COMP30120 Tutorial

Dimension Reduction and Feature Selection

Derek Greene

School of Computer Science and Informatics

Autumn 2015

Tutorial Q1(a)

In Weka, use filter-based feature selection with Information Gain to identify the 3 most discriminating and 3 least discriminating features in the *Wine* data set in the ARFF file provided.

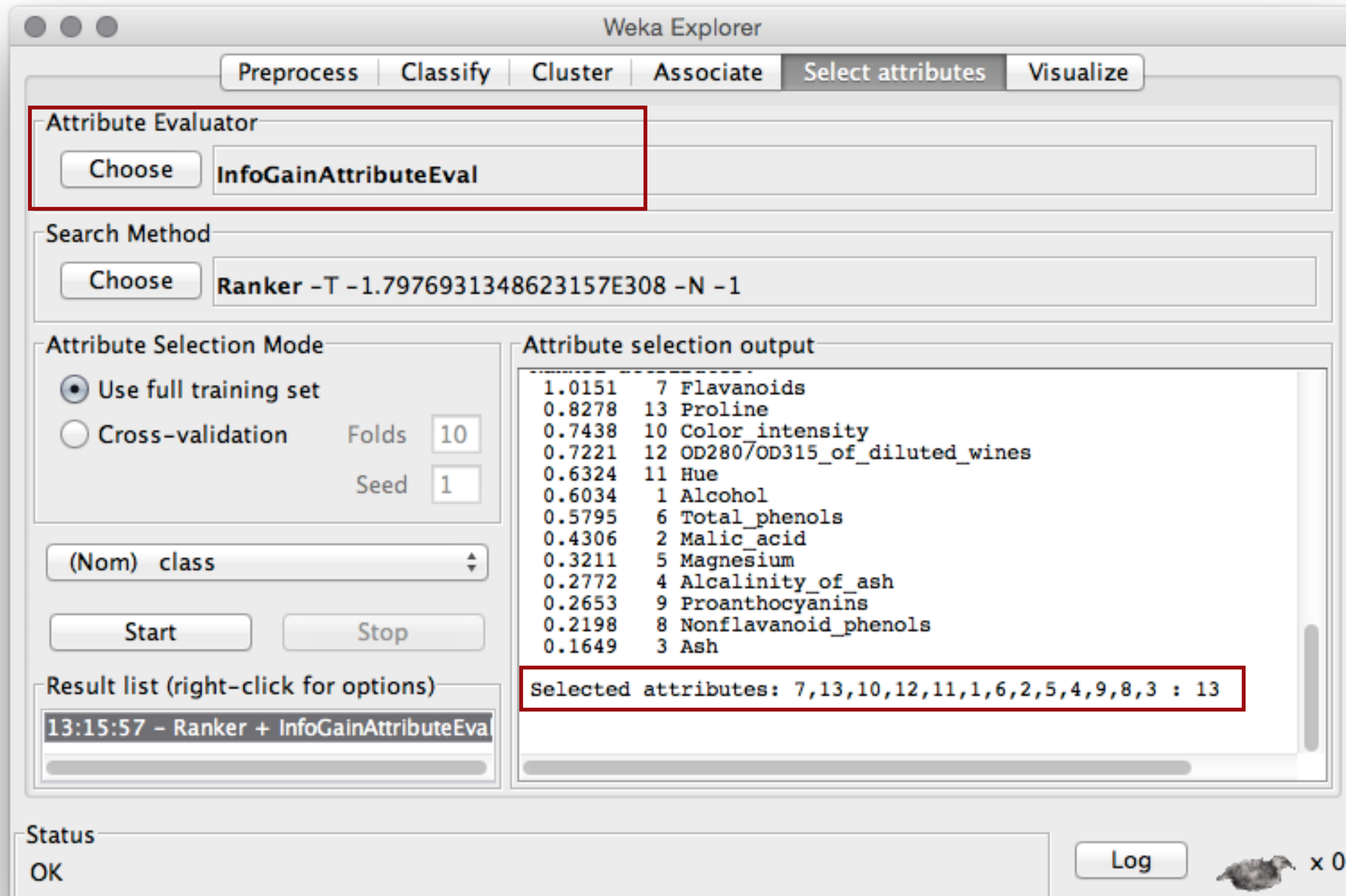
Assess the accuracy of a 1-nearest neighbour classifier with:

(i) only the 3 most discriminating features included.

(ii) only the 3 least discriminating features included.

Tutorial Q1(a)

In Weka *Select attributes* tab, choose *InfoGainAttributeEval* as the evaluator, *Ranker* as the method.



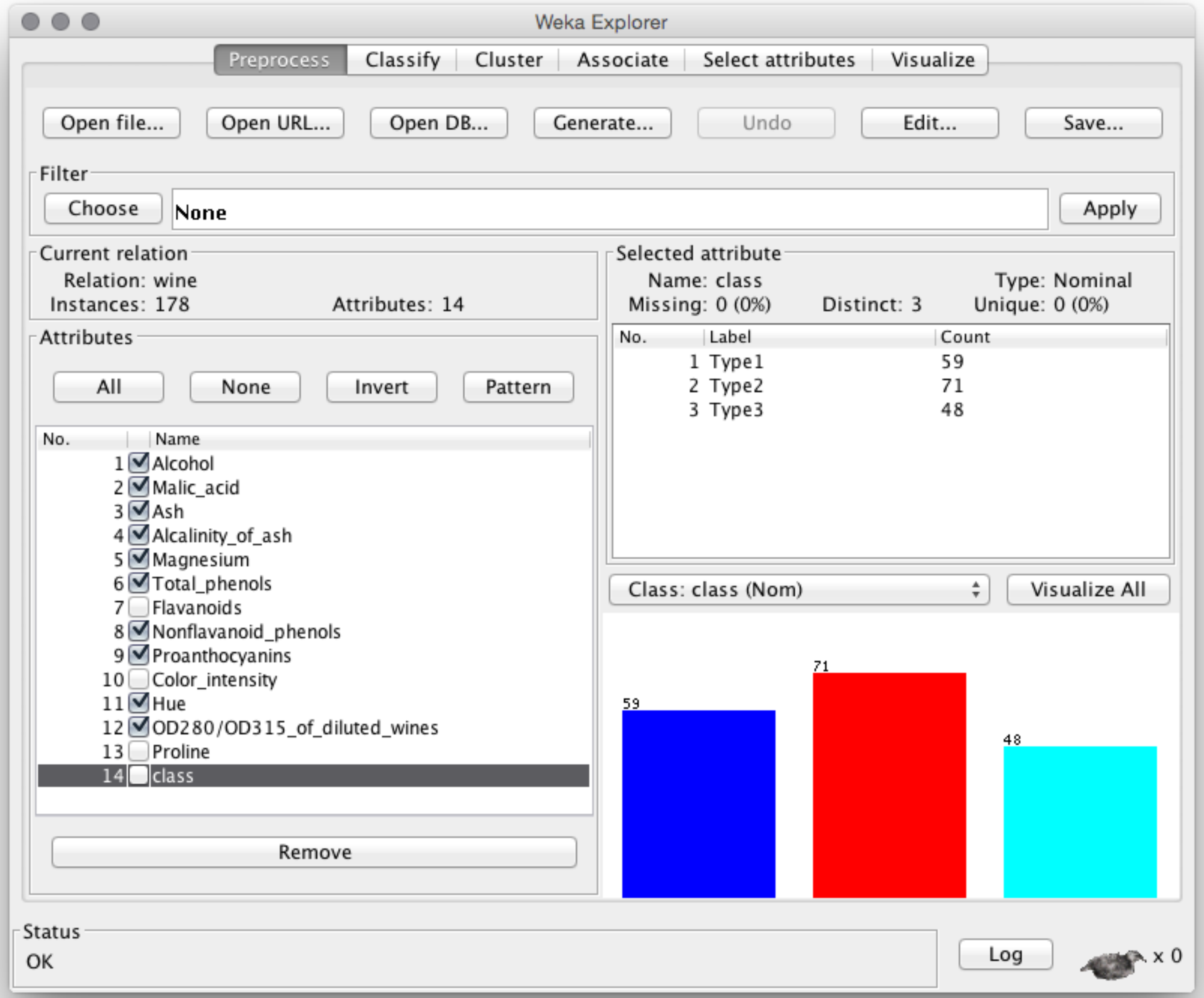
Tutorial Q1(a)

Assess the accuracy of a 1-nearest neighbour classifier with only the 3 most discriminating features included.

Most discriminating:
7, 13, 10

In the *Preprocess* tab, remove the unwanted features.

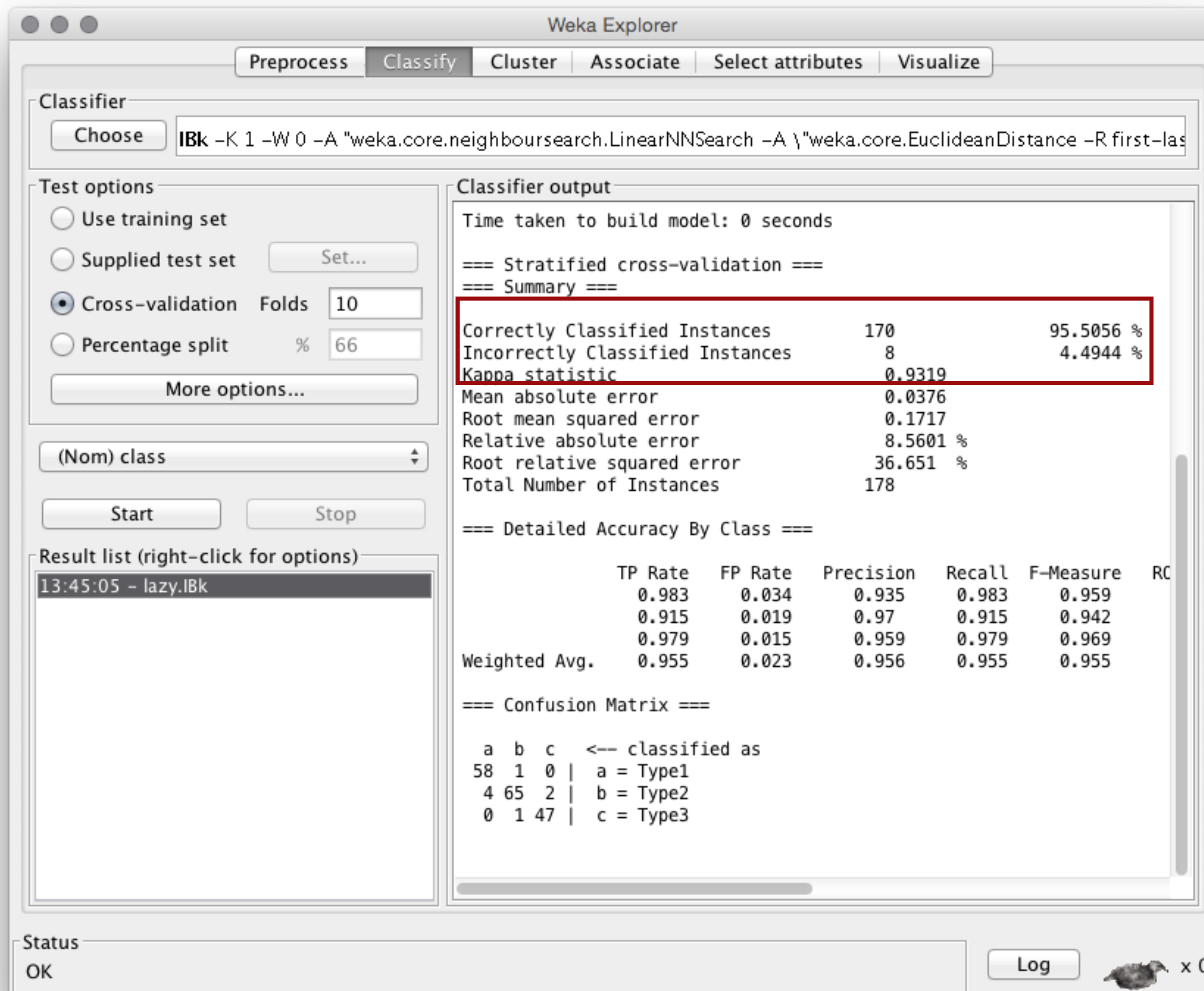
NB: Keep the "class" feature!



Tutorial Q1(a)

Assess the accuracy of a 1-nearest neighbour classifier with only the 3 most discriminating features included.

Re-run the 1NN classifier with the new feature subset.



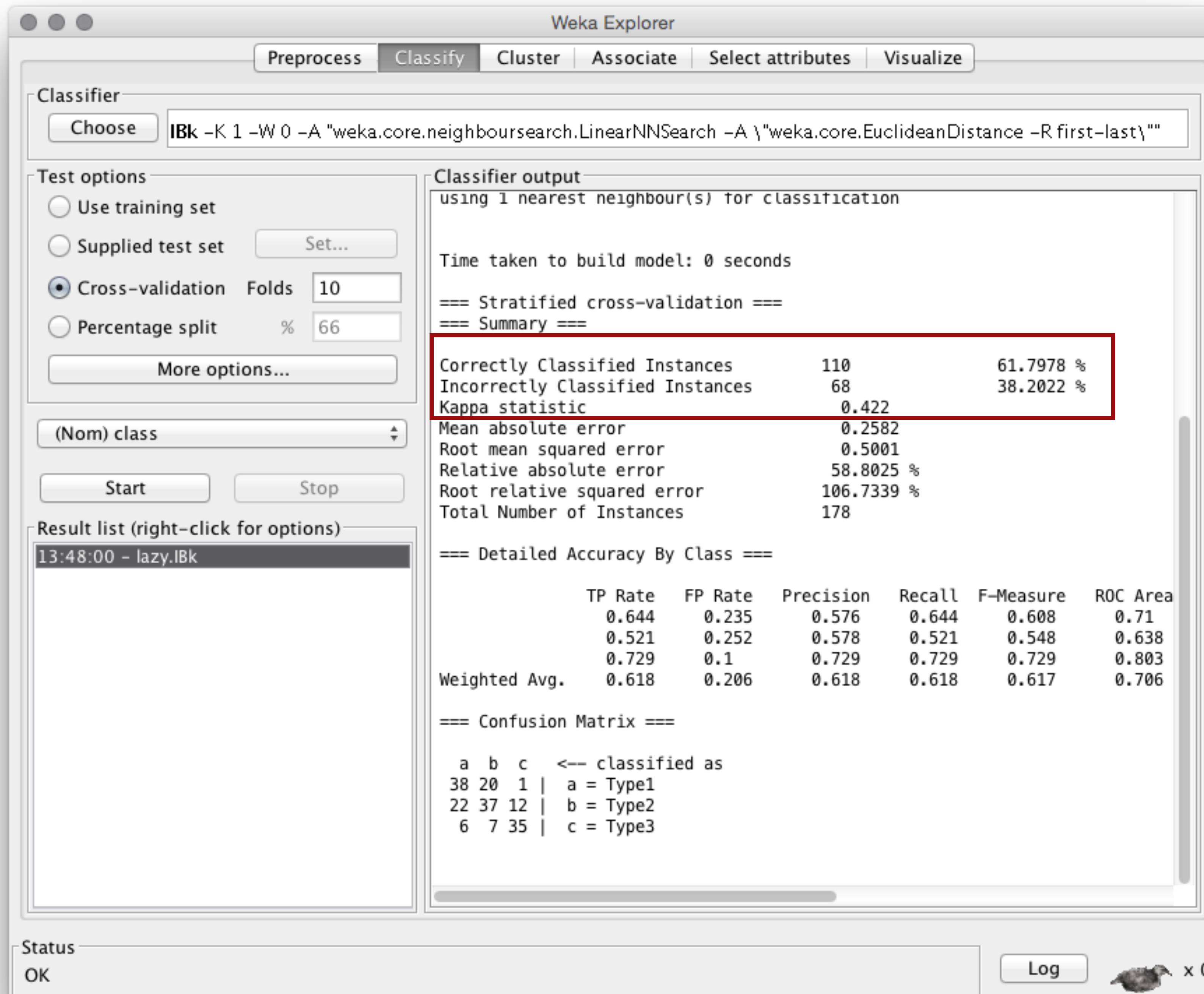
Tutorial Q1(a)

Assess the accuracy of a 1-nearest neighbour classifier with only the 3 least discriminating features included.

Least discriminating: 3, 8, 9

Load the ARFF file again, remove the unwanted features.

Re-run the 1NN classifier with the new feature subset.



Tutorial Q1(b)

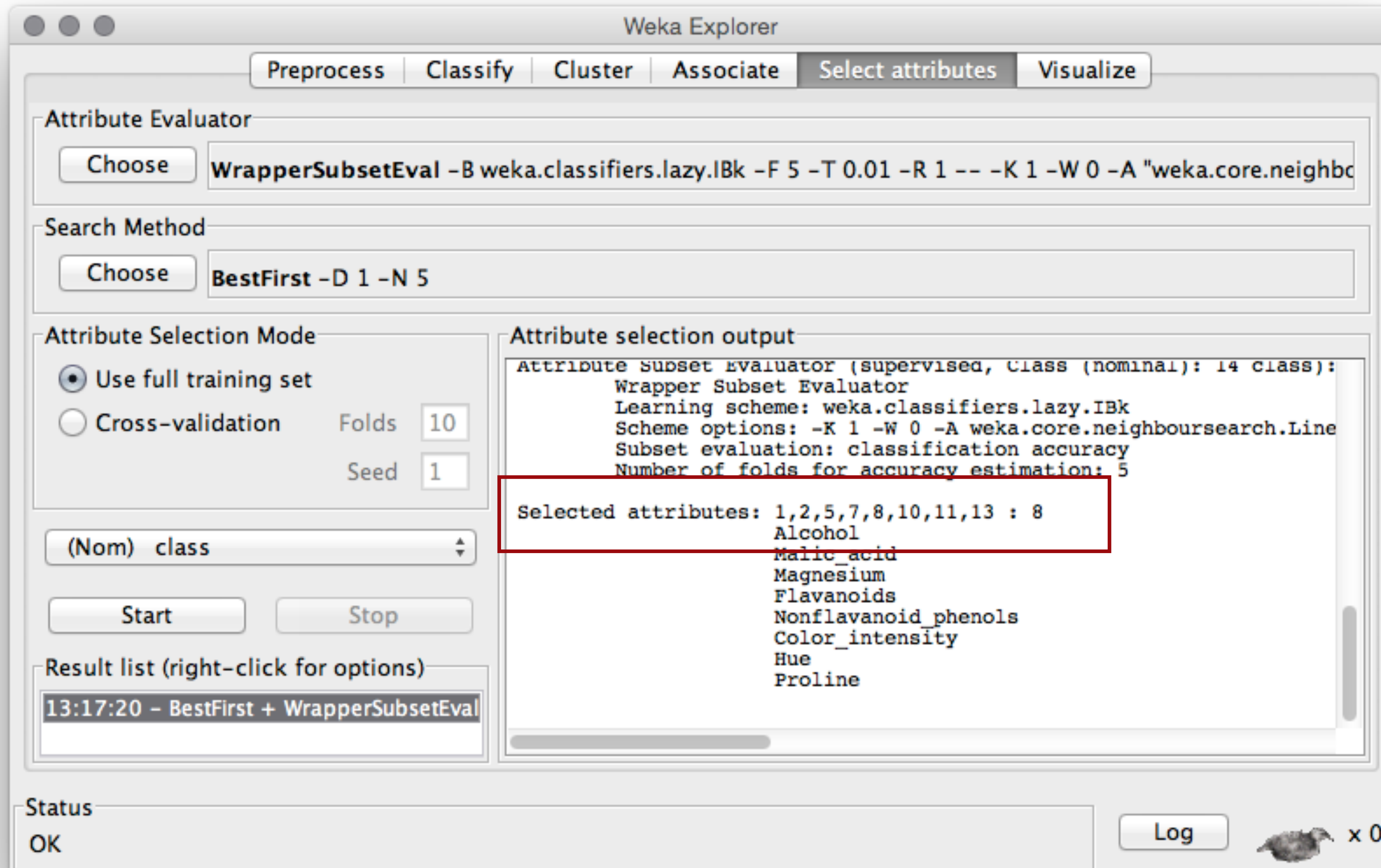
In Weka, apply wrapper-based feature selection to the *Wine* data set using a 1-nearest neighbour classifier and the following search strategies:

(i) forward sequential search

(ii) backward elimination

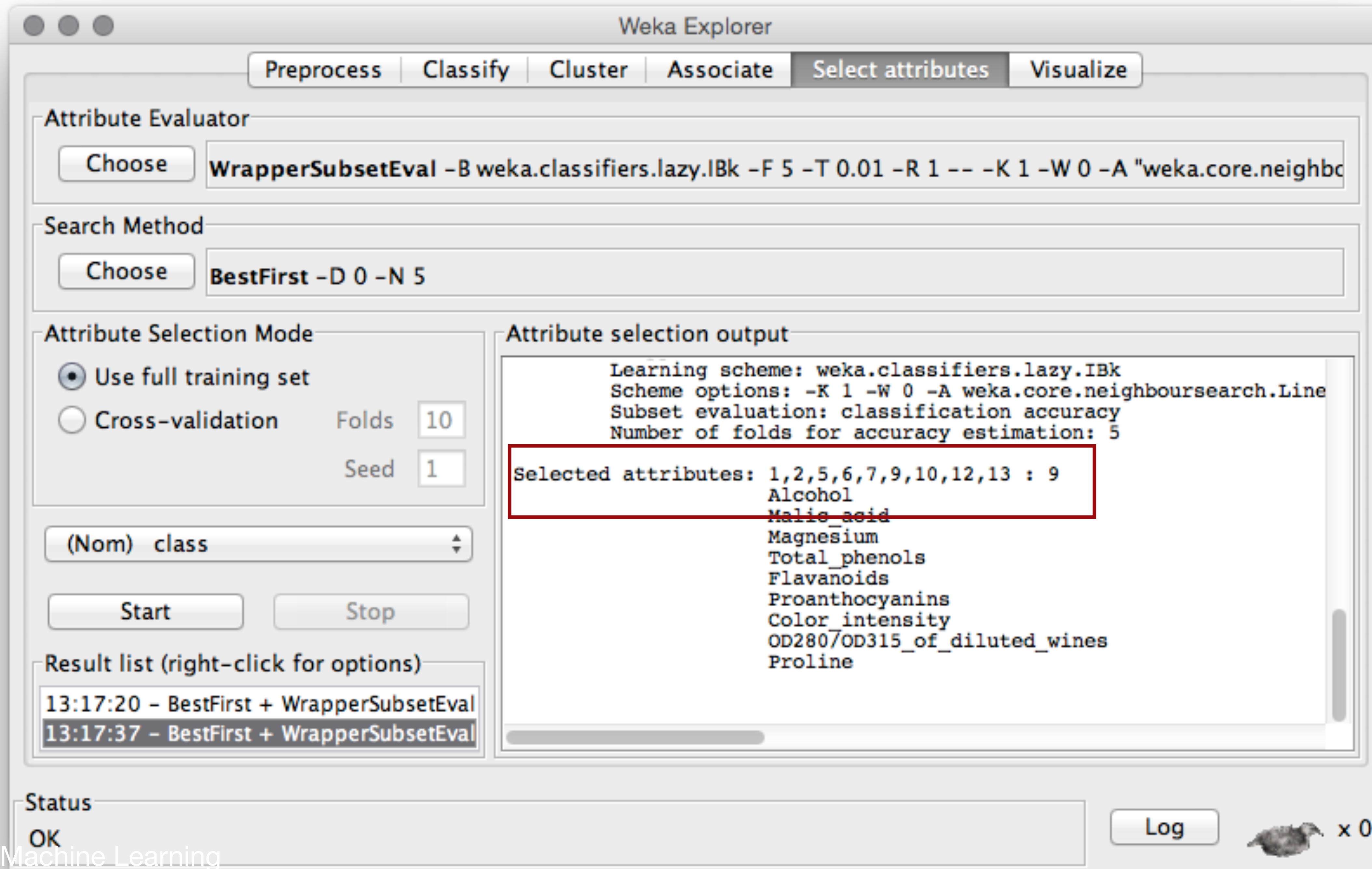
Tutorial Q1(b)

For Forward Sequential Search: In *Select attributes* tab, choose *WrapperSubsetEval* as the evaluator, *BestFirst* as the search method. Change options for search to *Forward* (1).



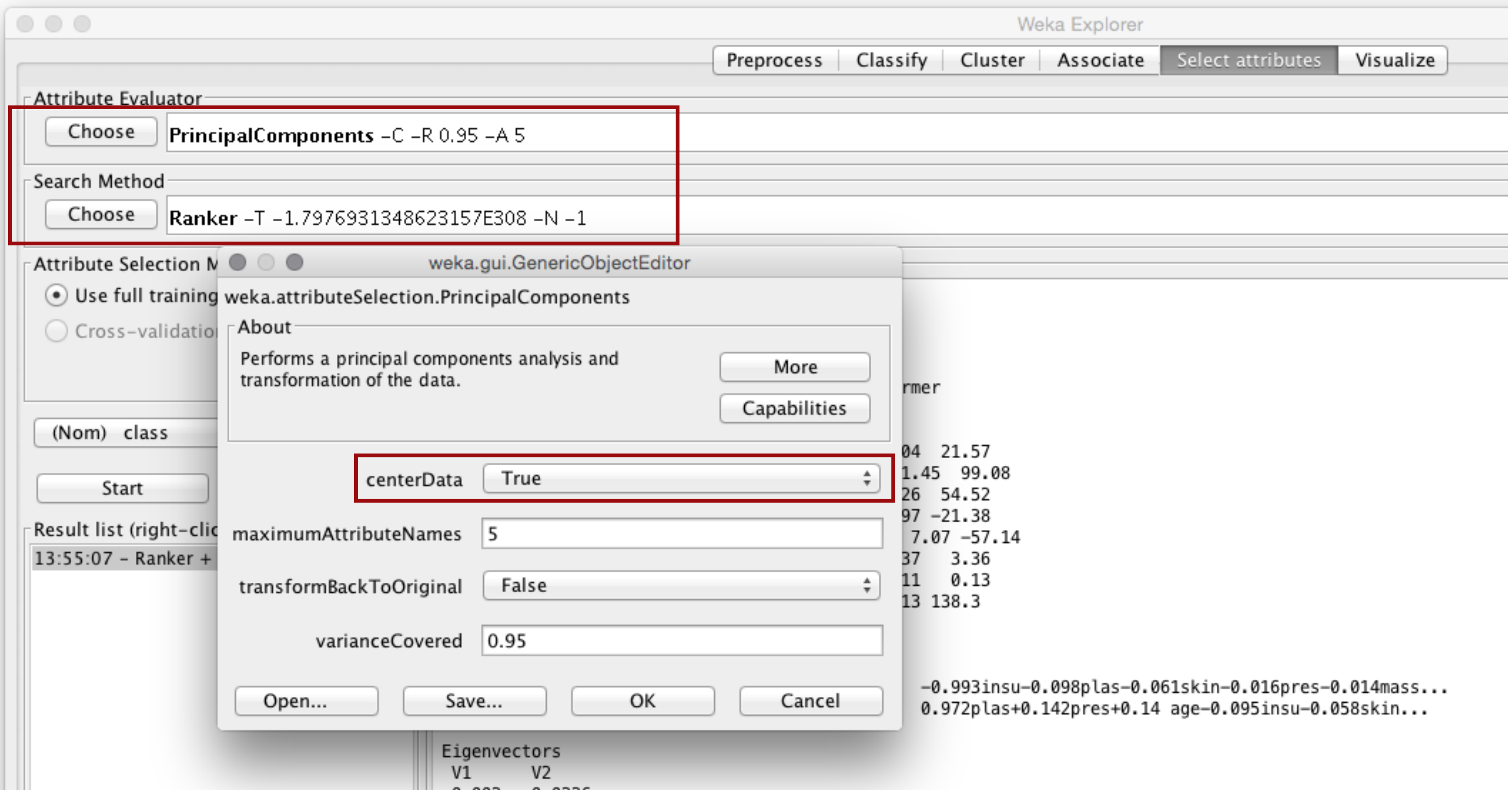
Tutorial Q1(b)

For Backward Elimination: In *Select attributes* tab, choose *WrapperSubsetEval* as the evaluator, *BestFirst* as the search method. Change options for search to *Backward* (0).



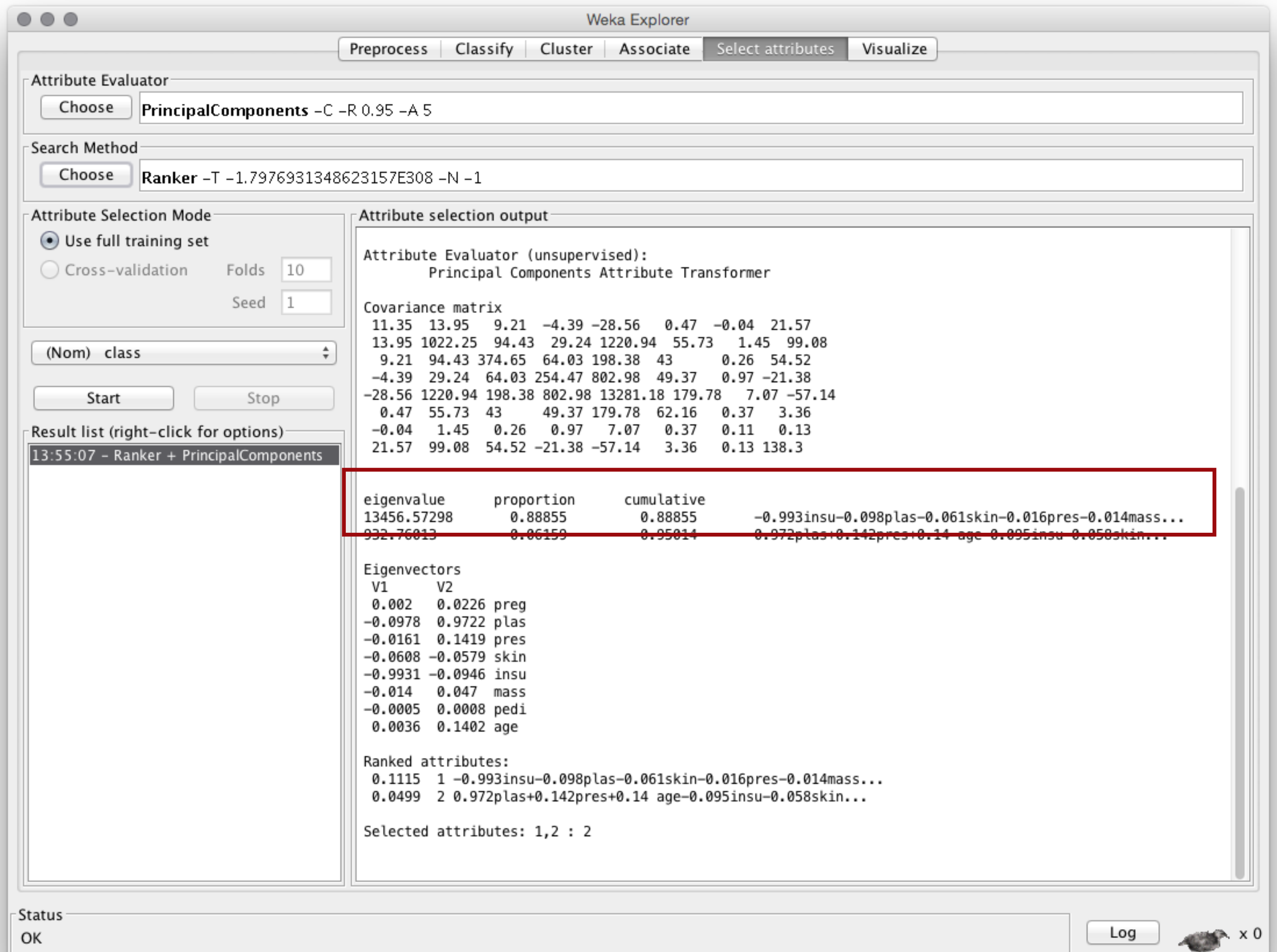
Tutorial Q2

In *Select attributes* tab, choose *PrincipalComponents* as the evaluator, and *Ranker* as the search method. Change options for PCA, set *centerData* to True to use the standard covariance matrix approach.



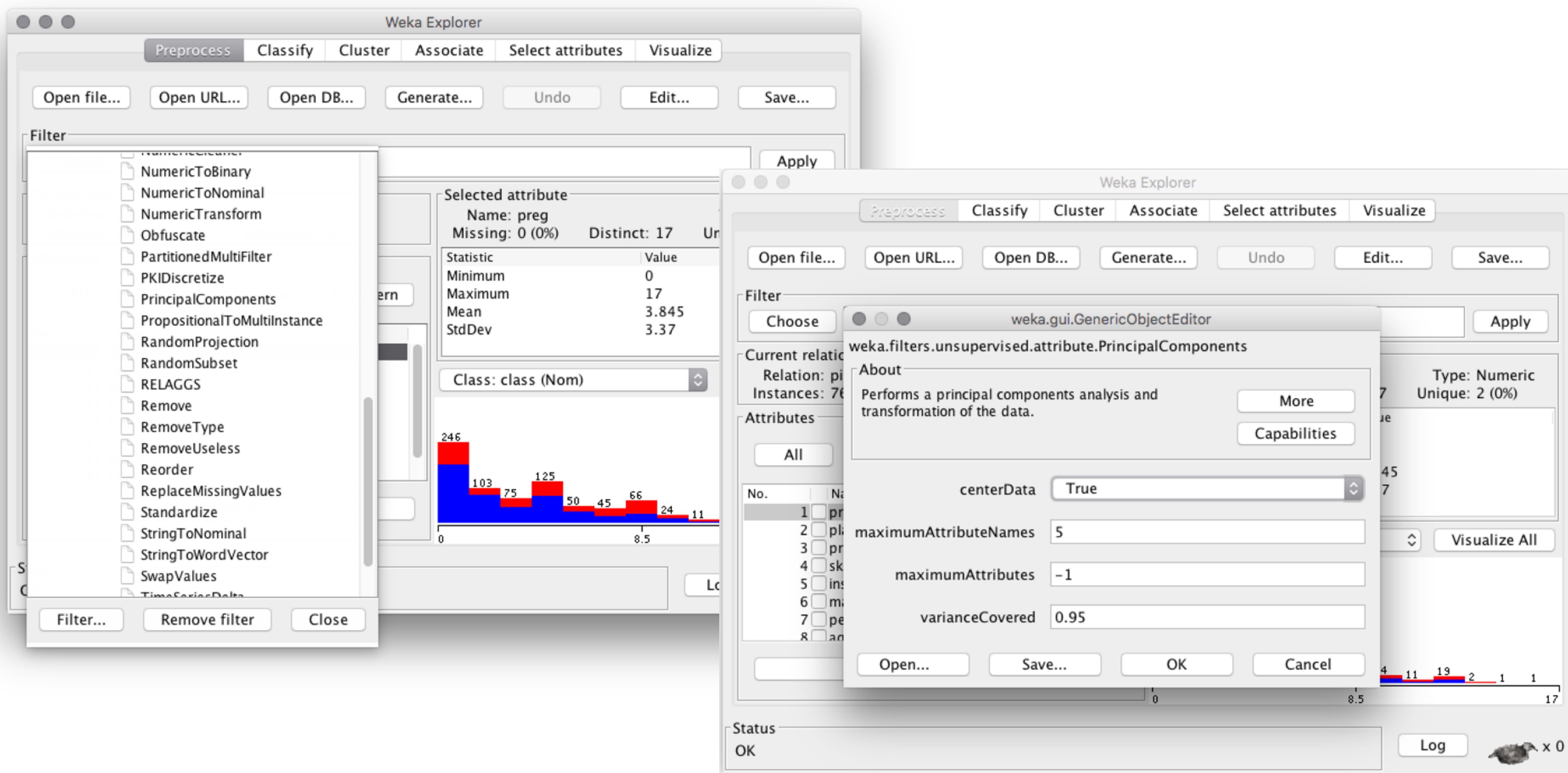
Tutorial Q2

PCA produces a new 2D feature space that accounts for > 95% of the variance of the data.



Tutorial Q2

- Can also apply PCA in the Weka *Preprocess* tab.
- Click *Filter*, then *Filters - Unsupervised - Attribute - Principal Components*. In Options, choose *Centre Data*, click *Apply*.



Tutorial Q3

(a) Explain why the feature subset selection problem with a k-Nearest Neighbour classifier is an exponential search problem.

(b) Describe in outline a Genetic Algorithm solution to this search problem.

(c) Describe crossover and mutation techniques for feature subset selection.

(d) Why is over-fitting a potential risk in wrapper feature subset selection?

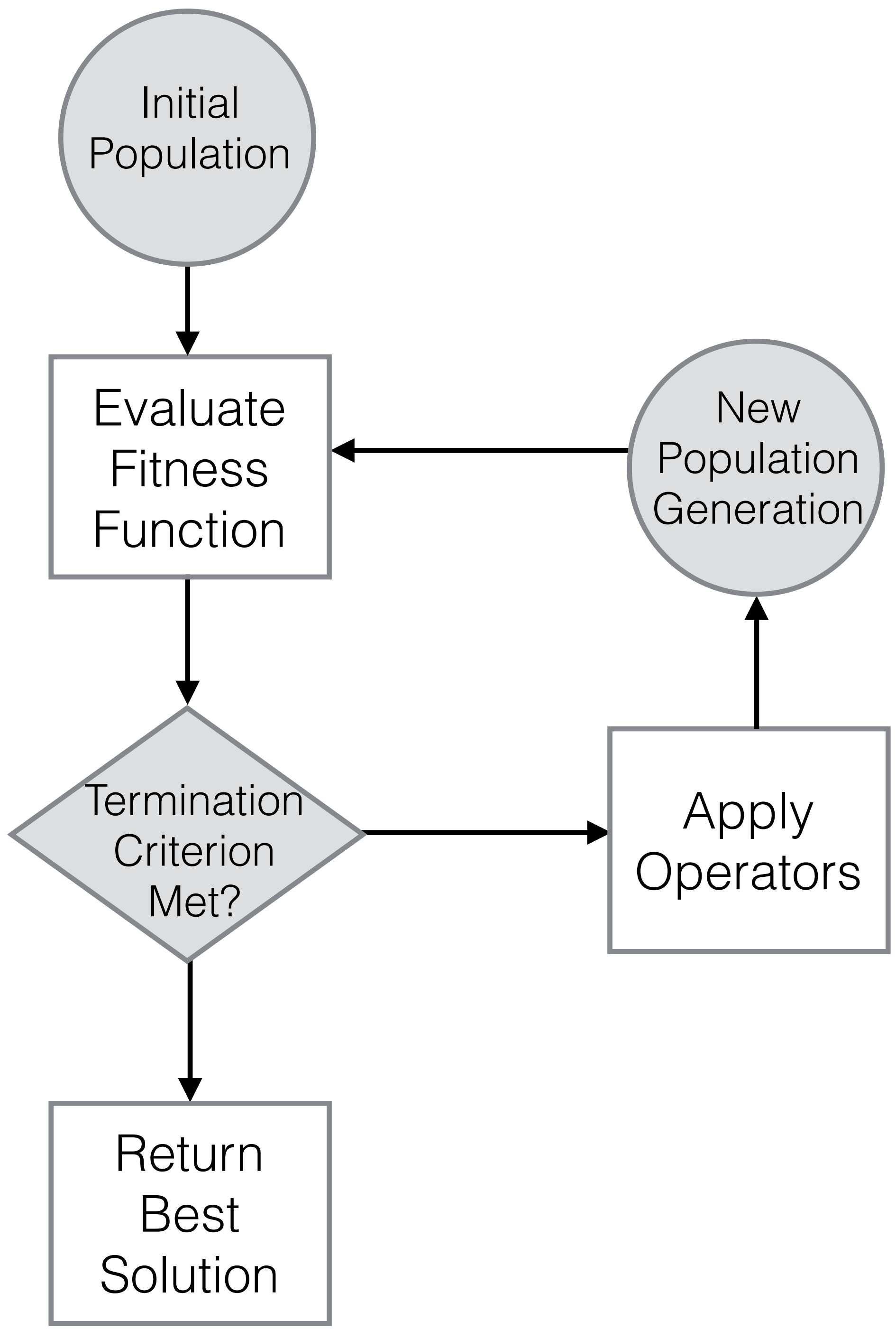
Tutorial Q3(a)

Explain why the feature selection problem with *k*-Nearest Neighbour Classifiers is an exponential search problem.

- Feature selection is in general NP-hard.
- Brute force evaluation of all feature subsets involves $\binom{d}{k}$ combinations if k is fixed, or 2^d subsets if not fixed.
- If we apply k-NN as a classifier in the context of a wrapper feature selection strategy, we need to evaluate accuracy.
- To measure generalisation accuracy, will typically want to apply k-fold cross validation
  e.g. 10-fold cross validation ⇒ 10 k-NN runs for every subset

Tutorial Q3(b)

Outline of a Genetic Algorithm solution…



- Involves applying creating "generations" of feature subset solutions, which are iteratively improved.

- *Population*: Different candidate feature subsets.
- *Fitness function*: Classification accuracy using a given classifier on each subset, based on 10 fold cross-validation

Tutorial Q3(c)

Describe crossover and mutation techniques for a Genetic Algorithm solution to the same problem.

Crossover Operator

Parents	1	1	1	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	0	0
Offspring	1	1	1	1	1	1	0	0	0	0

Mutation Operator

Before	1	0	0	1	1	0	0	1	1	1
After	0	0	0	1	1	0	0	1	1	1

10% mutation rate

Before	1	0	0	1	1	0	0	1	1	1
After	0	0	0	1	1	0	0	1	0	1

20% mutation rate

Tutorial Q3(d)

- Wrappers can be prone to overfitting when the chosen feature subsets are used to build classifiers on unseen data. We simply found the best feature subset for the training data!
- Using 10-fold cross validation and/or training-test splits to evaluate accuracy of the subsets can reduce overfitting to some extent.
- However, intensive search methods (e.g. Genetic Algorithms) can still identify feature subsets that "on average" favour the characteristics of the training data, and perform poorly on real unseen data that was never used in the feature selection process.

"Overfitting in feature selection appears to be exacerbated by the intensity of the search, since the more feature subsets that are visited the more likely the search is to find a subset that overfits"
- Loughrey & Cunningham, 2004

Tutorial Q3(d)

Journal of Machine Learning Research 3 (2003) 1371-1382 Submitted 5/02; Published 3/03

Overfitting in Making Comparisons Between Variable Selection Methods

Juha Reunanen JUHA.REUNANEN@FI.ABB.COM
ABB, Web Imaging Systems
P.O. Box 94, 00381 Helsinki, Finland

Editors: Isabelle Guyon and André Elisseeff

Abstract

This paper addresses a common methodological flaw in the comparison of variable selection meth-
ods. A practical approach to guide the search or the selection process is to compute cross-va
performance estimates of the different variable subsets. Used with computationally intensiv
algorithms, these estimates may overfit and yield biased predictions. Therefore, they cannot
reliably to compare two selection methods, as is shown by the empirical results of this pa
stead, like in other instances of the model selection problem, independent test sets should
for determining the final performance. The claims made in the literature about the superi
more exhaustive search algorithms over simpler ones are also revisited, and some of them ir

Keywords: Variable selection; Algorithm comparison; Overfitting; Cross-validation; *k*
neighbors

Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets·

John Loughrey, Pádraig Cunningham

Trinity College Dublin, College Green, Dublin 2, Ireland.
{John.Loughrey, Padraig.Cunningham}@cs.tcd.ie

Abstract. In Wrapper based feature selection, the more states that are visited during the search phase of the algorithm the greater the likelihood of finding a feature subset that has a high internal accuracy while generalizing poorly. When this occurs, we say that the algorithm has overfitted to the training data. We outline a set of experiments to show this and we introduce a modified genetic algorithm to address this overfitting problem by stopping the search before overfitting occurs. This new algorithm called GAWES (Genetic Algorithm With Early Stopping) reduces the level of overfitting and yields feature subsets that have a better generalization accuracy.