

# **COMP47460 Tutorial**

## **Dimension Reduction**

**Aonghus Lawlor**  
**Derek Greene**

**School of Computer Science**  
**Autumn 2018**



# Reminder - Selection v Transformation

---

- Two general strategies for dimension reduction:

## Feature Selection

- Tries to find a minimum subset of the original features that optimises one or more criteria, rather than producing an entirely new set of dimensions for the data.

e.g. Filters, Wrappers

## Feature Transformation (Feature Extraction)

- Transforms the original features of a dataset to a completely new, smaller, more compact feature set, while retaining as much information as possible.

e.g. Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA)

# Applying Filters and Wrappers

---

- **General filter approach:**

1. Choose the feature selection criterion (e.g. Information Gain)
2. Apply the filter to rank the features based on the criterion.
3. Select top- $k$  ranked features using an appropriate strategy.
4. Remove the other features from your dataset.
5. Apply the classifier on this new version of the dataset.

- **General wrapper approach:**

1. Choose the classifier, search strategy, and evaluation strategy.
2. Apply the wrapper to select a subset of features.
3. Remove the other features from your dataset.
4. Apply the same classifier on this new version of the dataset.

# Tutorial Q1

---

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

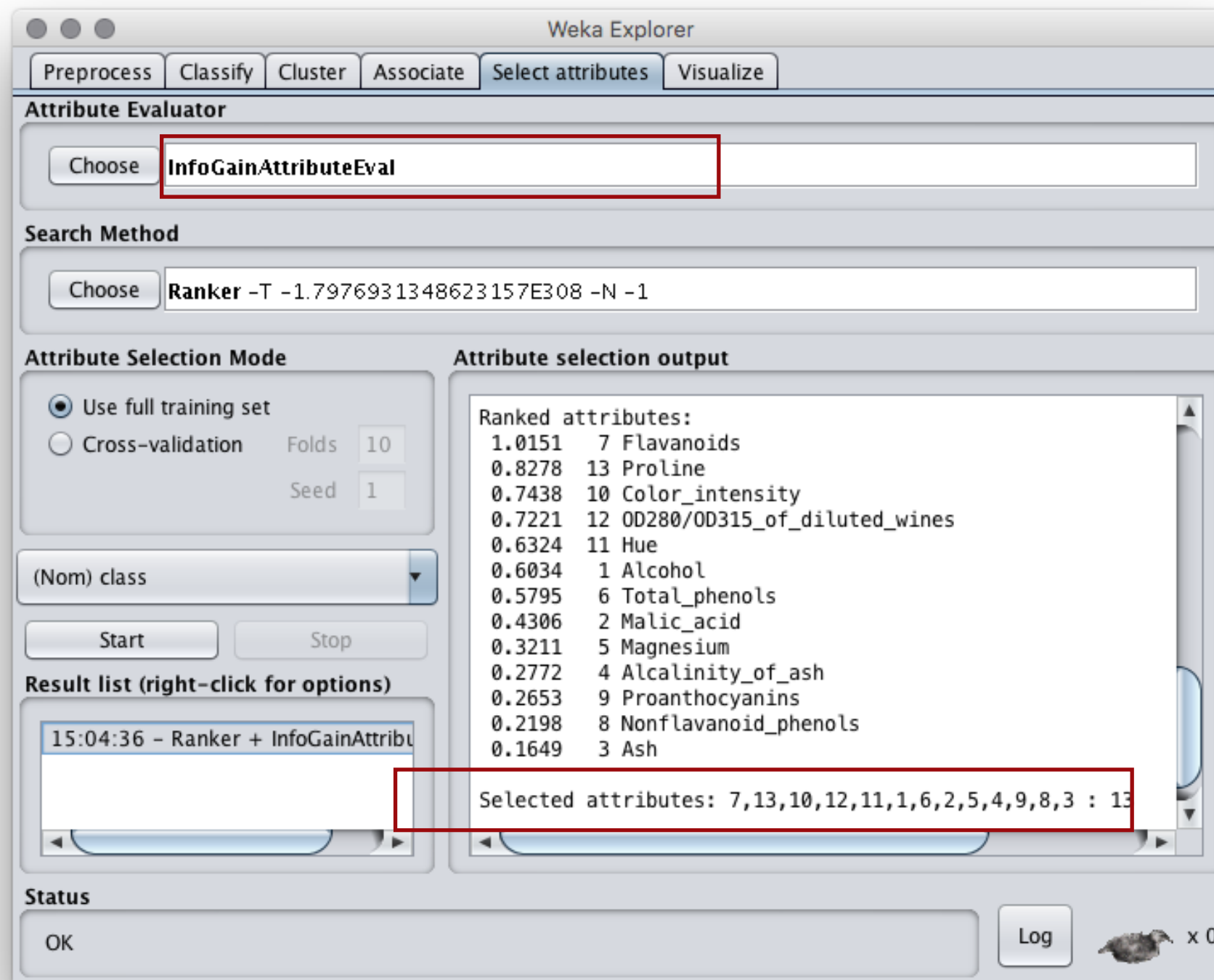
Based on these results, assess the 10-fold cross-validation classification accuracy of a 1-Nearest Neighbour classifier with:

- (i) only the 3 most discriminating features included
- (ii) only the 3 least discriminating features included



# Tutorial Q1

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



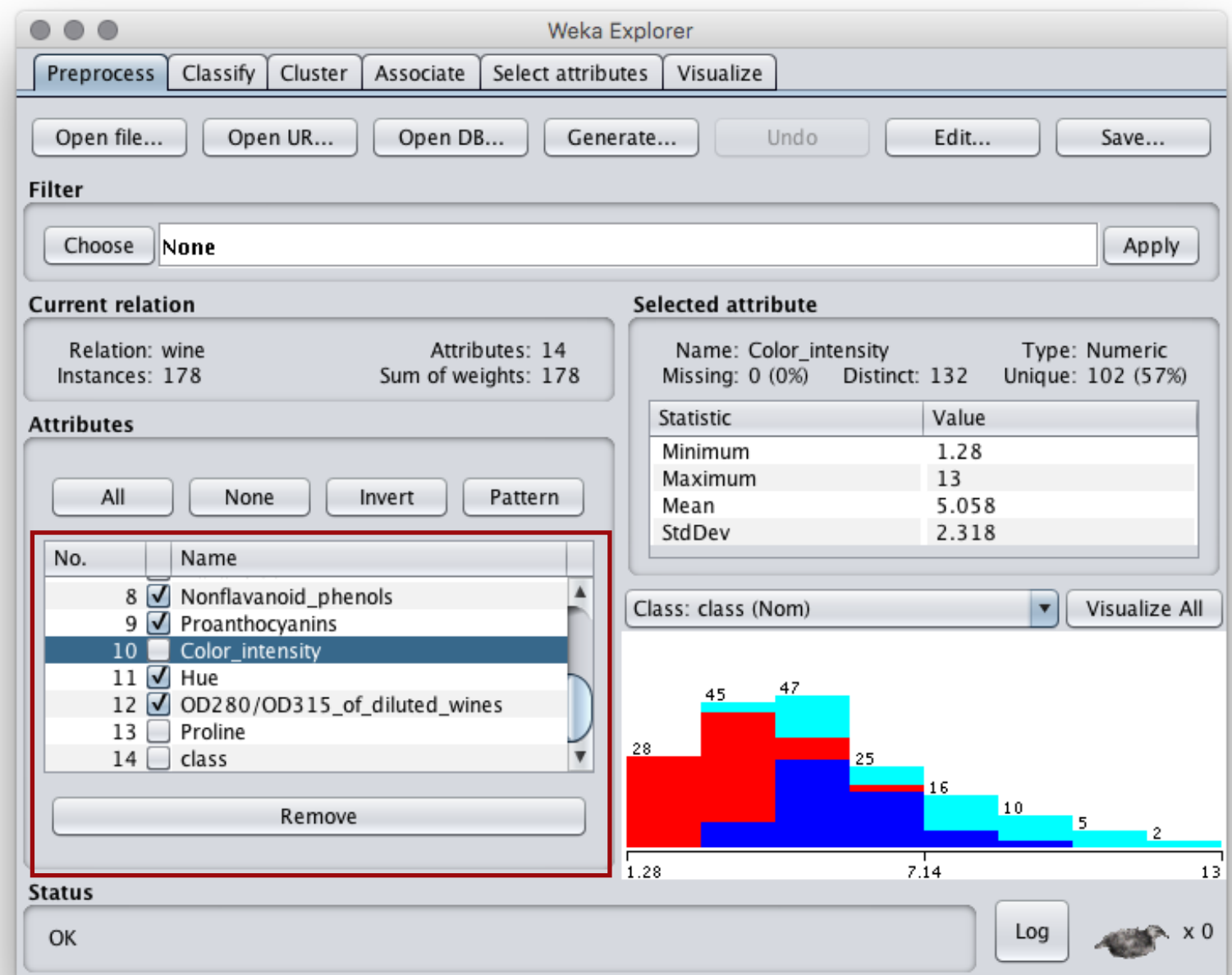
# Tutorial Q1(i)

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!



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open UR... | Open DB... | Generate... | Undo | Edit... | Save...

Filter: Choose None Apply

Current relation: Relation: wine, Instances: 178, Attributes: 14, Sum of weights: 178

Attributes: All | None | Invert | Pattern

No.	Name
8	<input checked="" type="checkbox"/> Nonflavanoid_phenols
9	<input checked="" type="checkbox"/> Proanthocyanins
10	<input type="checkbox"/> Color_intensity
11	<input checked="" type="checkbox"/> Hue
12	<input checked="" type="checkbox"/> OD280/OD315_of_diluted_wines
13	<input type="checkbox"/> Proline
14	<input type="checkbox"/> class

Remove

Selected attribute: Name: Color\_intensity, Type: Numeric, Missing: 0 (0%), Distinct: 132, Unique: 102 (57%)

Statistic	Value
Minimum	1.28
Maximum	13
Mean	5.058
StdDev	2.318

Class: class (Nom) Visualize All

28 45 47 25 16 10 5 2

1.28 7.14 13

Status: OK Log x 0

# Tutorial Q1(i)

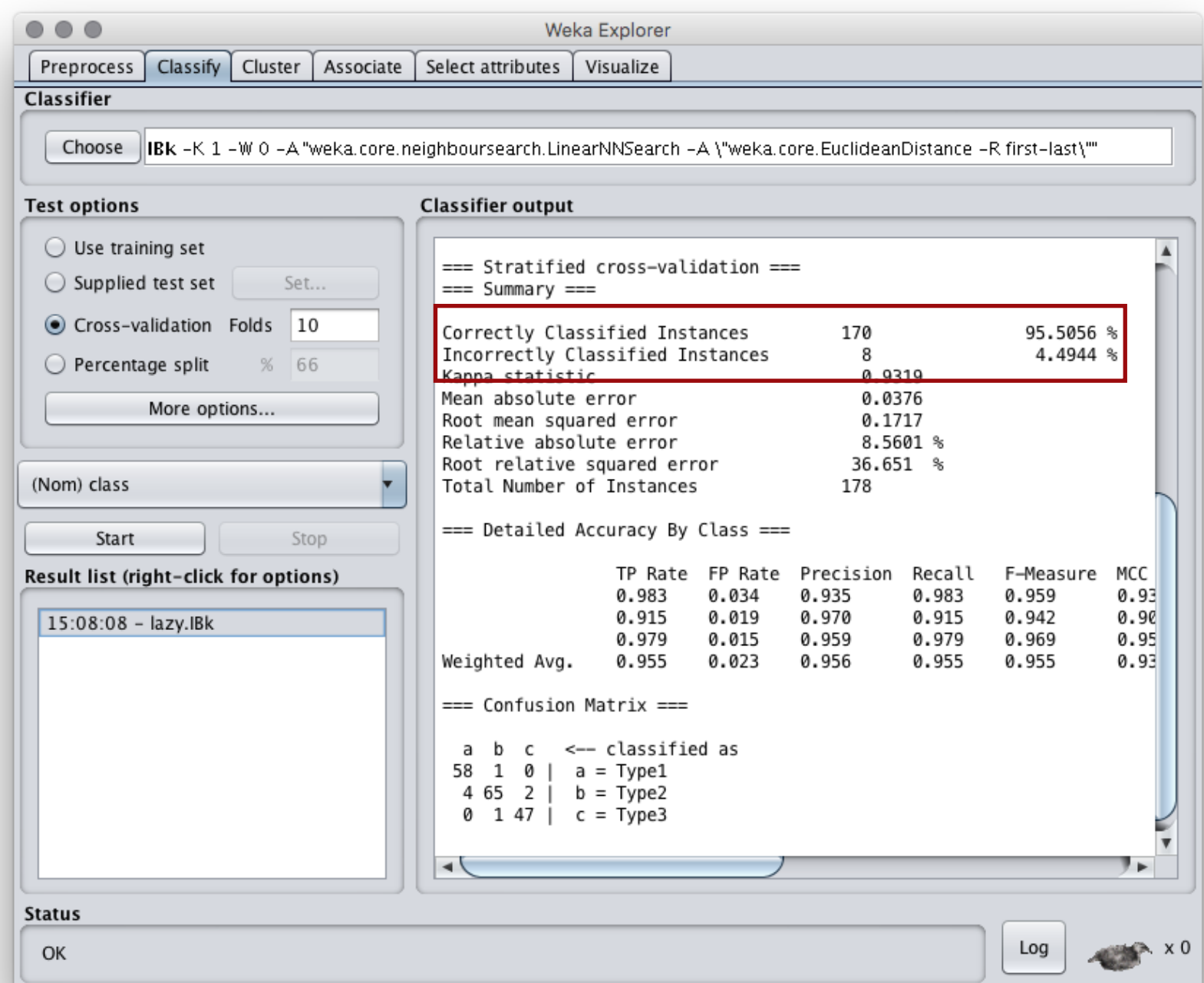
Assess the accuracy of a 1-nearest neighbour classifier with only the 3 most discriminating features included.

Most discriminating:  
7, 13, 10

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

NB: Keep the “class” feature!

Run the 1NN classifier with the new feature subset.



The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier is 'IBk -K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A {\"weka.core.EuclideanDistance -R first-last\"}'. The 'Test options' section shows 'Cross-validation' selected with 'Folds' set to 10. The 'Classifier output' section displays the results of the stratified cross-validation.

**Stratified cross-validation Summary**

Metric	Value	Percentage
Correctly Classified Instances	170	95.5056 %
Incorrectly Classified Instances	8	4.4944 %
Kappa statistic	0.9310	
Mean absolute error	0.0376	
Root mean squared error	0.1717	
Relative absolute error	8.5601 %	
Root relative squared error	36.651 %	
Total Number of Instances	178	

**Detailed Accuracy By Class**

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
Type1	0.983	0.034	0.935	0.983	0.959	0.93
Type2	0.915	0.019	0.970	0.915	0.942	0.90
Type3	0.979	0.015	0.959	0.979	0.969	0.95
Weighted Avg.	0.955	0.023	0.956	0.955	0.955	0.93

**Confusion Matrix**

a	b	c	<-- classified as
58	1	0	a = Type1
4	65	2	b = Type2
0	1	47	c = Type3

# Tutorial Q1(ii)

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

Least discriminating:  
3, 8, 9

Reload the ARFF file  
again, remove the  
unwanted features.

Run the 1NN classifier  
with the new feature  
subset.

The screenshot shows the Weka Explorer window with the 'Classify' tab selected. The classifier is 'IBk' with parameters '-K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\"'.

**Test options:**

- ☐ Use training set
- ☐ Supplied test set (Set...)
- ☒ Cross-validation Folds: 10
- ☐ Percentage split %: 66
- More options...

**Classifier output:**

=== Stratified cross-validation ===  
=== Summary ===

Correctly Classified Instances	110	61.7978 %
Incorrectly Classified Instances	68	38.2022 %
Kappa statistic	0.422	
Mean absolute error	0.2582	
Root mean squared error	0.5001	
Relative absolute error	58.8025 %	
Root relative squared error	106.7339 %	
Total Number of Instances	178	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MC
0	0.644	0.235	0.576	0.644	0.608	0.644
1	0.521	0.252	0.578	0.521	0.548	0.521
2	0.729	0.100	0.729	0.729	0.729	0.729
Weighted Avg.	0.618	0.206	0.618	0.618	0.617	0.618

=== Confusion Matrix ===

a	b	c	<-- classified as
38	20	1	a = Type1
22	37	12	b = Type2
6	7	35	c = Type3

**Result list (right-click for options):**

- 15:09:53 - lazy.IBk

**Status:** OK

Log x 0



# Tutorial Q2

---

In Weka, apply *wrapper-based feature selection* to the *Wine* dataset using a 3-Nearest Neighbour classifier and the following search strategies:

- (i) forward sequential search
- (ii) backward elimination

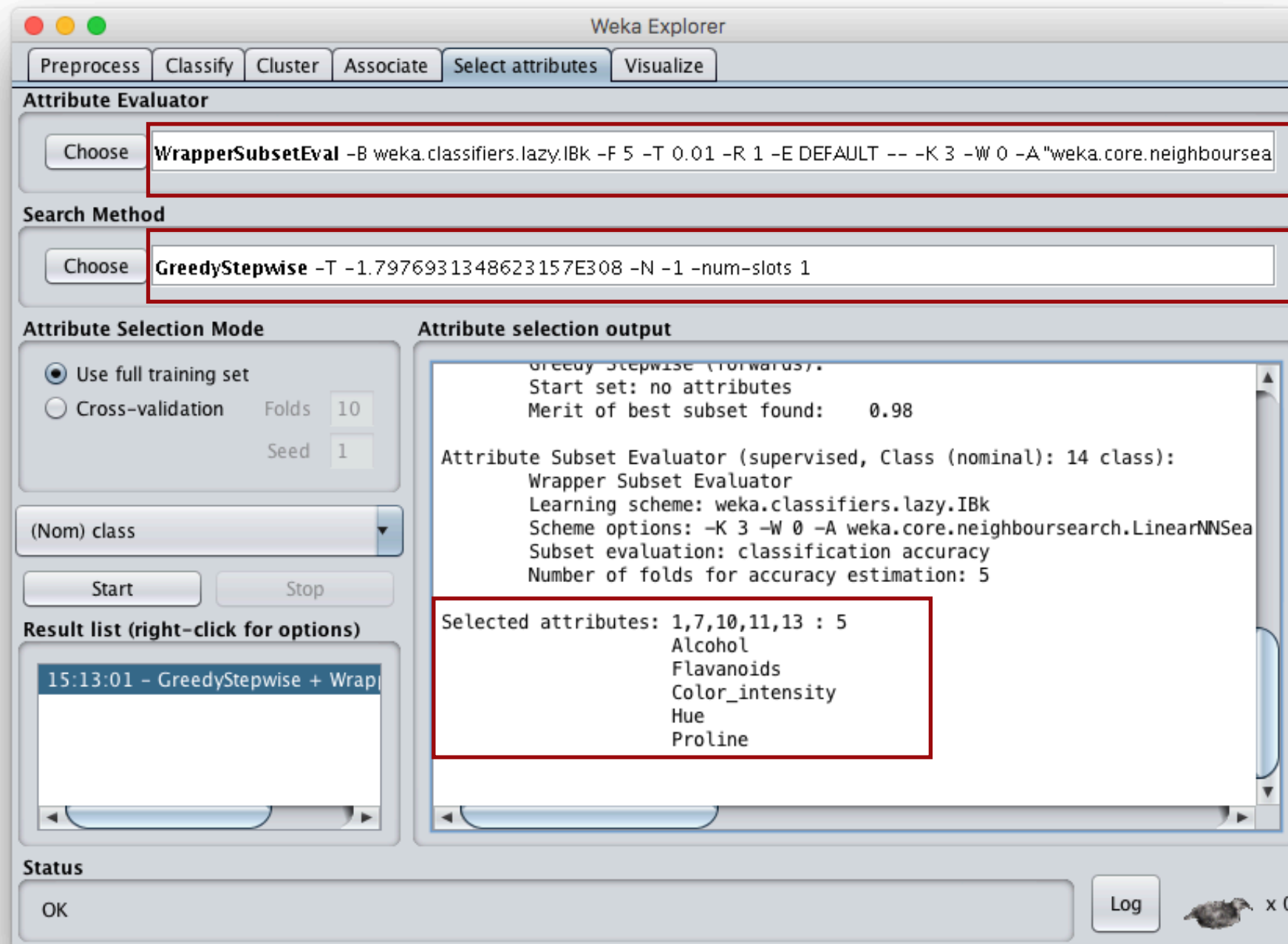
Which common features were selected by both search strategies?

Would it be appropriate to use either of the resulting feature subsets in conjunction with a Decision Tree classifier? Justify your answer.

NB: Reload the Wine dataset with all features.

# Tutorial Q2

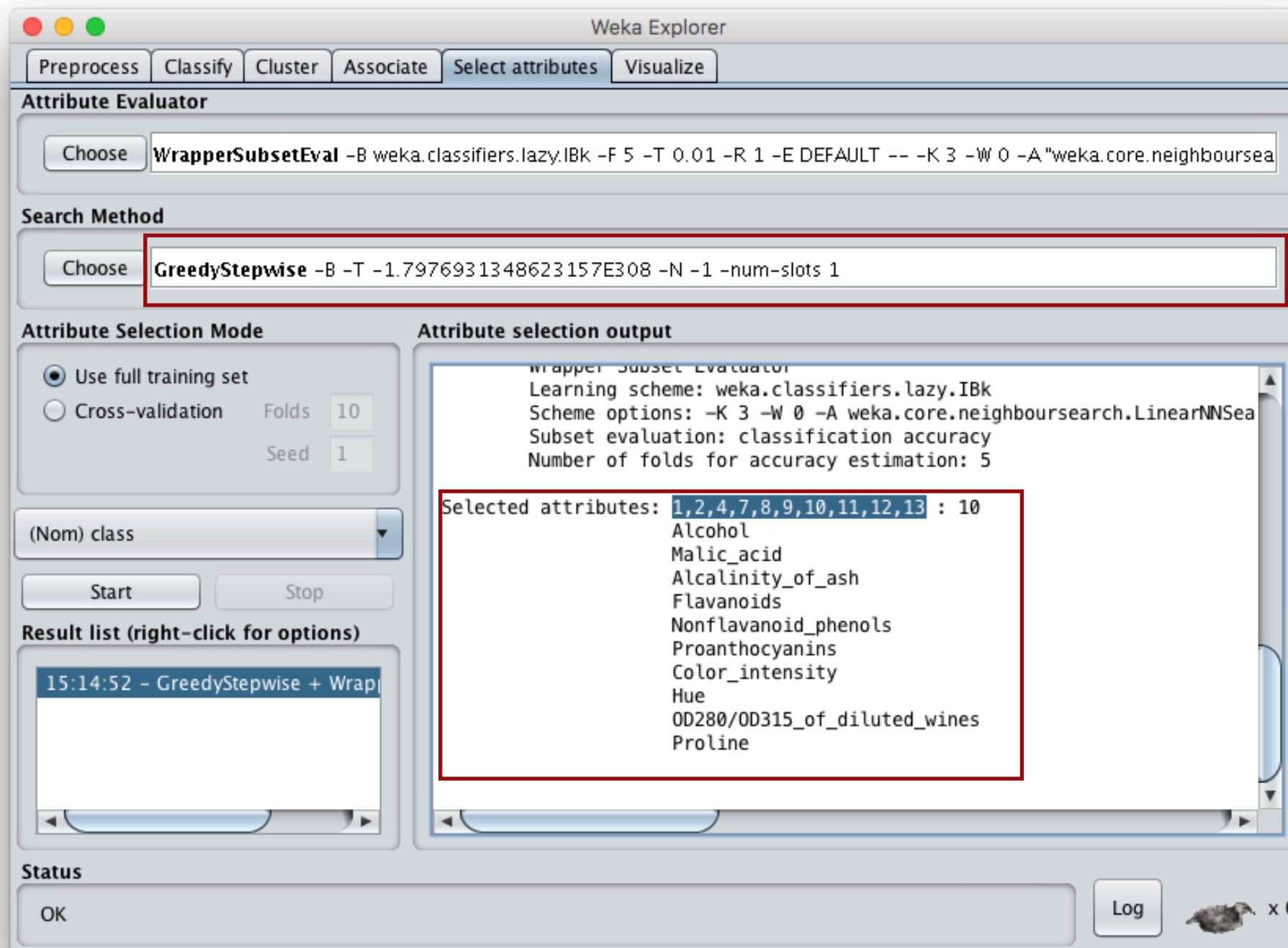
For **Forward Sequential** search: In *Select attributes* tab, choose *WrapperSubsetEval* as evaluator. Change options for wrapper to *IBk* (k=3). Choose *GreedyStepwise* as the search method, with option *SearchBackwards* = *False*.



Selected:  
1,7,10,11,13

# Tutorial Q2

For **Backward Elimination**: In *Select attributes* tab, choose *WrapperSubsetEval* as evaluator. Change options for wrapper to *IBk* (k=3). Choose *GreedyStepwise* as the search method, with option *SearchBackwards = True*.



Selected:  
1,2,4,7,8,9,10,  
11,12,13

# Reminder - PCA

---

- **Principal Components (PCs)**: New features constructed as linear combinations of the original features, uncorrelated with one another.
- **PCA Process**:
  1. Calculate the mean of the columns of  $\mathbf{X}$ .
  2. Subtract the column means from each row of  $\mathbf{X}$ , to create the **centred matrix**  $\mathbf{Y}$ .
  3. Calculate the **covariance matrix**  $\mathbf{C} = \mathbf{Y}^T \mathbf{Y} / (n - 1)$
  4. Calculate the eigenvectors of the covariance matrix  $\mathbf{C}$ .
  5. The Principal Components (PCs) are given by the eigenvectors of  $\mathbf{C}$ . The  $i$ -th PC is given by the eigenvector corresponding to the  $i$ -th largest eigenvalue of  $\mathbf{C}$ .
  6. Select an appropriate number of PCs  $k$  and use them as a new reduced  $n \times k$  representation of the dataset.

# Tutorial Q3

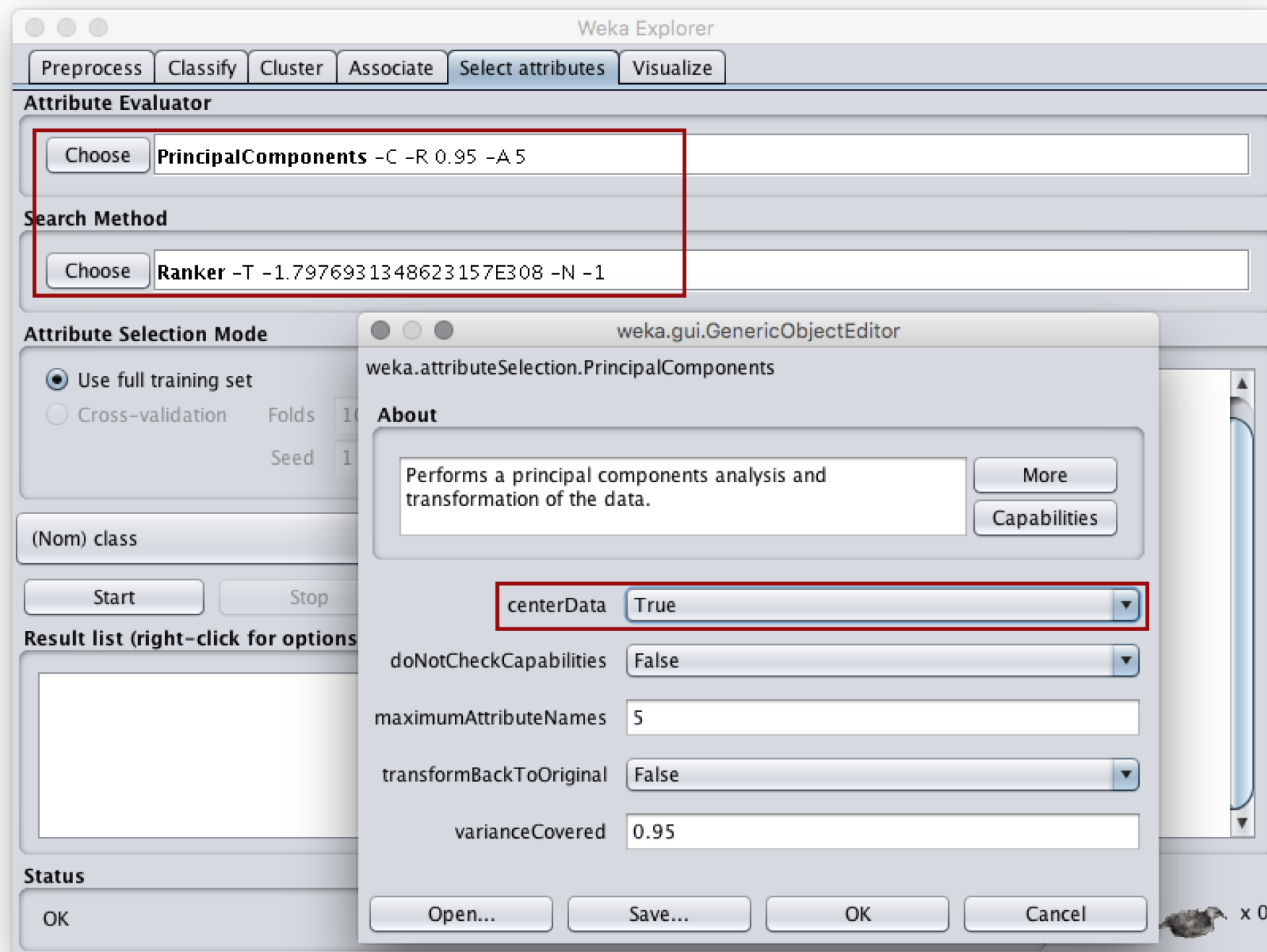
---

- Use Weka to apply PCA *feature transformation* to the Diabetes dataset in the ARFF file provided.  
(Use the Ranker search method in the 'Select Attributes' tab).
- How many Principal Components are selected by Weka?
- Produce a visualisation of the 2 leading Principal Components.



# Tutorial Q3

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 Q3

PCA in Weka selects 2 PCs, which account for  $> 95\%$  of the variance of the data (proportion is  $0.88855 + 0.06159 = 0.95014$ ).

Weka Explorer

Preprocess Classify Cluster Associate **Select attributes** Visualize

**Attribute Evaluator**

Choose

**Search Method**

Choose

**Attribute Selection Mode**

☒ Use full training set  
☐ Cross-validation Folds   
Seed

(Nom) class

Start Stop

**Result list (right-click for options)**

10:01:55 - Ranker + PrincipalComp

**Attribute selection output**

11.35	13.95	9.21	-4.39	-28.56	0.47	-0.04	21.57
13.95	1022.25	94.43	29.24	1220.94	55.73	1.45	99.08
9.21	94.43	374.65	64.03	198.38	43	0.26	54.52
-4.39	29.24	64.03	254.47	802.98	49.37	0.97	-21.38
-28.56	1220.94	198.38	802.98	13281.18	179.78	7.07	-57.14
0.47	55.73	43	49.37	179.78	62.16	0.37	3.36
-0.04	1.45	0.26	0.97	7.07	0.37	0.11	0.13
21.57	99.08	54.52	-21.38	-57.14	3.36	0.13	138.3

eigenvalue	proportion	cumulative
13456.57298	0.88855	0.88855
932.76013	0.06159	0.95014

-0.993insu-0.098plas-0.061skin-0.  
-0.972plas-0.142pres-0.14age+0.09

**Eigenvectors**

V1	V2
0.002	-0.0226
0.0078	0.0732

preg  
plas

Status

OK Log x 0

# Tutorial Q3

To visualise the PCs, right click the result on the *Result List* and choose *Visualize transformed data*. In the new window, click one of the small pairwise plots to zoom in.

