COMP30120 Tutorial

Ensembles

Derek Greene

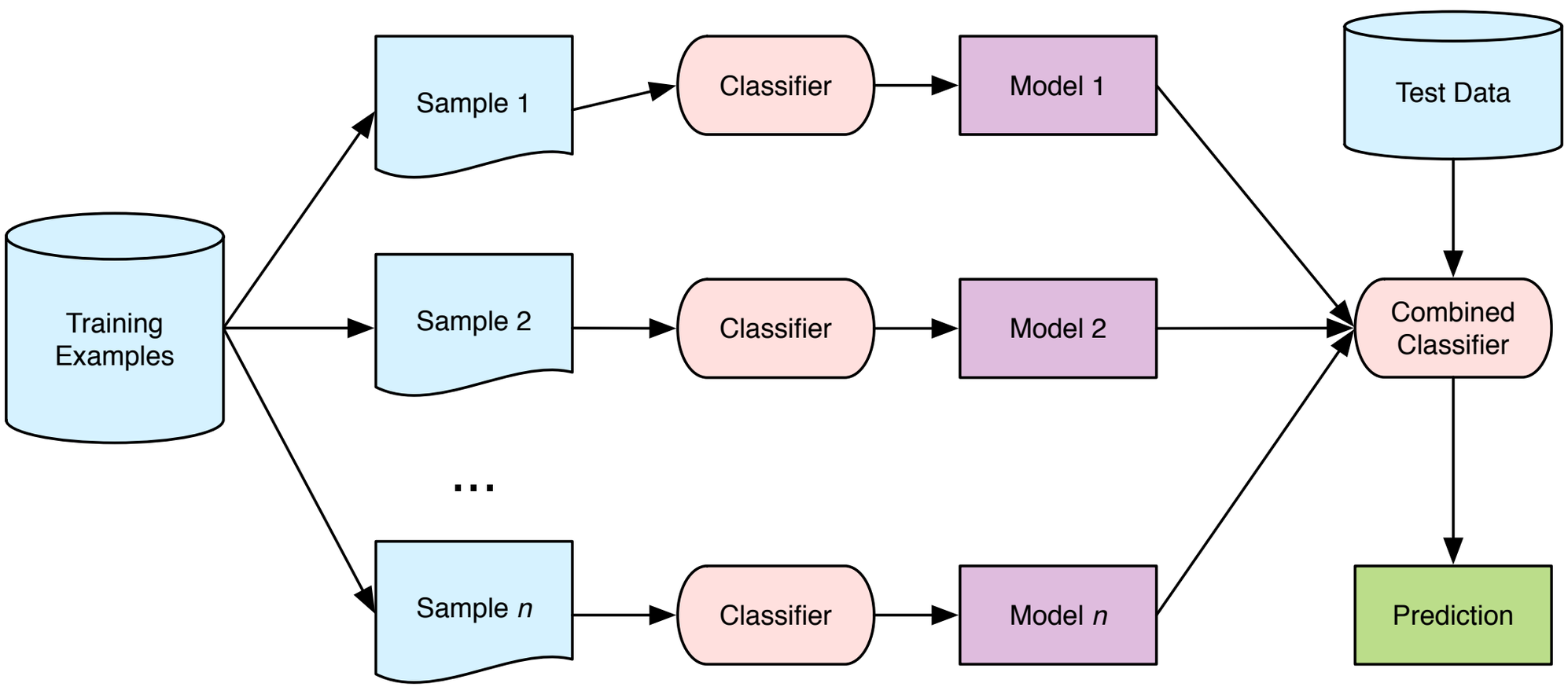
School of Computer Science and Informatics

Autumn 2015

Tutorial Q1(a)

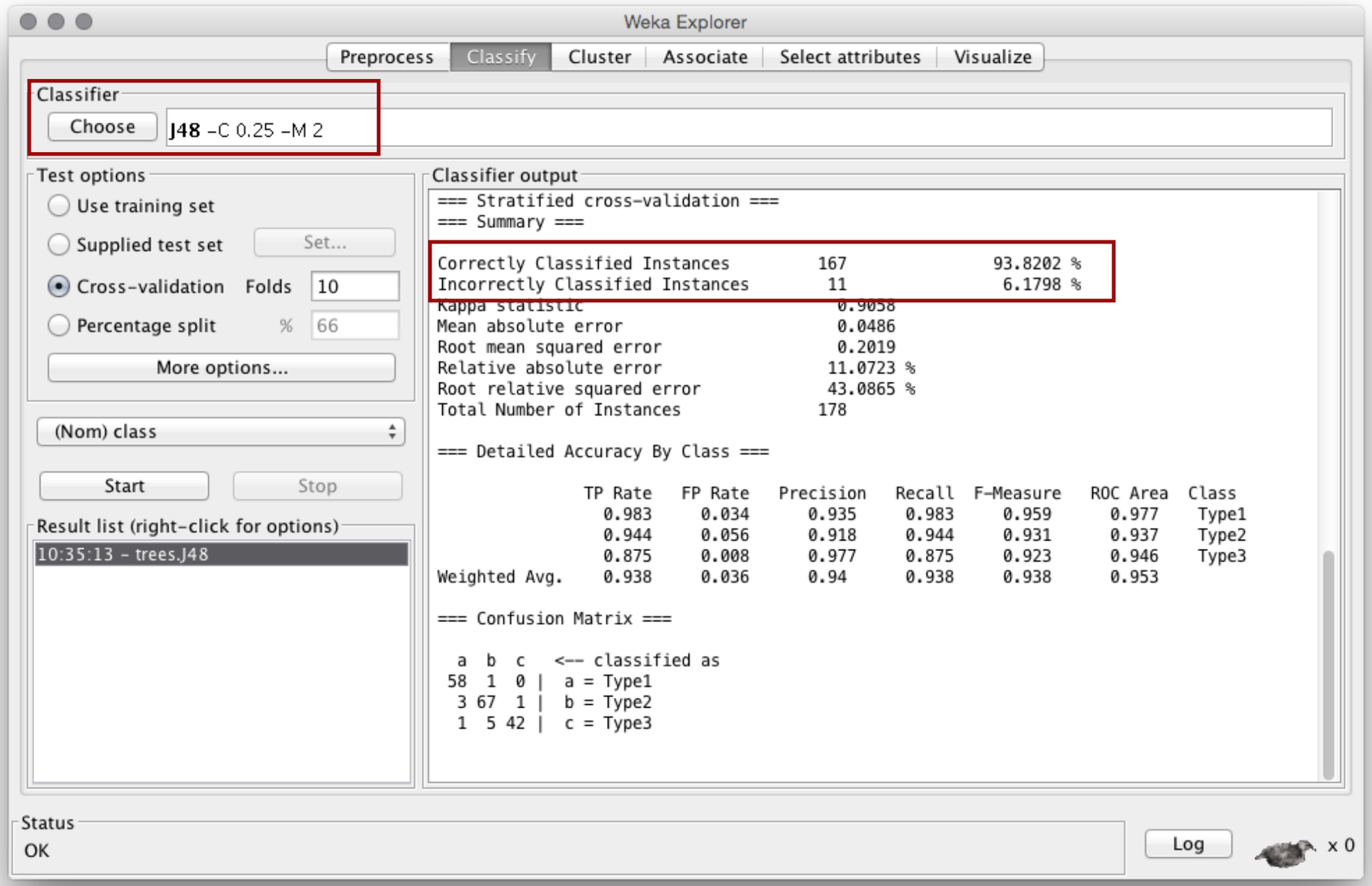
Q. *Bagging* (bootstrap aggregation) has a mechanism for achieving diversity for ensemble classifiers. Explain how it works.

1. Randomly sample from training data with replacement.
2. Apply a classifier to each sample independently.
3. Combine the outputs of the classifiers (e.g. majority voting).



Tutorial Q1(b)

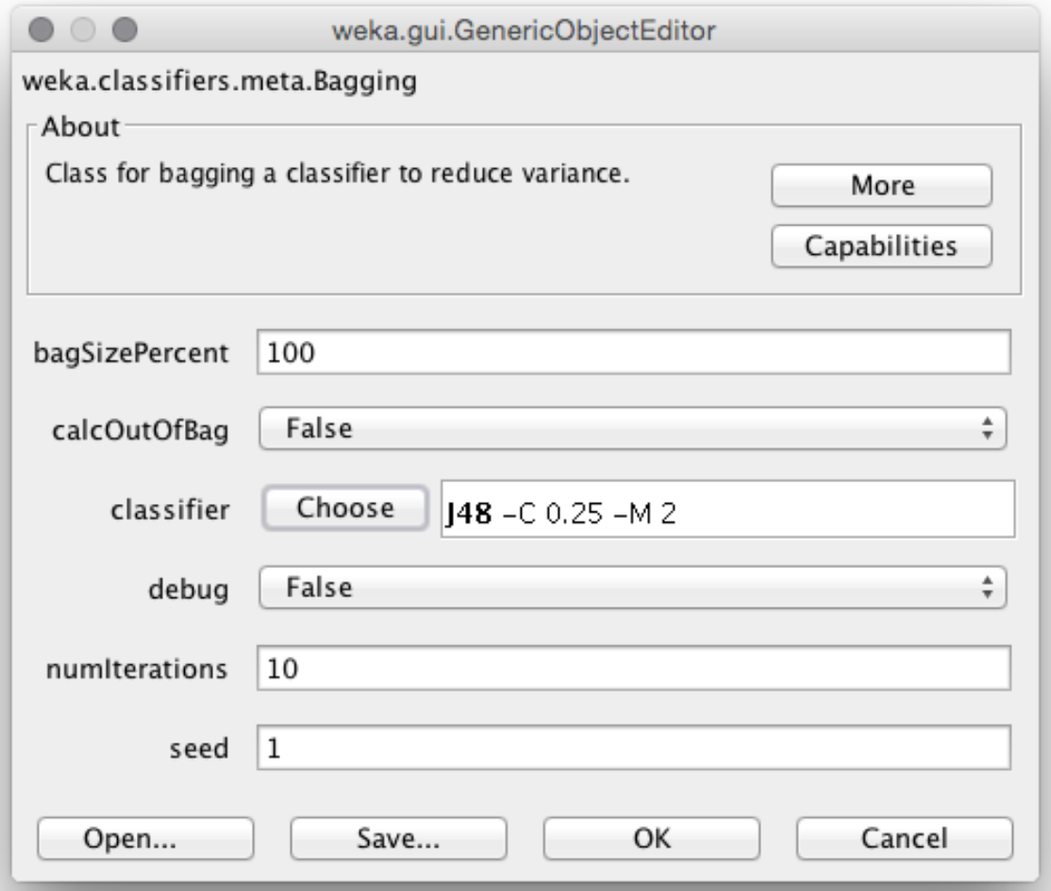
Q. In Weka, load the *Wine* data set using the ARFF file provided, and evaluate a decision tree classifier (J48) using 10-fold cross-validation. What percentage of instances are correctly classified?



Tutorial Q1(c)

Q. Apply ensemble classification using bagging to achieve diversity and with a decision tree classifier. What percentage of instances are now correctly classified with an ensemble of size 10?

1. Using Weka, click on the *Classify* tab.
2. Click *Choose*, select method *classifiers->meta->Bagging*.
3. Click *Bagging* in the box to the right. The configuration interface of the method appears.
4. Click *Choose*, select *J48*.
5. Set the *numIterations* to 10 (default).
6. Click *OK* button.
7. Click *Start* button to build the ensemble.



Tutorial Q1(c)

Weka Explorer

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

Classifier

Choose **Bagging** -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) class

Start | Stop

Result list (right-click for options)

- 10:35:13 - trees.J48
- 10:40:58 - meta.Bagging

Classifier output

```
Time taken to build model: 0.01 seconds

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

Correctly Classified Instances         168               94.382  %
Incorrectly Classified Instances        10                5.618  %
Kappa statistic                          0.915
Mean absolute error                      0.0693
Root mean squared error                  0.1657
Relative absolute error                 15.7898 %
Root relative squared error             35.3632 %
Total Number of Instances              178

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.966     0.034      0.934     0.966     0.95       0.996    Type1
                 0.901     0.019      0.97      0.901     0.934      0.992    Type2
                 0.979     0.031      0.922     0.979     0.949      0.995    Type3
Weighted Avg.    0.944     0.027      0.945     0.944     0.944      0.994

=== Confusion Matrix ===

  a  b  c   <-- classified as
 57  1  1 |  a = Type1
  4 64  3 |  b = Type2
  0  1 47 |  c = Type3
```

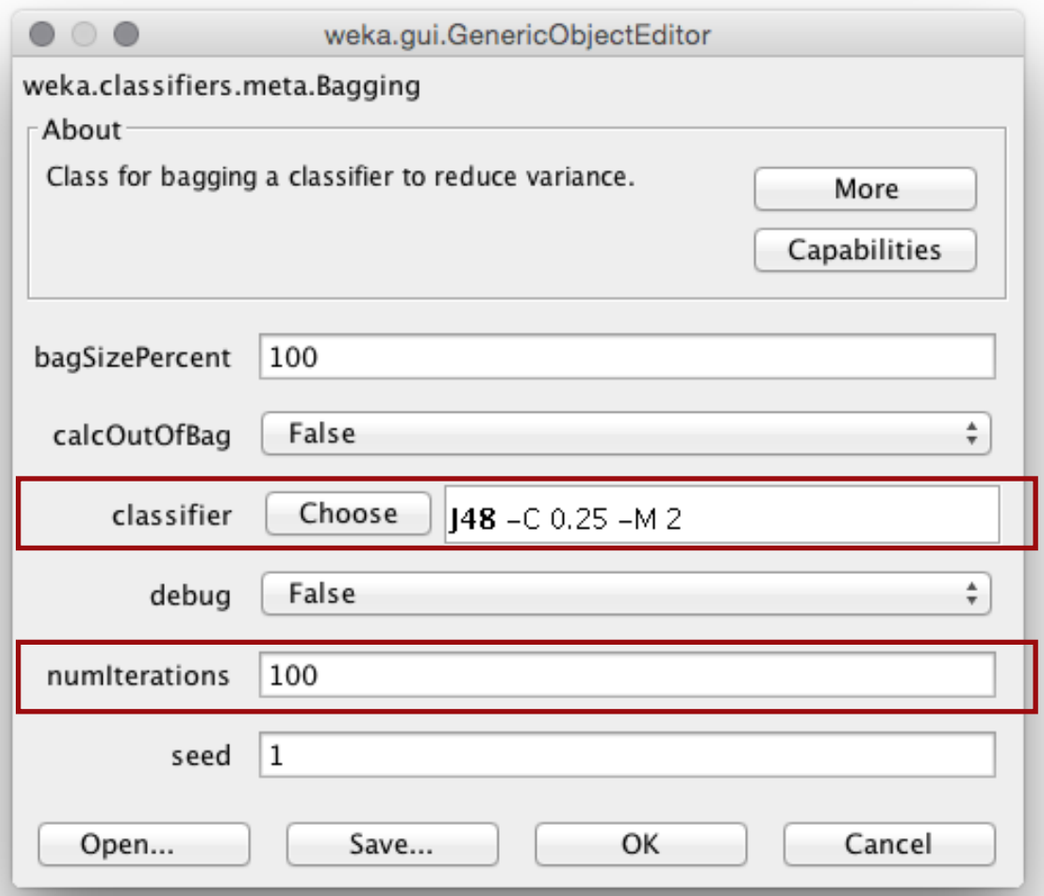
Status

OK

Improvement: 93.82% → 94.38%

Tutorial Q1(d)

Q. Repeat (c), but increase the ensemble size to 100, 200, then 300 classifiers. What level of improvement does this provide, in terms of percentage of instances correctly classified?

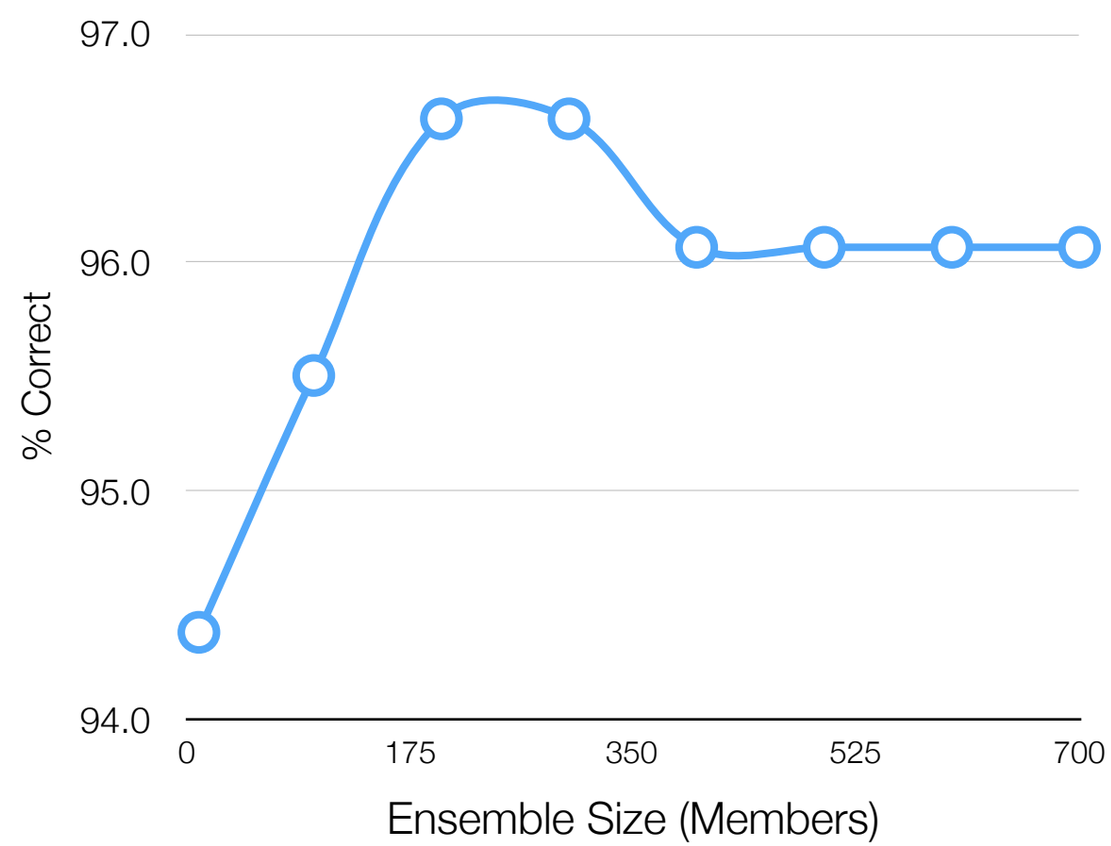


Ensemble Size	% Correct	% Incorrect
10	94.38	5.62
100	95.51	4.49
200	96.63	3.37
300	96.63	3.37

Tutorial Q1(e)

Q. Why does the level of improvement in accuracy often “level off” after an ensemble has been increased to a certain size?

Ensemble Size	% Correct	% Incorrect
10	94.38	5.62
100	95.51	4.49
200	96.63	3.37
300	96.63	3.37
400	96.07	3.93
500	96.07	3.93
600	96.07	3.93
700	96.07	3.93



➡ Eventually, new ensemble members will have prediction patterns collinear with existing members. No new diversity is added, so ensemble accuracy will plateau.

Tutorial Q2(a)

Q. Explain what differentiates the ensemble members in a *boosting* classifier ensemble.

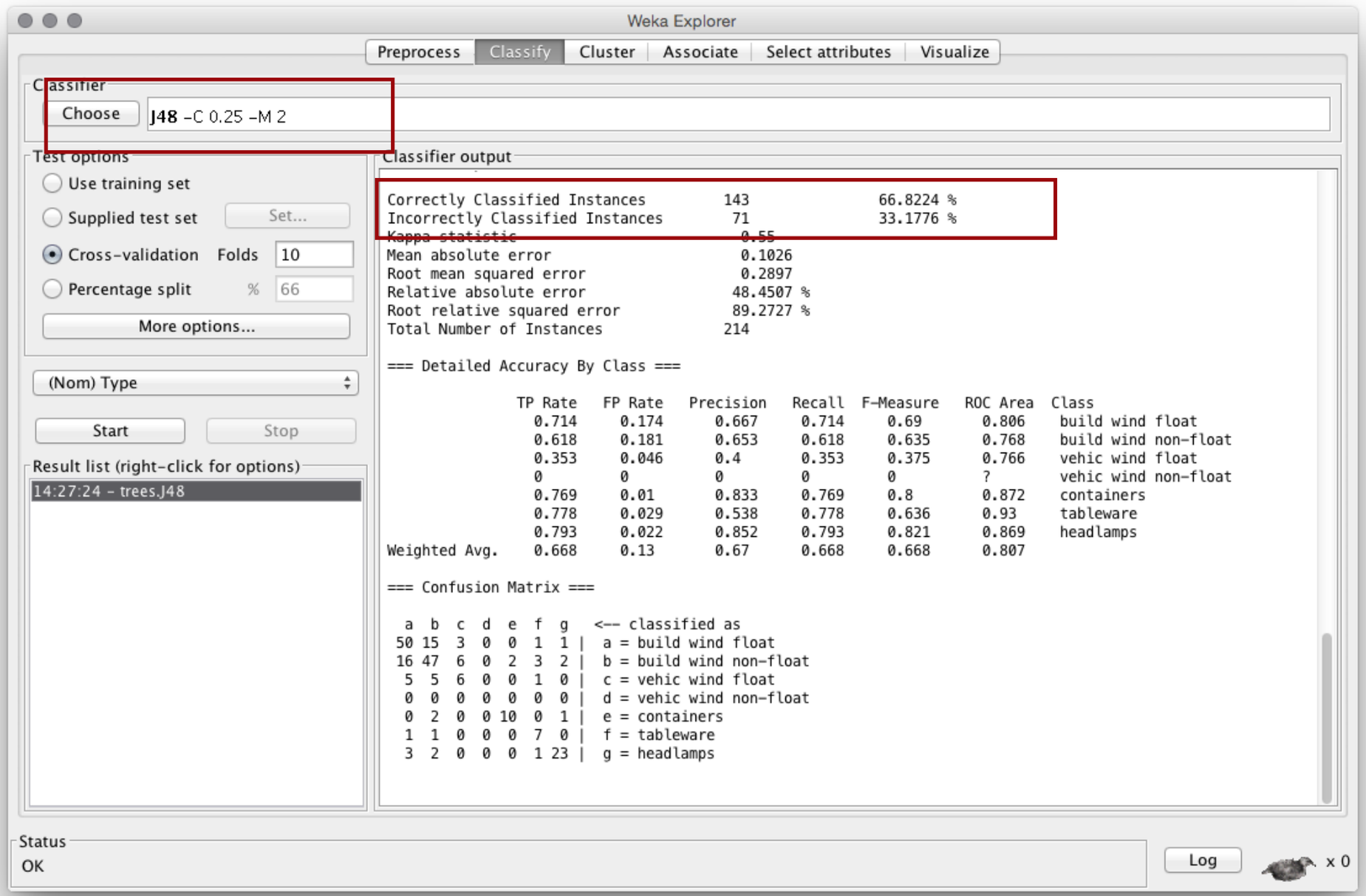
➡ *Boosting:* Train a series of classifiers such that later classifiers are trained to better predict on examples that earlier ones perform poorly on.

➡ Focus on previous errors when building next ensemble member by adjusting weights for all examples.

➡ This leads to both diversity (disagreement) between the base classifiers and increase in overall ensemble accuracy.

Tutorial Q2(b)

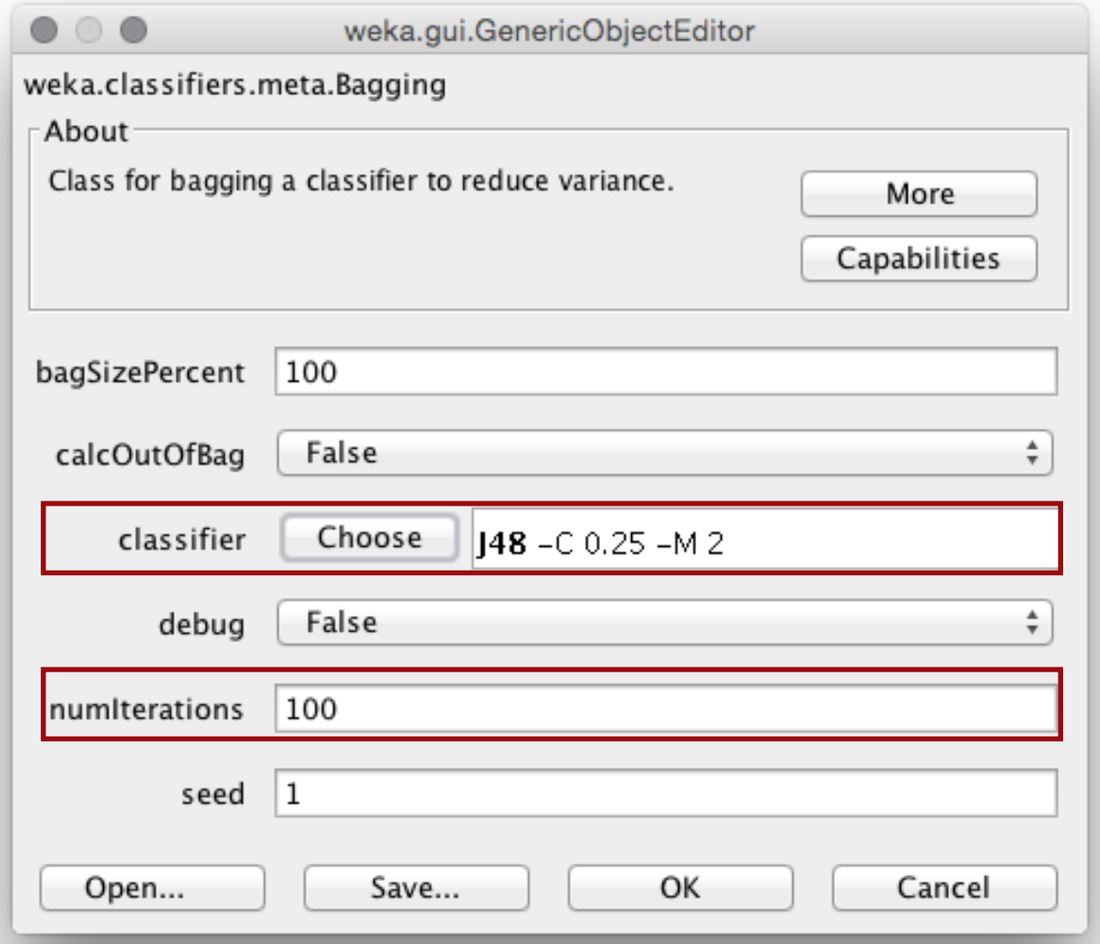
Q. In Weka, load the *Glass* data set using the ARFF file provided, and evaluate a decision tree classifier (J48) using 10-fold cross-validation. What percentage of instances are correctly classified?



Tutorial Q2(c)

Q. Apply bagging with a decision tree classifier for an ensemble size of 100. What is the improvement over a single tree?

1. Using Weka, click on the *Classify* tab.
2. Click *Choose*, select method *classifiers->meta->Bagging*.
3. Click *Bagging* in the box to the right. The configuration interface of the method appears.
4. Click *Choose*, select *J48*.
5. Set the *numIterations* to 100.
6. Click *OK* button.
7. Click *Start* button to build the ensemble.



Tutorial Q2(c)

Weka Explorer

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

Classifier

Choose: **Bagging** -P 100 -S 1 -I 100 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) Type

Start | Stop

Result list (right-click for options)

14:27:24 - trees.J48
14:30:05 - meta.Bagging

Classifier output

```
Correctly Classified Instances         161               75.2336 %
Incorrectly Classified Instances        53               24.7664 %
Kappa statistic                          0.6605
Mean absolute error                      0.1034
Root mean squared error                  0.225
Relative absolute error                 48.8089 %
Root relative squared error             69.3144 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.843     0.146      0.738     0.843     0.787      0.919    build wind float
                 0.724     0.13       0.753     0.724     0.738      0.884    build wind non-float
                 0.294     0.015      0.625     0.294     0.4        0.879    vehic wind float
                 0         0          0         0         0          ?        vehic wind non-float
                 0.692     0.01       0.818     0.692     0.75       0.944    containers
                 0.889     0.024      0.615     0.889     0.727      0.993    tableware
                 0.862     0.022      0.862     0.862     0.862      0.944    headlamps
Weighted Avg.    0.752     0.1        0.751     0.752     0.744      0.912

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 59  8  1  0  0  1  1 |  a = build wind float
 13 55  2  0  2  2  2 |  b = build wind non-float
  6  5  5  0  0  1  0 |  c = vehic wind float
  0  0  0  0  0  0  0 |  d = vehic wind non-float
  0  3  0  0  9  0  1 |  e = containers
  1  0  0  0  0  8  0 |  f = tableware
  1  2  0  0  0  1 25 |  g = headlamps
```

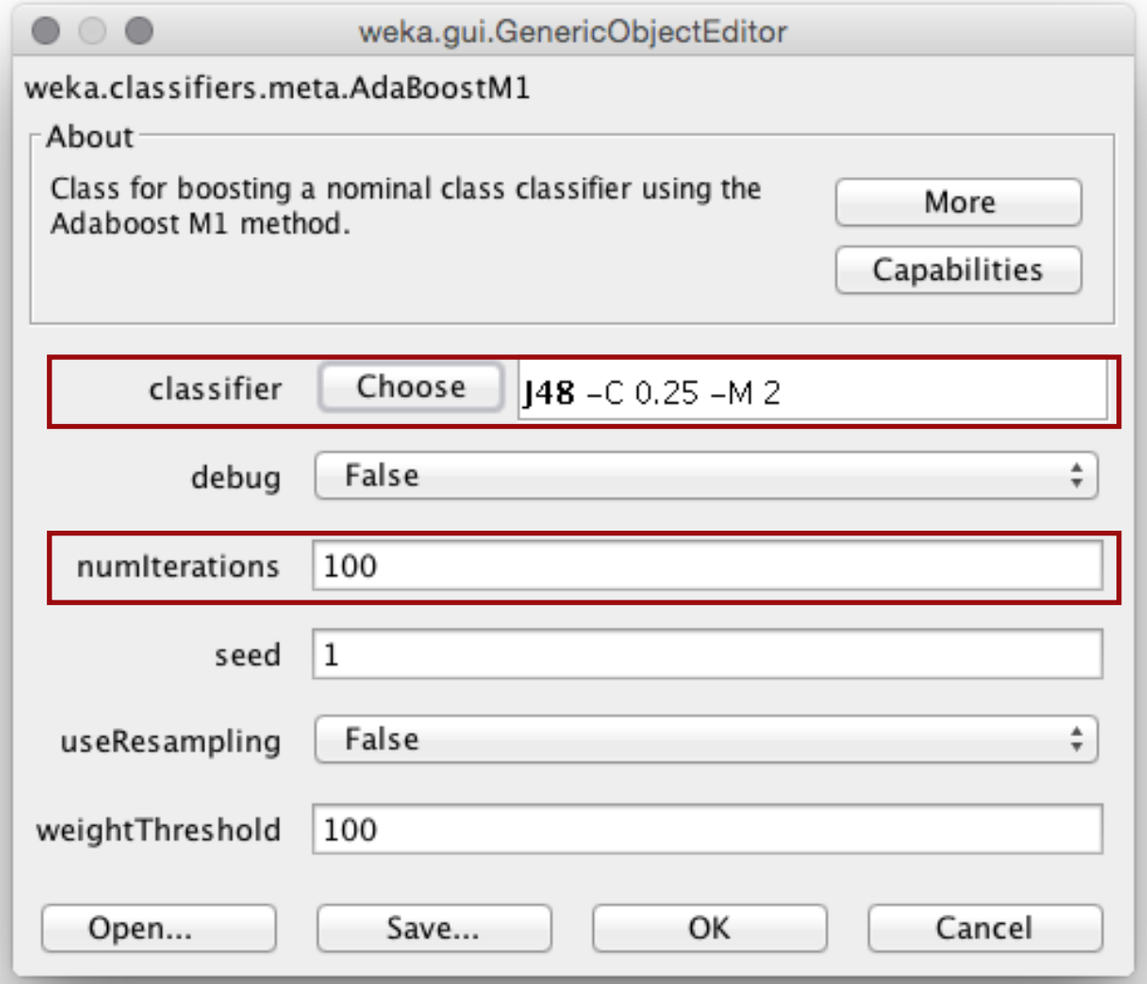
Status: OK

Improvement: 66.82% → 75.23%

Tutorial Q2(d)

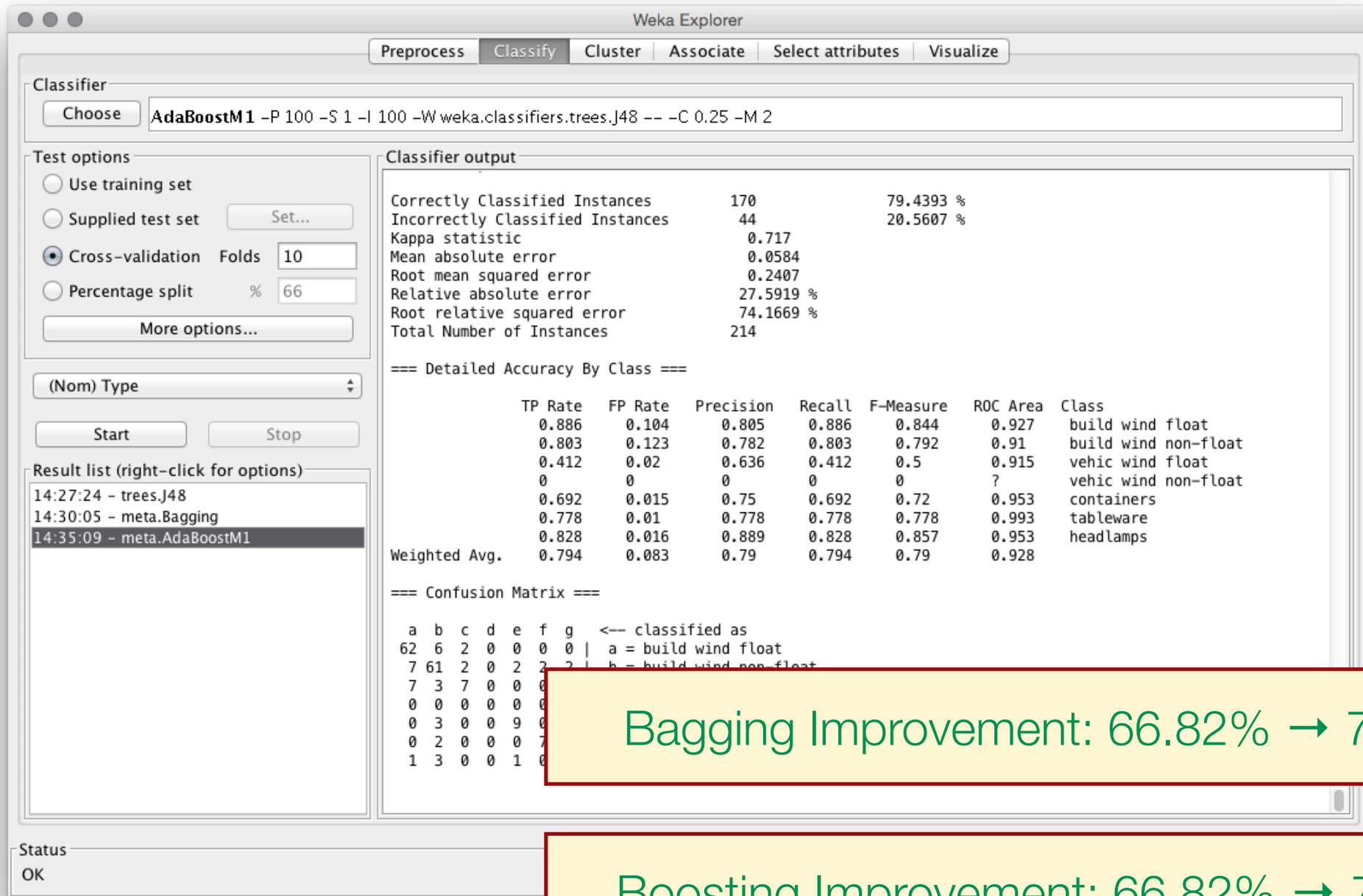
Q. Now apply *boosting* with a decision tree classifier for an ensemble size of 100. How does it compare to the results from (c)? How do you explain this difference?

1. Using Weka, click on the *Classify* tab.
2. Click *Choose*, select method *classifiers->meta->AdaBoostM1*.
3. Click *AdaBoostM1* in the box to the right. The configuration interface of the method appears.
4. Click *Choose*, select *J48*.
5. Set the *numIterations* to 100.
6. Click *OK* button.
7. Click *Start* button to build the ensemble.



Tutorial Q2(d)

- Boosting adds diversity to the ensemble, while also improving accuracy on the difficult examples.



Bagging Improvement: 66.82% → 75.23%

Boosting Improvement: 66.82% → 79.44%

Tutorial Q3(a)

Q. Applying bagging to a “stable” classifier is generally not a good idea. Why is this?

- If a classification algorithm is “unstable”, small changes in the training set lead to larger changes in the model built on the data.

  ➡Ensemble is more diverse.

  - e.g. bagging on decision trees, neural networks.

- If a classification algorithm is “stable”, small changes in the training set only lead to small changes in the model.

  ➡Little diversity in the ensemble, so little improvement in accuracy.

  - e.g. bagging on k-NN classifiers

Tutorial Q3(b)

Q. Explain how diversity is generated using a *random subspacing* classifier ensemble.

1. A subset of features is randomly selected without replacement.
2. Train a classifier using only selected features to represent training data.
3. Combine outputs of many classifiers trained on different subsets.

Original data set (5 features)

X	Col	H	W	Wgt	HW
1	210	60	62	186	0.97
2	220	70	53	180	1.32
3	215	55	50	152	1.10
4	180	76	40	152	1.90
...	...	...	...	...	...
10	190	60	58	175	1.03

Subset 1 (3 features)

X	H	W	HW
1	60	62	0.97
2	70	53	1.32
3	55	50	1.10
4	76	40	1.90
...	...	...	...
10	60	58	1.03

Classifier → Model 1

Subset 2 (3 features)

X	Col	H	HW
1	210	60	3.50
2	220	70	3.14
3	215	55	3.91
4	180	76	2.37
...	...	...	...
10	190	60	3.17

Classifier → Model 2

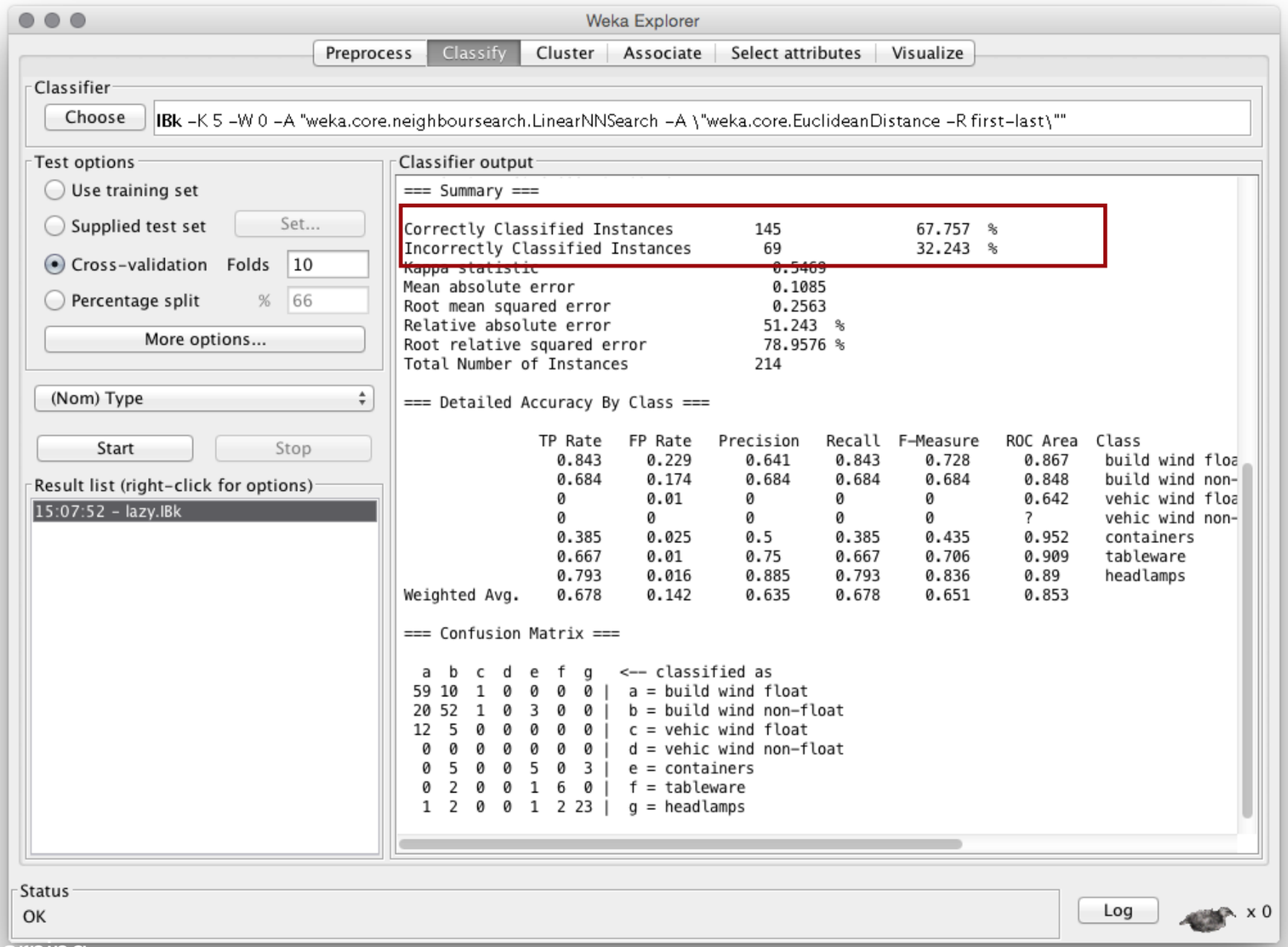
Subset 3 (3 features)

X	Col	H	W
1	210	60	62
2	220	70	53
3	215	55	50
4	180	76	40
...	...	...	...
10	190	60	58

Classifier → Model 3

Tutorial Q3(c)

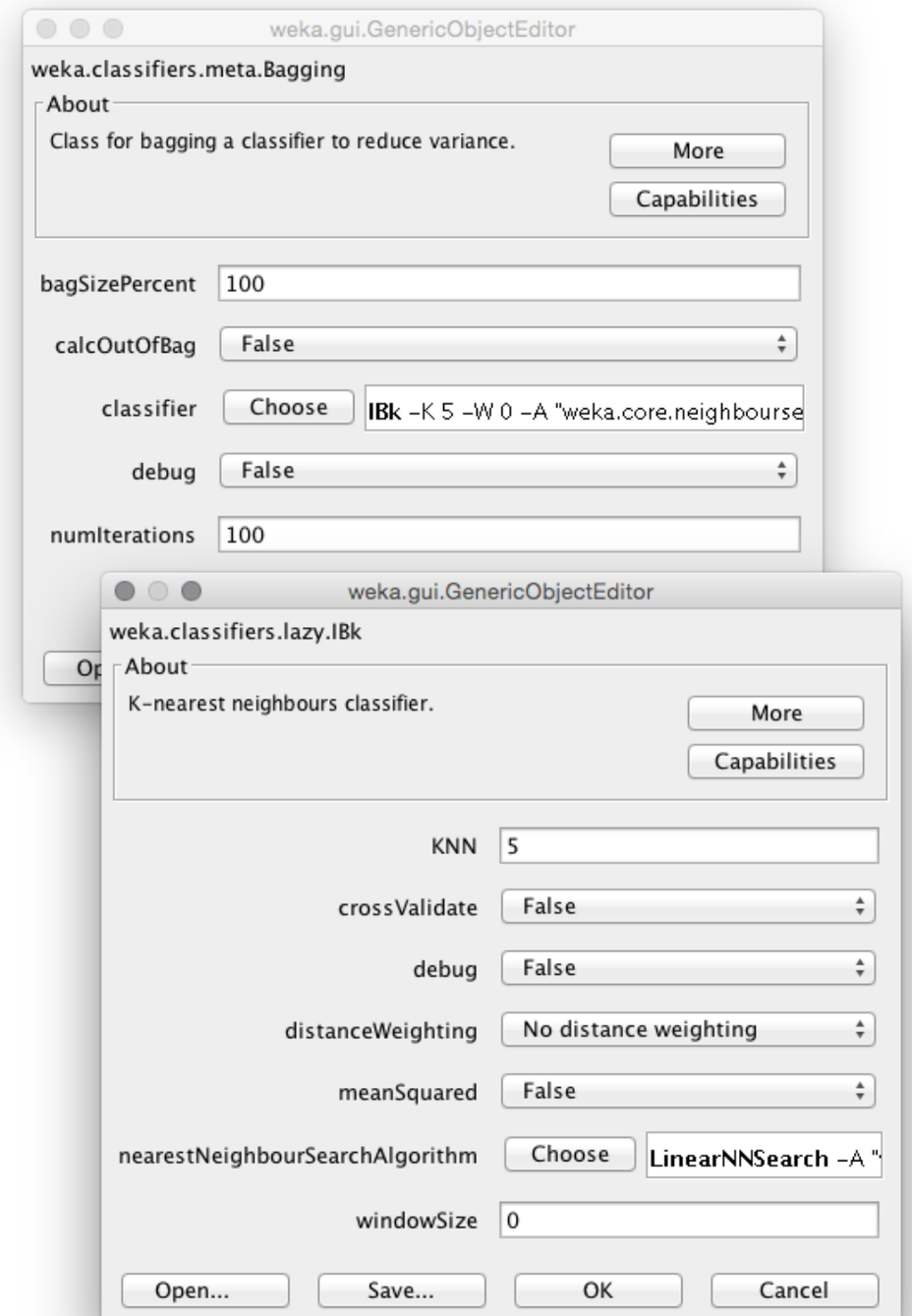
Q. In Weka, load the *Glass* data set. Evaluate a k-NN classifier with *k*=5 neighbours using 10-fold cross-validation. What percentage of instances are correctly classified?



Tutorial Q3(d)

Q. Apply bagging with with a k-NN classifier (*k*=5) for an ensemble size of 100. What is the improvement?

1. Using Weka, click on the *Classify* tab.
2. Click *Choose*, select method *classifiers->meta->Bagging*.
3. Click *Bagging* in the box to the right. The configuration interface of the method appears.
4. Click *Choose*, select *IBk*.
5. Click *IBk*. In the configuration window, set *KNN* to 5. Close window.
6. Set the *numIterations* to 100.
7. Click *OK* button.
8. Click *Start* button to build the ensemble.



Tutorial Q3(d)

Weka Explorer

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

Classifier

Choose: **Bagging** -P 100 -S 1 -I 100 -W weka.classifiers.lazy.IBk -- -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

(Nom) Type

Start | Stop

Result list (right-click for options)

- 15:07:52 - lazy.IBk
- 15:12:16 - meta.Bagging

Classifier output

```
Correctly Classified Instances         146               68.2243 %
Incorrectly Classified Instances        68               31.7757 %
Kappa statistic                          0.5563
Mean absolute error                      0.111
Root mean squared error                  0.2478
Relative absolute error                 52.3983 %
Root relative squared error             76.3409 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.814     0.194      0.671     0.814     0.735      0.89     build wind floa
                 0.684     0.196      0.658     0.684     0.671      0.874    build wind non-
                 0.118     0.005      0.667     0.118     0.2        0.799    vehic wind floa
                 0         0          0         0         0          ?        vehic wind non-
                 0.462     0.035      0.462     0.462     0.462      0.913    containers
                 0.667     0.01       0.75      0.667     0.706      0.981    tableware
                 0.793     0.016      0.885     0.793     0.836      0.951    headlamps
Weighted Avg.    0.682     0.138      0.686     0.682     0.666      0.891

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 57 12  1  0  0  0  0 |  a = build wind float
 19 52  0  0  5  0  0 |  b = build wind non-float
  8  7  2  0  0  0  0 |  c = vehic wind float
  0  0  0  0  0  0  0 |  d = vehic wind non-float
  0  4  0  0  6  0  3 |  e = containers
  0  2  0  0  1  6  0 |  f = tableware
  1  2  0  0  1  2 23 |  g = headlamps
```

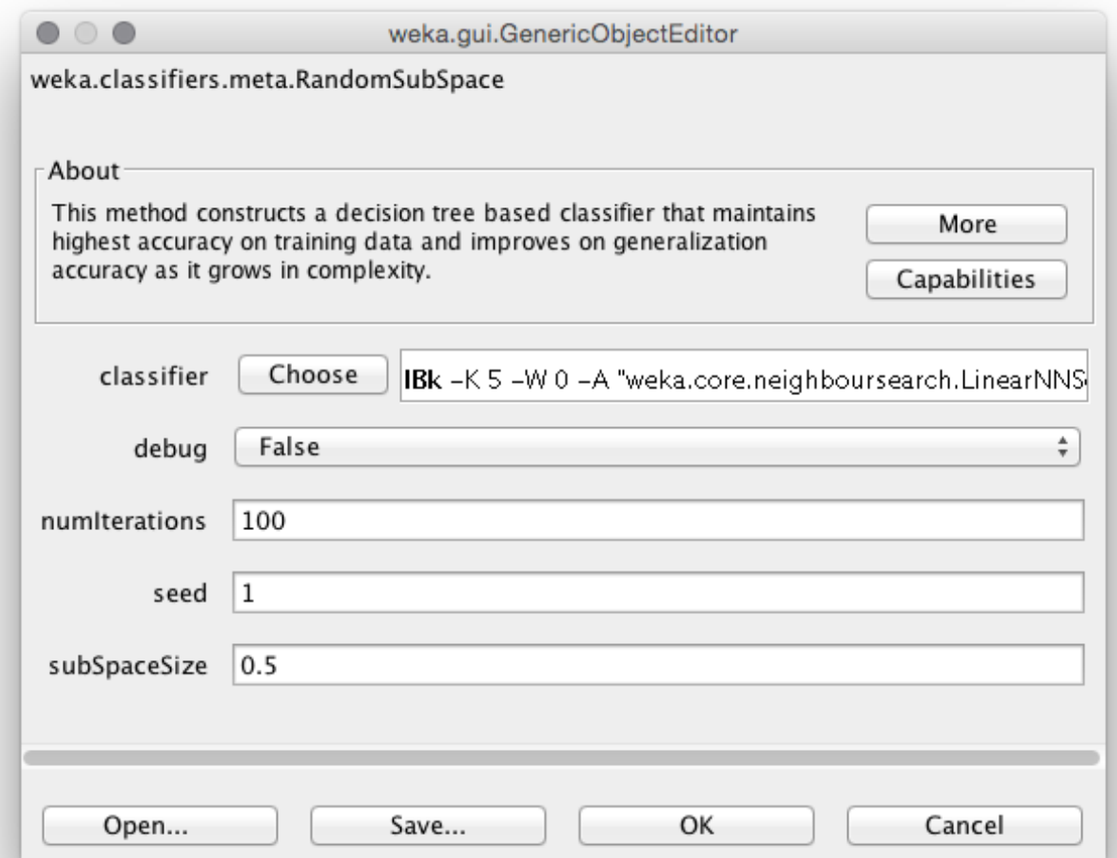
Status: OK

Improvement: 67.76% → 68.22%

Tutorial Q3(d)

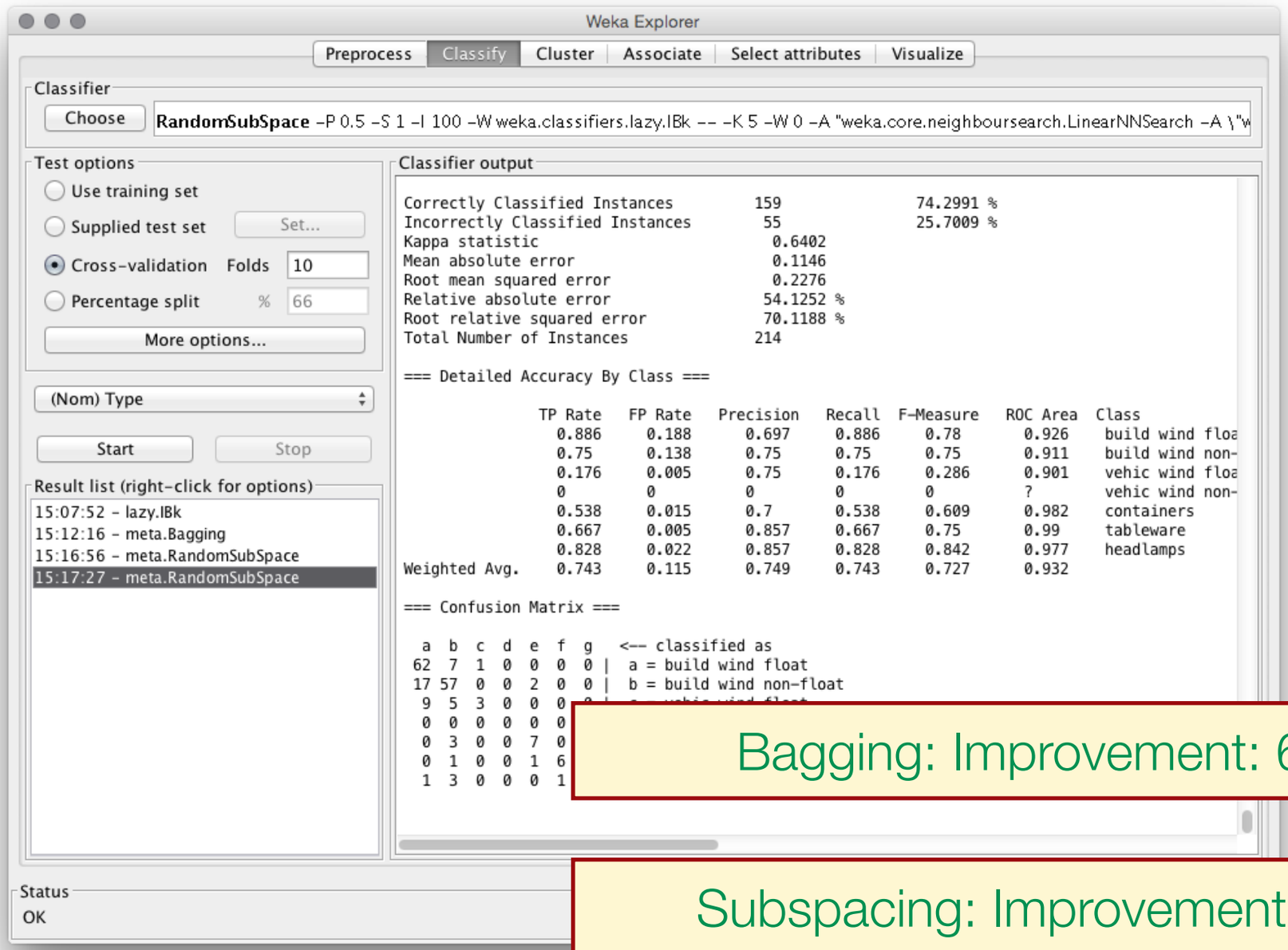
Q. Now apply *random subspacing* with a k-NN classifier (*k*=5) for an ensemble size of 100. How does it compare to the results from (d)? How do you explain this difference?

1. Using Weka, click on the *Classify* tab.
2. Click *Choose*, select method *classifiers->meta->RandomSubSpace*.
3. Click *RandomSubSpace* in the box to the right. The configuration interface of the method appears.
4. Click *Choose*, select *IBk*.
5. Click *IBk*. In the configuration window, set *KNN* to 5. Close window.
6. Set the *numIterations* to 100.
7. Click *OK* button.
8. Click *Start* button to build the ensemble.



Tutorial Q3(d)

- Bagging is ineffective using a stable classifier (k-NN).
- Random subspacing adds more instability into the classifier (diversity) ⇒ different features used when calculating distances.



Bagging: Improvement: 67.76% → 68.22%

Subspacing: Improvement: 67.76% → 74.23%