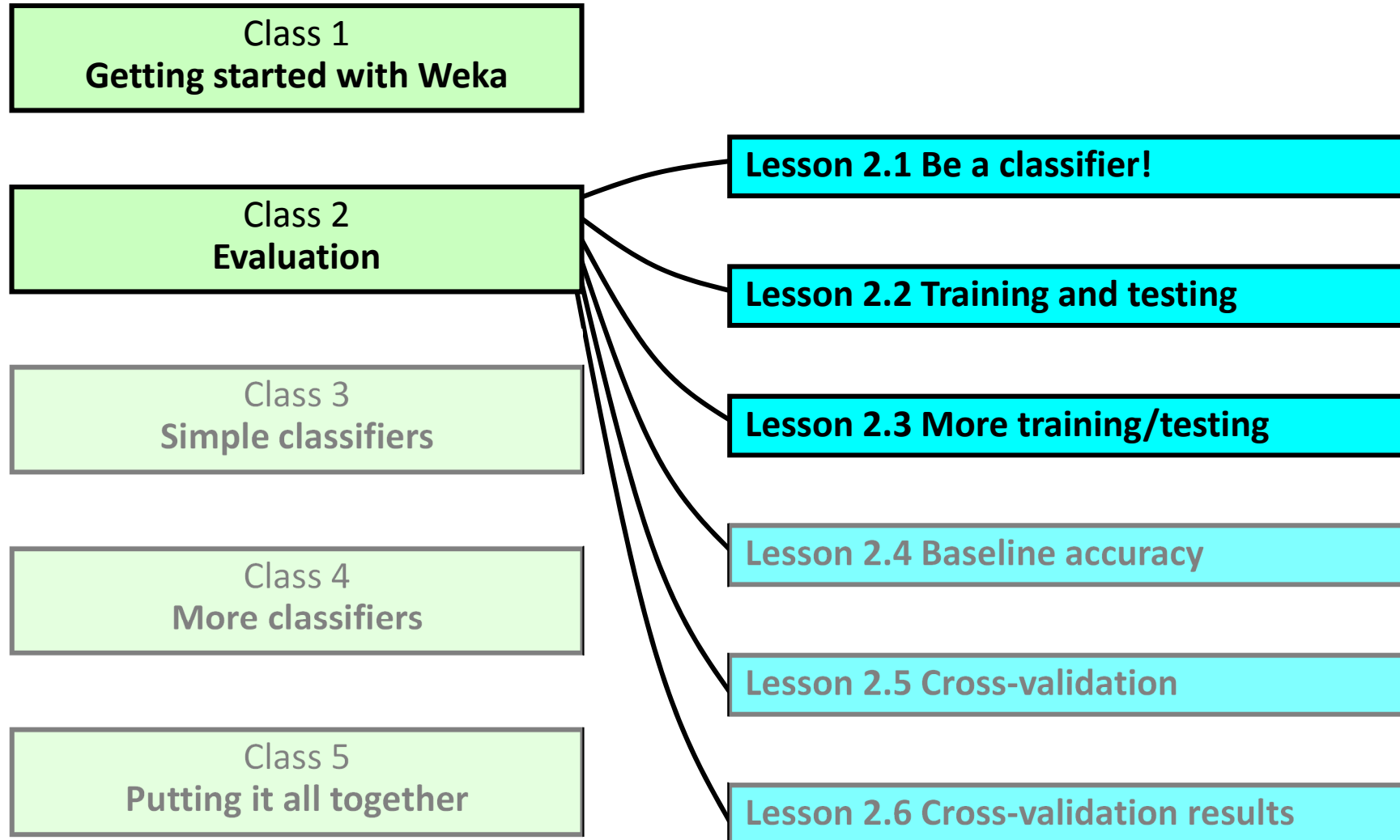
Repeated training and testing

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

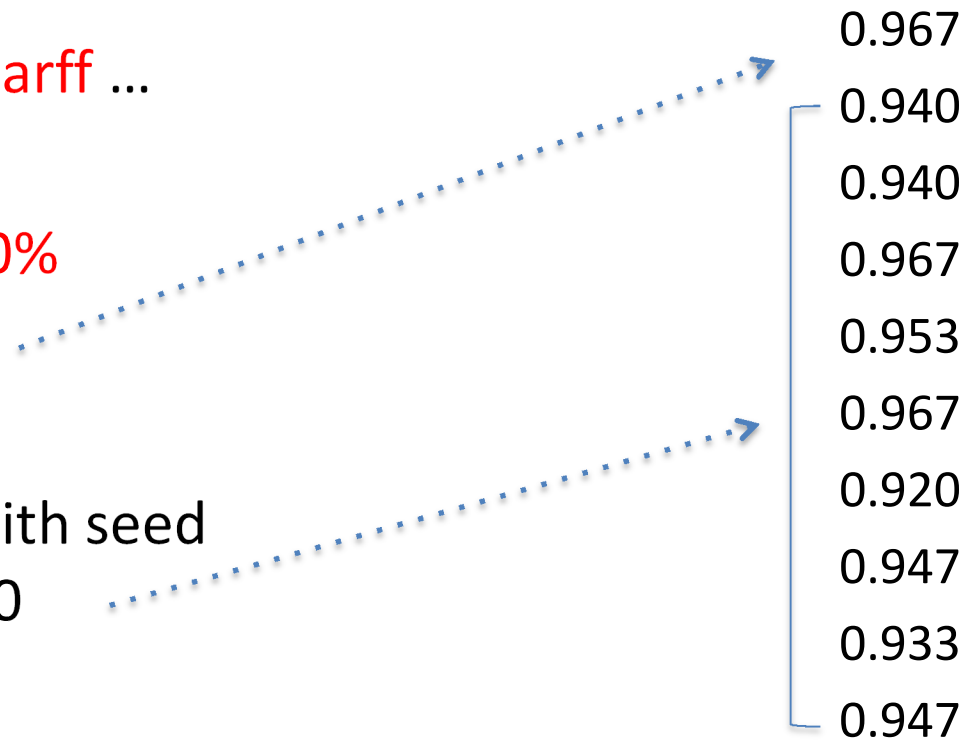
Lesson 2.3: Repeated training and testing



Lesson 2.3: Repeated training and testing

Evaluate J48 on segment-challenge

- ❖ With `segment-challenge.arff` ...
- ❖ and J48 (`trees>J48`)
- ❖ Set `percentage split` to `90%`
- ❖ Run it: 96.7% accuracy
- ❖ Repeat
- ❖ [`More options`] Repeat with seed
2, 3, 4, 5, 6, 7, 8, 9 10



Lesson 2.3: Repeated training and testing

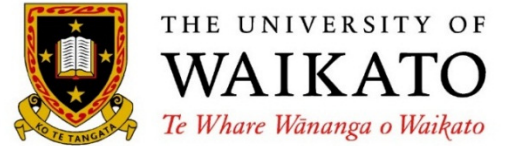
Evaluate J48 on segment-challenge

Sample mean	$\bar{x} = \frac{\sum x_i}{n}$	0.967
		0.940
		0.940
		0.967
Variance	$\sigma^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$	0.953
		0.967
		0.920
Standard deviation	σ	0.947
		0.933
		0.947

$$\bar{x} = 0.949, \sigma = 0.018$$

Lesson 2.3: Repeated training and testing

- ❖ Basic assumption:
training and test sets sampled independently from an infinite population
- ❖ Expect slight variation in results ...
- ❖ ... get it by setting the random-number seed
- ❖ Can calculate mean and standard deviation experimentally



Data Mining with Weka

Class 2 – Lesson 4

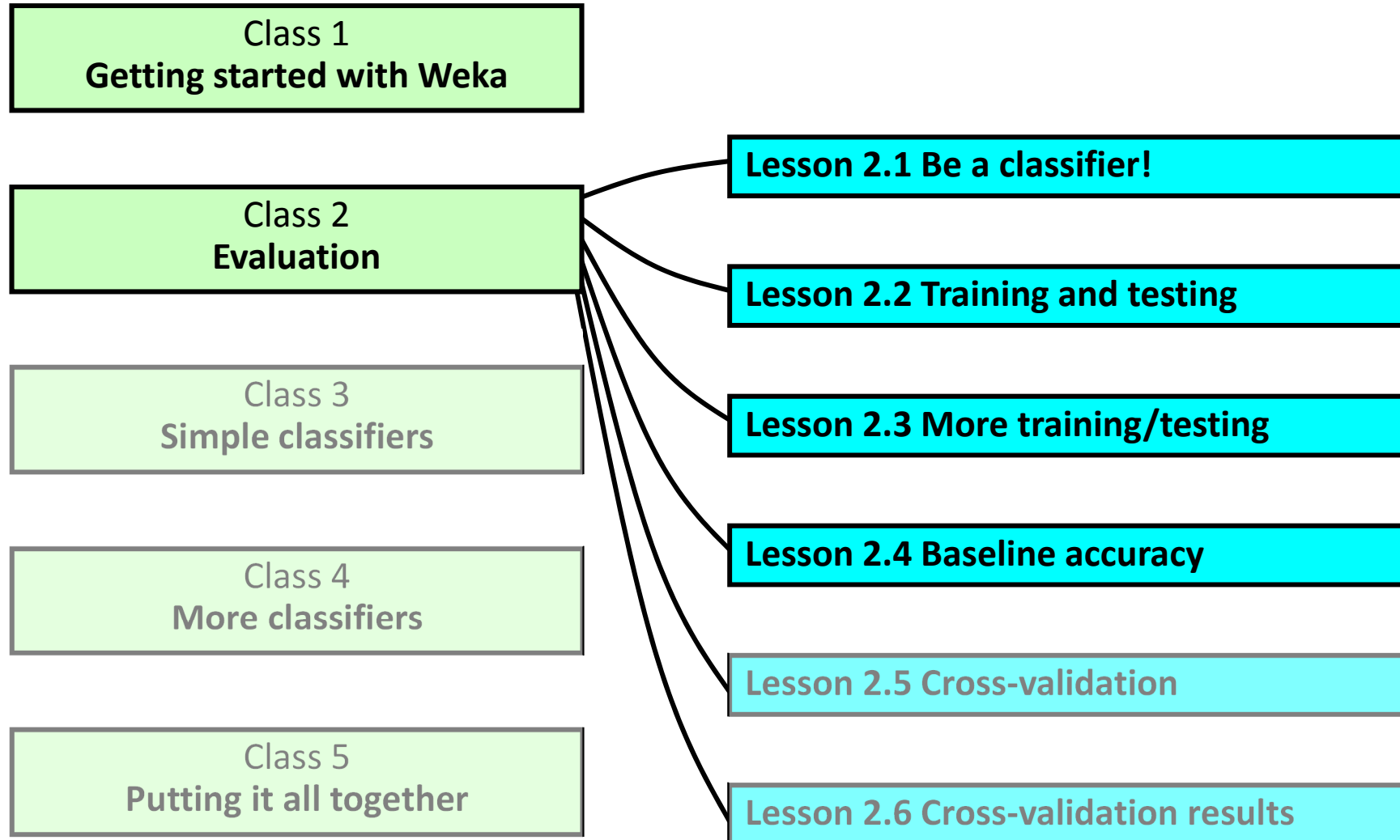
Baseline accuracy

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 2.4: Baseline accuracy



Lesson 2.4: Baseline accuracy

Use diabetes dataset and default holdout

- ❖ Open file **diabetes.arff**
- ❖ Test option: Percentage split
- ❖ Try these classifiers:
 - **trees > J48** **76%**
 - **bayes > NaiveBayes** **77%**
 - **lazy > IBk** **73%**
 - **rules > PART** **74%**

(we'll learn about them later)
- ❖ 768 instances (500 negative, 268 positive)
- ❖ Always guess "negative": 500/768 **65%**
- ❖ **rules > ZeroR**: most likely class!

Lesson 2.4: Baseline accuracy

Sometimes baseline is best!

- ❖ Open **supermarket.arff** and blindly apply
 - rules > ZeroR* 64%
 - trees > J48* 63%
 - bayes > NaiveBayes* 63%
 - lazy > IBk* 38% (!!)
 - rules > PART* 63%
- ❖ Attributes are not informative
- ❖ Don't just apply Weka to a dataset:
you need to understand what's going on!

Lesson 2.4: Baseline accuracy

- ❖ Consider whether differences are likely to be significant
- ❖ Always try a simple baseline,
e.g. **rules** > **ZeroR**
- ❖ Look at the dataset
- ❖ Don't blindly apply Weka:
try to understand what's going on!



Data Mining with Weka

Class 2 – Lesson 5

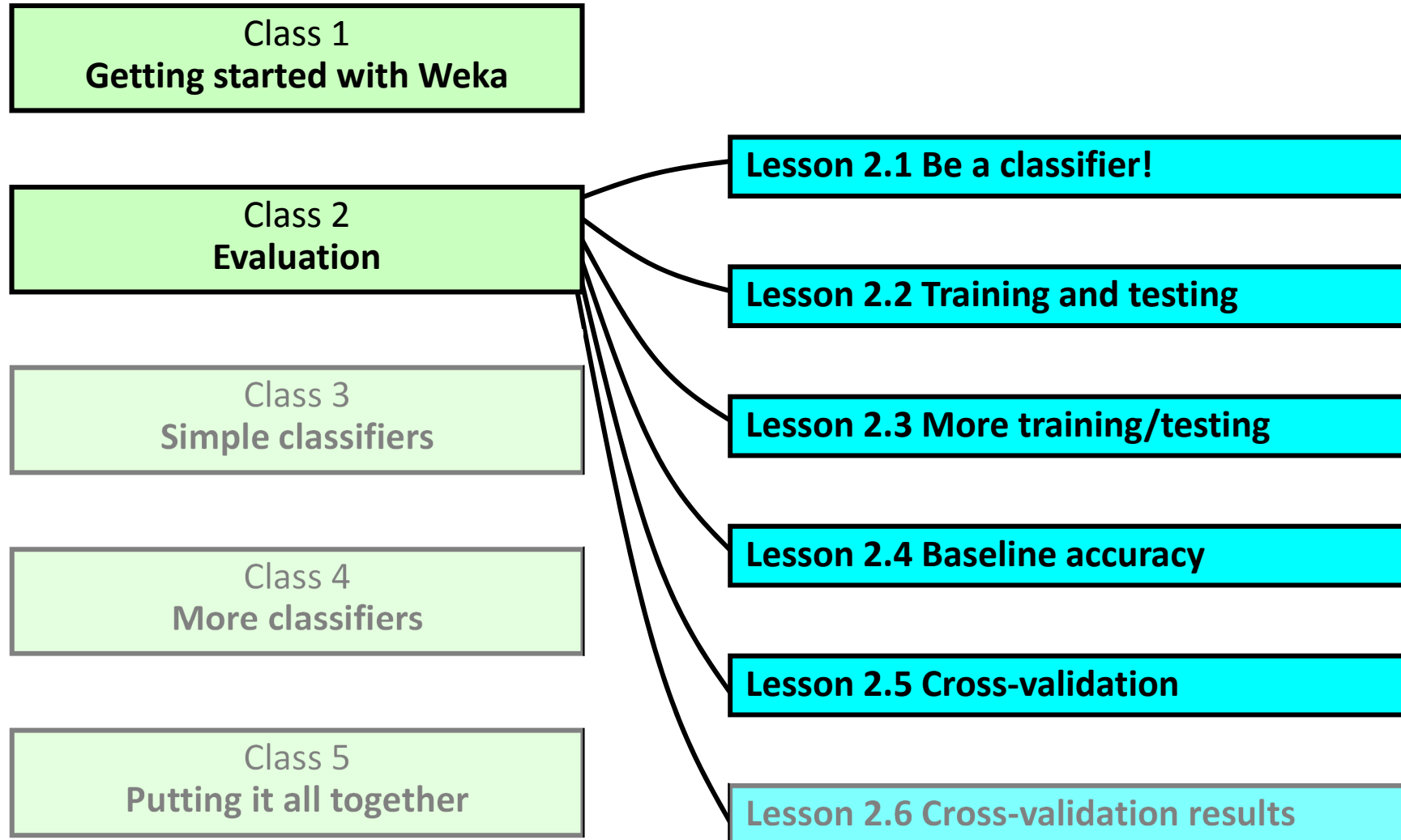
Cross-validation

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 2.5: Cross-validation

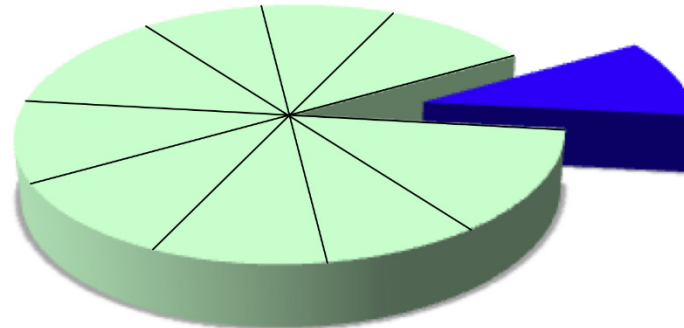
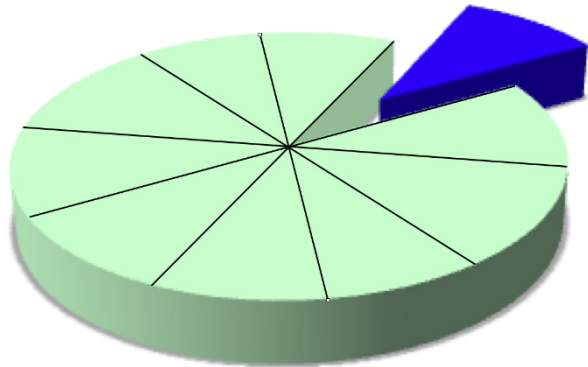
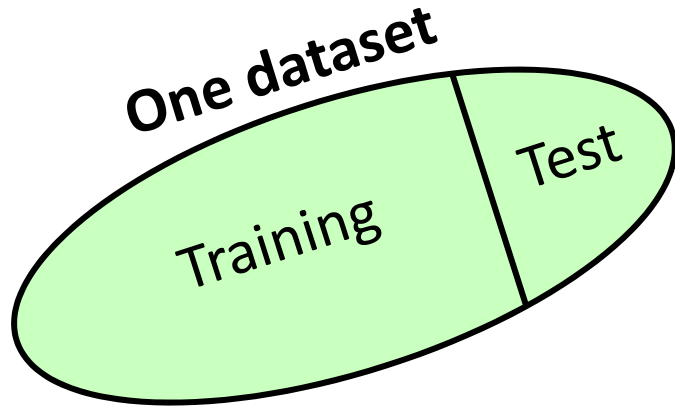


Lesson 2.5: Cross-validation

- ❖ Can we improve upon repeated holdout?
(i.e. reduce variance)
- ❖ Cross-validation
- ❖ Stratified cross-validation

Lesson 2.5: Cross-validation

- ❖ Repeated holdout
(in Lesson 2.3, hold out 10% for testing, repeat 10 times)

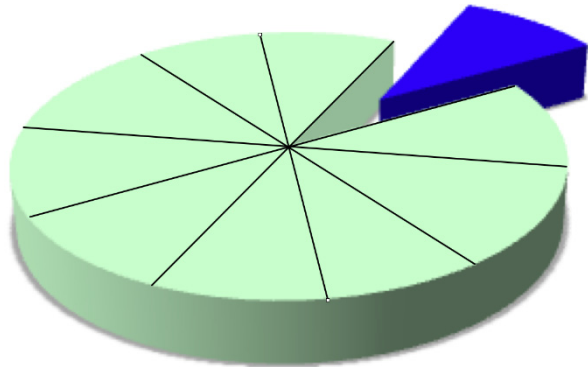


(repeat 10 times)

Lesson 2.5: Cross-validation

10-fold cross-validation

- ❖ Divide dataset into 10 parts (folds)
- ❖ Hold out each part in turn
- ❖ Average the results
- ❖ Each data point used once for testing, 9 times for training

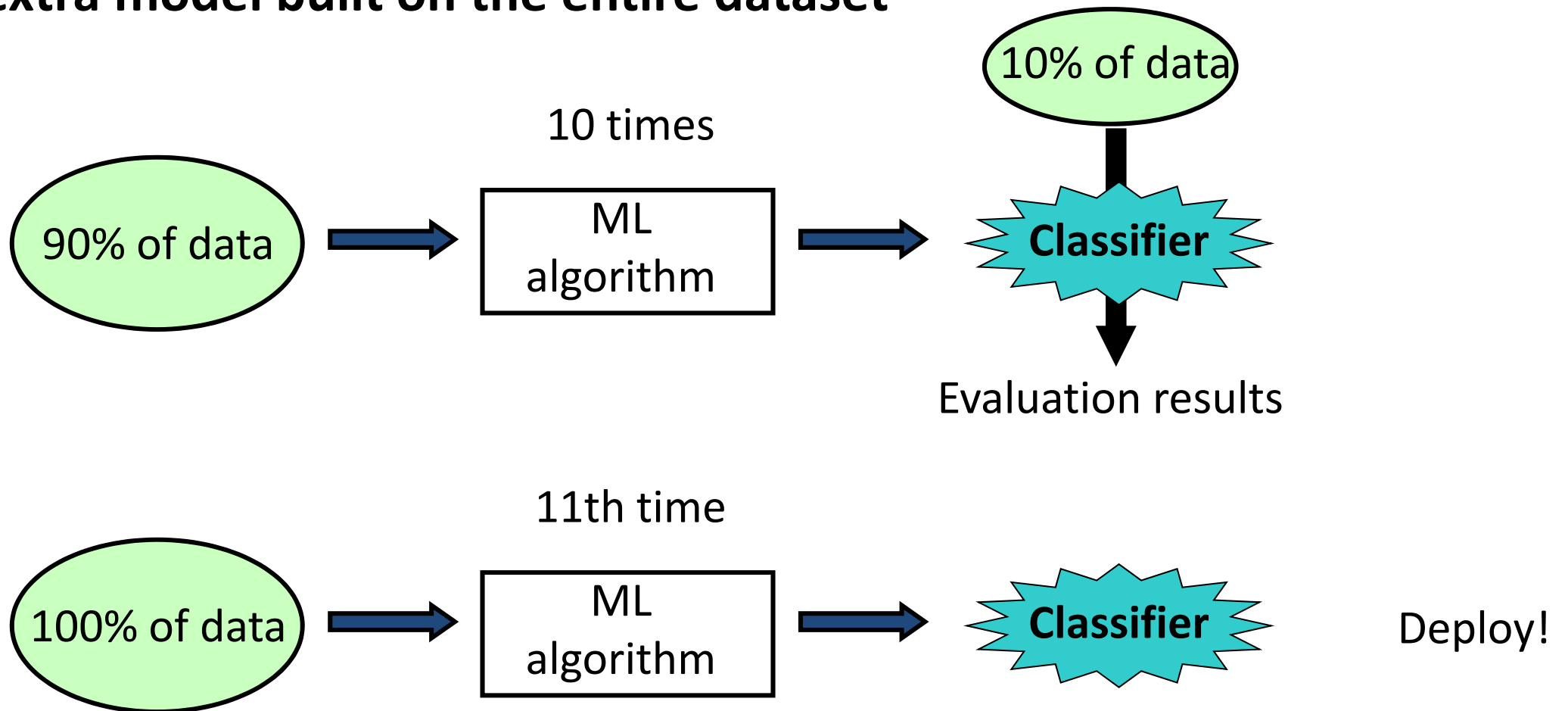


Stratified cross-validation

- ❖ Ensure that each fold has the right proportion of each class value

Lesson 2.5: Cross-validation

After cross-validation, Weka outputs an extra model built on the entire dataset

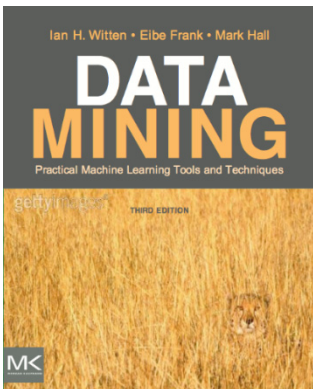


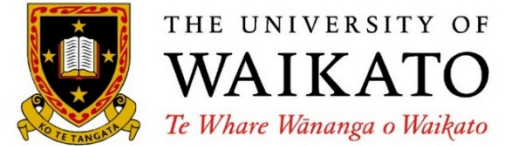
Lesson 2.5: Cross-validation

- ❖ Cross-validation better than repeated holdout
- ❖ Stratified is even better
- ❖ With 10-fold cross-validation, Weka invokes the learning algorithm 11 times
- ❖ **Practical rule of thumb:**
- ❖ Lots of data? – use percentage split
- ❖ Else stratified 10-fold cross-validation

Course text

- ❖ Section 5.3 *Cross-validation*





Data Mining with Weka

Class 2 – Lesson 6

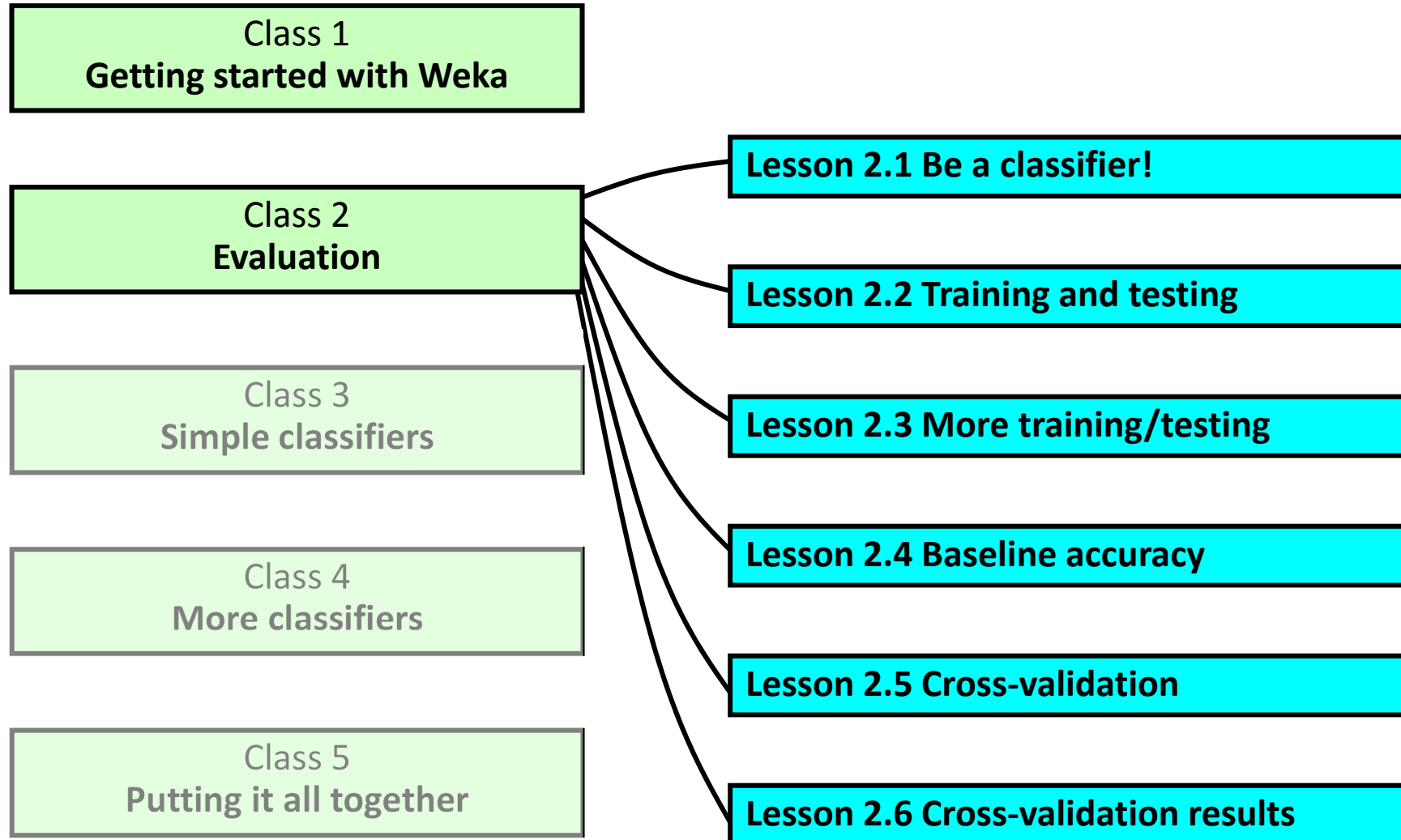
Cross-validation results

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 2.6: Cross-validation results



Lesson 2.6: Cross-validation results

Is cross-validation really better than repeated holdout?

❖ **Diabetes** dataset

❖ Baseline accuracy (**rules > ZeroR**): 65.1%

❖ **trees > J48**

❖ 10-fold cross-validation 73.8%

❖ ... with different random number seed

1	2	3	4	5	6	7	8	9	10
73.8	75.0	75.5	75.5	74.4	75.6	73.6	74.0	74.5	73.0

Lesson 2.6: Cross-validation results

		holdout (10%)	cross-validation (10-fold)
Sample mean	$\bar{x} = \frac{\sum x_i}{n}$	75.3	73.8
		77.9	75.0
		80.5	75.5
		74.0	75.5
Variance	$\sigma^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$	71.4	74.4
		70.1	75.6
		79.2	73.6
Standard deviation	σ	71.4	74.0
		80.5	74.5
		67.5	73.0
		$\bar{x} = 74.8$	$\bar{x} = 74.5$
		$\sigma = 4.6$	$\sigma = 0.9$

Lesson 2.6: Cross-validation results

- ❖ Why 10-fold? E.g. 20-fold: 75.1%
- ❖ Cross-validation really is better than repeated holdout
- ❖ It reduces the variance of the estimate



Data Mining with Weka

Department of Computer Science
University of Waikato
New Zealand



Creative Commons Attribution 3.0 Unported License



creativecommons.org/licenses/by/3.0/

weka.waikato.ac.nz