THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

Class 1 – Lesson 1

Introduction

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Advanced Data Mining with Weka

… a practical course on how to use popular "packages" in Weka for data mining

… follows on from earlier courses

Data Mining with Weka

More Data Mining with Weka

… will pick up some basic principles along the way

... and look at some specific application areas

Ian H. Witten + Waikato data mining team

University of Waikato, New Zealand

Advanced Data Mining with Weka

- **As you know, a Weka is**
  - *a bird found only in New Zealand?*
  - ***Data mining workbench**:*
    Waikato Environment for Knowledge Analysis

Machine learning algorithms for data mining tasks

- classification, data preprocessing
- feature selection, clustering, association rules, etc

Weka 3.7/3.8: Cleaner core, plus package system for new functionality

- some packages do things that were standard in Weka 3.6
- many others
- users can distribute their own packages

Advanced Data Mining with Weka

What will you learn?

- *How to use packages*
- *Time series forecasting: the time series forecasting package*
- *Data stream mining: incremental classifiers*
- *The MOA system for Massive Online Analysis*
- *Weka's MOA package*
- *Interface to R: using R facilities from Weka*
- *Distributed processing using Apache SPARK*
- *Scripting Weka in Python: the Jython package and the Python Weka wrapper*
- *Applications: analyzing soil samples, neuroimaging with functional MRI data, classifying tweets and images, signal peptide prediction*

Use Weka on your own data … and understand what you're doing!

Advanced Data Mining with Weka

- This course assumes that you know about data mining
  ... and are an advanced user of Weka
- See *Data Mining with Weka*
  and *More Data Mining with Weka*
- (Refresher: see videos on YouTube WekaMOOC channel)

The Waikato data mining team (in order of appearance)

Ian Witten
(Class 1)

Geoff Holmes
(Lesson 1.6)

Albert Bifet
(Class 2)

Bernhard Pfahringer
(Lesson 2.4)

Tony Smith
(Lesson 2.6)

Eibe Frank
(Class 3)

Pamela Douglas
(Lesson 3.6)

Mark Hall
(Class 4)

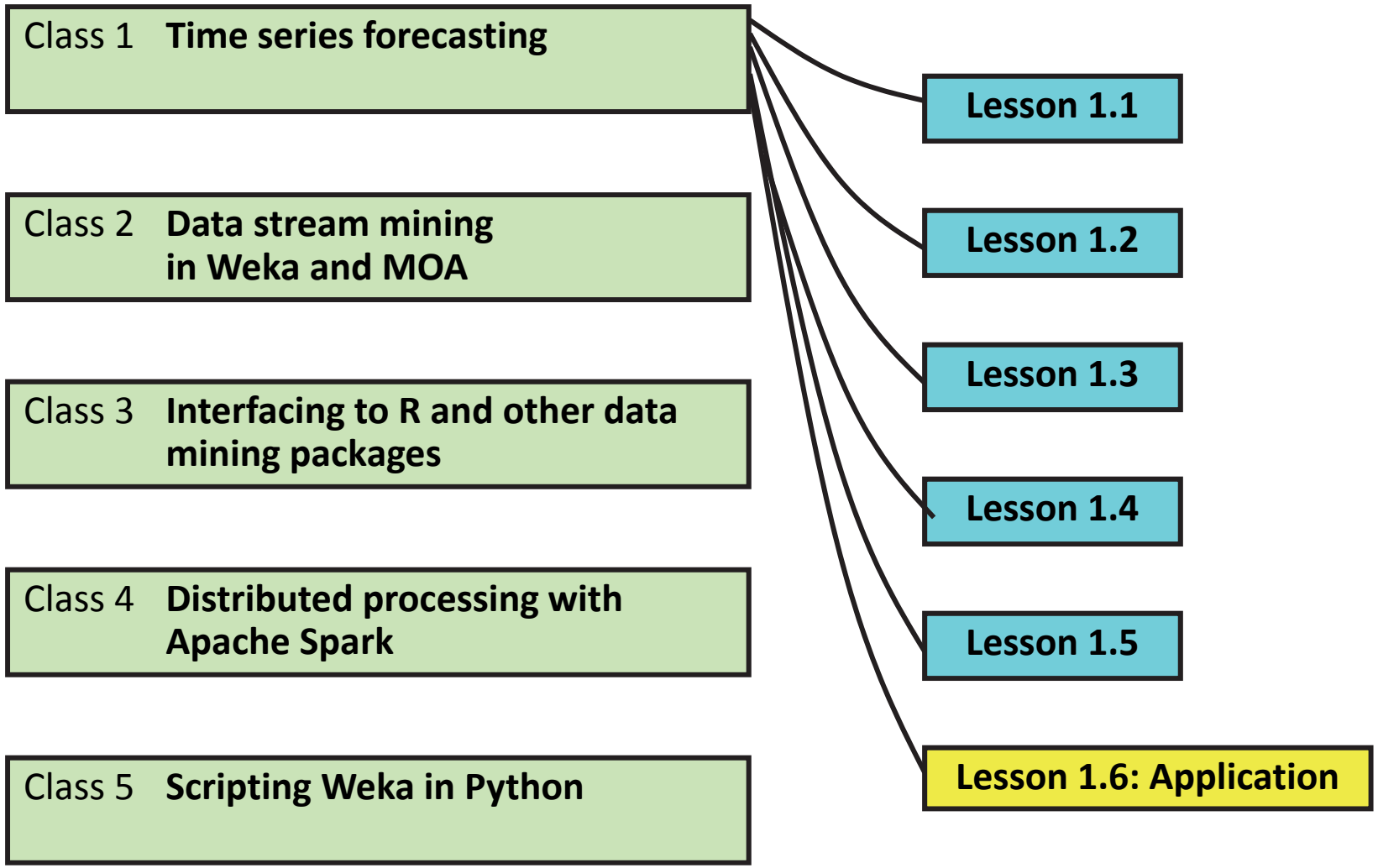
Mike Mayo
(Lesson 4.6)

Peter Reutemann
(Class 5)

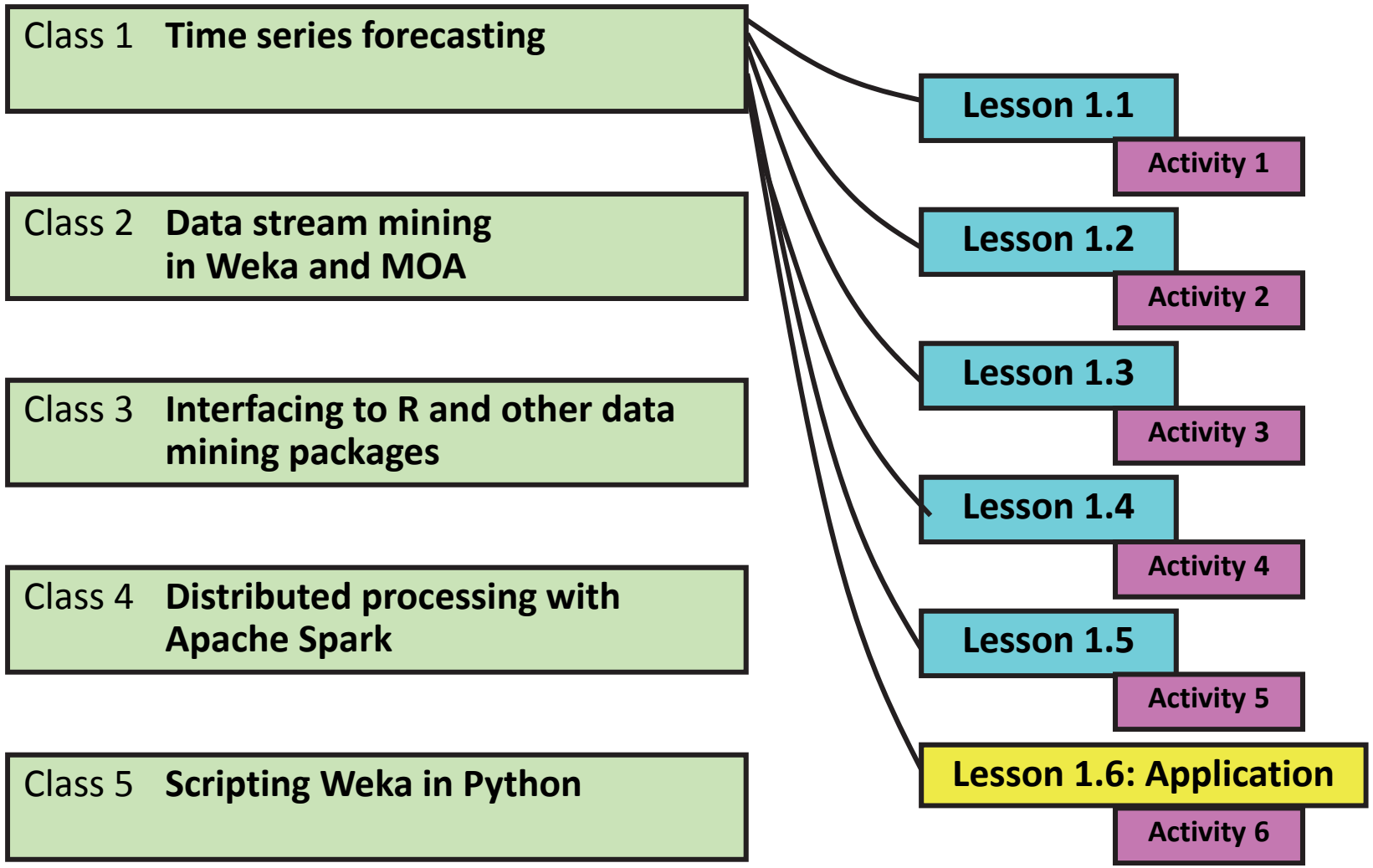
Course organization

Class 1	**Time series forecasting**
Class 2	**Data stream mining in Weka and MOA**
Class 3	**Interfacing to R and other data mining packages**
Class 4	**Distributed processing with Apache Spark**
Class 5	**Scripting Weka in Python**

Course organization



Course organization



Course organization

Class 1 **Time series forecasting**

Class 2 **Data stream mining in Weka and MOA**

→ **Mid-class assessment** 1/3

Class 3 **Interfacing to R and other data mining packages**

Class 4 **Distributed processing with Apache Spark**

Class 5 **Scripting Weka in Python**

→ **Post-class assessment** 2/3

Download Weka 3.7/3.8 now!

Download from

http://www.cs.waikato.ac.nz/ml/weka

for Windows, Mac, Linux

Weka 3.7 or 3.8 (or later)

the latest version of Weka

includes datasets for the course

do not use Weka 3.6!

Even numbers (3.6, 3.8) are *stable* versions

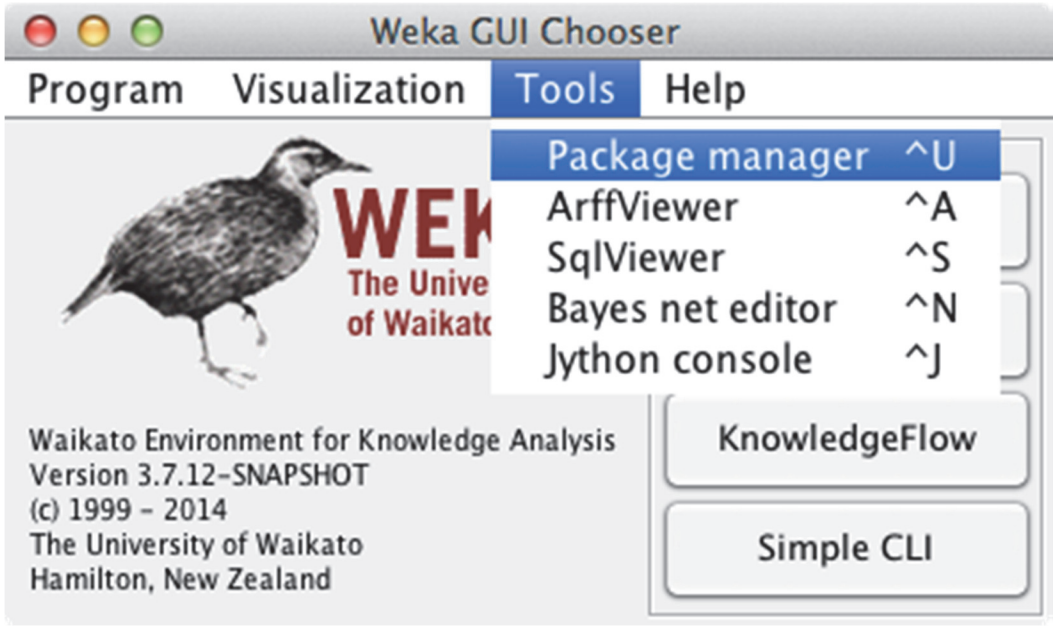
Odd numbers (3.7, 3.9) are *development* versions

Weka 3.7/3.8

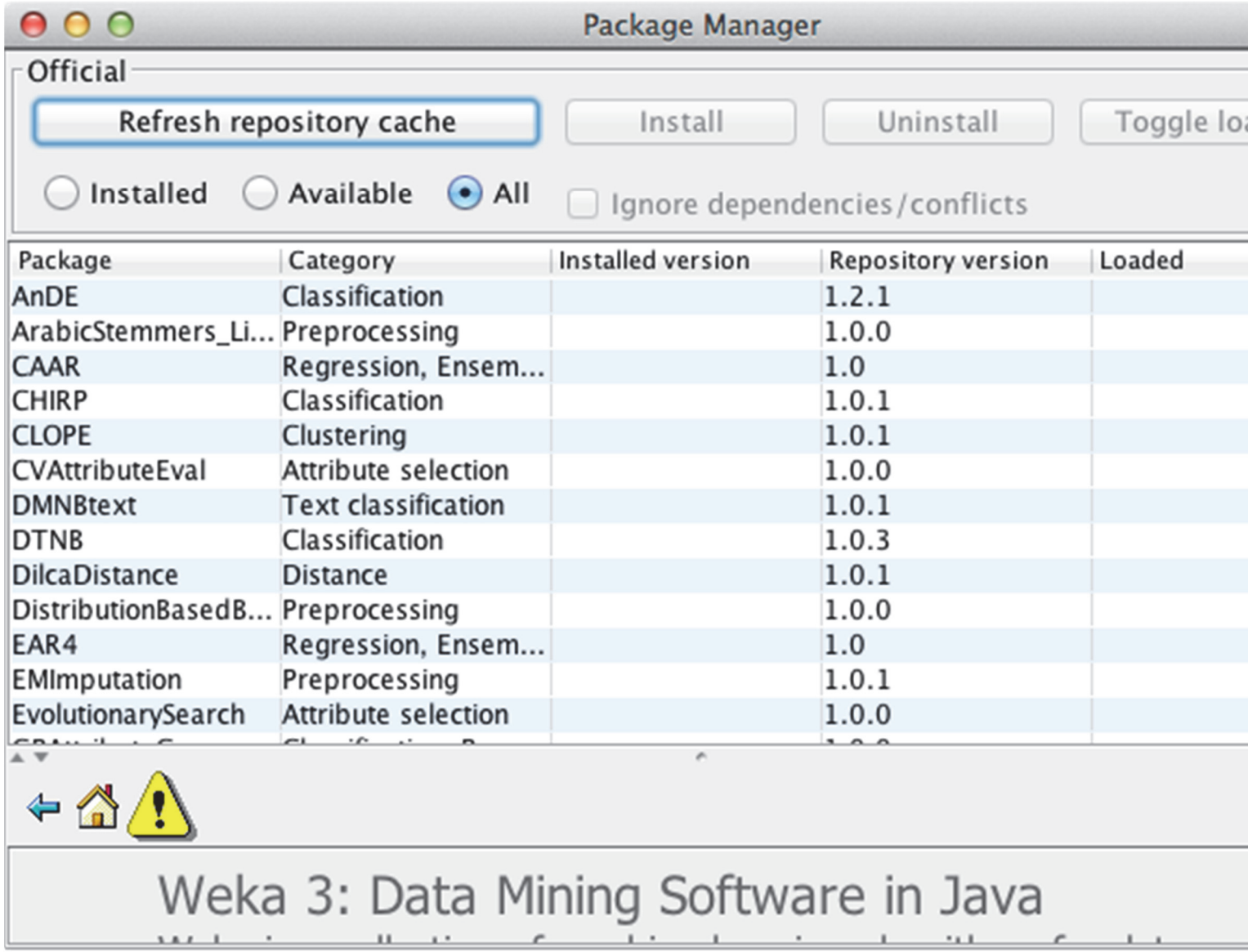
Core:

- some additional filters
- little-used classifiers moved into packages
  - e.g. multiInstanceLearning, userClassifier packages
- ... also little-used clusterers, association rule learners
- some additional feature selection methods

Packages:



Weka 3.7/3.8



- Official packages: 154
  - list is on the Internet
  - need to be connected!
- Unofficial packages
  - user supplied
  - listed at https://weka.wikispaces.com/Unofficial+packages+for+WEKA+3.7

Class 1: Time series forecasting

Lesson 1.1 Installing Weka and Weka packages

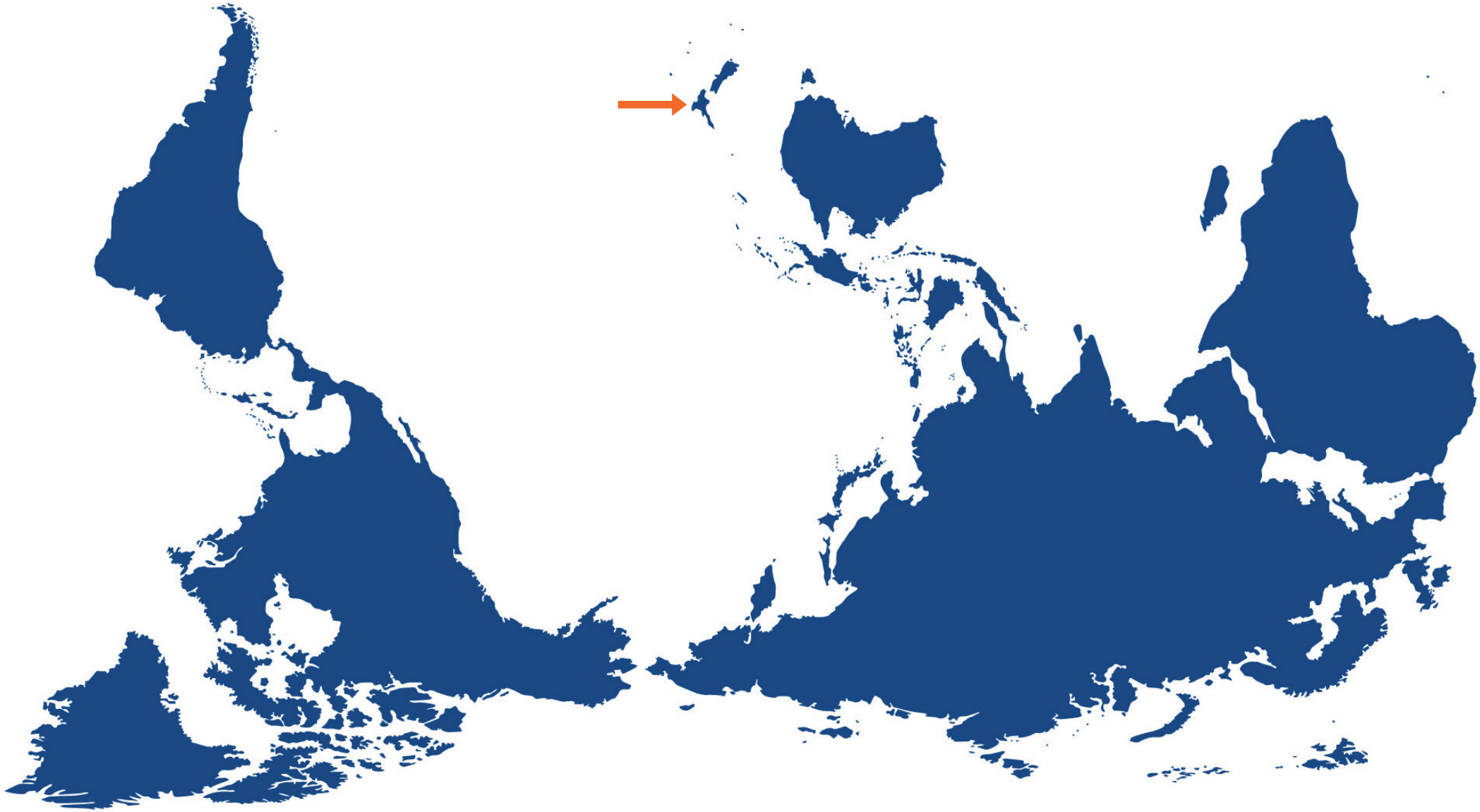
Lesson 1.2 Time series: linear regression with lags

Lesson 1.3 Using the *timeseriesForecasting* package

Lesson 1.4 Looking at forecasts

Lesson 1.5 Lag creation, and overlay data

Lesson 1.6 Application: analysing infrared data from soil samples



THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

Class 1 – Lesson 2

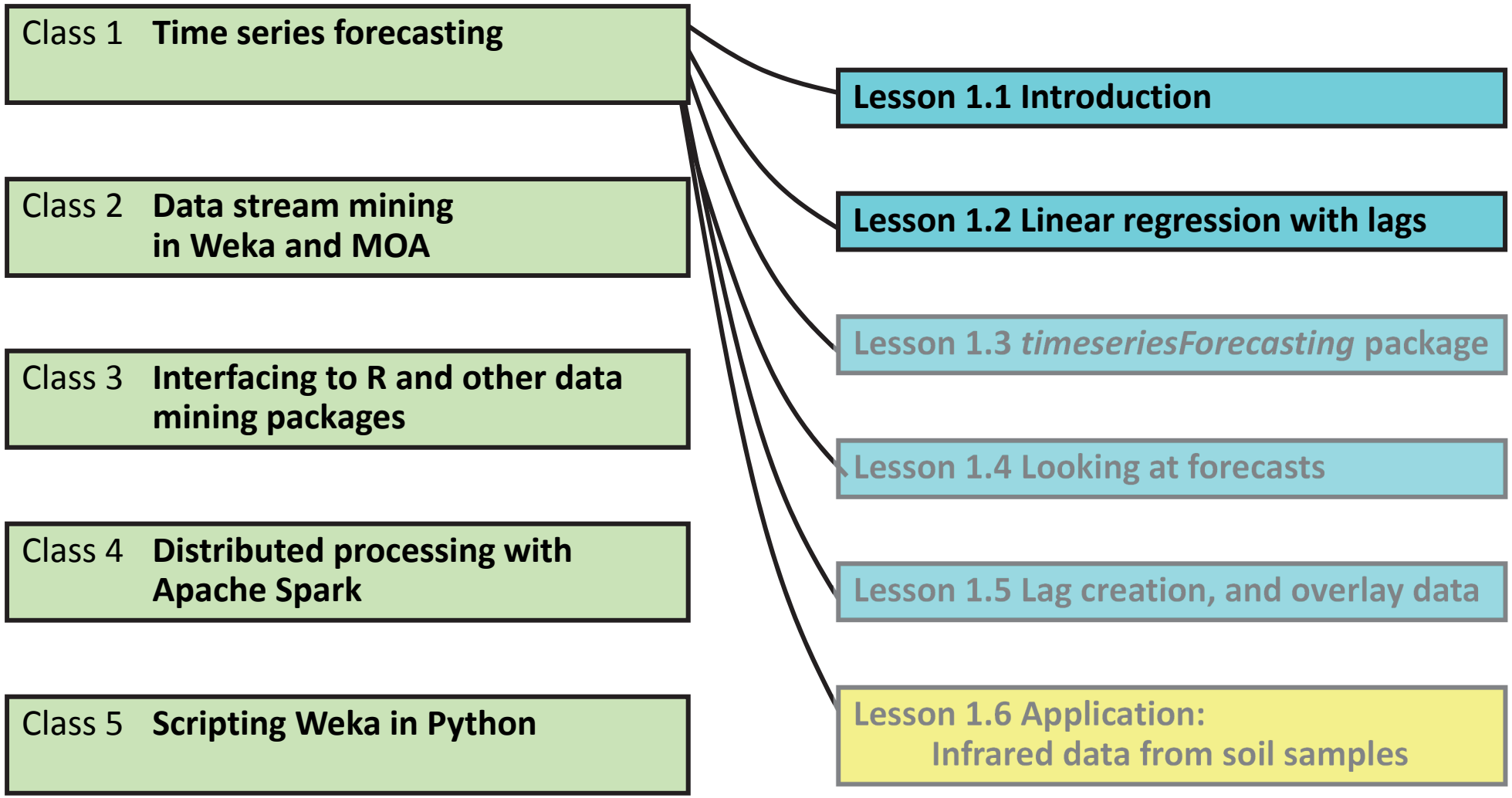
Linear regression with lags

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 1.2: Linear regression with lags

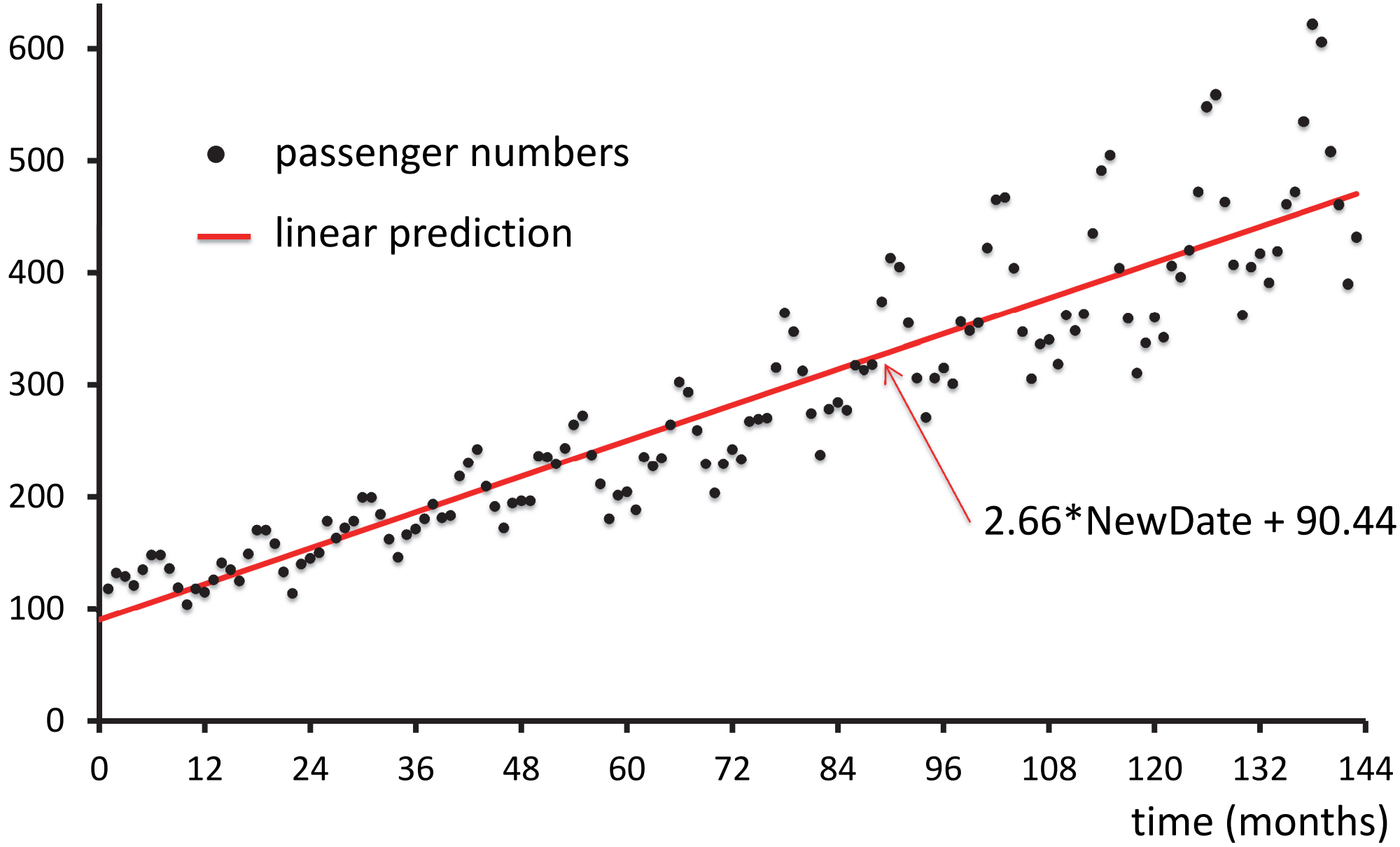


Linear regression with lags

Load airline.arff

- Look at it; visualize it
- Predict passenger_numbers: classify with *LinearRegression* (RMS error 46.6)
- Visualize classifier errors using right-click menu
- Re-map the date: msec since Jan 1, 1970 -> months since Jan 1, 1949
  - *AddExpression* (a2/(1000*60*60*24*365.25) + 21)*12; call it NewDate
    [it's approximate: think about leap years]
- Remove Date
- Model is 2.66*NewDate + 90.44

Linear regression with lags

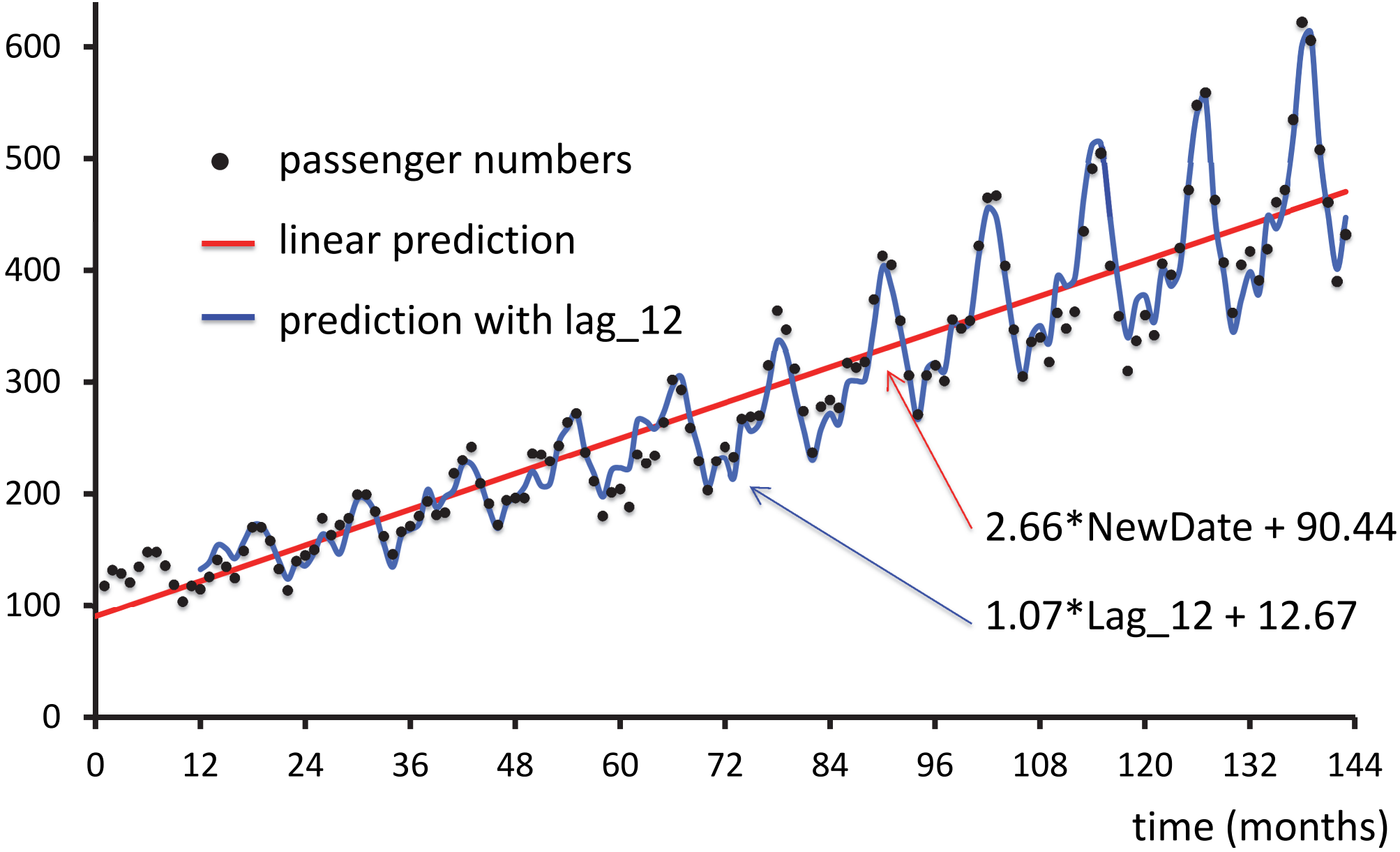


Linear regression with lags

- *Copy* passenger_numbers and apply *TimeSeriesTranslate* by –12
- Predict passenger_numbers: classify with *LinearRegression* (RMS error 31.7)
- Model is 1.54*NewDate + 0.56*Lag_12 + 22.09

- The model is a little crazy, because of missing values
  - in fact, *LinearRegression* first applies *ReplaceMissingValues* to replace them by their mean
  - this is a very bad thing to do for this dataset
- Delete the first 12 instances using the *RemoveRange* instance filter
- Predict with *LinearRegression* (RMS error 16.0)
- Model is 1.07*Lag_12 + 12.67
- Visualize – using *AddClassification*??

Linear regression with lags



Linear regression with lags

Pitfalls and caveats

- Remember to set the class to *passenger_numbers* in the *Classify* panel
- Before we renormalized *Date*, the model's *Date* coefficient was truncated to 0
- Use *MathExpression* instead of *AddExpression* to convert the date *in situ*?
- *Months* are inaccurate because one should take account of leap years
- in *AddClassification*, be sure to set *LinearRegression* and *outputClassification*
- *AddClassification* needs to know the class, so set it in the *Preprocess* panel
- *AddClassification* uses a model built from training data — inadvisable!
  - instead, could output classifications from the Classify panel's *More options...* menu
  - choose *PlainText* for Output predictions
  - to output additional attributes, click *PlainText* and configure appropriately
- Weka visualization cannot show multiple lines on a graph — export to Excel
- *TimeSeriesTranslate* does not operate on the class attribute — so unset it
- Can delete instances in Edit panel by right-clicking

Linear regression with lags

- Linear regression can be used for time series forecasting
- Lagged variables yield more complex models than “linear”
- We chose appropriate lag by eyeballing the data
- Could include >1 lagged variable with different lags
- What about seasonal effects? (more passengers in summer?)
- Yearly, quarterly, monthly, weekly, daily, hourly data?
- **Doing this manually is a pain!**

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

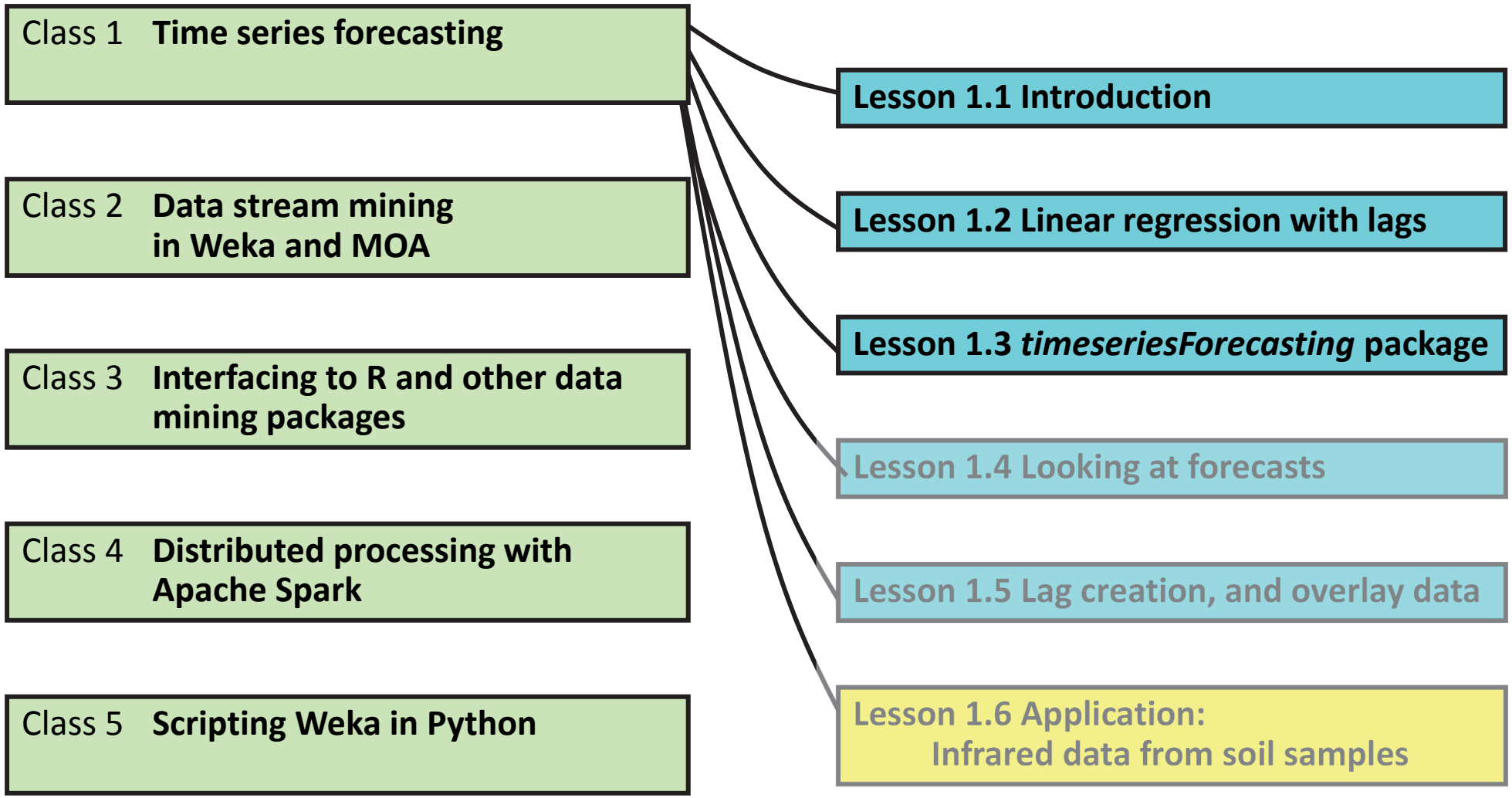
Class 1 – Lesson 3

timeseriesForecasting package

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

Lesson 1.3: Using the timeseriesForecasting package



Using the timeseriesForecasting package

Install the *timeseriesForecasting* package

... it's near the end of the list of packages that you get via the *Tools* menu

- Reload the original airline.arff
- Go to the Forecast panel, click *Start*
- Training data is transformed into a large number of attributes
  - depends on the periodicity of the data – here, <Detect automatically> gives *Monthly*
  - *Date-remapped* is months since Jan 1, 1949, as in the last lesson (but better)
- Model is very complex
- ... but (turn on "Perform evaluation") looks good!
  — RMS error 10.6 (vs. 16.0 before)

Using the timeseriesForecasting package

Making a simpler model

- Cannot edit the generated attributes, unfortunately
- Go to Advanced configuration, select Base Learner
- Choose *FilteredClassifier* with *LinearRegression* classifier and *Remove* filter, configured to remove all attributes EXCEPT:

  1 (passenger_numbers), 4 (Date-remapped), 16 (Lag-12)
- Model is 1.55*NewDate + 0.56*Lag_12 + 22.04 (we saw this in last lesson!)
- Delete the first 12 instances

  ?? use *Multifilter*, with *Remove* (–V –R 1,4,16) followed by *RemoveRange* (–R 1-12)
- Instead, use “More options” on the Lag creation panel to remove instances
- Model is 1.07*Lag_12 + 12.67, with RMS error 15.8 (as before)

Using the timeseriesForecasting package

Simple vs complex model

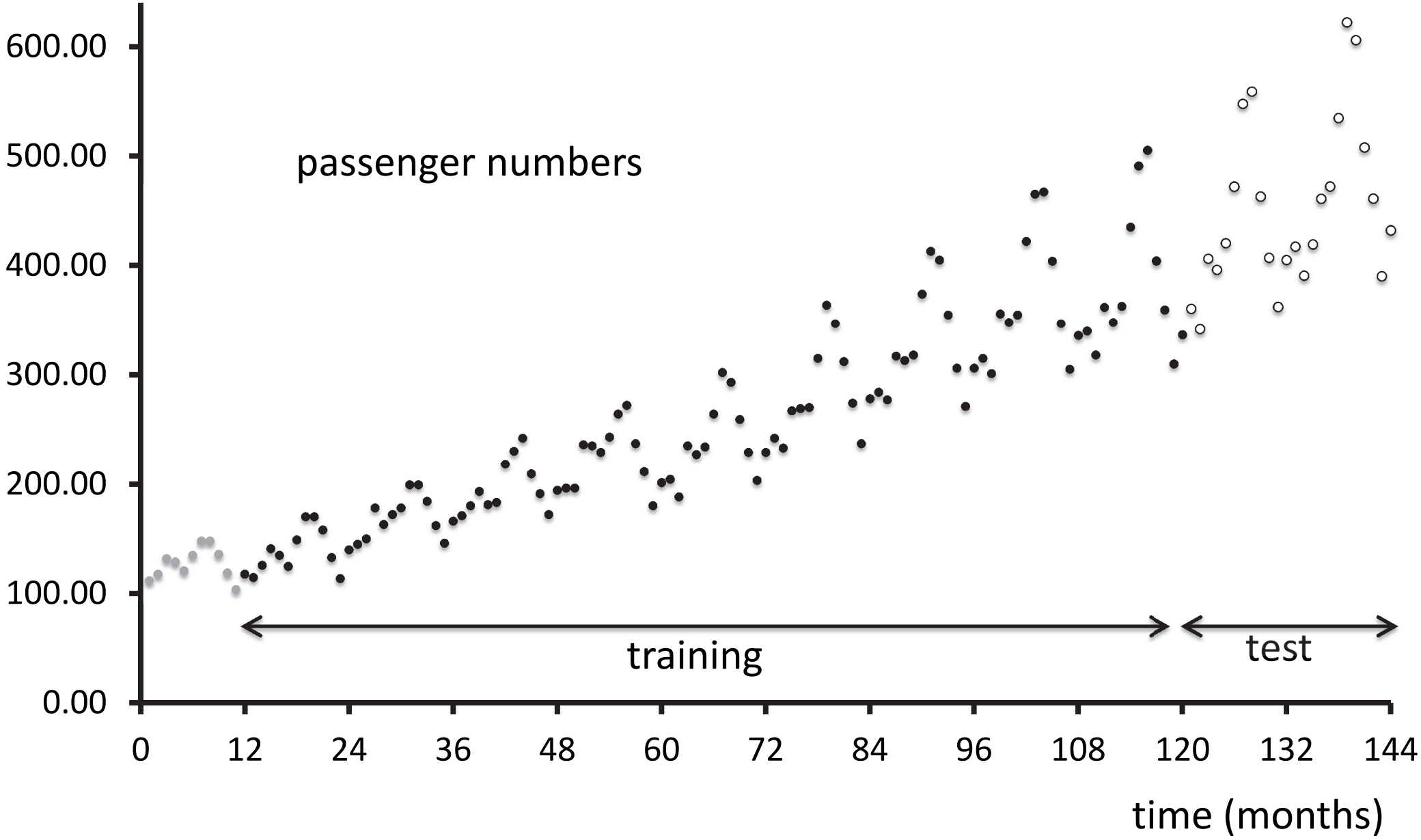
- Return to full model (but removing first 12 instances):
  RMS error is 8.7 (vs. 15.6 for simple model) – on the training data
- Model looks very complex! – is it over-fitted?
- Evaluate on held-out training (specify 24 instances in Evaluation panel)
  - data covers 12 years, lose 1 year at beginning: train on 9 years, test on 2 years
- RMS error is 58.0! (vs 6.4 for simple model)
- Training/test error very different for complex model (similar for simple one)

Using the timeseriesForecasting package

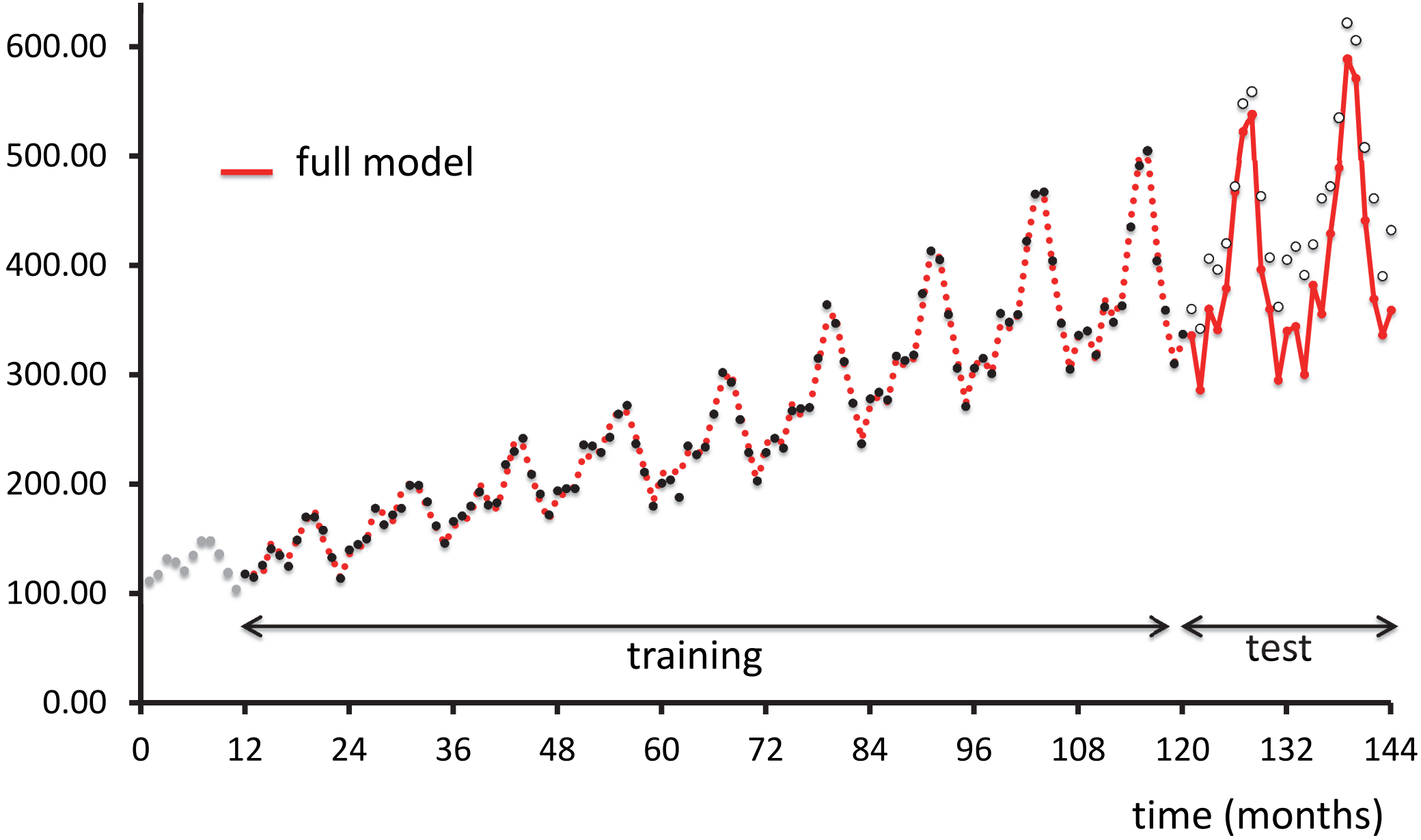
Overfitting: Training/test RMS error differs

LinearRegression classifier	*training error*	*test error*
❖ full model (all attributes)	6.4	58.0
simple model (2 attributes)	15.6	18.7
❖ AttributeSelectedClassifier: default settings 4 attributes: Month, Quarter, Lag-1, Lag-12 (wrapper-based attribute selection doesn't make sense)	11.0	19.8
❖ Use Lag creation/Periodic attributes panels to reduce attributes to 2 simple model, above	same as	

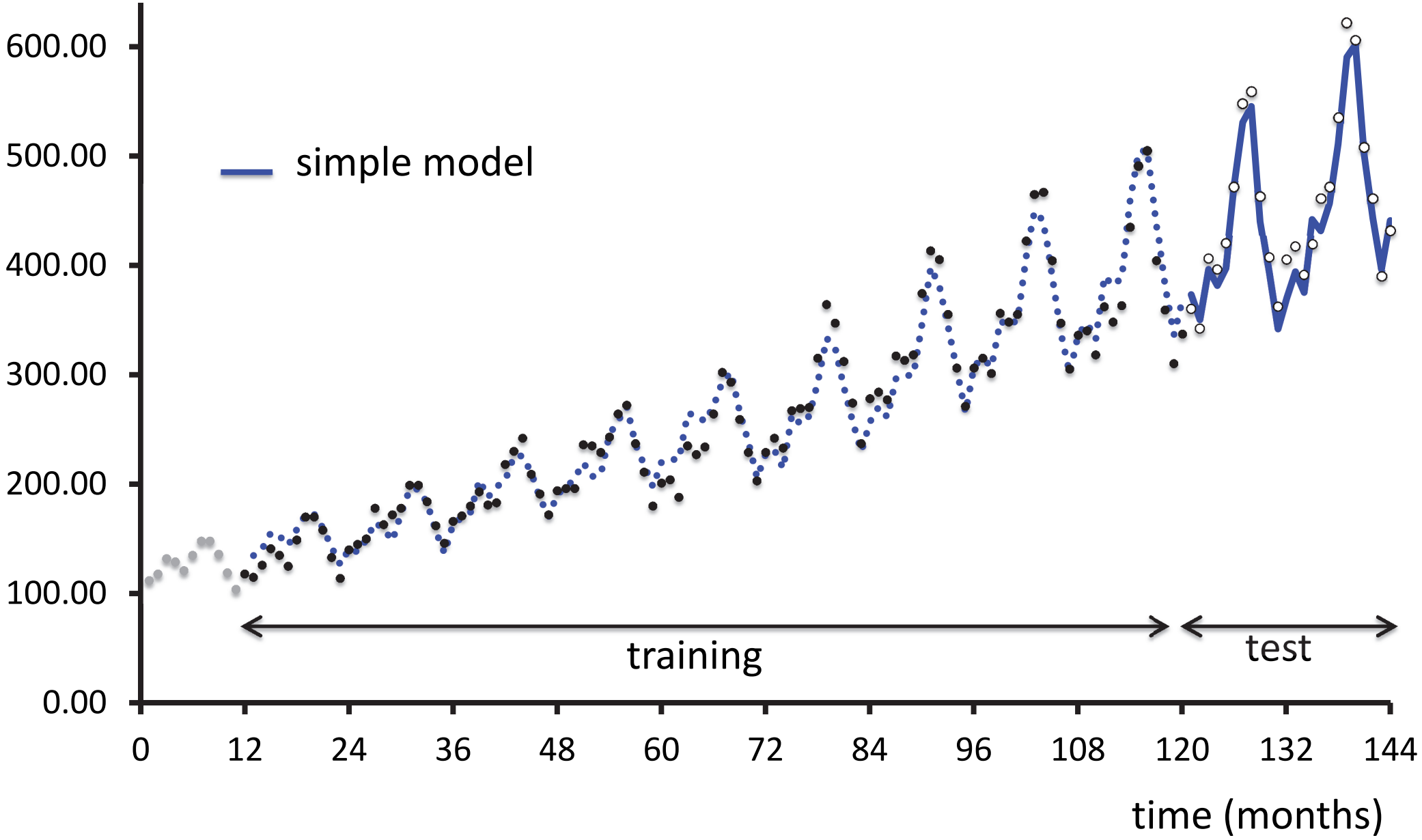
Using the timeseriesForecasting package



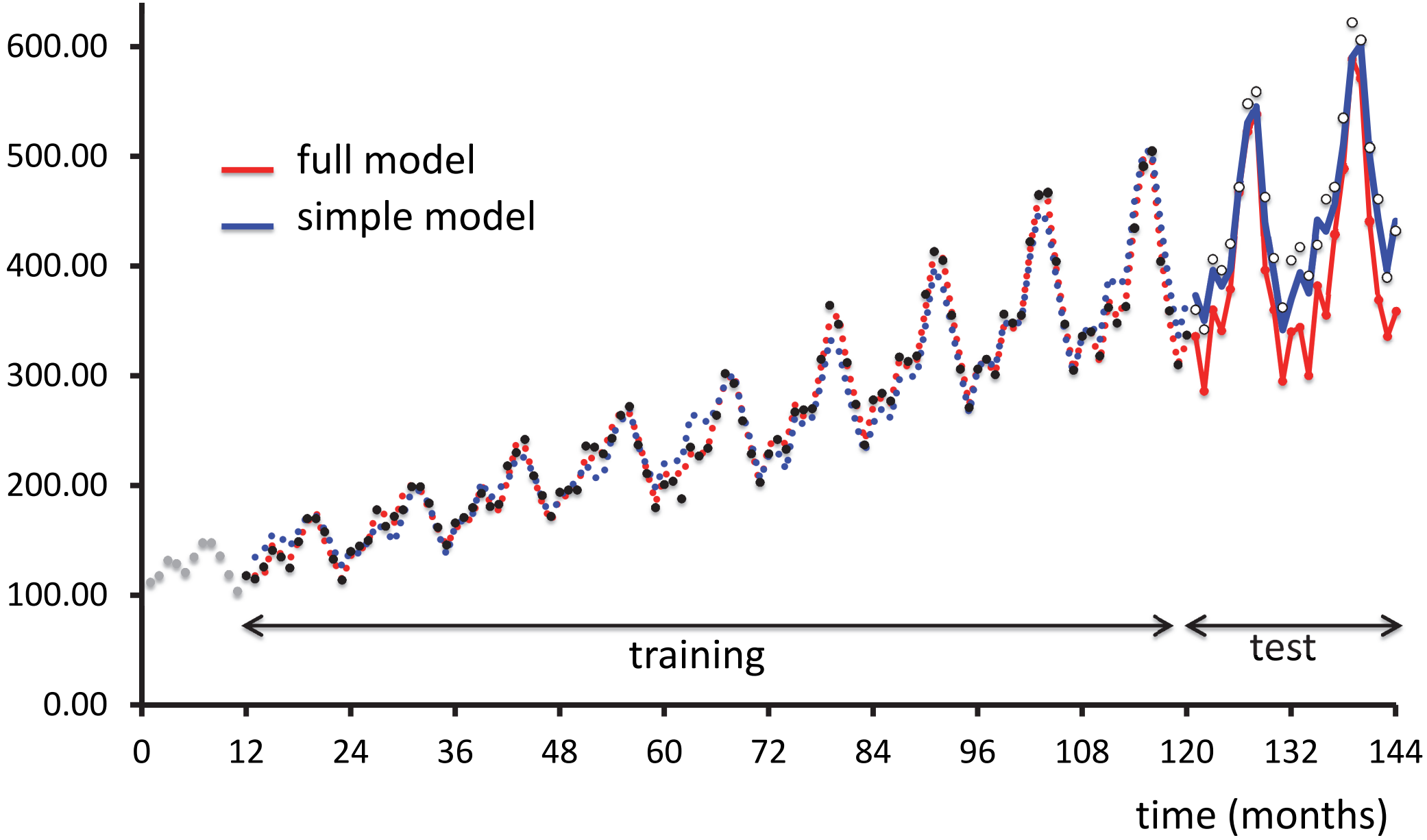
Using the timeseriesForecasting package



Using the timeseriesForecasting package



Using the timeseriesForecasting package



Using the timeseriesForecasting package

- Weka's *timeseriesForecasting* package makes it easy
- Automatically generates many attributes (e.g. lagged variables)
- Too many? – try simpler models, using the *Remove* filter
  - or use *Lag creation* and *Periodic attributes* in "Advanced configuration"
- Beware of evaluation based on training data!
  - hold out data using the *Evaluation* tab (fraction, or number of instances)
- Evaluate time series by repeated 1-step-ahead predictions
  - errors propagate!

Reference: Richard Darlington, "A regression approach to time series analysis"
http://node101.psych.cornell.edu/Darlington/series/series0.htm

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

Class 1 – Lesson 4

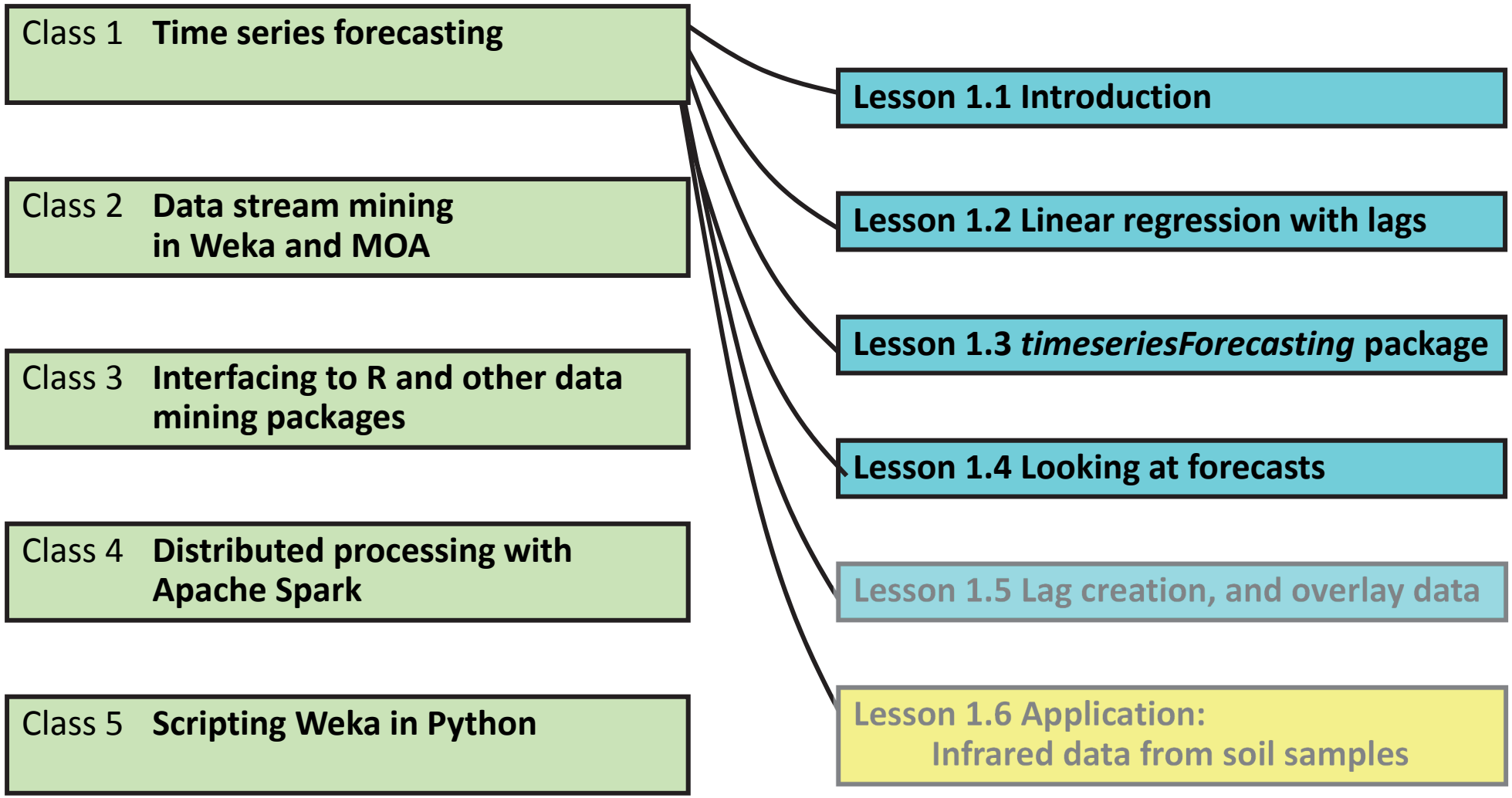
Looking at forecasts

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 1.4: Looking at forecasts

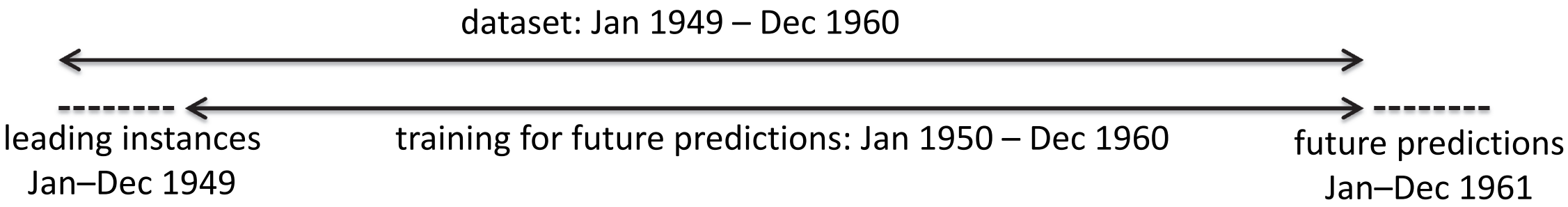


Looking at forecasts

The timeseriesForecasting package produces visualizations ...

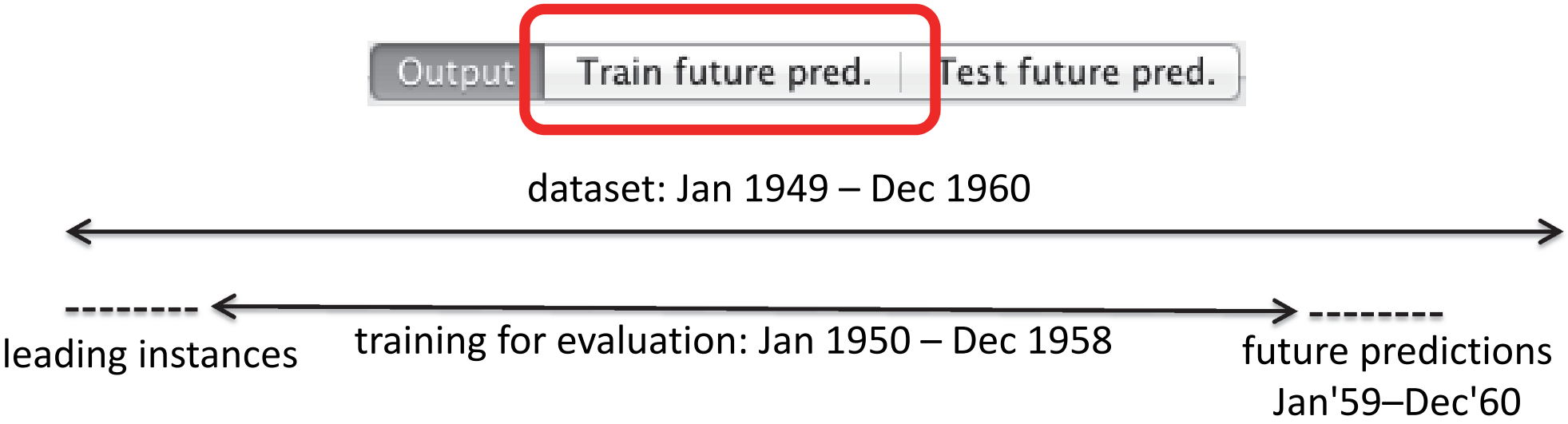
- Restart the Explorer; load airline.arff; Forecast panel; click *Start*

- Look at *Train future pred.* (training data plus forecast)
- Forecast 12 units ahead (dashed line, round markers)
- Lag creation: remove leading instances



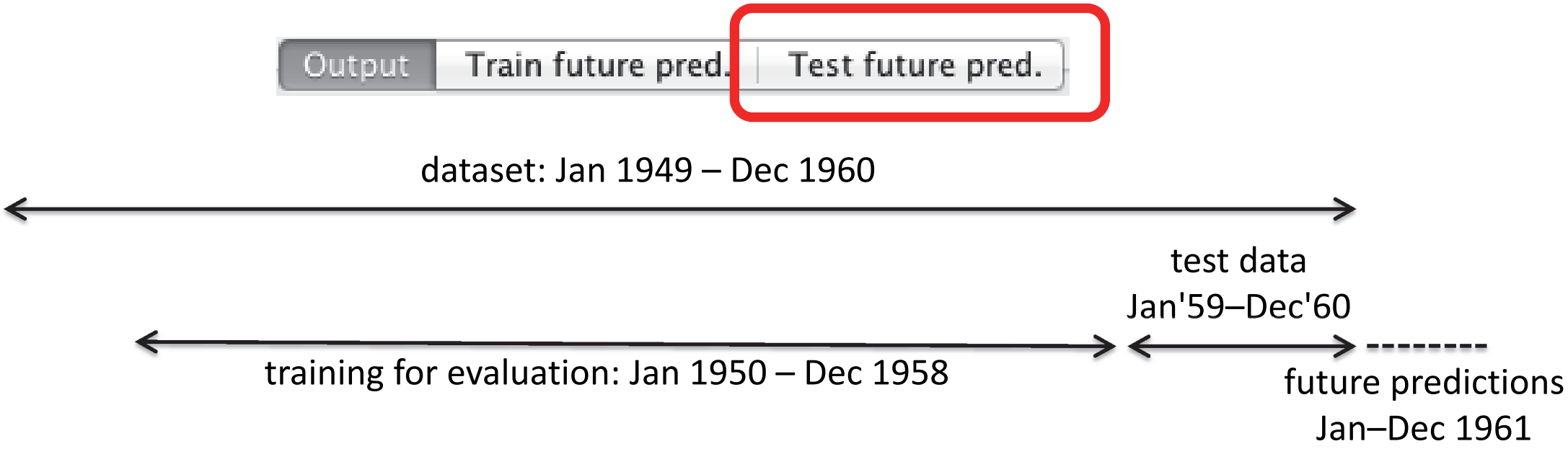
Looking at forecasts

- Advanced configuration; Evaluate on training and on 24 held out instances



Looking at forecasts

- Advanced configuration; Evaluate on training and on 24 held out instances



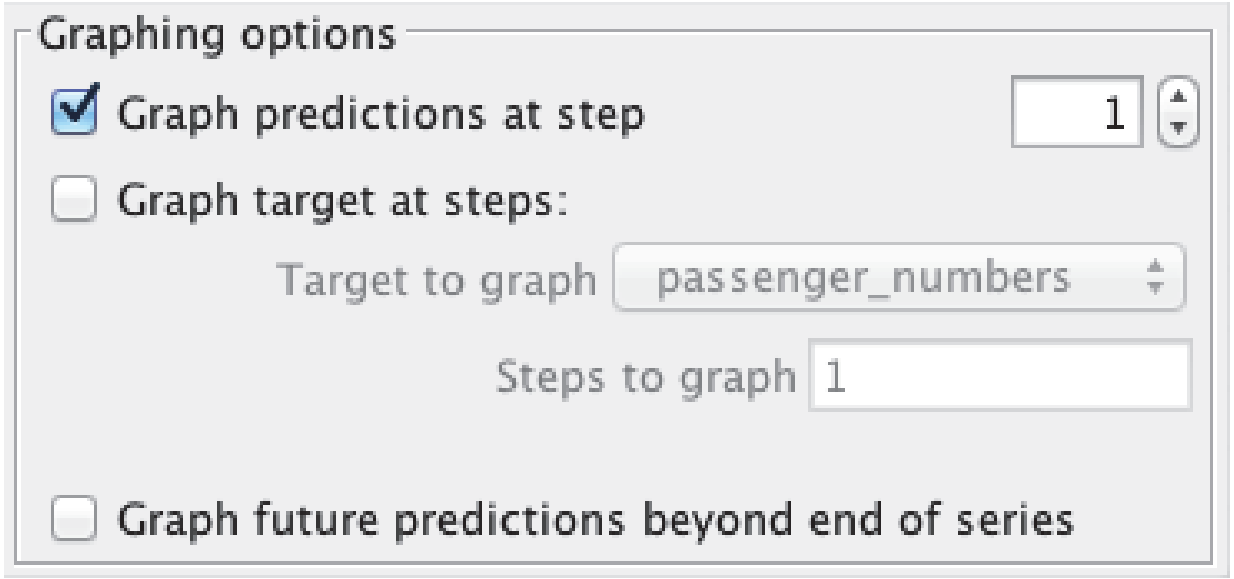
- *Test future pred*: test data plus forecast

  ... but would be nice to see 1-step-ahead estimates for test data too

Looking at forecasts

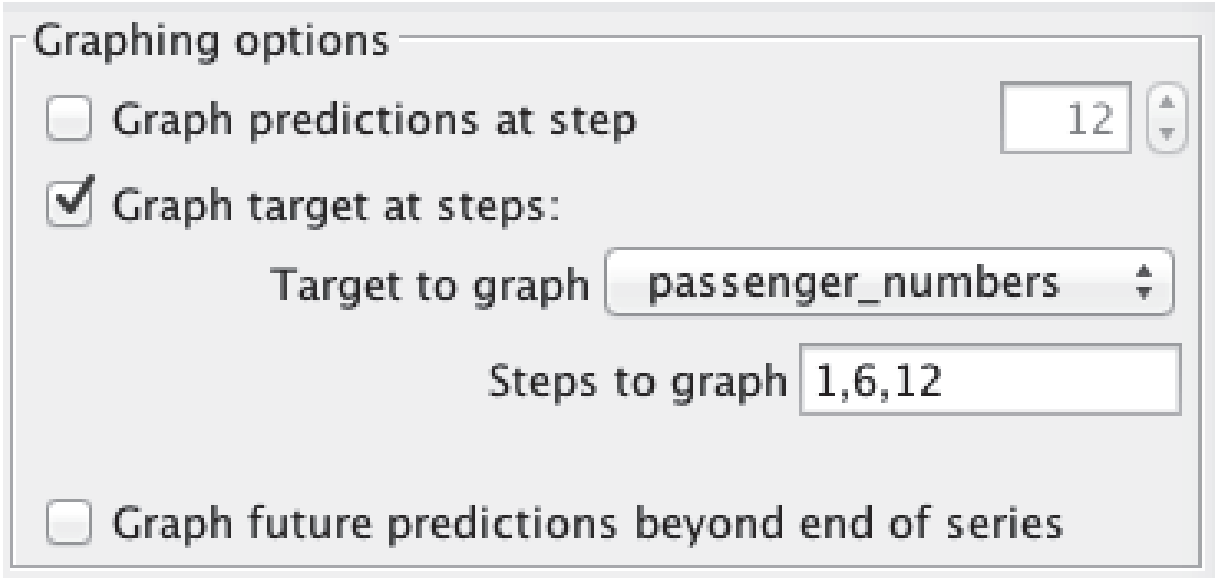
Output: Graphing options

- Turn off *Evaluate on training*
- Turn off *Graph future predictions*
- Run; no graphical output
- Turn on *Graph predictions at step* 1



- Shows 1-step-ahead predictions for test data

Looking at forecasts



Multi-step forecasts

- Graph predictions at step 12
- Graph target at step 12
- Compare 1-step-ahead, 6-steps-ahead, and 12 steps-ahead predictions
- Change base learner to SMOreg and see the difference
- Get better predictions by reducing attributes (see last lesson's Activity):
  - minimum lag of 12
  - turn off powers of time, products of time and lagged vbls
  - customize to no periodic attributes

Looking at forecasts

- Many different options for visualizing time series predictions
- Need to distinguish different parts of the timeline
  - initialization: time period for leading instances
  - extrapolation: future predictions
  - full training data
  - test data (if specified)
  - training data with test data held out
- Number of steps ahead when making predictions

Reference: Mark Hall, "Time Series Analysis and Forecasting with Weka"
http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

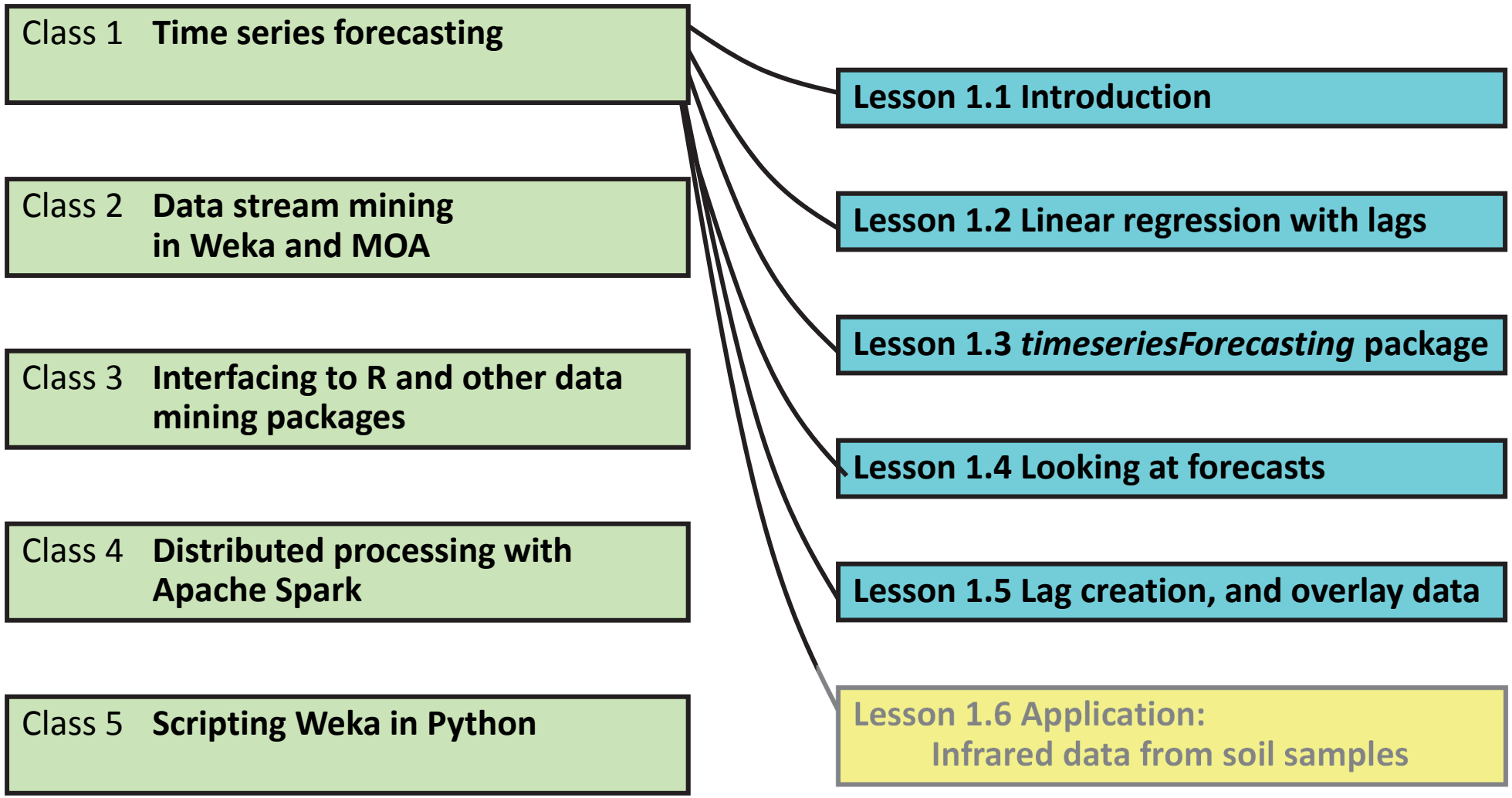
Class 1 – Lesson 5

Lag creation and overlay data

Ian H. Witten

Department of Computer Science
University of Waikato
New Zealand

Lesson 1.5: Lag creation, and overlay data



Lag creation, and overlay data

Basic configuration: parameters

- Time stamp: “date” attribute used by default (can be overridden)
- Periodicity: *Detect automatically* is recommended
  - or you can specify hourly, daily, weekly, monthly ...
  - possibly useful if the date field contains many unknown values
  - interpolates new instances if you specify a shorter periodicity
  - e.g. airline data: Monthly 144 instances; Weekly 573 (= 144×4 – 3); Hourly 104,449
- Periodicity also determines what attributes are created
  - always includes Class, Date, Date’, Date’^{2}, Date’^{3}
  - lagged variables: Monthly 12; Weekly 52; Daily 7; Hourly 24
  - plus product of date and lagged variables
  - if Daily, include DayOfWeek, Weekend; if Hourly include AM/PM
- These attributes can be overridden using the Advanced Configuration panel

Lag creation, and overlay data

appleStocks2011: daily *High*, *Low*, *Open*, *Close*, *Volume*

- Target selection
  - data contains more than one thing to predict
  - most days from 3 Jan 2011 – 10 Aug 2011
  - forecast *Close*
  - generates lags up to 12 (Monthly??); set to Daily (lags up to 7)
  - no instances for Jan 8/9, 15/16/17, 22/23. 29/30 ... weekends + a few holidays
  - these “missing values” are interpolated – but perhaps they shouldn’t be!
- Skip list:
  - e.g. weekend, sat, tuesday, mar, october, 2011-07-04@yyyy-MM-dd
  - specify
    weekend, 2011-01-17@yyyy-MM-dd, 2011-02-21, 2011-04-22, 2011-05-30, 2011-07-04
  - set max lag of 10 (2 weeks)

Lag creation, and overlay data

Playing around with the data

- Evaluation: hold out 0.3 of the instances
- Output: graph target: *Close*
- Mean absolute error:
- Remove leading instances

	training	*test*
	3.4	9.1
	3.5	7.7

Multiple targets

- Can predict more than one target
  - lagged versions of one attribute may help predictions of another
  - ... or maybe just cause overfitting
- Basic configuration select *Close* and *High*:

	error for *Close*	3.4	8.0
select them all	error for *Close*	2.5	9.6

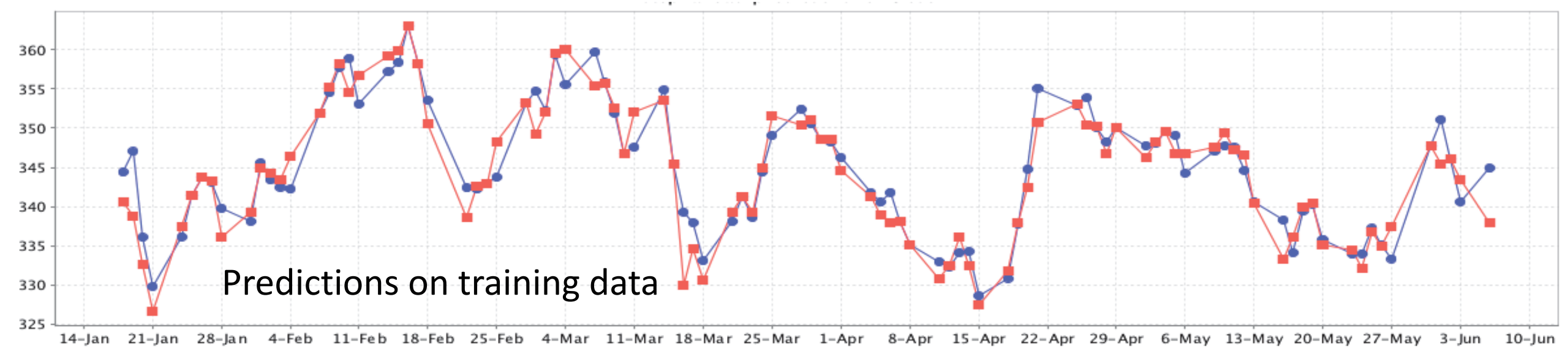
Lag creation, and overlay data

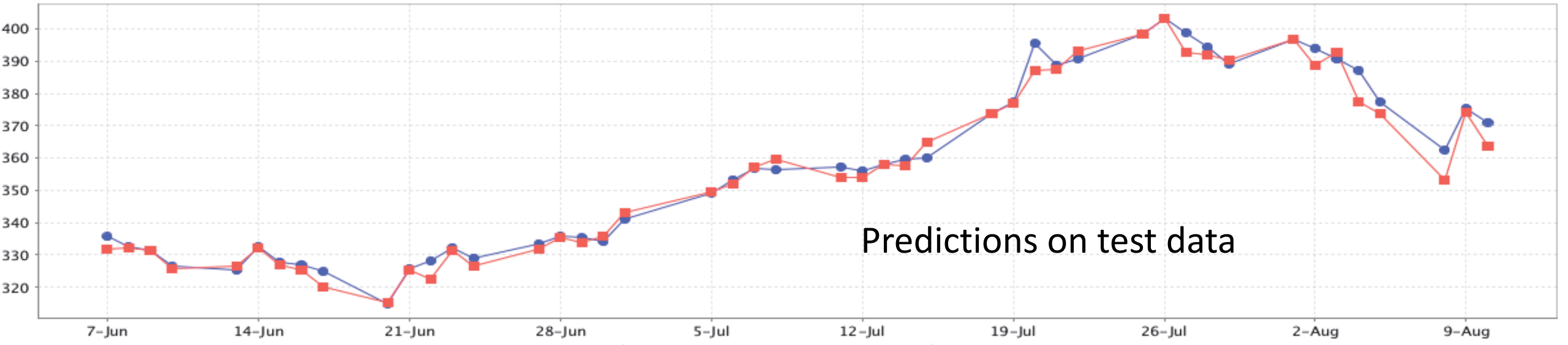
Overlay data

- Additional data that may be relevant to the prediction
  - e.g. weather data, demographic data, lifestyle data
- Not to be forecast (can't be predicted)
- Available to assist future predictions
- Simulate this using appleStocks data
  - revert to single target: *Close*
  - (turn off "output future predictions")

		training	*test*
Overlay data:	*Open*	3.0	5.9
	Open and *High*	1.9	2.9
Base learner SMOreg		1.7	2.4

Lag creation, and overlay data





Lag creation, and overlay data

- Many different parameters and options
- Getting the time axis right – days, hours, weeks, months, years
  - automatic interpolation for missing instances
  - skip facility to ensure that time increases “linearly”
- Selecting target or targets
- Overlay data – can help a lot (obviously!)
- We haven’t looked at:
  - confidence intervals, adjust for variance, fine tune lag selection, average consecutive long lags, custom periodic fields, evaluation metrics

Reference: Richard Darlington, “A regression approach to time series analysis” http://node101.psych.cornell.edu/Darlington/series/series0.htm

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

Class 1 – Lesson 6

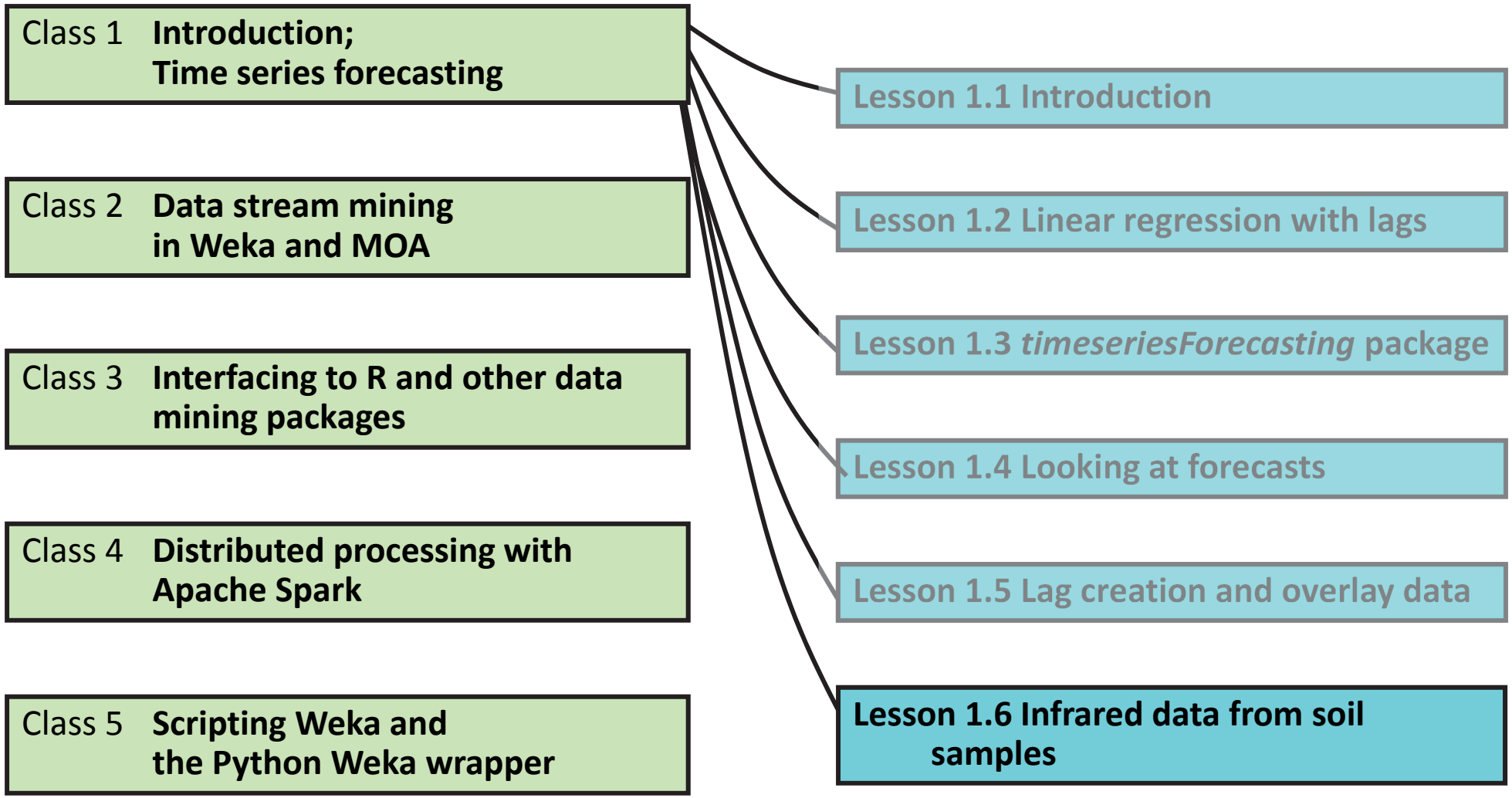
Application: Infrared data from soil samples

Geoff Holmes

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz

Lesson 1.6: Application: Infrared data from soil samples



Infrared data from soil samples

A word about applications in general

- The top academic conference in machine learning is called ICML.
- In 2012 a paper was published at this conference as a wake-up call.
- The author was Kiri Wagstaff from the Jet Propulsion Lab in Pasadena
- The paper is accessible to anyone with an interest in machine learning
- **Machine Learning that Matters** http://icml.cc/2012/papers/298.pdf
- The paper suggests 6 challenges for machine learning applications

Infrared data from soil samples

- A law passed or legal decision made that relies on the result of an ML analysis.
- **$100M saved through improved decision making provided by an ML system.**
- A conflict between nations averted through high-quality translation provided by an ML system.
- A 50% reduction in cybersecurity break-ins through ML defences.
- A human life saved through a diagnosis or intervention recommended by an ML system.
- Improvement of 10% in one country's Human Development Index (HDI)

Infrared data from soil samples

Taking a step back

- Let’s simplify what Machine Learning is in terms of input and output:

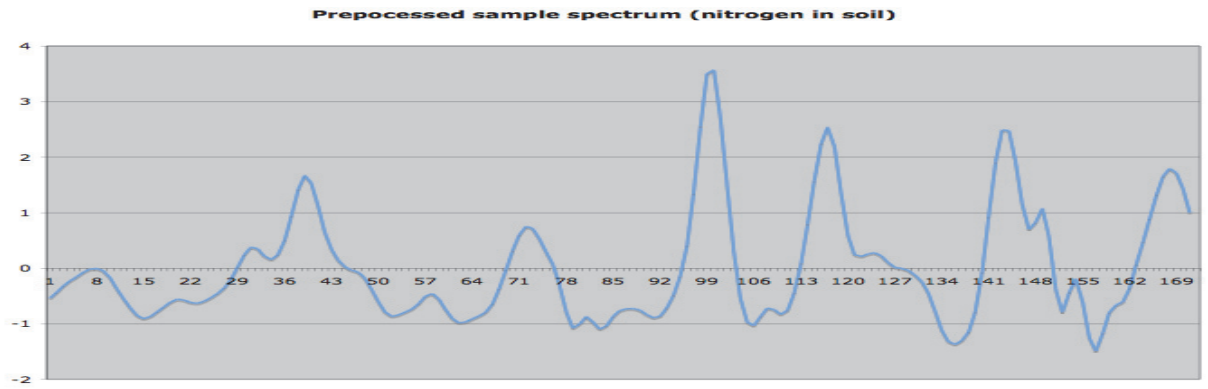
- Input is a set of samples (instances) X and an output target Y (one value per sample)
- Problem is to learn a mapping that describes the relationship between the input and the output. This mapping is termed a model.
- We use the model on unseen observations to predict the target (key is generalisation error).

Infrared data from soil samples

- Now let's see where we get X and Y from for this application.
- Soil samples have traditionally been analysed using "wet chemistry" techniques in order to determine their properties (eg available nitrogen, organic carbon, etc.). These techniques take days. The properties are our Y values or targets.
- The soil from a "soil bank" is re-used to form the input X. We need to record a unique identifier for each sample because we need to match up with the right target(s) established by wet chemistry on that sample. To actually get the input we put each sample through a device called a near-infrared spectrometer. If you Google "NIR machine" you will see lots of machines and also lots of uses for the device.

Infrared data from soil samples

- The NIR device produces a “signature” for the soil sample, like the one below.
- These values form our input, in the sense of an ARFF file they are reflectance values for a given wavelength band of the light spectrum (the first attribute covers 350 nanometers, the next 400, 450, etc.)
- To build any meaningful model from this data we need at least a few hundred samples. Recall that we get our targets from wet chemistry so it is expensive to put together a decent training set.



Infrared data from soil samples

Why is it worth the trouble of re-processing the soil?

- While it is true that the training set is expensive to produce it is worth it because once we have our model we can use it to predict the “available nitrogen” say of a new sample within the time it takes to run it through an NIR device (which is milli-seconds for NIR, days for wet chemistry).
- It is also true that if we have several target values for the same soil sample then we can use the X input against different Y output to produce a range of models, one per target. When predicting we simply use the same NIR spectra as input to each model, producing multiple predictions (nitrogen, carbon, potassium, etc.) for that single sample.

Infrared data from soil samples

Modelling

- The training set comprises (X=numeric values per wavelength, Y=numeric value for say nitrogen). So this is a regression problem.
- Classifiers of interest: *LinearRegression, RepTree, M5P, RandomForest, SMOReg, GaussianProcesses, etc.*
- Applying the above to our X and Y values will produce models but as you will see in the Activity, pre-processing can help each classifier to improve.
- Typical pre-processing for NIR revolves around downsampling, removing baseline effects (signal creep) and spectral smoothing.

Infrared data from soil samples

Experimentation

- In the Activity you will look at applying the first four classifiers in the list on the last slide to a large (4K samples) soil data set where you will develop a model for OrganicCarbon. The data set also contains targets for OrganicNitrogen (which you can look at separately).
- You will process the data raw then look at what happens when you apply the three pre-processing techniques mentioned above.
- Note that you are about to enter experimental ML – you have 4 classifiers, each have parameters to tweak, you have 4 pre-processing methods (including the raw spectra) some with parameters, each can be combined, … **the space of experiments is large!**

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Advanced Data Mining with Weka

Department of Computer Science
University of Waikato
New Zealand

weka.waikato.ac.nz