COMP47490 Tutorial

Ensembles

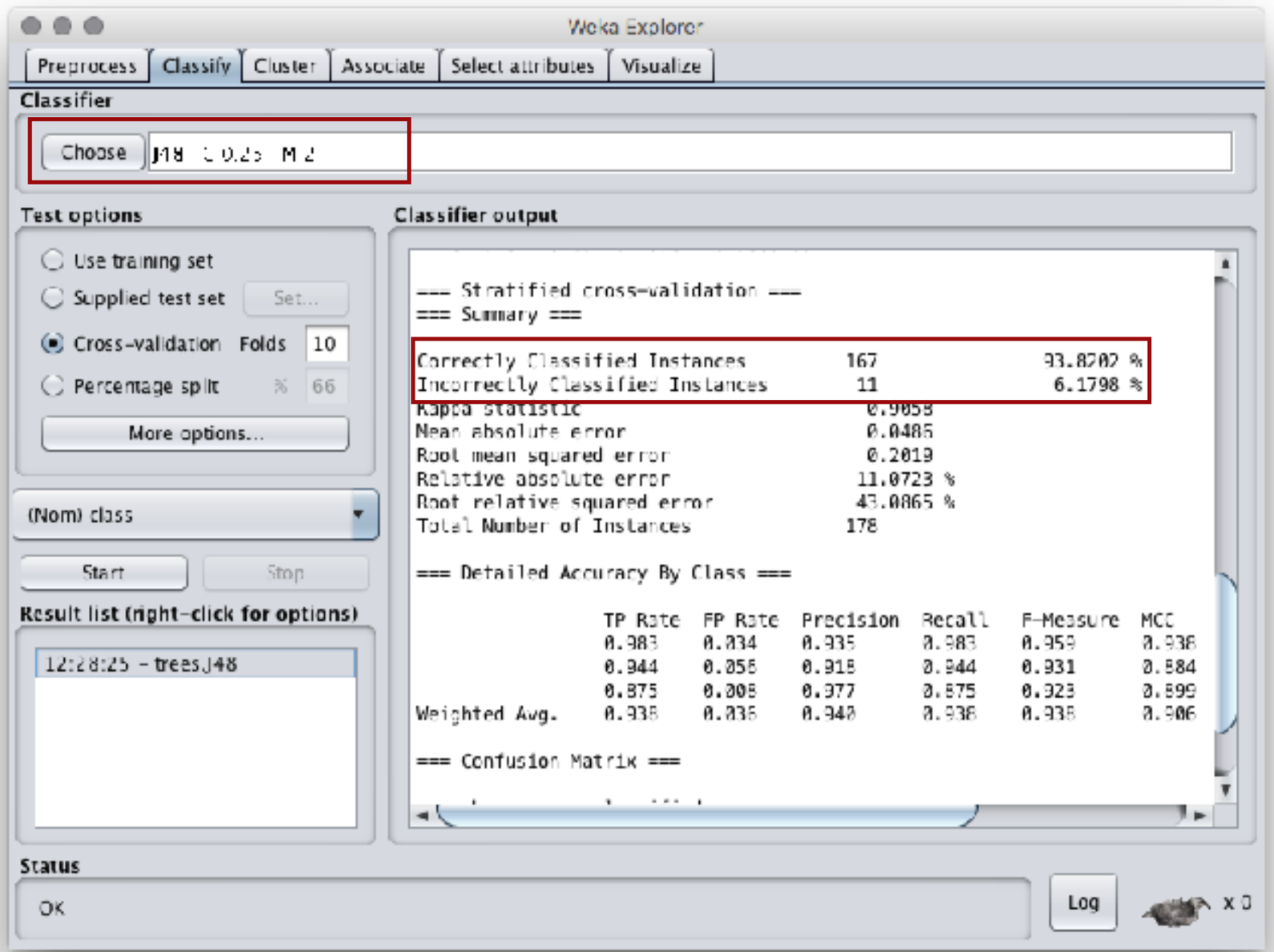
Aonghus Lawlor

School of Computer Science

Autumn 2019

Tutorial Q1(a)

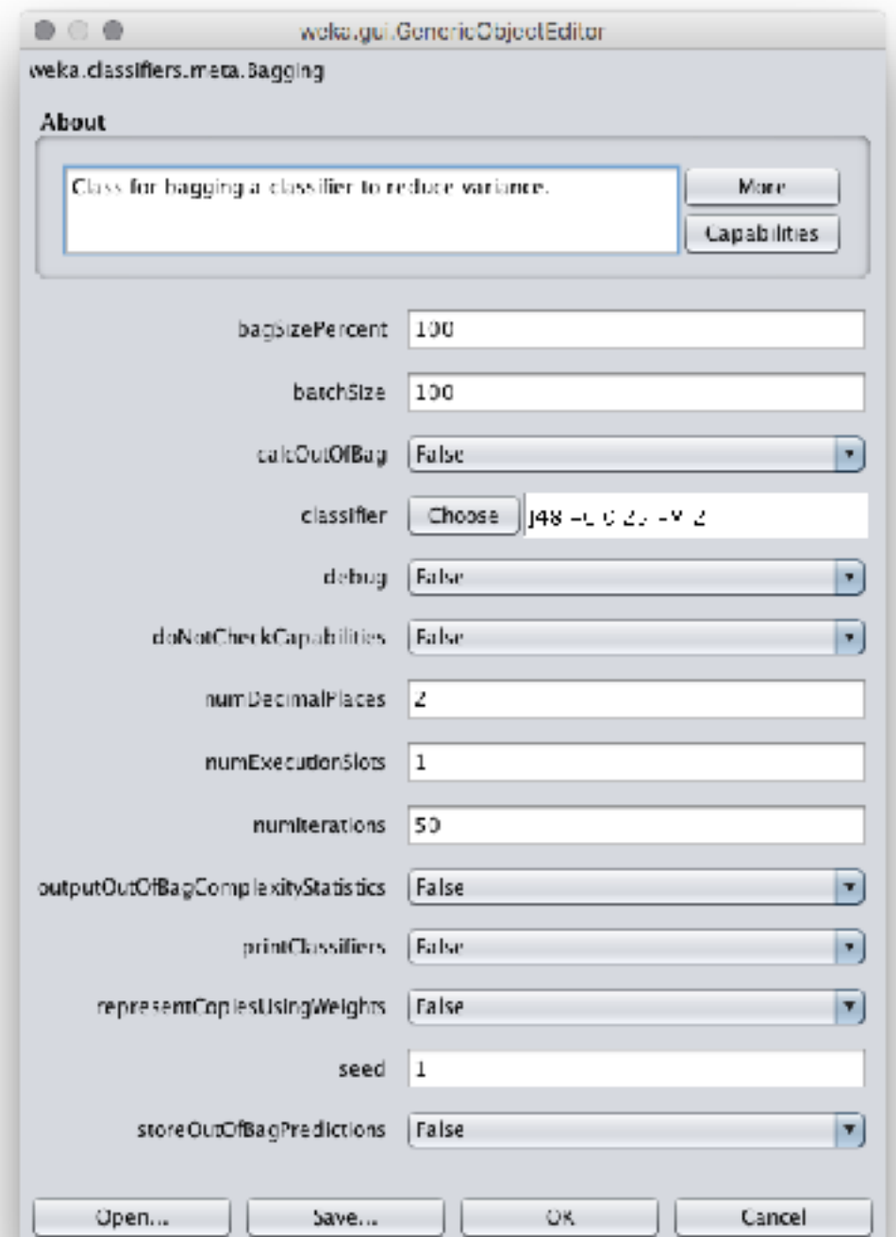
Q. In Weka, load the *Wine* data set using the ARFF file provided, and evaluate a decision tree classifier (J48) using 10-fold cross-validation. What percentage of instances are correctly classified?



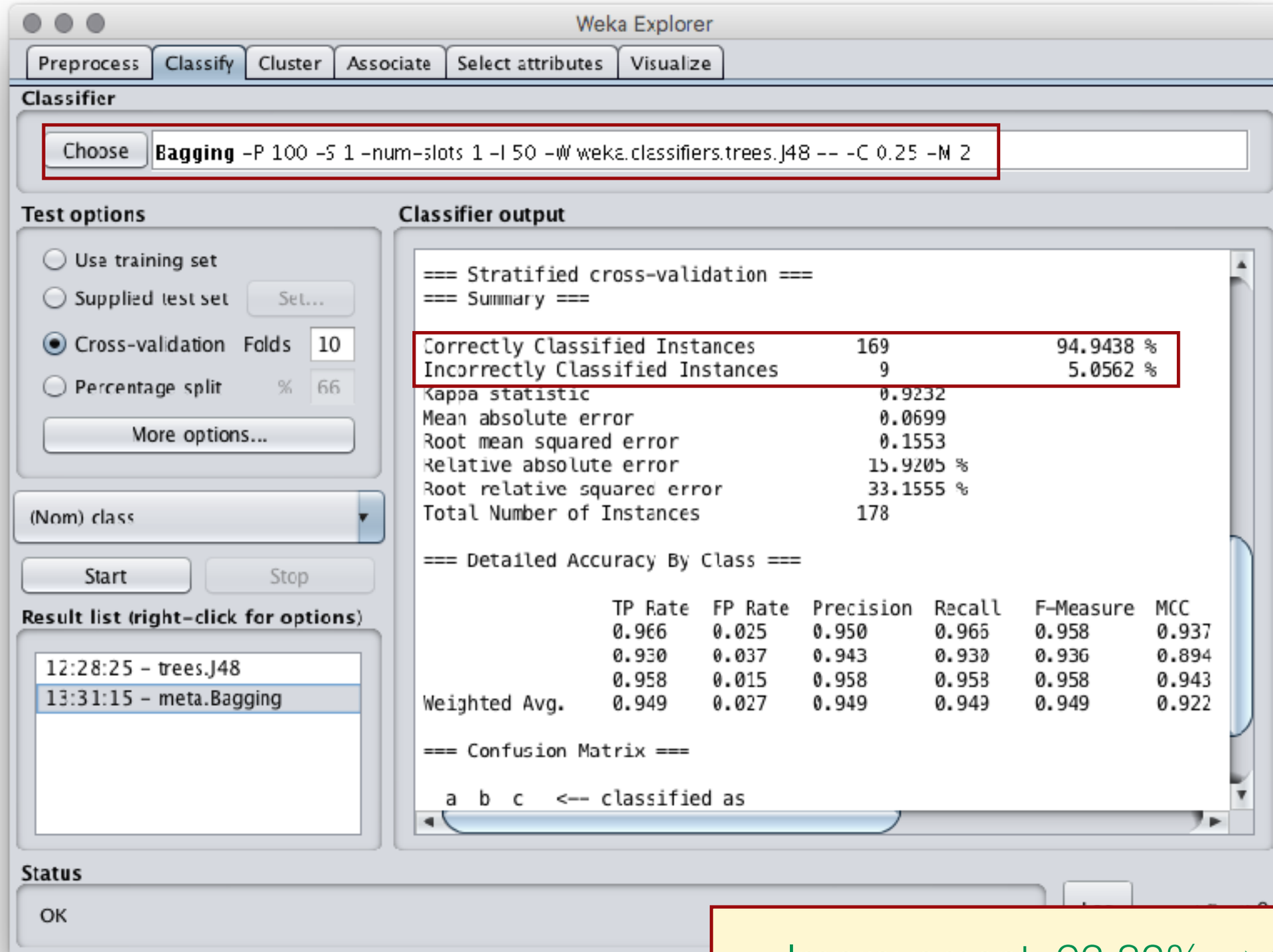
Tutorial Q1(b)

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 50?

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 50 Click *OK* button.
6. Click *Start* button to build the ensemble.



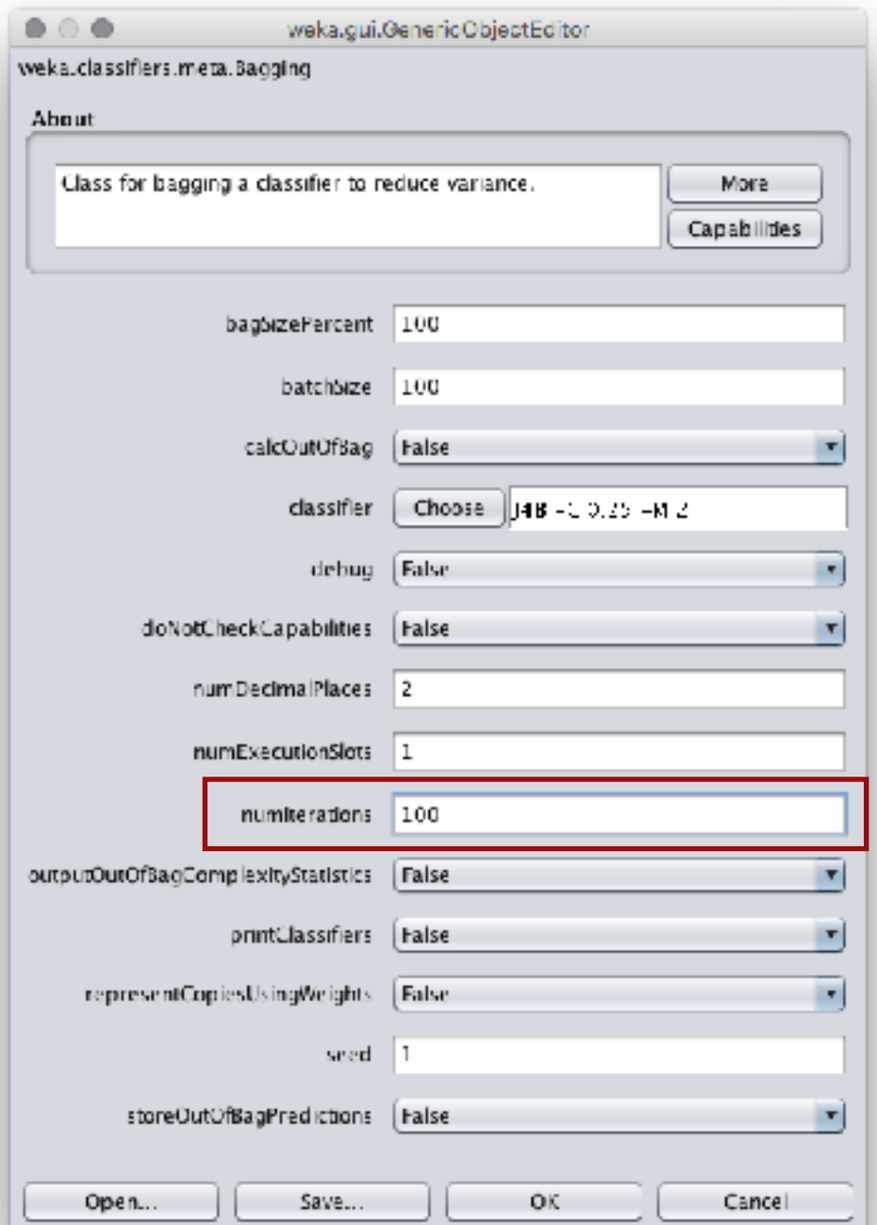
Tutorial Q1(b)



Improvement: 93.82% → 94.94%

Tutorial Q1(c)

Q. Repeat (b), but increase the ensemble size to 100, 250, then 500 classifiers. What level of improvement does this provide, in terms of percentage of instances correctly classified?

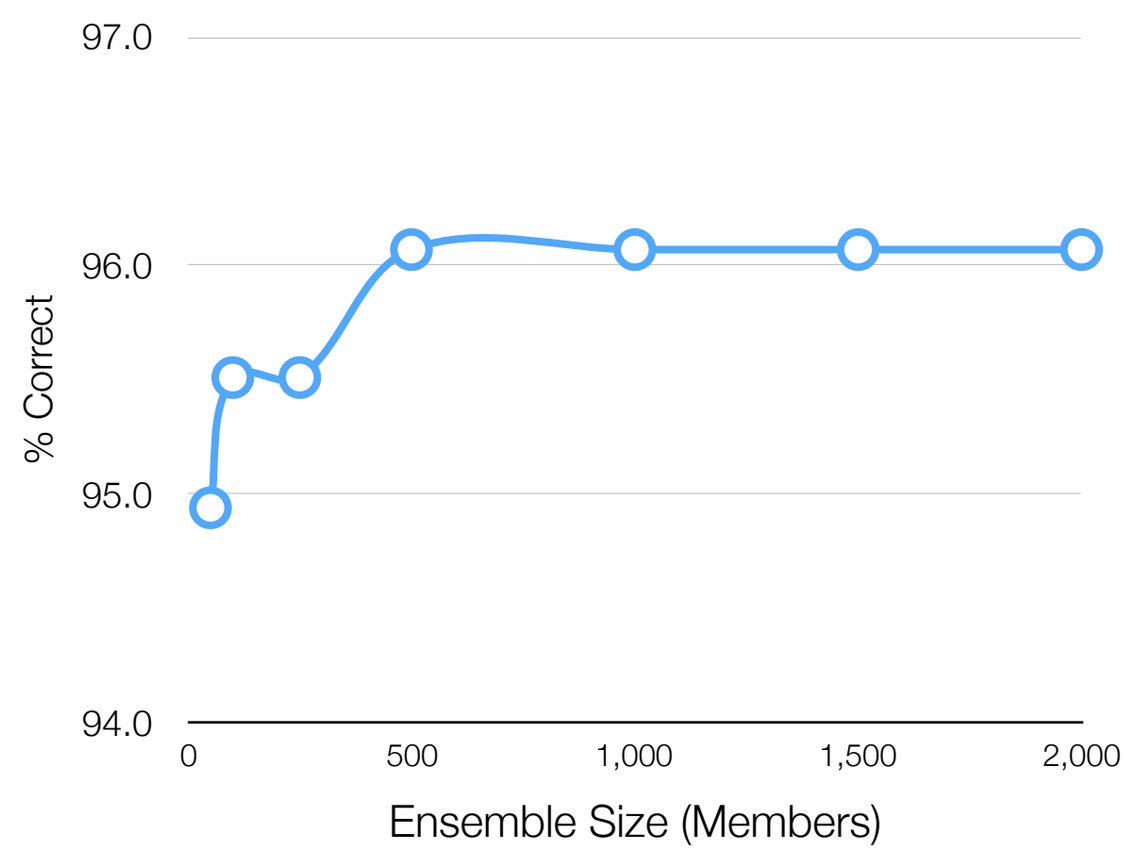


Ensemble Size	% Correct	% Incorrect
50	94.94	5.06
100	95.51	4.49
250	95.51	4.49
500	96.07	3.93

Tutorial Q1(d)

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

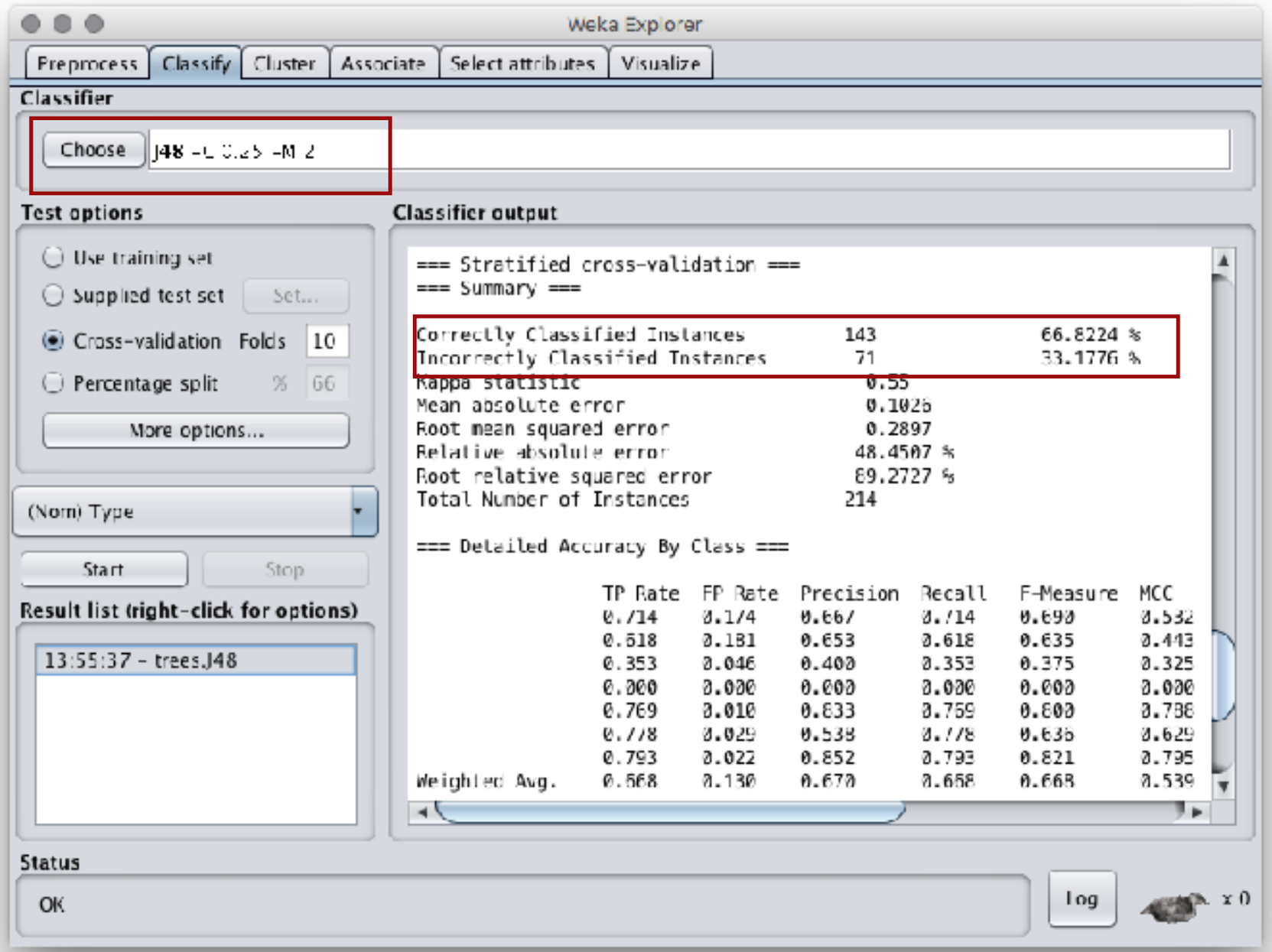
Ensemble Size	% Correct	% Incorrect
50	94.94	5.06
100	95.51	4.49
250	95.51	4.49
500	96.07	3.93
1000	96.07	3.93
1500	96.07	3.93
2000	96.07	3.93



➡ 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)

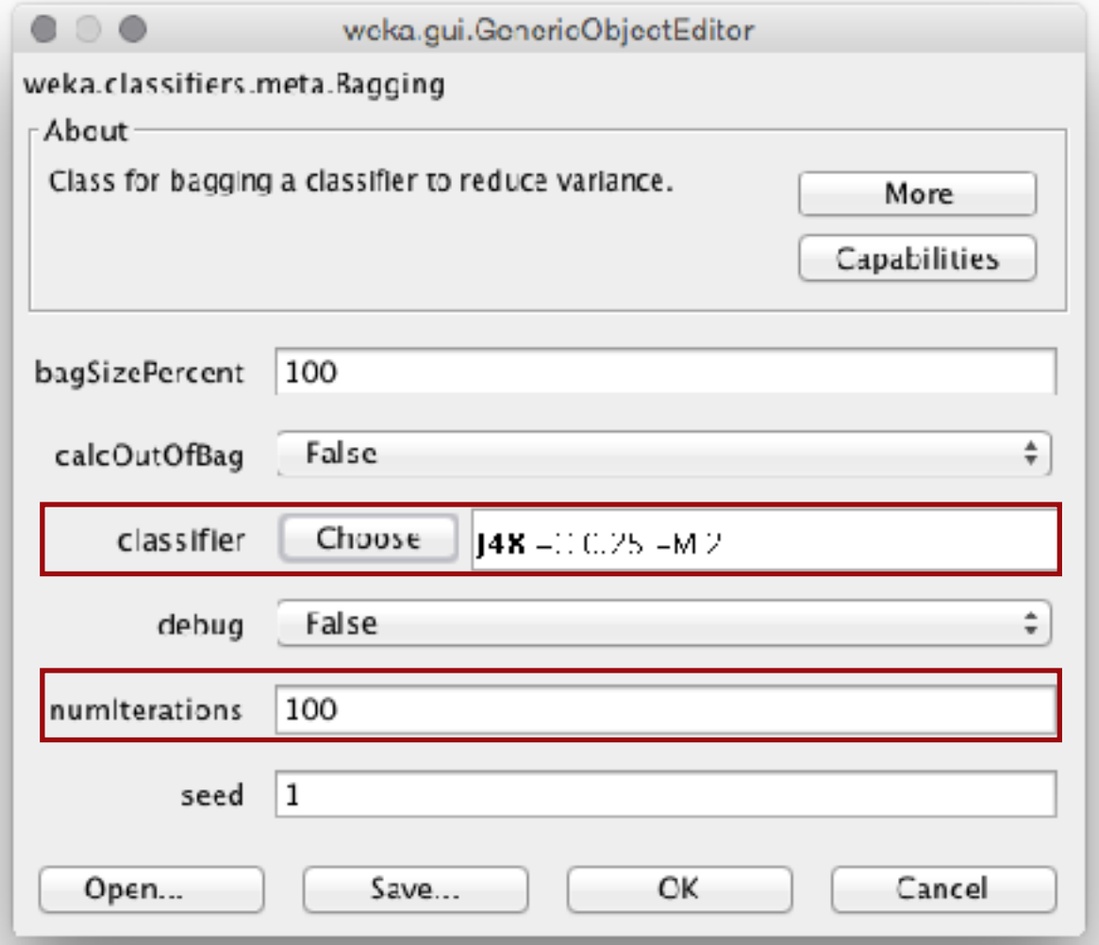
Q. In Weka, load the *Glass* data set using the ARFF file provided, and evaluate a decision tree classifier (J48) using 10-fold cross-validation. What percentage of instances are correctly classified?



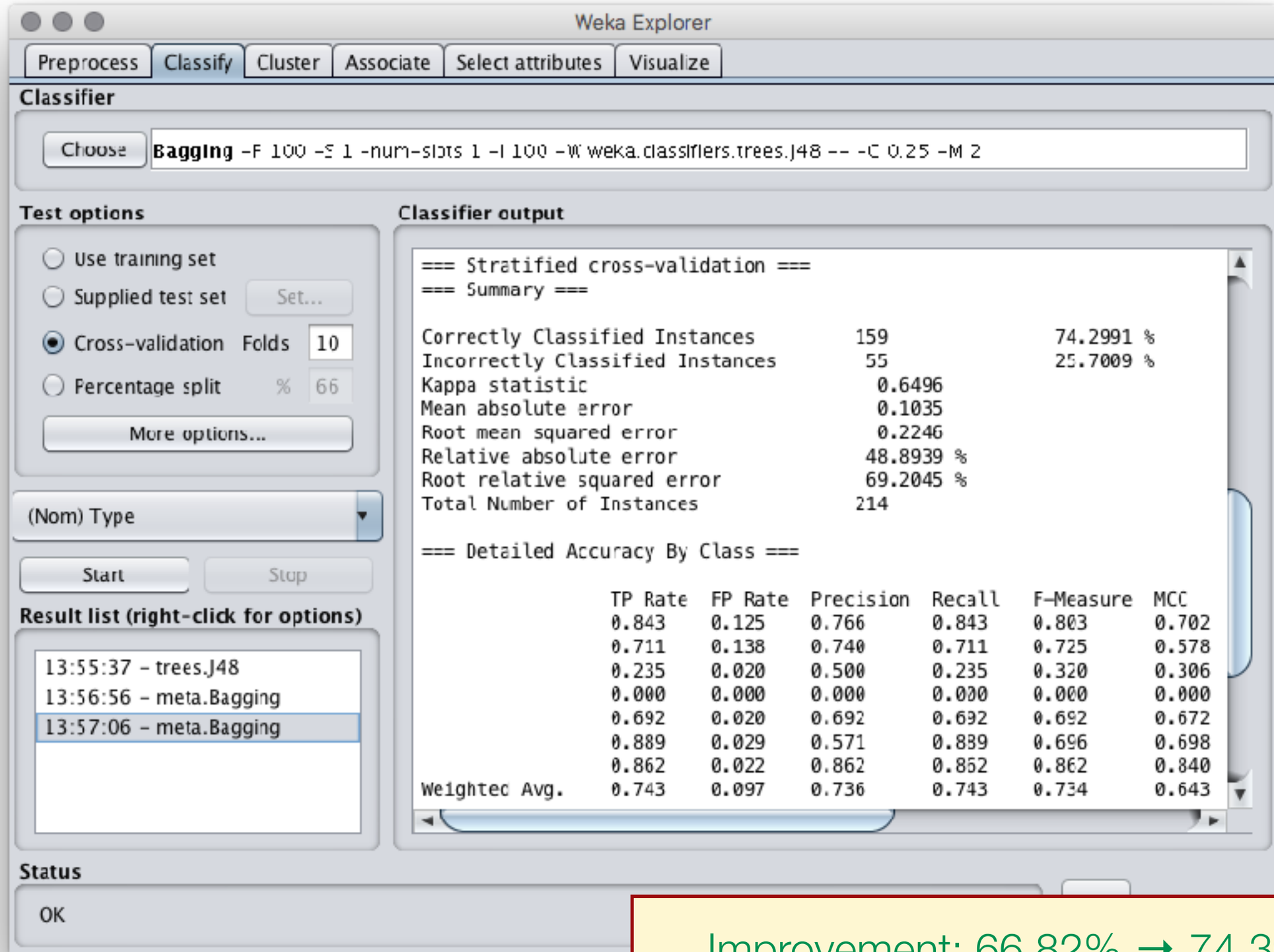
Tutorial Q2(b)

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(b)

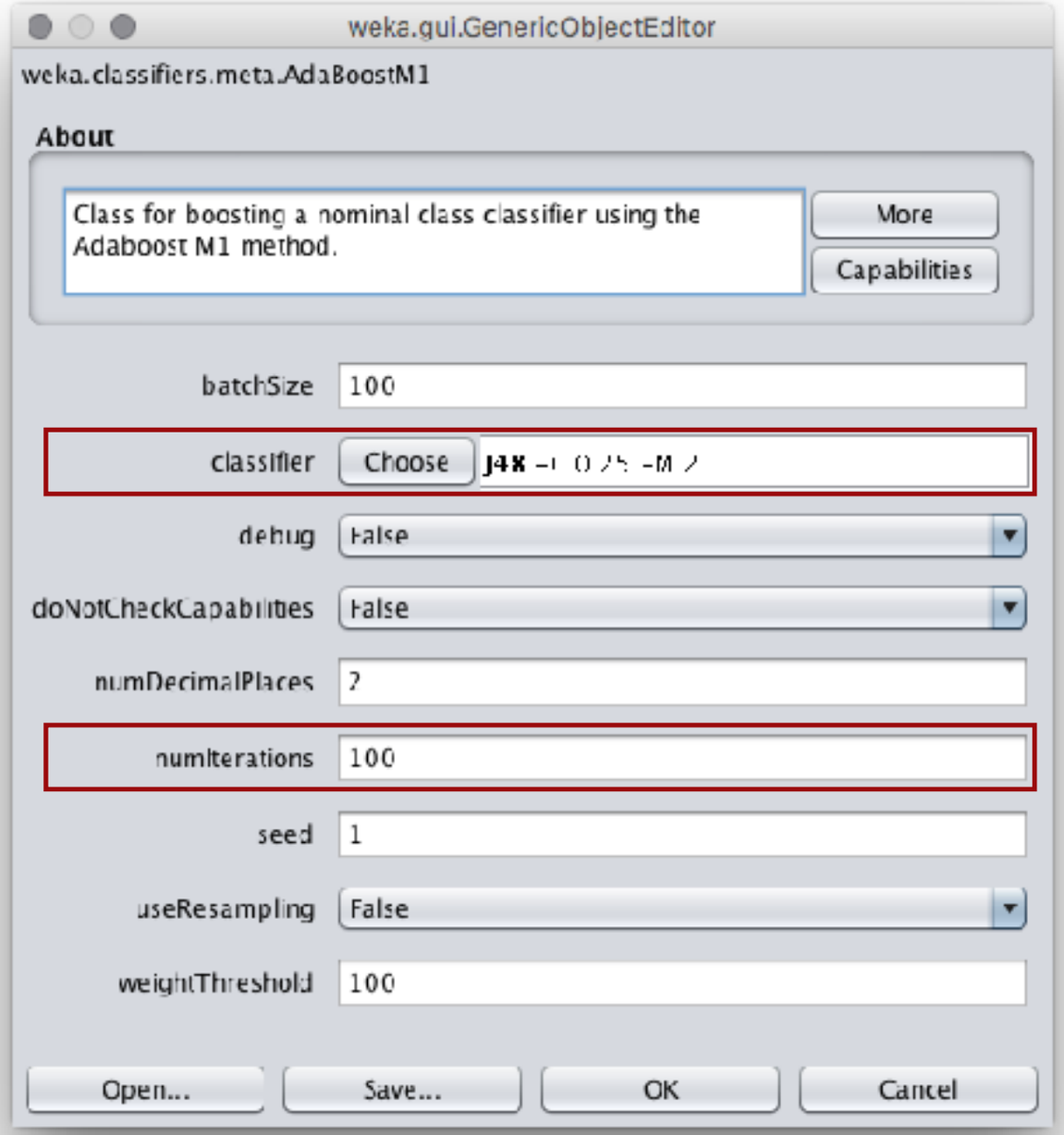


Improvement: 66.82% → 74.30%

Tutorial Q2(c)

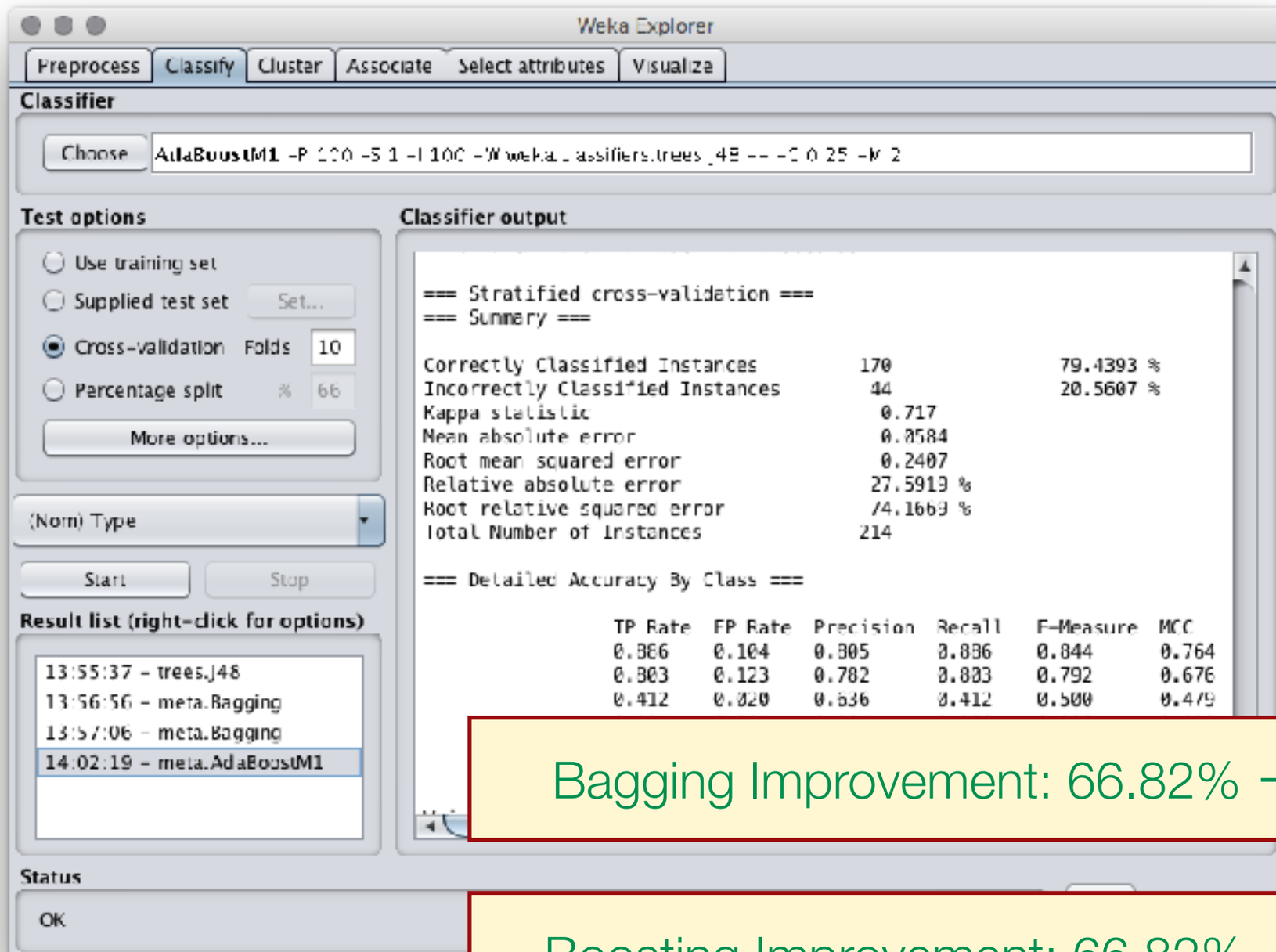
Q. Now apply *boosting* with a decision tree classifier for an ensemble size of 100. How does it compare to the results from (b)? How do you explain this difference?

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(c)

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

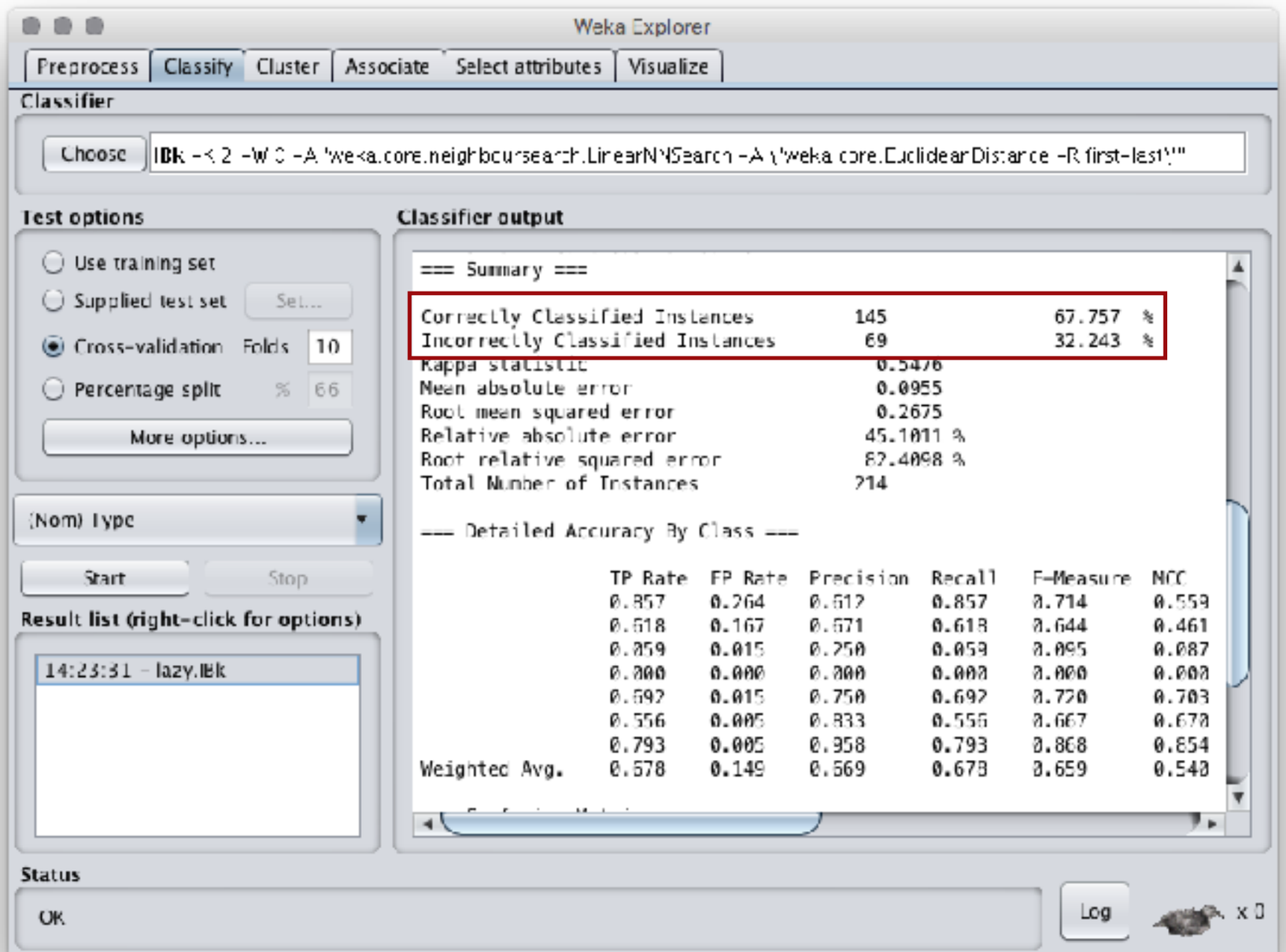


Bagging Improvement: 66.82% → 74.30%

Boosting Improvement: 66.82% → 79.44%

Tutorial Q3(a)

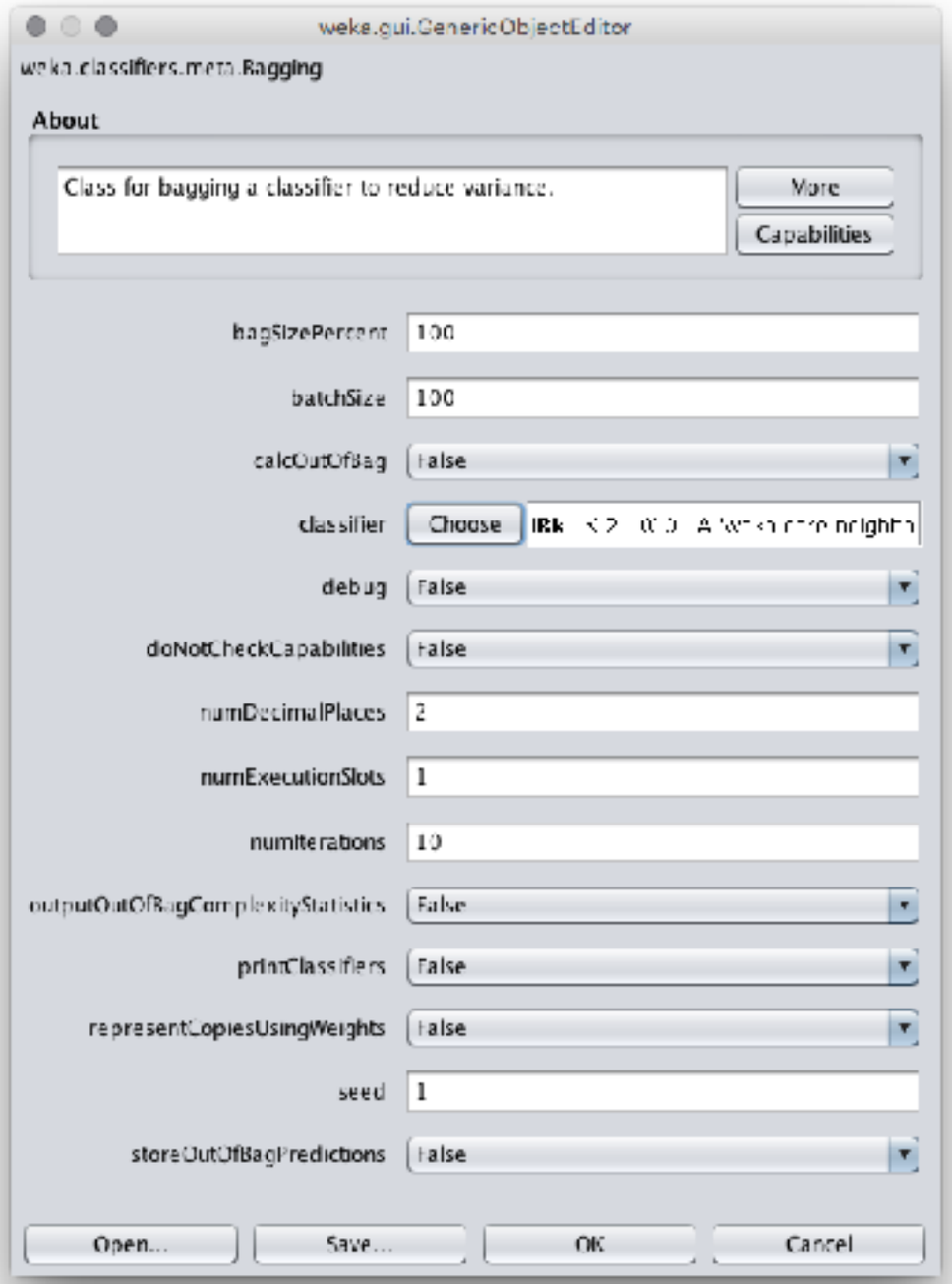
Q. In Weka, load the *Glass* data set. Evaluate a k-NN classifier with *k=2* neighbours using 10-fold cross-validation. What percentage of instances are correctly classified?



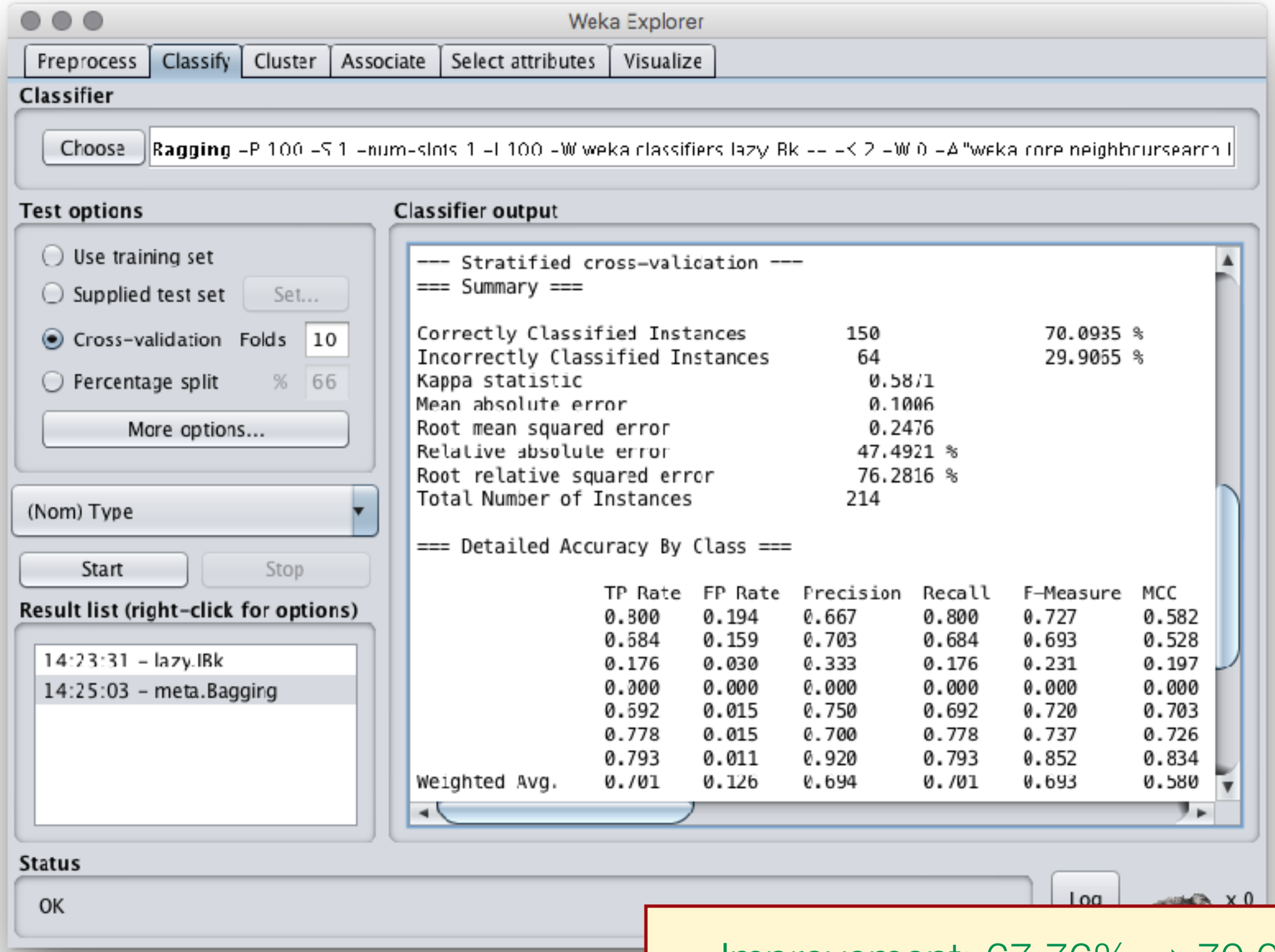
Tutorial Q3(b)

Q. Apply bagging with with a k-NN classifier (*k*=2) 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 2. Close window.
6. Set the *numIterations* to 100.
7. Click *OK* button.
8. Click *Start* button to build the ensemble.



Tutorial Q3(b)

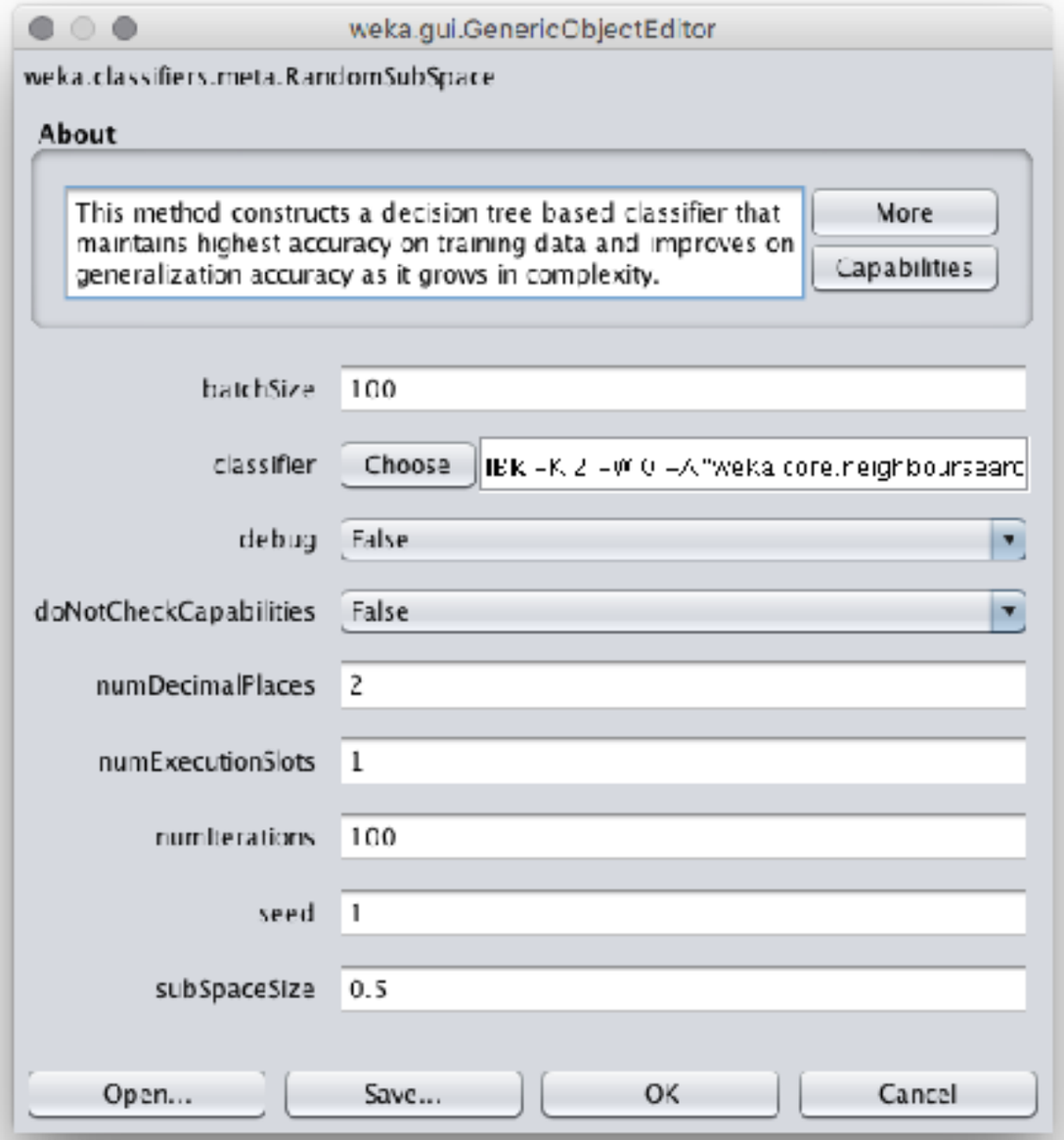


Improvement: 67.76% → 70.09%

Tutorial Q3(c)

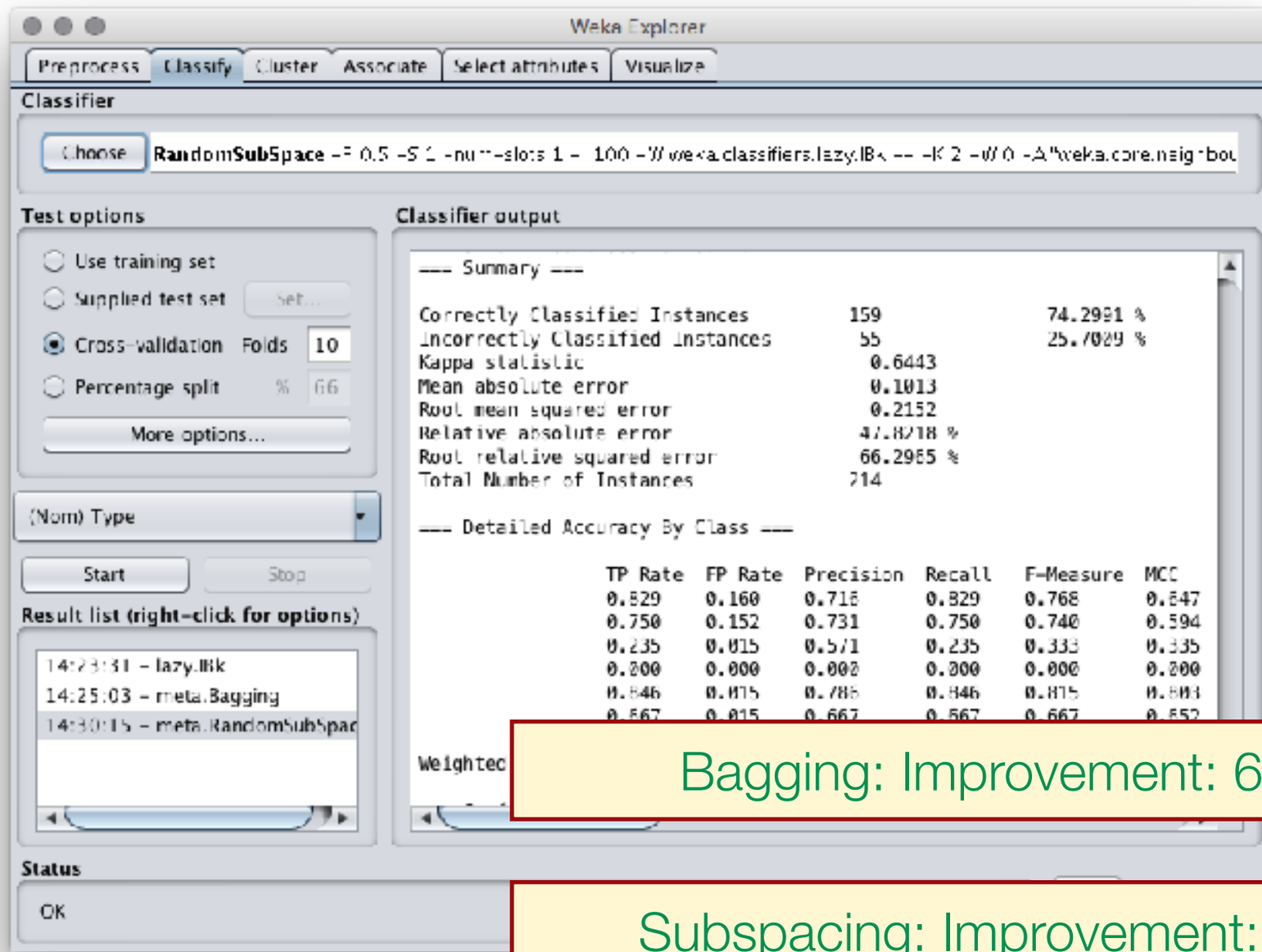
Q. Now apply *random subspacing* with a k-NN classifier (*k=3*) for an ensemble size of 100. How does it compare to the results from (b)? How do you explain this difference?

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 3. Close window.
6. Set the *numIterations* to 100.
7. Click *OK* button.
8. Click *Start* button to build the ensemble.



Tutorial Q3(c)

- 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%