Boston University

CS699 Data Mining

Project Report

PC1 Software Defect Classification

-Paritosh Shirodkar and Kevin Rodrigues

paritosh@bu.edu

kevin96@bu.edu

Introduction

One of the NASA Metrics Data Program defect data sets. Data from flight software for earth orbiting satellite. Data comes from McCabe and Halstead features extractors of source code. These features were defined in the 70s in an attempt to objectively characterize code features that are associated with software quality.

“McCabe's cyclomatic complexity is a software quality metric that quantifies the complexity of a software program. Complexity is inferred by measuring the number of linearly independent paths through the program. The higher the number the more complex the code.”

(http://www.chambers.com.au/glossary/mc_cabe_cyclomatic_complexity.php)

“Halstead’s metrics depends upon the actual implementation of the program and its measures, which are computed directly from the operators and operands from source code, in a static manner. It allows evaluating testing time, vocabulary, size, difficulty, errors, and efforts for source code.”
(https://www.tutorialspoint.com/software_engineering/software_design_complexity.htm)

Attribute Information

1. *loc:* numeric % McCabe's line count of code
2. *v(g):* numeric % McCabe "cyclomatic complexity"
3. *ev(g):* numeric % McCabe "essential complexity"
4. *iv(g):* numeric % McCabe "design complexity"
5. *n:* numeric % Halstead total operators + operands
6. *v:* numeric % Halstead "volume"
7. *l:* numeric % Halstead "program length"
8. *d:* numeric % Halstead "difficulty"
9. *i:* numeric % Halstead "intelligence"
10. *e:* numeric % Halstead "effort"
11. *b:* numeric % Halstead
12. *t:* numeric % Halstead's time estimator
13. *lOCode:* numeric % Halstead's line count
14. *lOComment:* numeric % Halstead's count of lines of comments
15. *lOBlank:* numeric % Halstead's count of blank lines
16. *lOCodeAndComment:* numeric
17. *uniq_Op:* numeric % unique operators
18. *uniq_Opnd:* numeric % unique operands
19. *total_Op:* numeric % total operators
20. *total_Opnd:* numeric % total operands
21. *branchCount:* numeric % of the flow graph
22. *branchCount:* numeric % of the flow graph
23. *defects :* {false,true} % module has/has not one or more reported defects

Data Mining Goal

To classify whether the module has one or more defect or not i.e true or false.

Data-Preprocessing

- Binning

  Performed equal width binning on each attribute of the dataset using the R Script which can be found in the attached documents or over here.

Attribute	Minimum Value	Maximum Value	Number of Bins
Loc	0	602	10
v.g.	1	136	10
ev.g.	1	123	10
iv.G.	1	123	10
N	1	2785	10
V	0	25942.69	10
L	0	2	3
D	0	270.66	10
I	0	598.33	10

E	0	4279633	10
B	0	8.65	10
T	0	237757.4	10
I	0	598.33	10
lOCode	0	600	6
lOComment	0	159	10
IoCodeandComment	0	48	5
IoBlank	0	225	9
Uniq_Op	1	99	10
Uniq_Opnd	0	538	9
total_Op	1	1641	10
total_Opnd	0	1144	10
branchCount	1	236	10

Splitting the dataset

Splitting the dataset into training and test dataset (2/3 of the dataset is the training set = 744 instances, 1/3 of the dataset is the test set = 365 instances) out of these in order to avoid the class imbalance problem 6.94 % of the total instances should be 'TRUE' the remaining would be 'FALSE'.

Training dataset

'TRUE' instances:	52
'FALSE' instance:	692
Total number of instances:	744

Test dataset

'TRUE' instances:	25
'FALSE' instances:	340
Total number of instances:	365

Resulting datasets

Training dataset: pc1_bin_training_set_weka.arff

Test dataset: pc1_bin_test_set_weka.arff

Note: For splitting the dataset using the WEKA software the instructions given in the "Splitting Dataset While Preserving Class Distribution" document are followed.

Attribute selection algorithms used

Gain Ratio: To reduce a bias towards multi-valued attributes, we use gain ratio. In this technique, we take into consideration the number and size of branches whenever choosing an attribute.

(Han, J., Kamber, M., Pei, J., "Data mining: concepts and techniques," 3rd Ed., Morgan Kaufmann, 2012)

WrapperSubsetEval: In the wrapper approach to feature subset selection, a search for an optimal set of features is made using the induction algorithm as a black box. The estimated future performance of the algorithm is the heuristic guiding the search. Statistical methods for feature subset selection including forward selection, backward elimination, and their stepwise variants can be viewed as simple hill-climbing techniques in the space of feature subsets".

(http://www.aaai.org/Papers/KDD/1995/KDD95-049.pdf)

SymetricalUncertAttributeEval: Evaluates the worth of an attribute by measuring the symmetrical uncertainty with respect to the class.
(http://weka.sourceforge.net/doc.dev/weka/attributeSelection/SymmetricalUncertAttributeEval.html)

CfsSubsetEval: Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.
(http://weka.sourceforge.net/doc.dev/weka/attributeSelection/CfsSubsetEval.html)

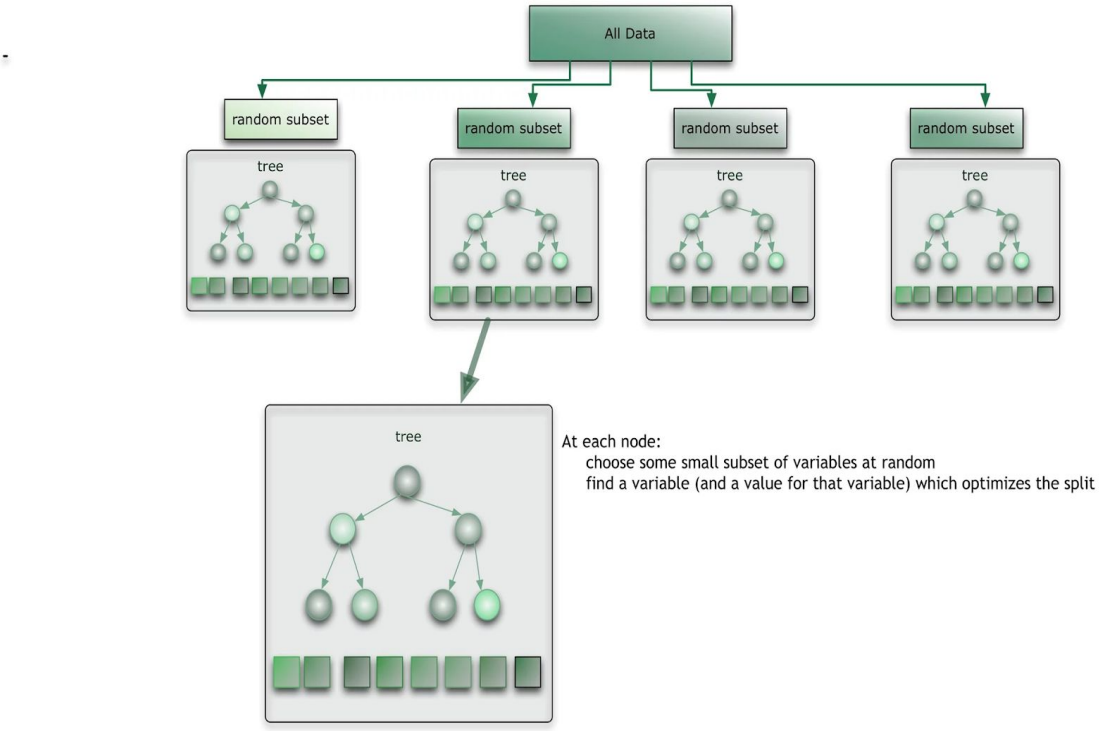
Classification algorithms used

Multilayer Perceptron: A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer is capable of approximating any continuous function. Multilayer perceptrons are often applied to supervised learning problems. They train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Backpropagation is used to make those weights and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE). (https://skymind.ai/wiki/multilayer-perceptron)

J48: The C4.5 algorithm for building decision trees is implemented in WEKA as a classifier called J48. First, C4.5 uses information gain when generating the decision tree. Second, although other systems also incorporate pruning, C4.5 uses a single-pass pruning process to mitigate over-fitting. Pruning results in many improvements. Third, C4.5 can work with both continuous and discrete data. Our understanding is it does this by specifying ranges or thresholds for continuous data thus turning continuous data into discrete data. Finally, incomplete data is dealt with in its own way.

(https://hackerbits.com/data/c4-5-data-mining-algorithm/)

Random Forest: The random forest starts with a standard machine learning technique called a “decision tree” which, in ensemble terms, corresponds to our weak learner. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets. Thus, in ensemble terms, the trees are weak learners and the random forest is a strong learner. (http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics)



IBK: The k-nearest neighbors algorithm (KNN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. In KNN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to

the class of that single nearest neighbor. In KNN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

KNN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. (http://sciencewise.info/resource/Ibk_algorithm/Ibk_algorithm_by_Wikipedia)

Naive Bayes: Naive Bayesian classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved in this algorithm.

Bayesian Theorem:

• The purpose of the Bayesian theorem is to predict the class label for a given tuple.

• Let X be a data tuple.

• In Bayesian terms, X is considered "evidence."

• It is described by measurements made on a set of n attributes.

• Let H be some hypothesis, such as that the data tuple X belongs to a specified class C.

• For classification problems, we are looking for the probability that tuple X belongs to class C, given that we know the attribute description of X.

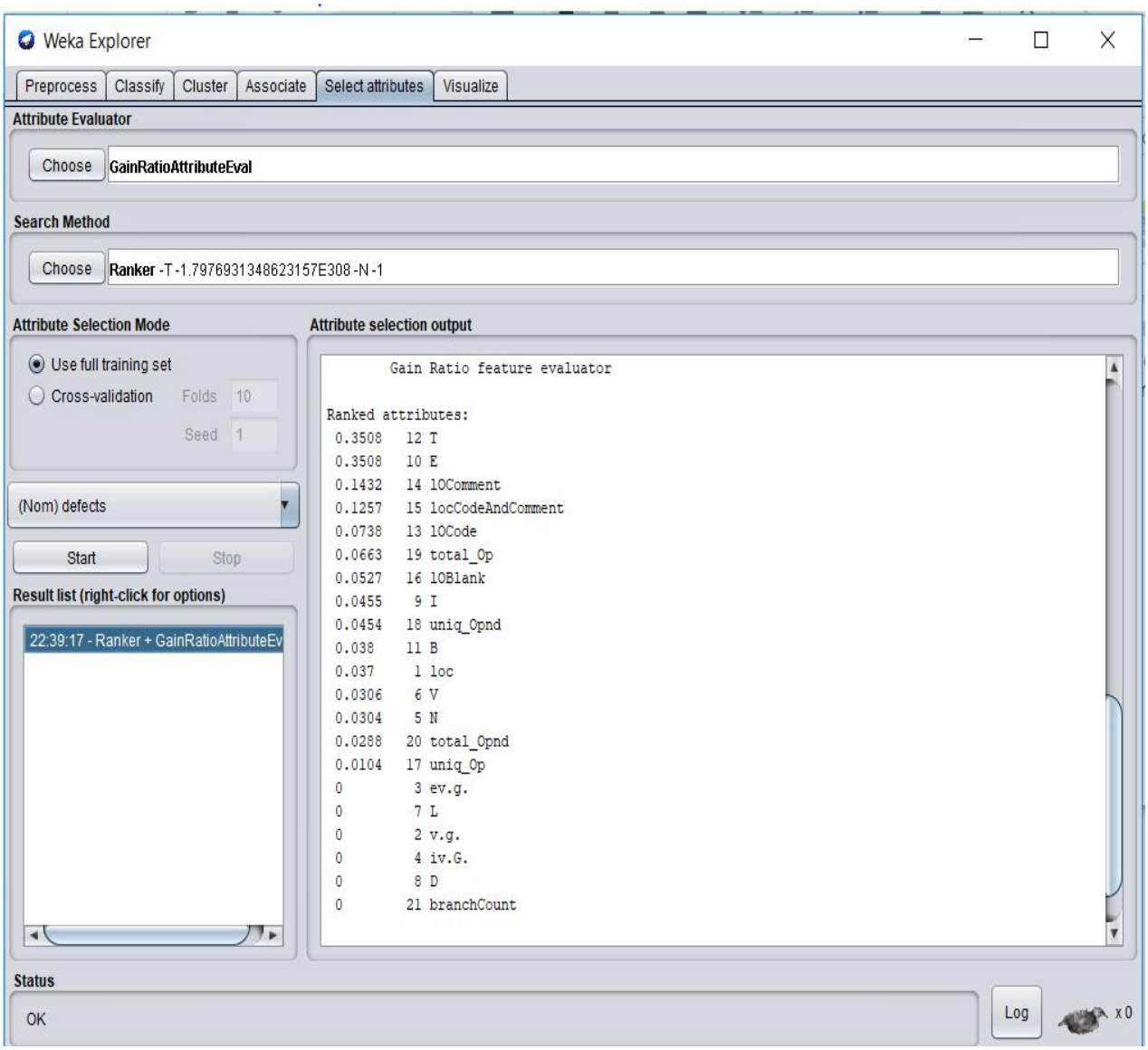
Bayesian theorem is $P(H\backslash X) = \frac{P\left(\frac{X}{H}\right)P(H)}{P(X)}$

(http://www.ques10.com/p/166/write-short-note-on-bayesian-classification/)

Building Classifier Models

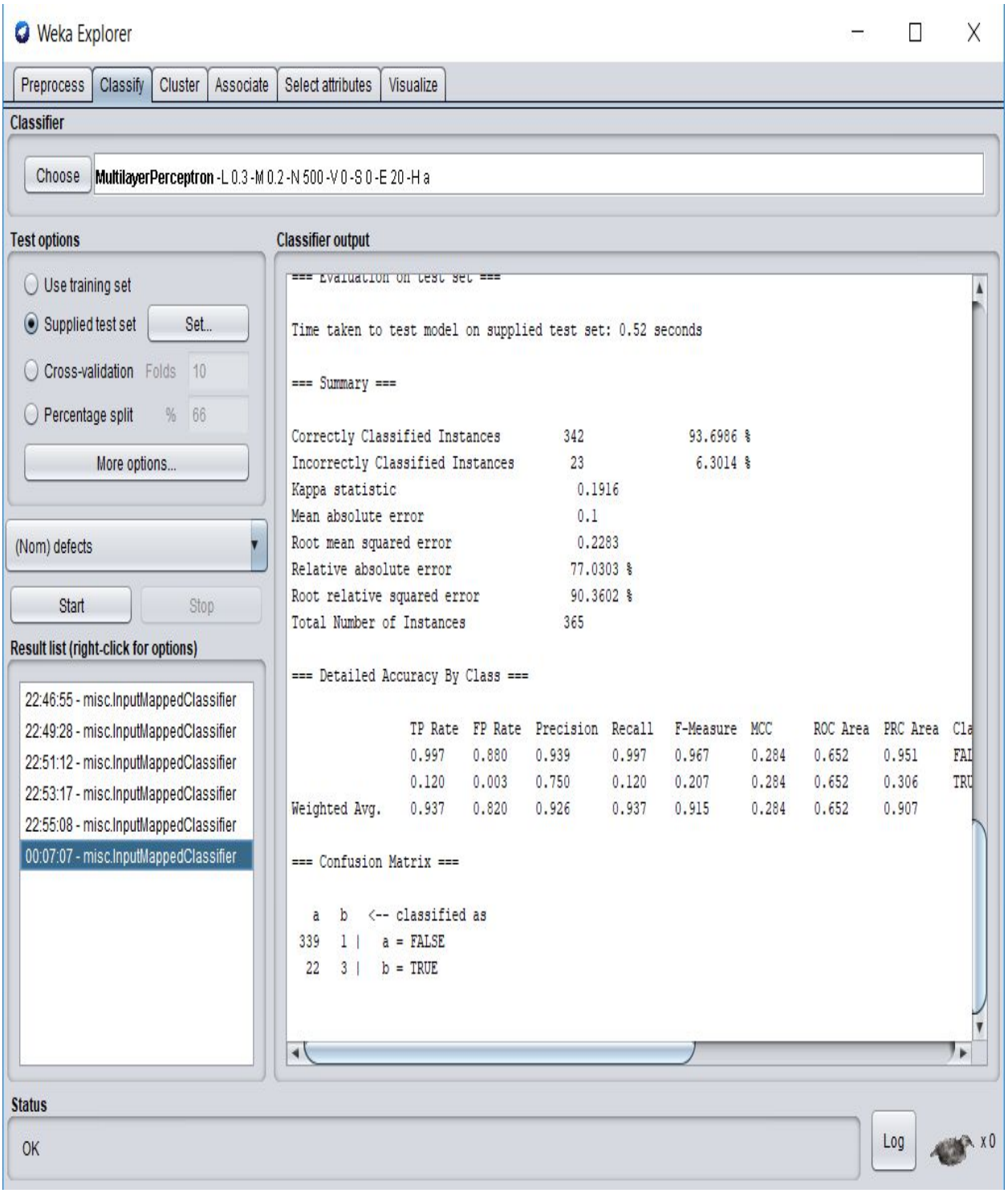
Screenshots of Attribute Selection Algorithms and their corresponding Classifier models

Gain Ratio

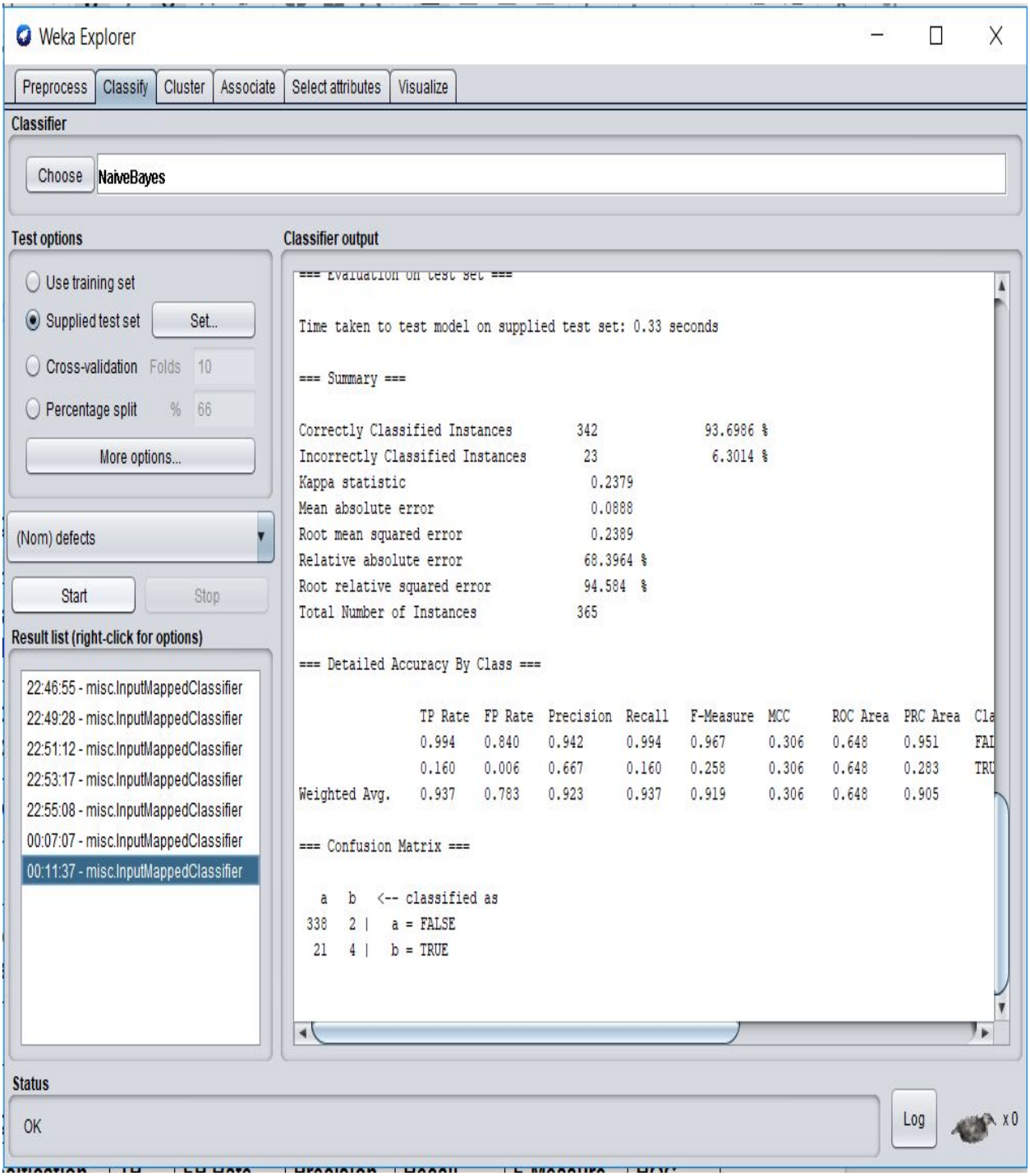


Selected Attributes: T, E, lOComment, lOCodeAndComment and lOCode

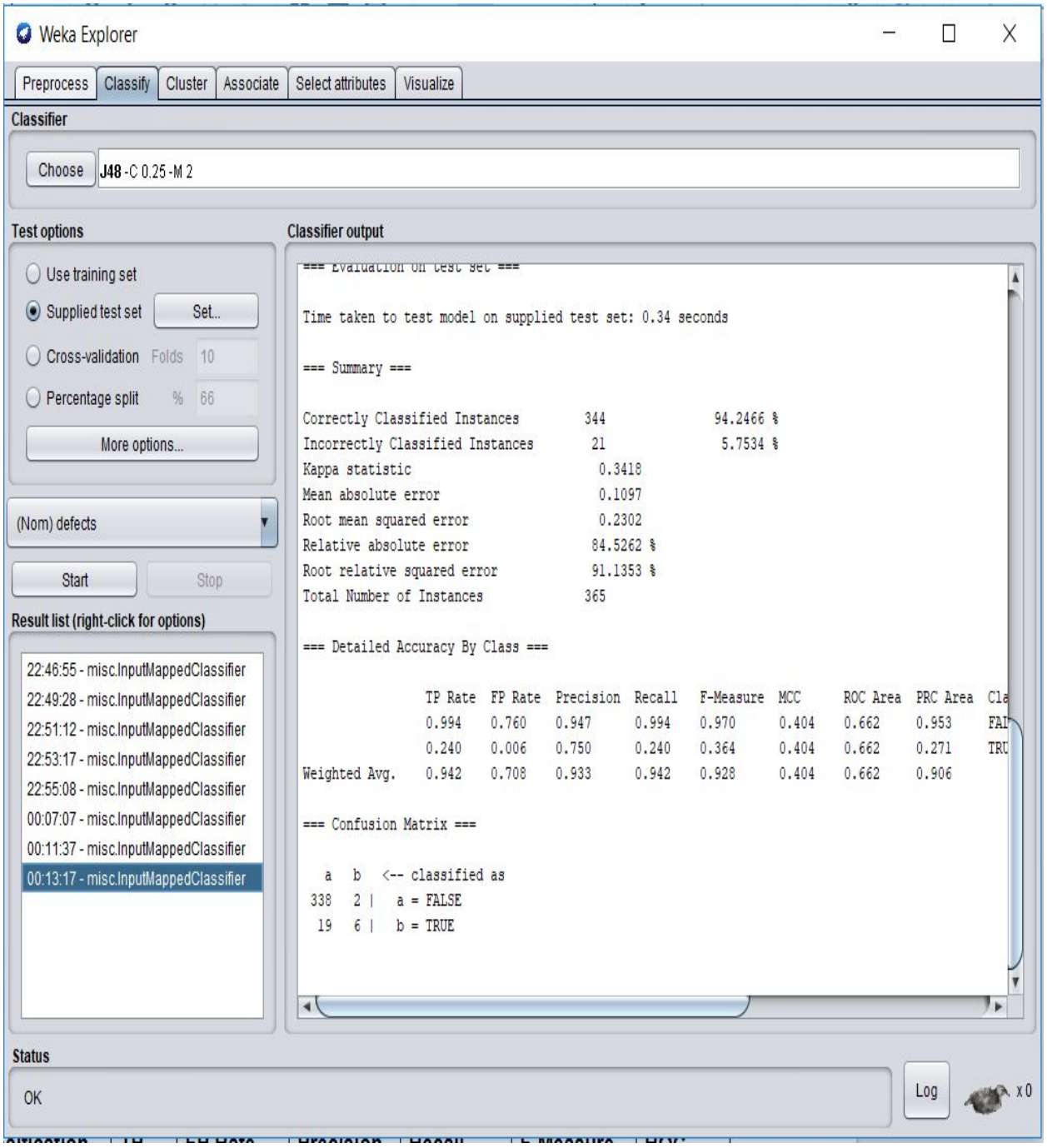
Multilayer Perceptron



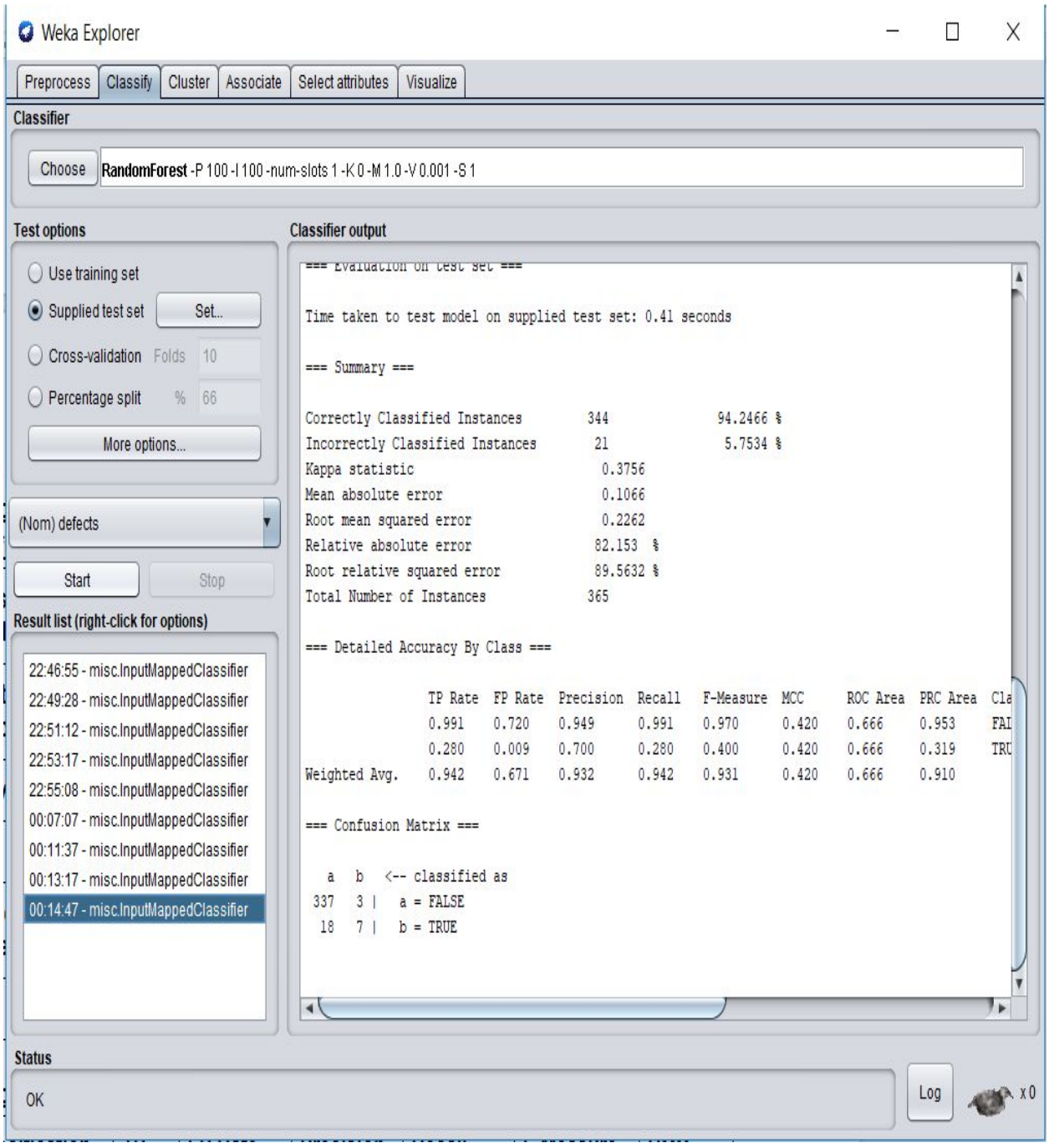
Naive Bayes


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) defects
Start
Stop
Result list (right-click for options)
22:46:55 - misc.InputMappedClassifier
22:49:28 - misc.InputMappedClassifier
22:51:12 - misc.InputMappedClassifier
22:53:17 - misc.InputMappedClassifier
22:55:08 - misc.InputMappedClassifier
00:07:07 - misc.InputMappedClassifier
00:11:37 - misc.InputMappedClassifier
Classifier output
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.33 seconds
=== Summary ===
Correctly Classified Instances 342 93.6986 %
Incorrectly Classified Instances 23 6.3014 %
Kappa statistic 0.2379
Mean absolute error 0.0888
Root mean squared error 0.2389
Relative absolute error 68.3964 %
Root relative squared error 94.584 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Cla
0.994 0.840 0.942 0.994 0.967 0.306 0.648 0.951 FAL
0.160 0.006 0.667 0.160 0.258 0.306 0.648 0.283 TRU
Weighted Avg. 0.937 0.783 0.923 0.937 0.919 0.306 0.648 0.905
=== Confusion Matrix ===
a b <-- classified as
338 2 | a = FALSE
21 4 | b = TRUE
Status
OK
Log
x 0

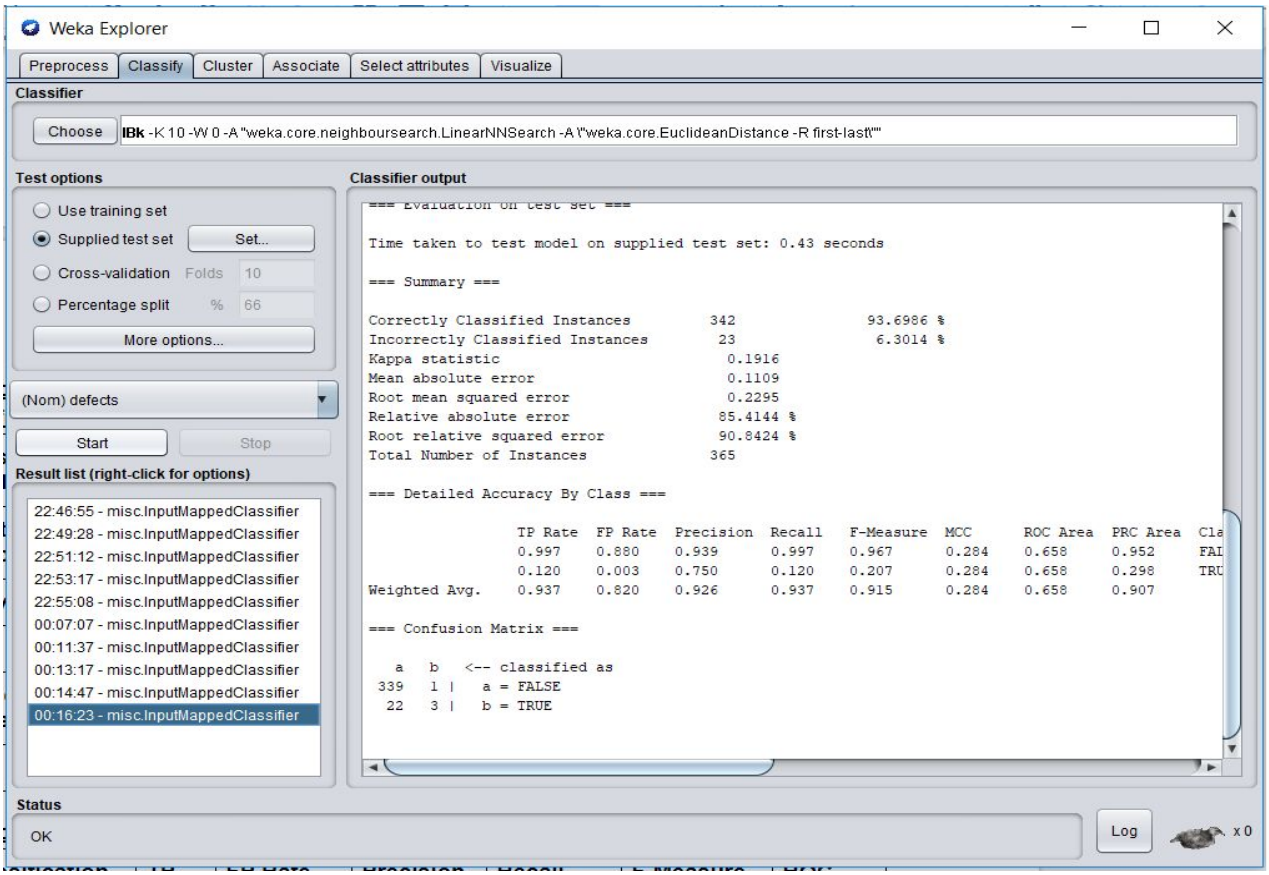
J48


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
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) defects
Start
Stop
Result list (right-click for options)
22:46:55 - misc.InputMappedClassifier
22:49:28 - misc.InputMappedClassifier
22:51:12 - misc.InputMappedClassifier
22:53:17 - misc.InputMappedClassifier
22:55:08 - misc.InputMappedClassifier
00:07:07 - misc.InputMappedClassifier
00:11:37 - misc.InputMappedClassifier
00:13:17 - misc.InputMappedClassifier
Classifier output
Time taken to test model on supplied test set: 0.34 seconds
=== Summary ===
Correctly Classified Instances 344 94.2466 %
Incorrectly Classified Instances 21 5.7534 %
Kappa statistic 0.3418
Mean absolute error 0.1097
Root mean squared error 0.2302
Relative absolute error 84.5262 %
Root relative squared error 91.1353 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area
0.994 0.760 0.947 0.994 0.970 0.404 0.662 0.953
0.240 0.006 0.750 0.240 0.364 0.404 0.662 0.271
Weighted Avg. 0.942 0.708 0.933 0.942 0.928 0.404 0.662 0.906
=== Confusion Matrix ===
a b <-- classified as
338 2 | a = FALSE
19 6 | b = TRUE
Status
OK
Log
x 0

Random Forest



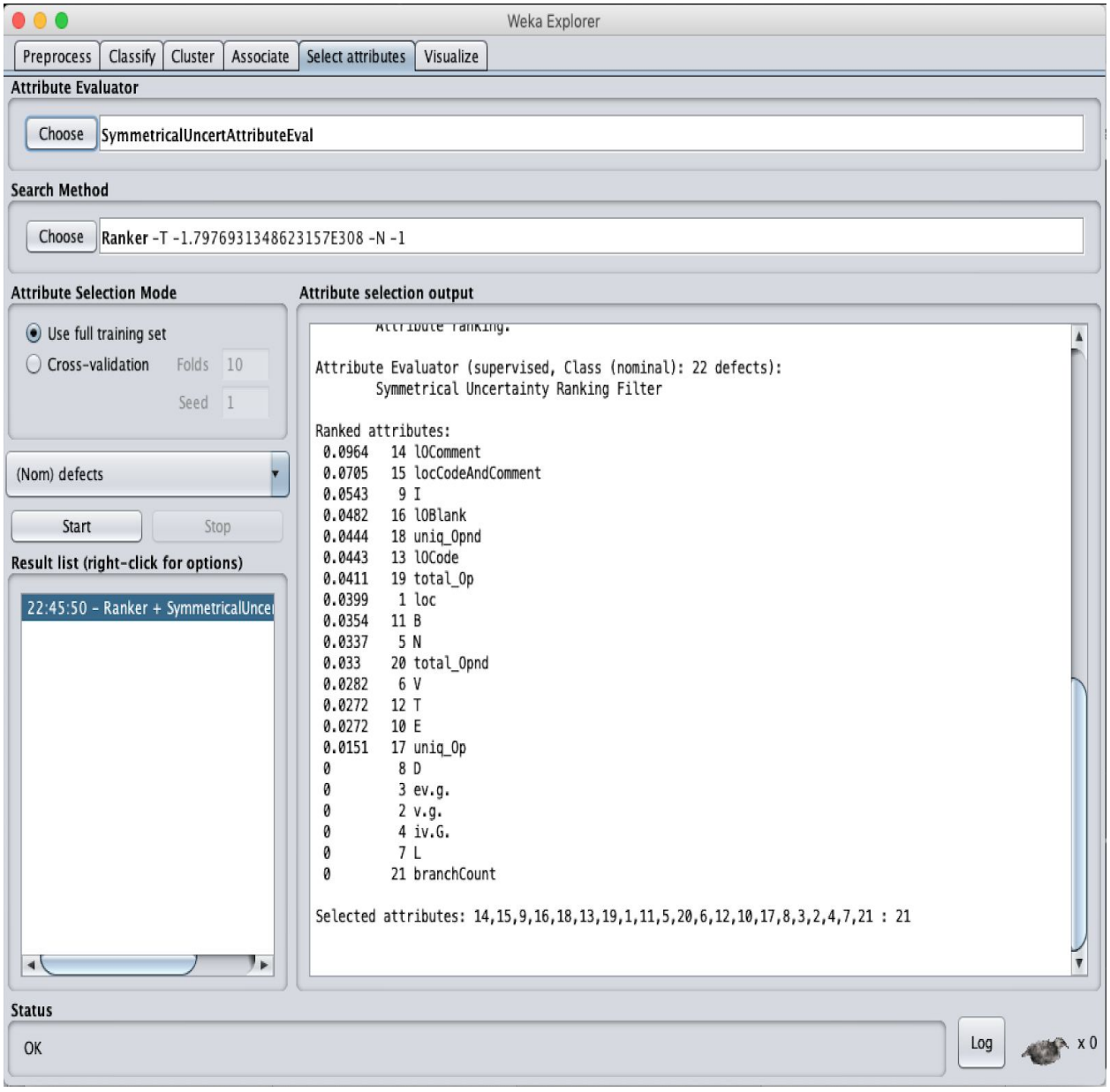
Instance-based KNN (K = 10)



Classification Model	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Accuracy
Multilayer Perceptron	0.937	0.820	0.926	0.937	0.915	0.652	93.69
Naive Bayes	0.937	0.783	0.923	0.937	0.919	0.648	93.69
J48	0.942	0.708	0.933	0.942	0.928	0.662	94.25
Random Forest	0.942	0.671	0.932	0.942	0.931	0.666	94.25
IBK (K=10)	0.937	0.820	0.926	0.937	0.915	0.658	93.69

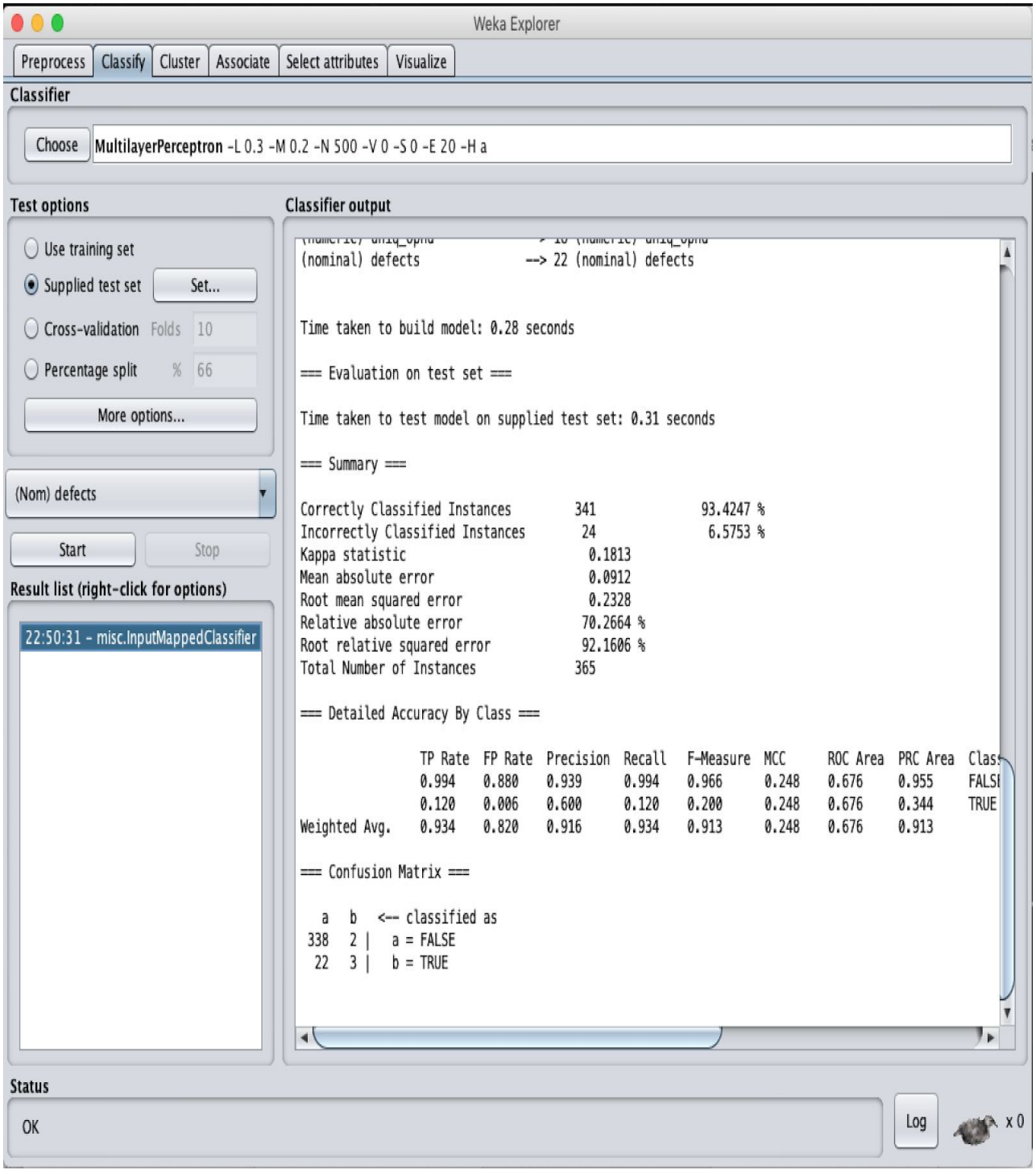
Note:- Highlighted model gives the best results for this Attribute selection Algorithm

SymetricalUncertAttributeEval

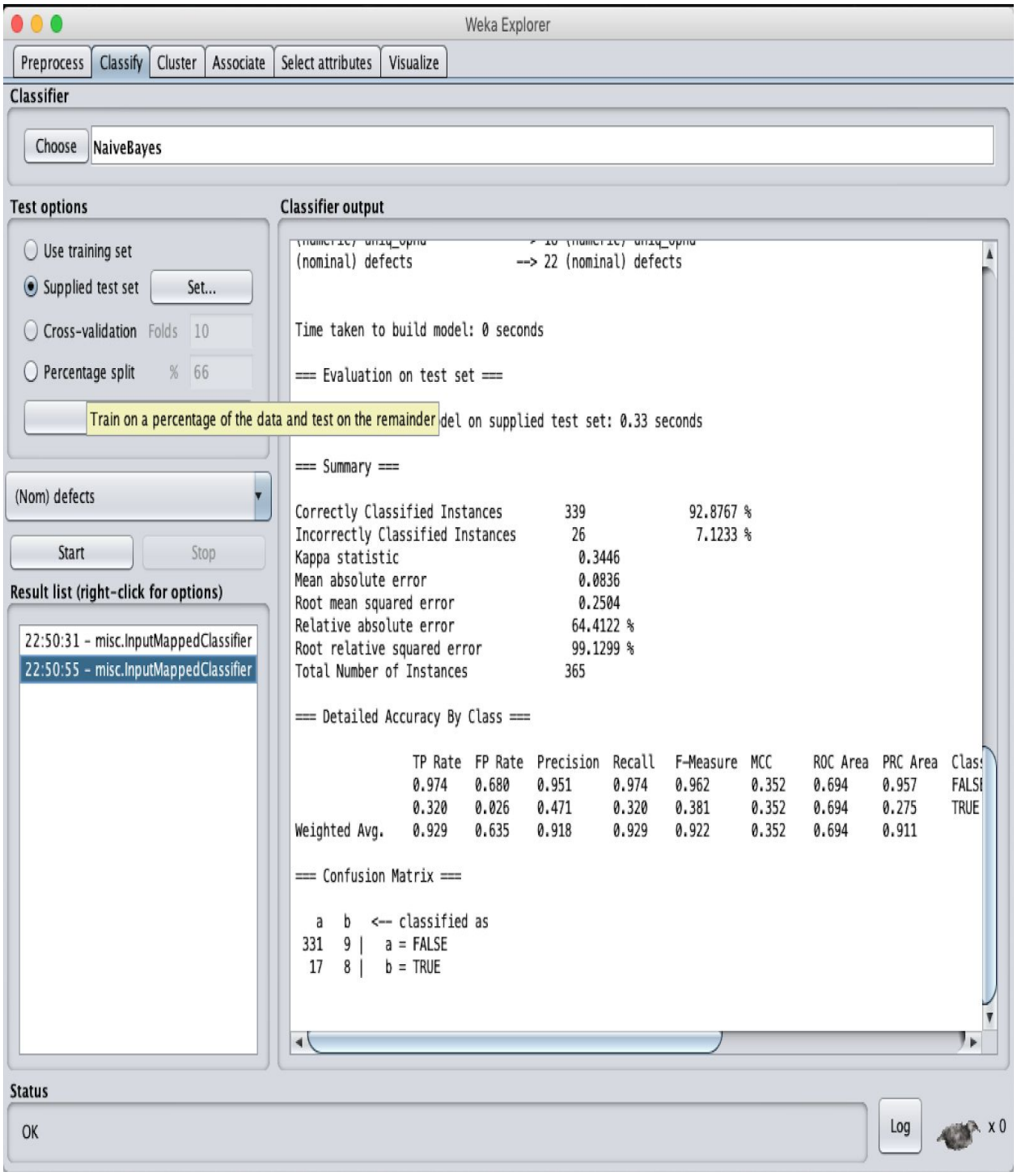


Attributes selected: loComment, locCodeandComment, I, loBlank, Uniq_opnd

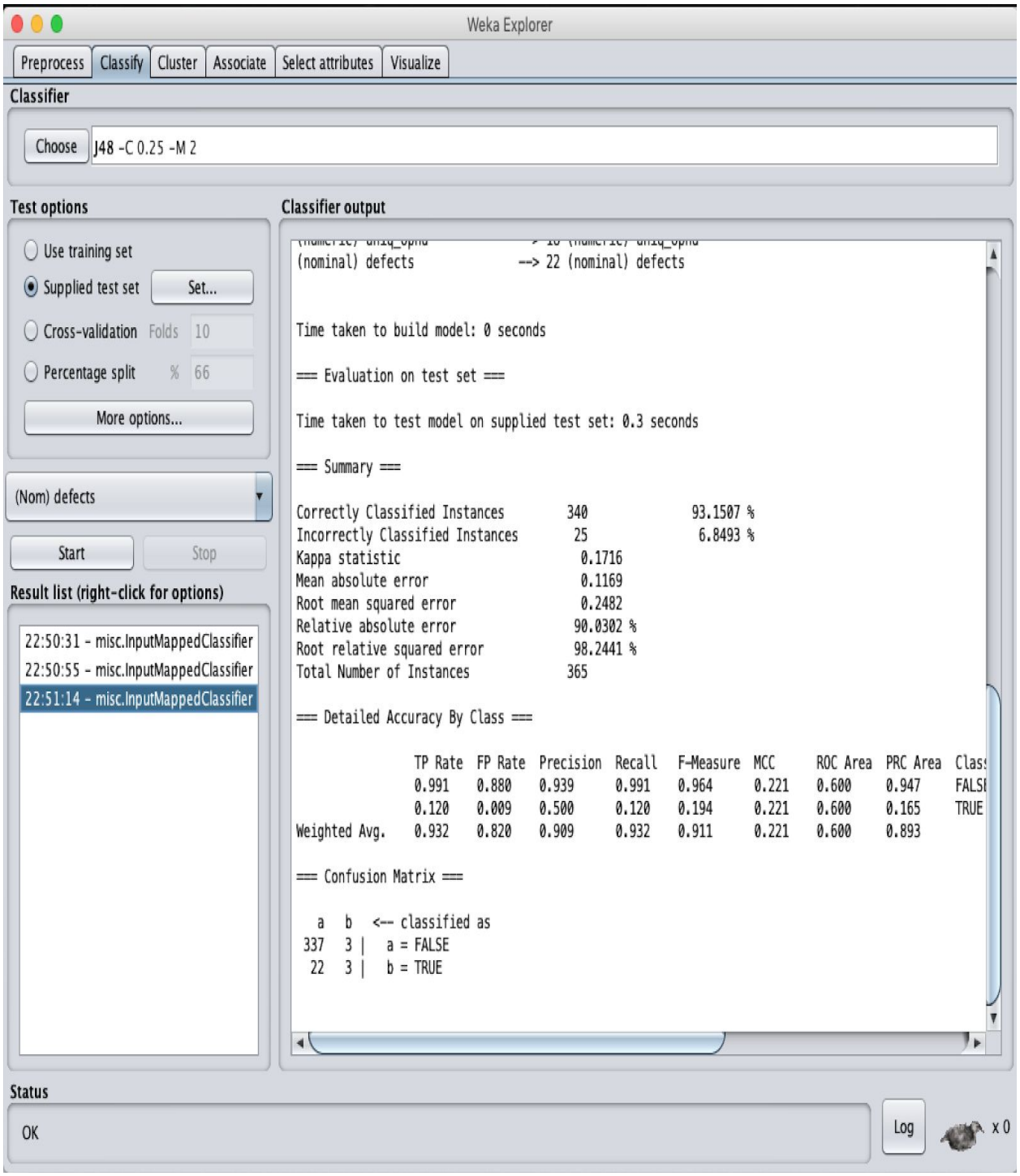
Multilayer Perceptron



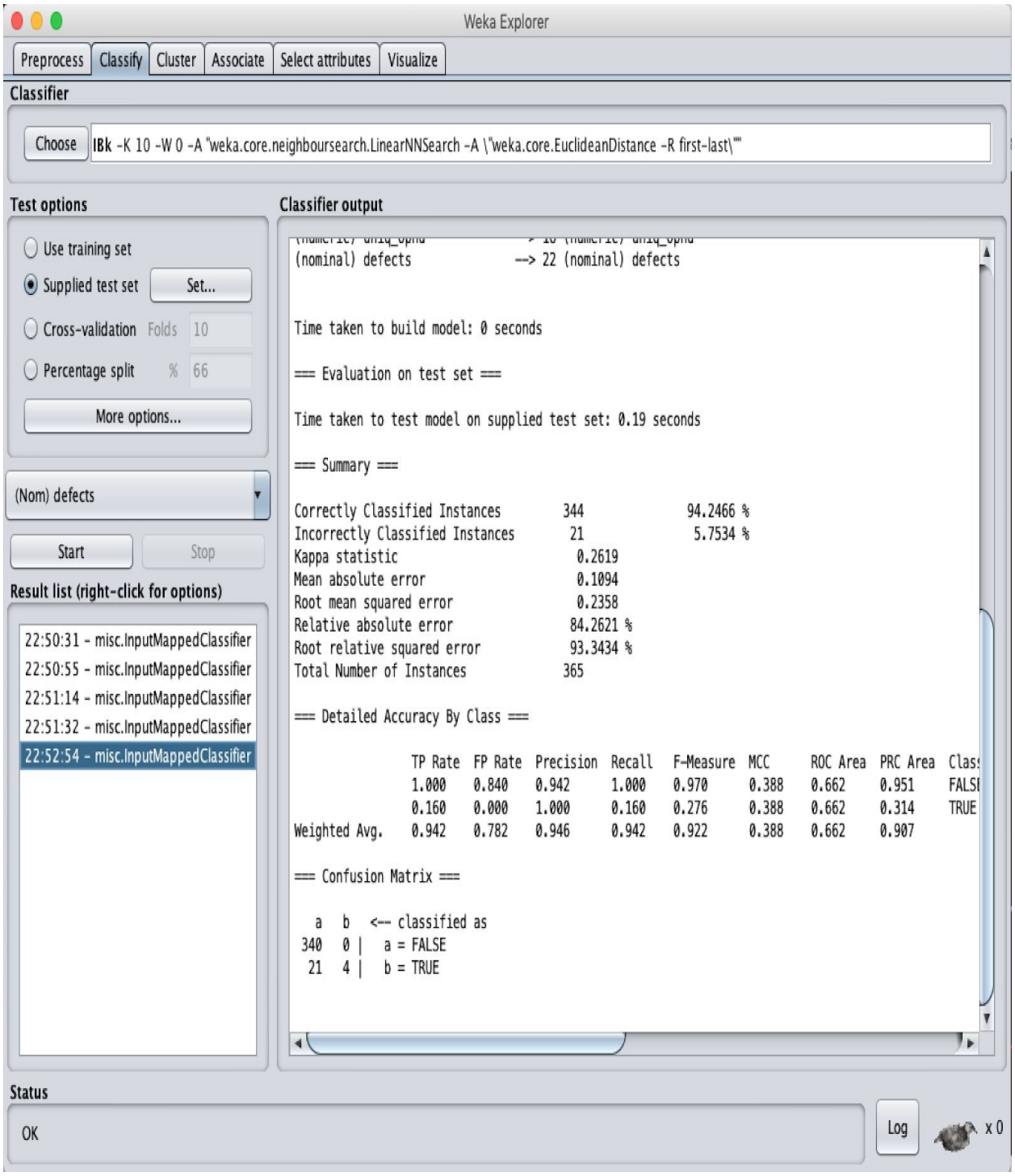
Naïve Bayes


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
Train on a percentage of the data and test on the remainder
(Nom) defects
Start
Stop
Result list (right-click for options)
22:50:31 - misc.InputMappedClassifier
22:50:55 - misc.InputMappedClassifier
Classifier output
(nominal) defects --> 22 (nominal) defects
Time taken to build model: 0 seconds
=== Evaluation on test set ===
del on supplied test set: 0.33 seconds
=== Summary ===
Correctly Classified Instances 339 92.8767 %
Incorrectly Classified Instances 26 7.1233 %
Kappa statistic 0.3446
Mean absolute error 0.0836
Root mean squared error 0.2504
Relative absolute error 64.4122 %
Root relative squared error 99.1299 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.974 0.680 0.951 0.974 0.962 0.352 0.694 0.957 FALSE
0.320 0.026 0.471 0.320 0.381 0.352 0.694 0.275 TRUE
Weighted Avg. 0.929 0.635 0.918 0.929 0.922 0.352 0.694 0.911
=== Confusion Matrix ===
a b <-- classified as
331 9 | a = FALSE
17 8 | b = TRUE
Status
OK
Log
x 0

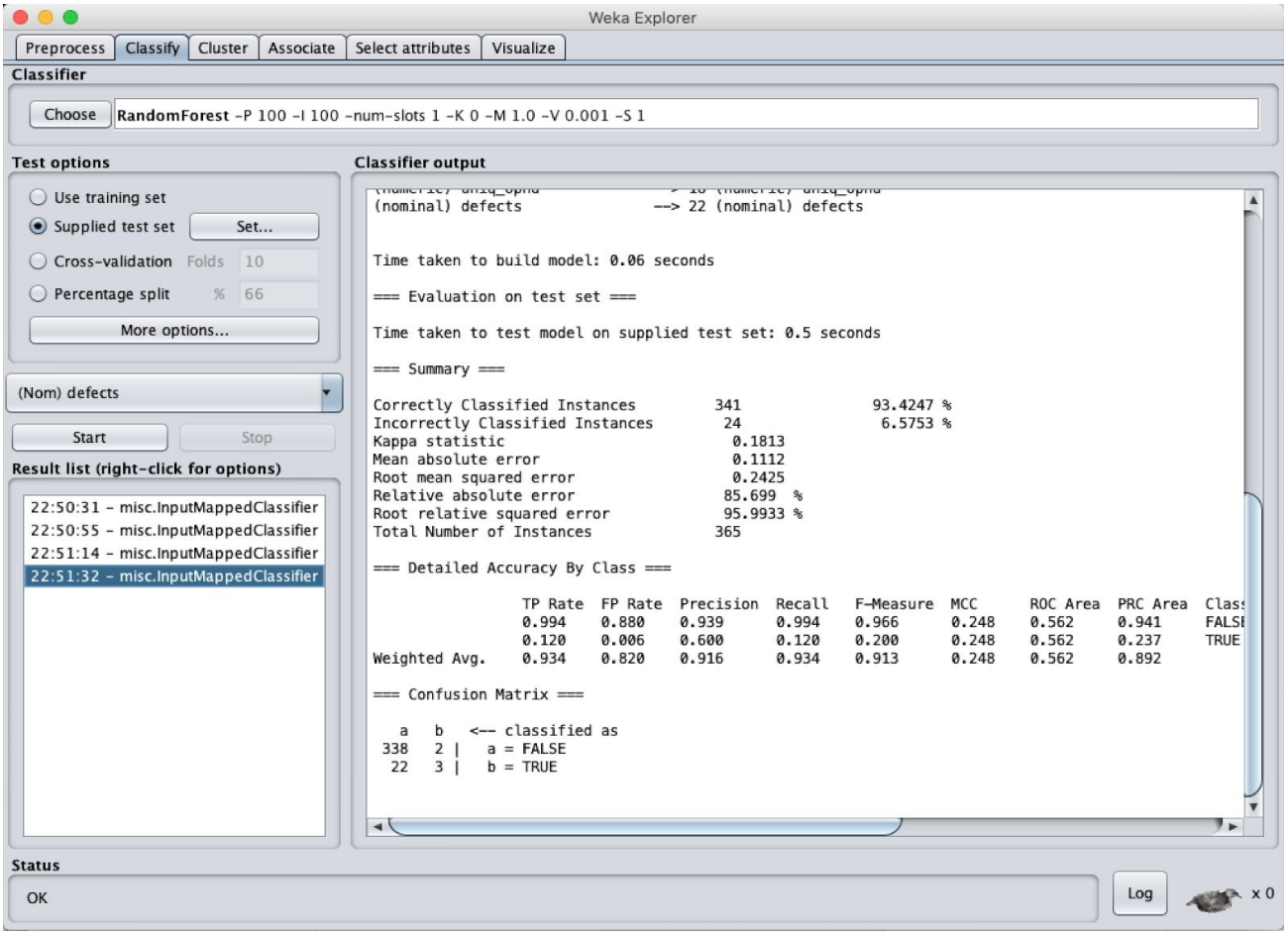
J48


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
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) defects
Start
Stop
Result list (right-click for options)
22:50:31 - misc.InputMappedClassifier
22:50:55 - misc.InputMappedClassifier
22:51:14 - misc.InputMappedClassifier
Classifier output
(nominal) defects --> 22 (nominal) defects
Time taken to build model: 0 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.3 seconds
=== Summary ===
Correctly Classified Instances 340 93.1507 %
Incorrectly Classified Instances 25 6.8493 %
Kappa statistic 0.1716
Mean absolute error 0.1169
Root mean squared error 0.2482
Relative absolute error 90.0302 %
Root relative squared error 98.2441 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.991 0.880 0.939 0.991 0.964 0.221 0.600 0.947 FALSE
0.120 0.009 0.500 0.120 0.194 0.221 0.600 0.165 TRUE
Weighted Avg. 0.932 0.820 0.909 0.932 0.911 0.221 0.600 0.893
=== Confusion Matrix ===
a b <-- classified as
337 3 | a = FALSE
22 3 | b = TRUE
Status
OK
Log
x 0

Random Forest



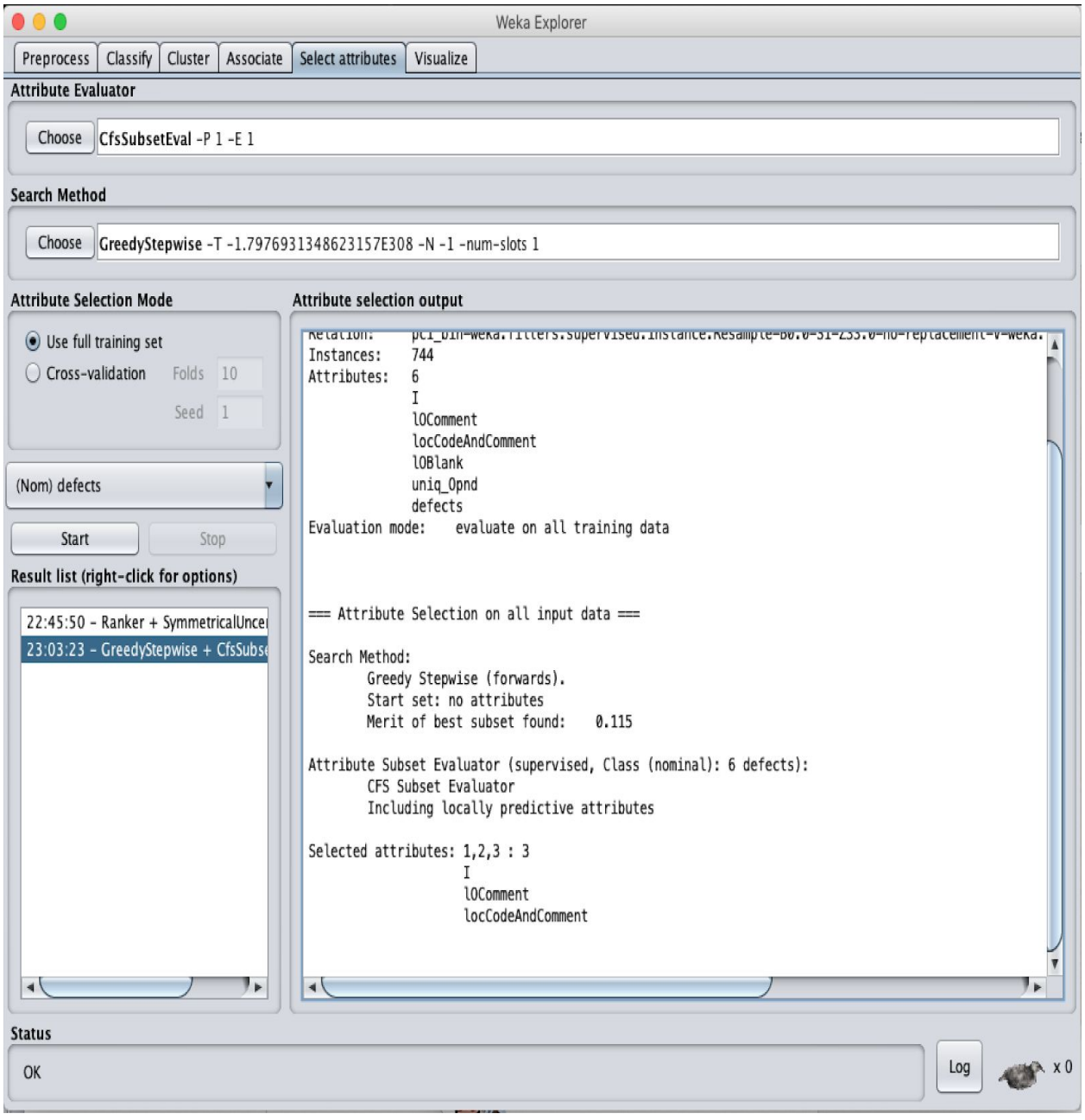
Instance-based KNN (K = 10)



Classification Model	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Accuracy
Multilayer Perceptron	0.934	*0.820*	*0.916*	*0.934*	*0.913*	*0.676*	*93.43*
Naive Bayes	0.929	*0.635*	*0.918*	*0.929*	*0.922*	*0.694*	*92.87*
J48	*0.932*	*0.820*	*0.909*	*0.932*	*0.911*	*0.600*	*93.15*
Random Forest	*0.934*	*0.820*	*0.916*	*0.934*	*0.913*	*0.562*	*93.42*
IBK (K=10)	*0.942*	*0.782*	*0.946*	*0.942*	*0.922*	*0.662*	*94.25*

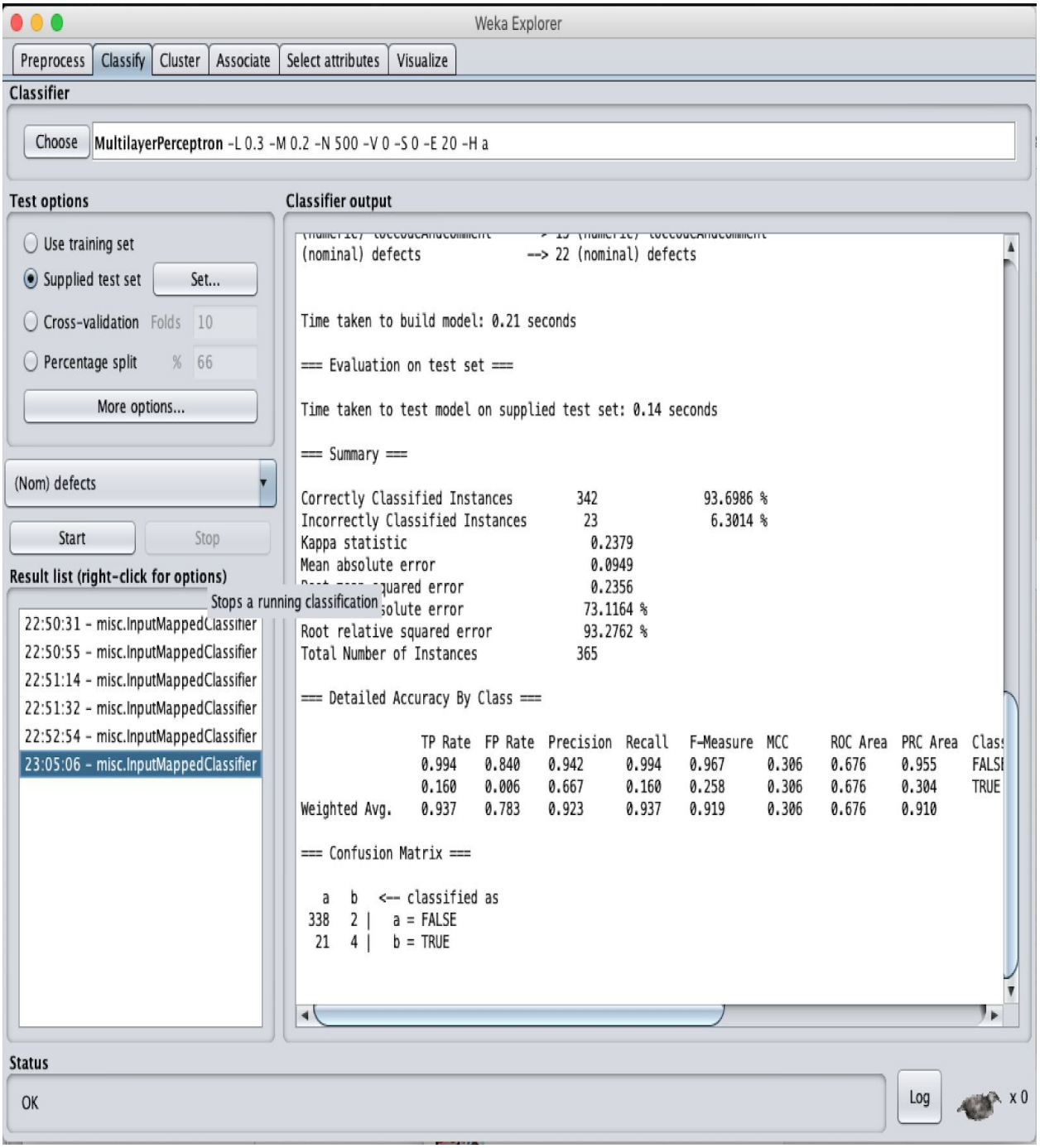
Note:- Highlighted model gives the best results for this Attribute selection Algorithm

CfsSubsetEval

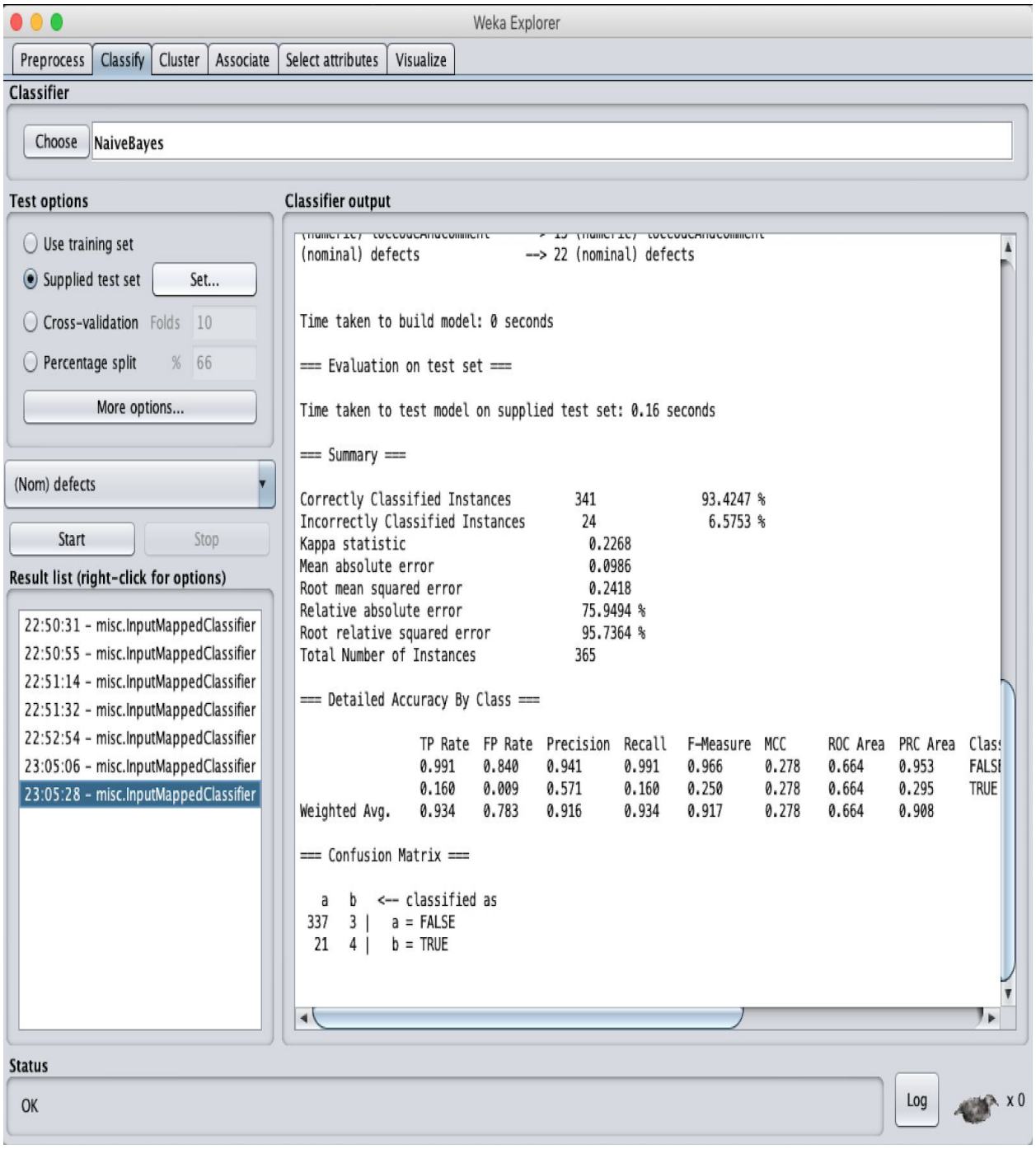


Attributes selected: lOComment, locCodeandComment, I

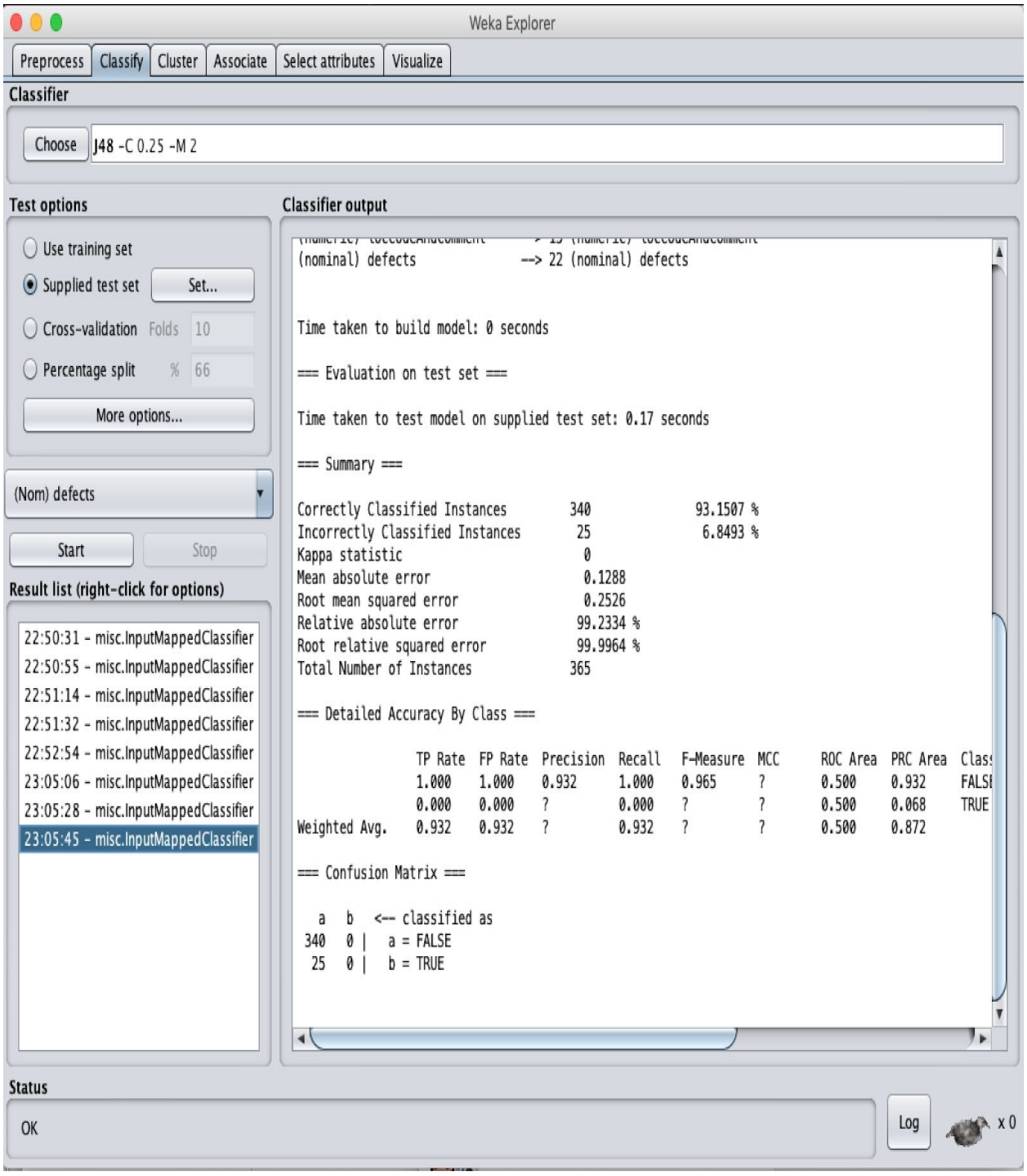
Multilayer Perceptron



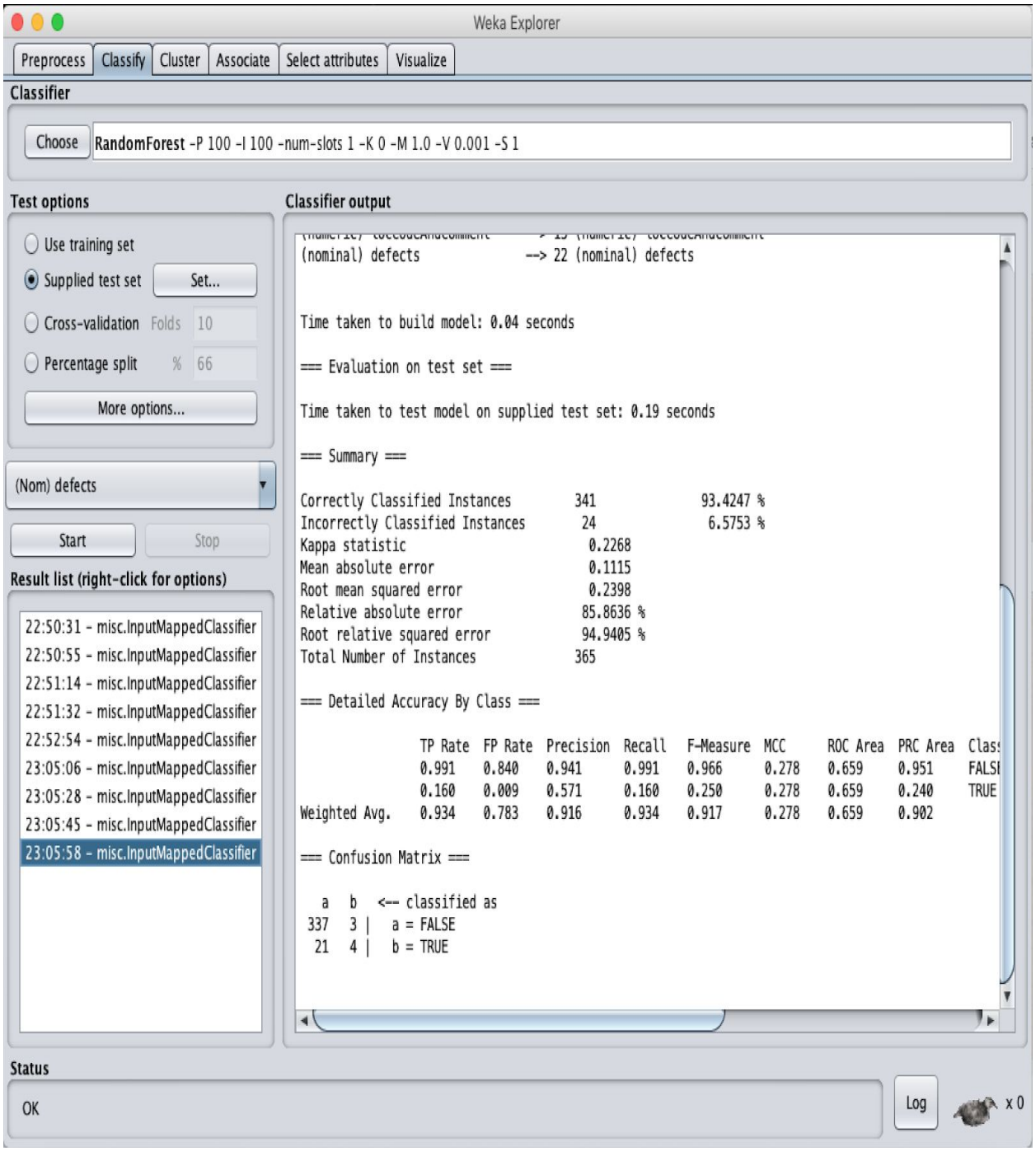
Naïve Bayes


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) defects
Start
Stop
Result list (right-click for options)
22:50:31 - misc.InputMappedClassifier
22:50:55 - misc.InputMappedClassifier
22:51:14 - misc.InputMappedClassifier
22:51:32 - misc.InputMappedClassifier
22:52:54 - misc.InputMappedClassifier
23:05:06 - misc.InputMappedClassifier
23:05:28 - misc.InputMappedClassifier
Classifier output
(nominal) defects --> 22 (nominal) defects
Time taken to build model: 0 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.16 seconds
=== Summary ===
Correctly Classified Instances 341 93.4247 %
Incorrectly Classified Instances 24 6.5753 %
Kappa statistic 0.2268
Mean absolute error 0.0986
Root mean squared error 0.2418
Relative absolute error 75.9494 %
Root relative squared error 95.7364 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.991 0.840 0.941 0.991 0.966 0.278 0.664 0.953 FALSE
0.160 0.009 0.571 0.160 0.250 0.278 0.664 0.295 TRUE
Weighted Avg. 0.934 0.783 0.916 0.934 0.917 0.278 0.664 0.908
=== Confusion Matrix ===
a b <-- classified as
337 3 | a = FALSE
21 4 | b = TRUE
Status
OK
Log
x 0

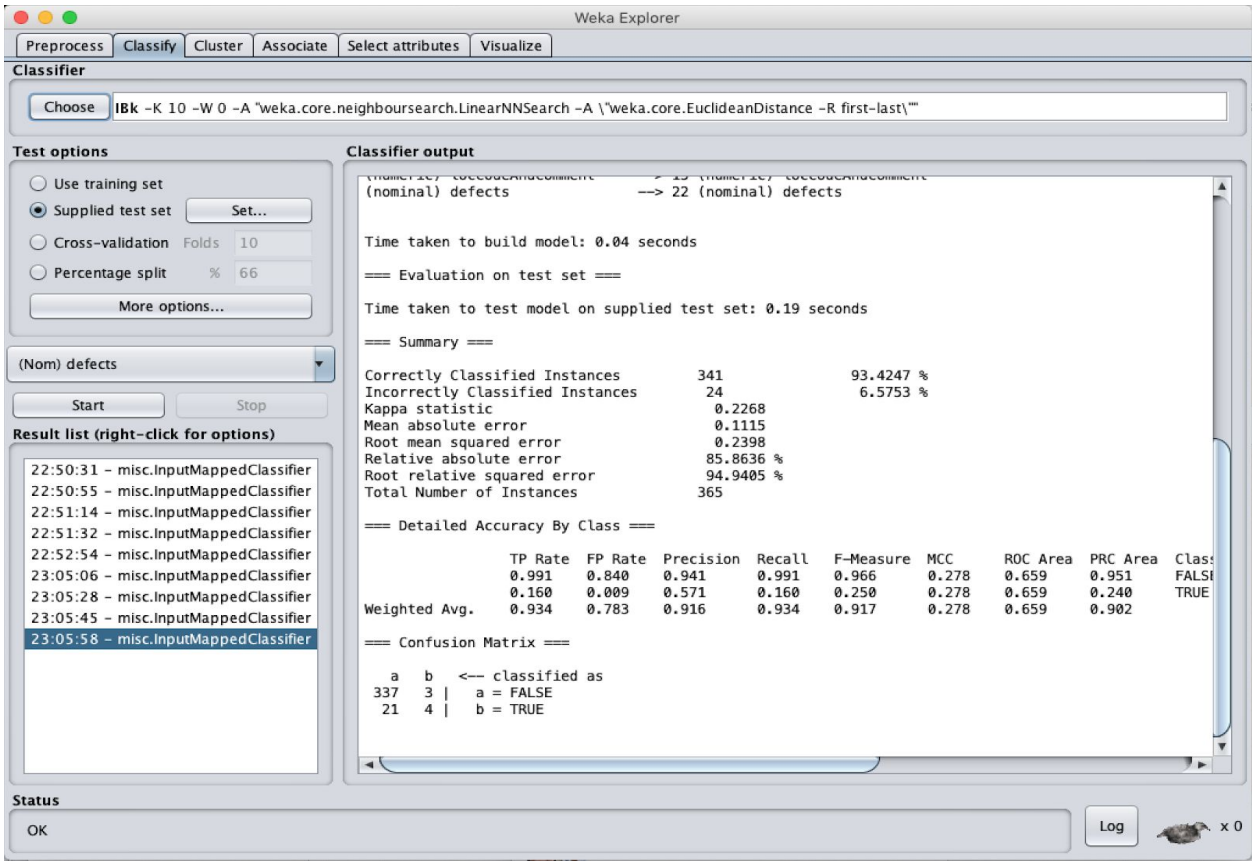
J48


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
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) defects
Start
Stop
Result list (right-click for options)
22:50:31 - misc.InputMappedClassifier
22:50:55 - misc.InputMappedClassifier
22:51:14 - misc.InputMappedClassifier
22:51:32 - misc.InputMappedClassifier
22:52:54 - misc.InputMappedClassifier
23:05:06 - misc.InputMappedClassifier
23:05:28 - misc.InputMappedClassifier
23:05:45 - misc.InputMappedClassifier
Classifier output
(nominal) defects --> 22 (nominal) defects
Time taken to build model: 0 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.17 seconds
=== Summary ===
Correctly Classified Instances 340 93.1507 %
Incorrectly Classified Instances 25 6.8493 %
Kappa statistic 0
Mean absolute error 0.1288
Root mean squared error 0.2526
Relative absolute error 99.2334 %
Root relative squared error 99.9964 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area
1.000 1.000 0.932 1.000 0.965 ? 0.500 0.932 FALSE
0.000 0.000 ? 0.000 ? ? 0.500 0.068 TRUE
Weighted Avg. 0.932 0.932 ? 0.932 ? ? 0.500 0.872
=== Confusion Matrix ===
a b <-- classified as
340 0 | a = FALSE
25 0 | b = TRUE
Status
OK
Log
x 0

Random Forest



Instance-Based KNN (K = 10)



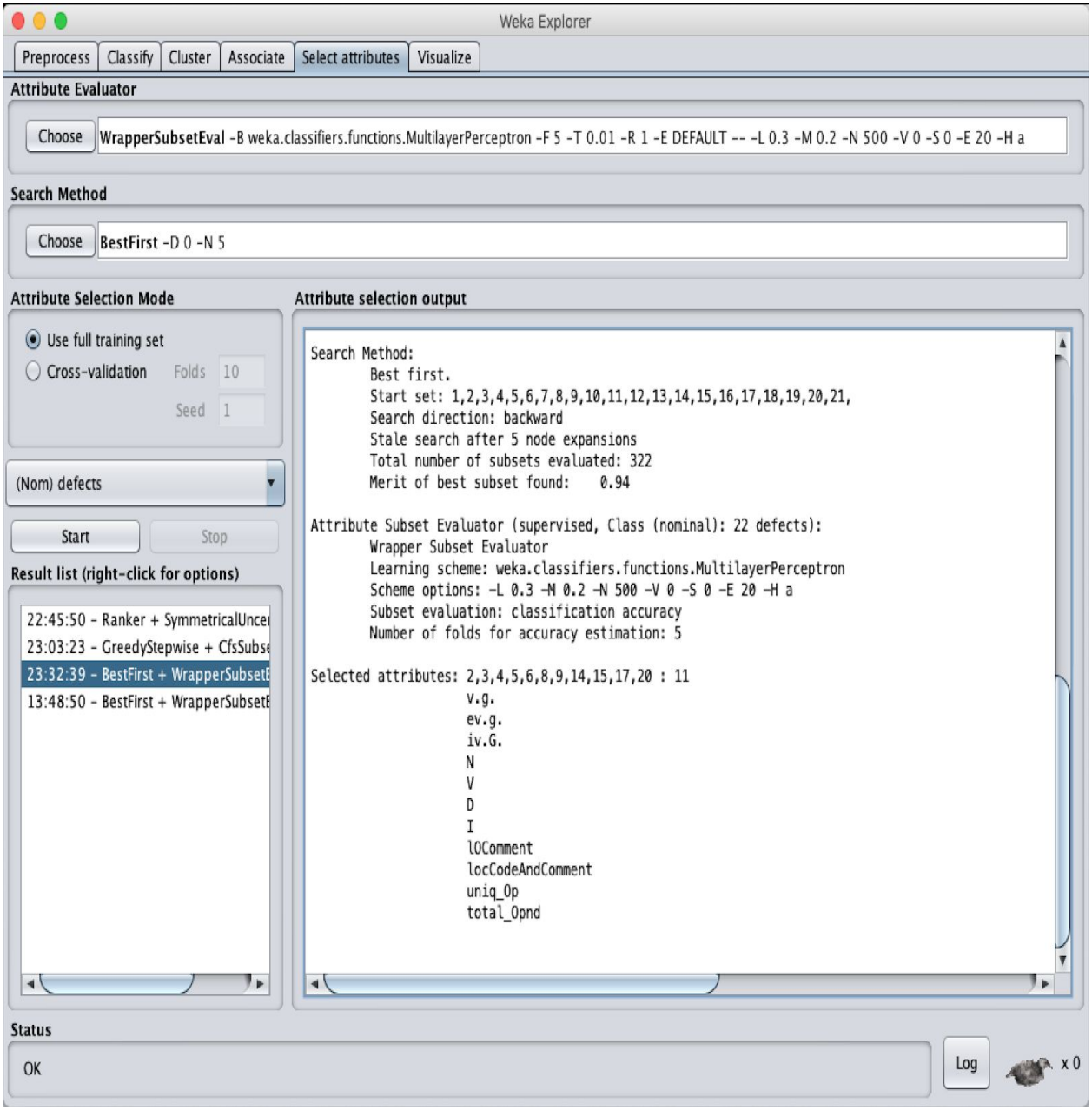
Classification Model	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Accuracy
Multilayer Perceptron	0.937	0.783	0.923	0.937	0.919	0.671	93.69
Naive Bayes	0.934	0.783	0.916	0.934	0.917	0.664	93.43
J48	0.932	0.932	ud	0.932	ud	0.500	93.15
Random Forest	0.934	0.783	0.916	0.934	0.917	0.659	93.43
IBK (K=10)	0.934	0.783	0.916	0.934	0.917	0.659	93.43

The value of Precision and F-Measure is undefined (ud) since the value of TP and FP for J48 classifier is 0.

Note:- Highlighted model gives the best results for this Attribute selection Algorithm

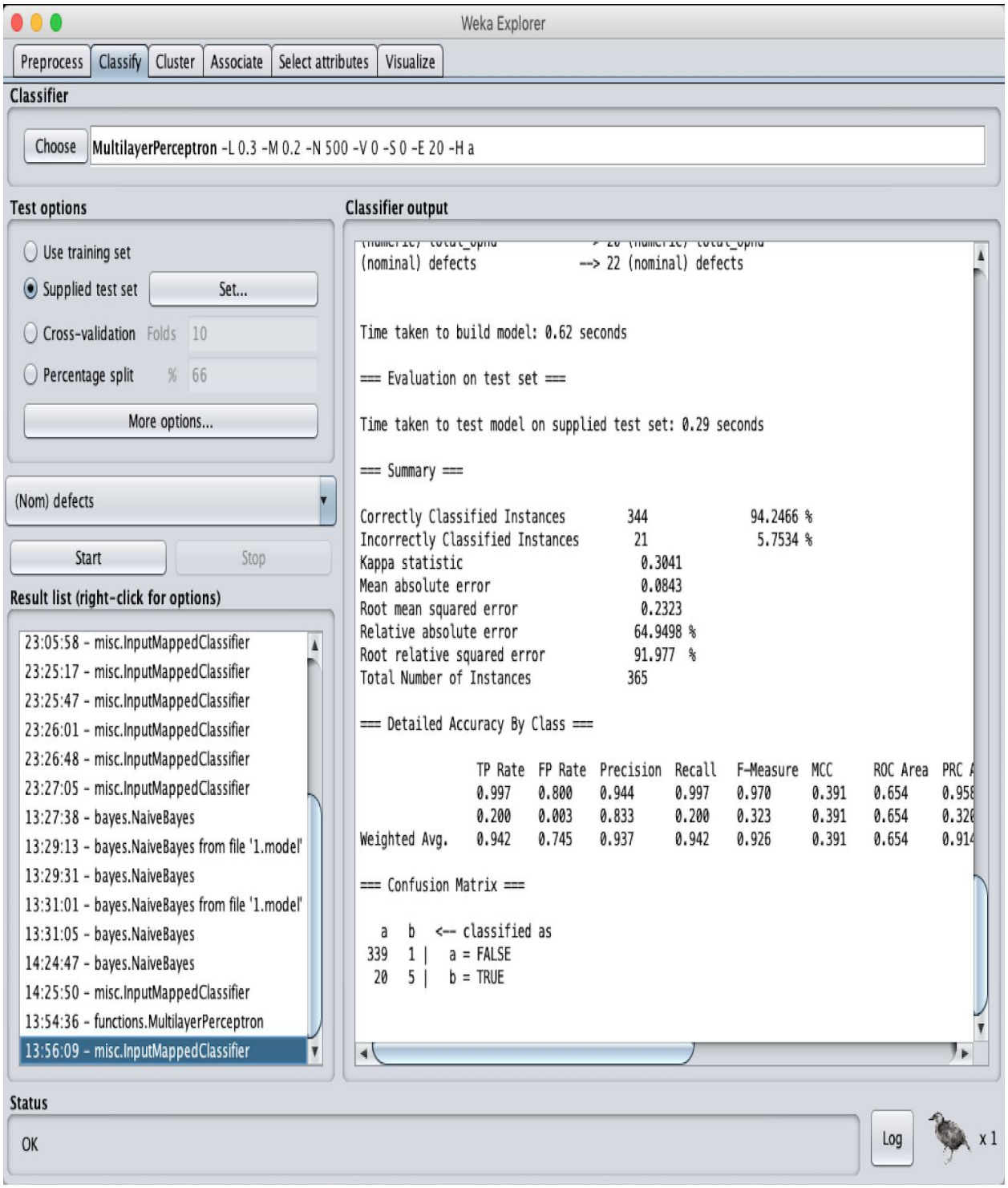
WraperSubsetEval

Multilayer Perceptron

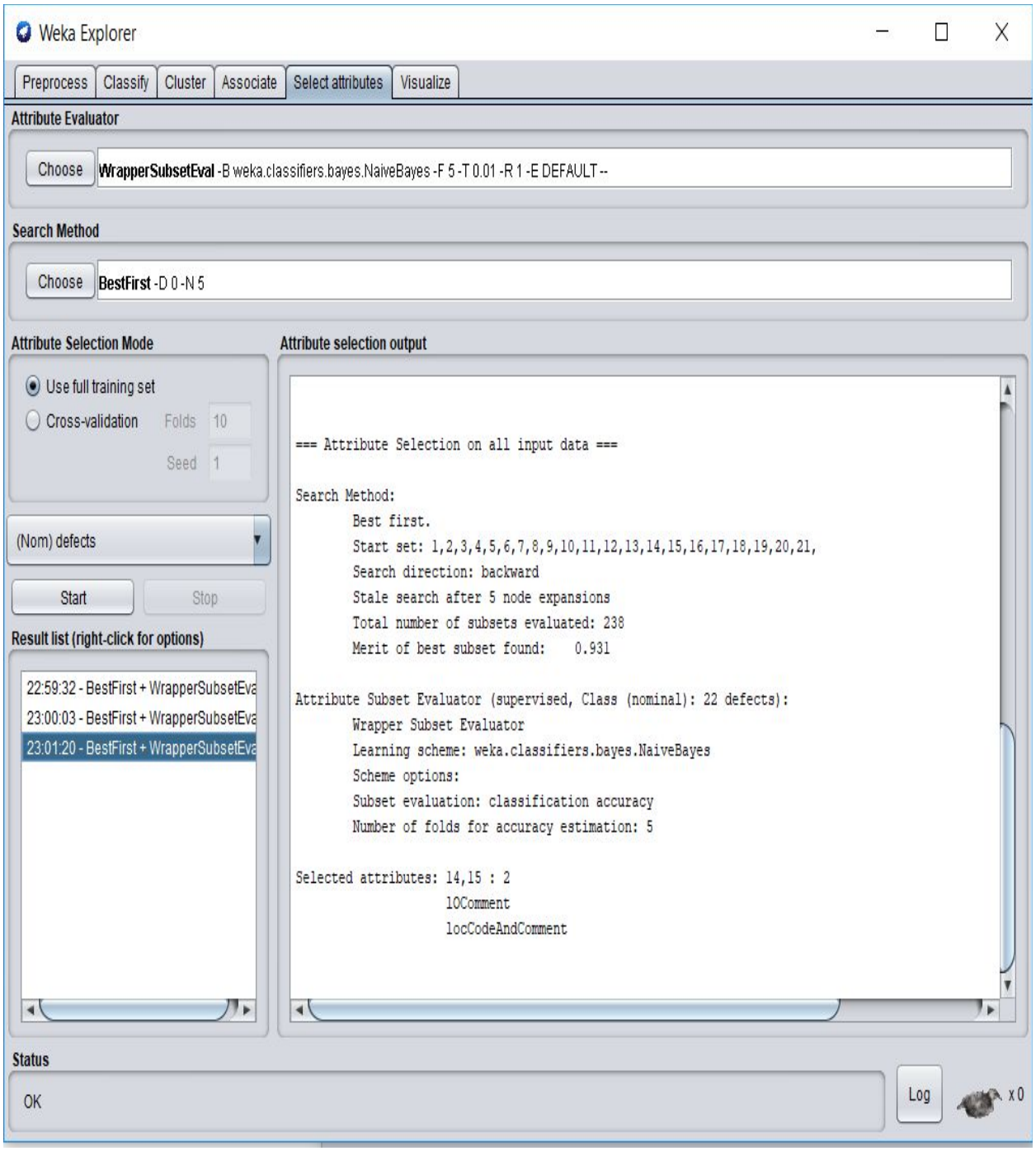


Attributes Selected: v.g., ev.g., iv.G., N, V, D, I, lOComent, locCodeAndComment, unique_Op and total_Opnd.

Multilayer Perceptron Classifier

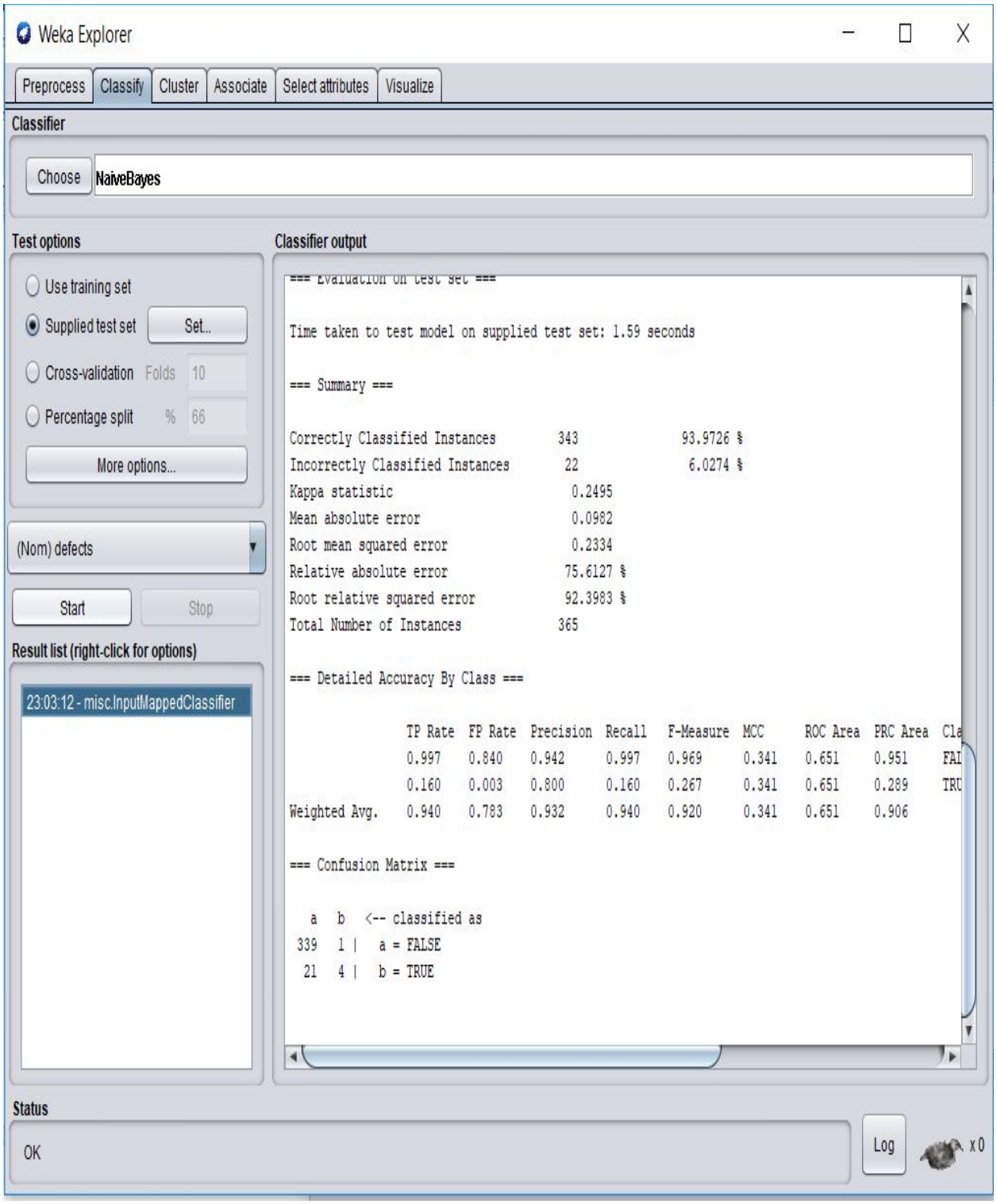


Naïve Bayes

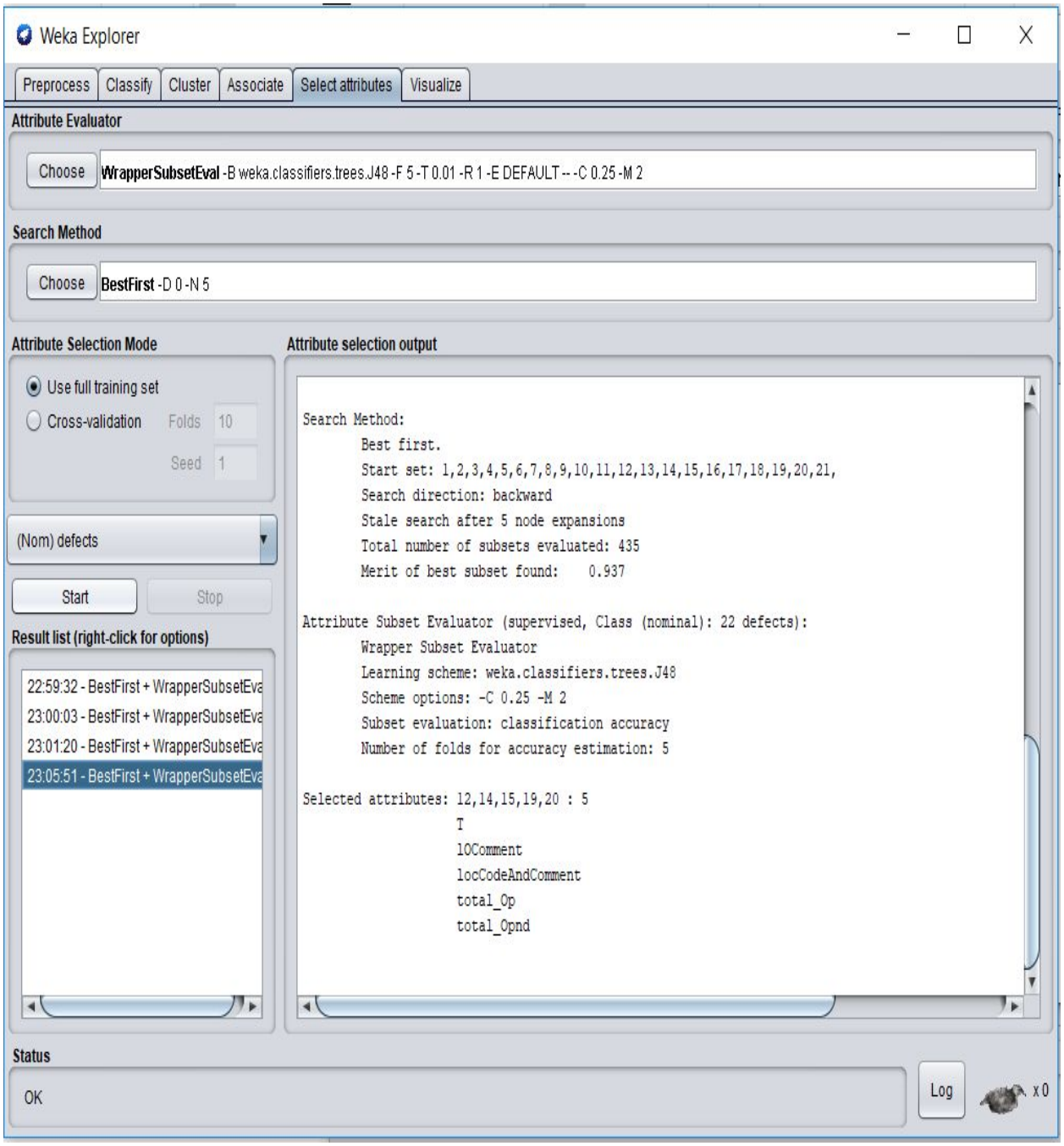


Attributes Selected: lOComment and locCodeAndComment

Naïve Bayes Classifier

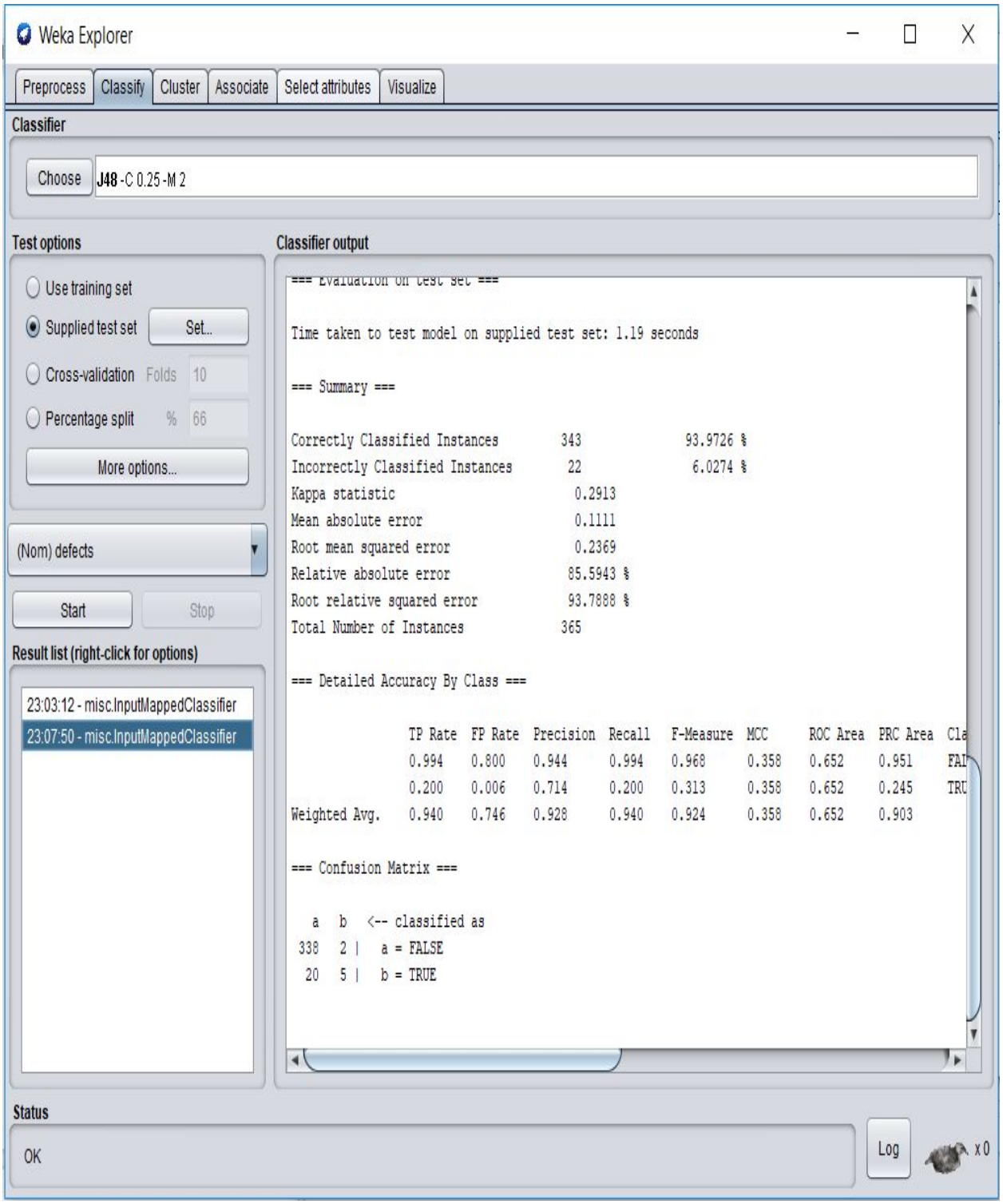


J48

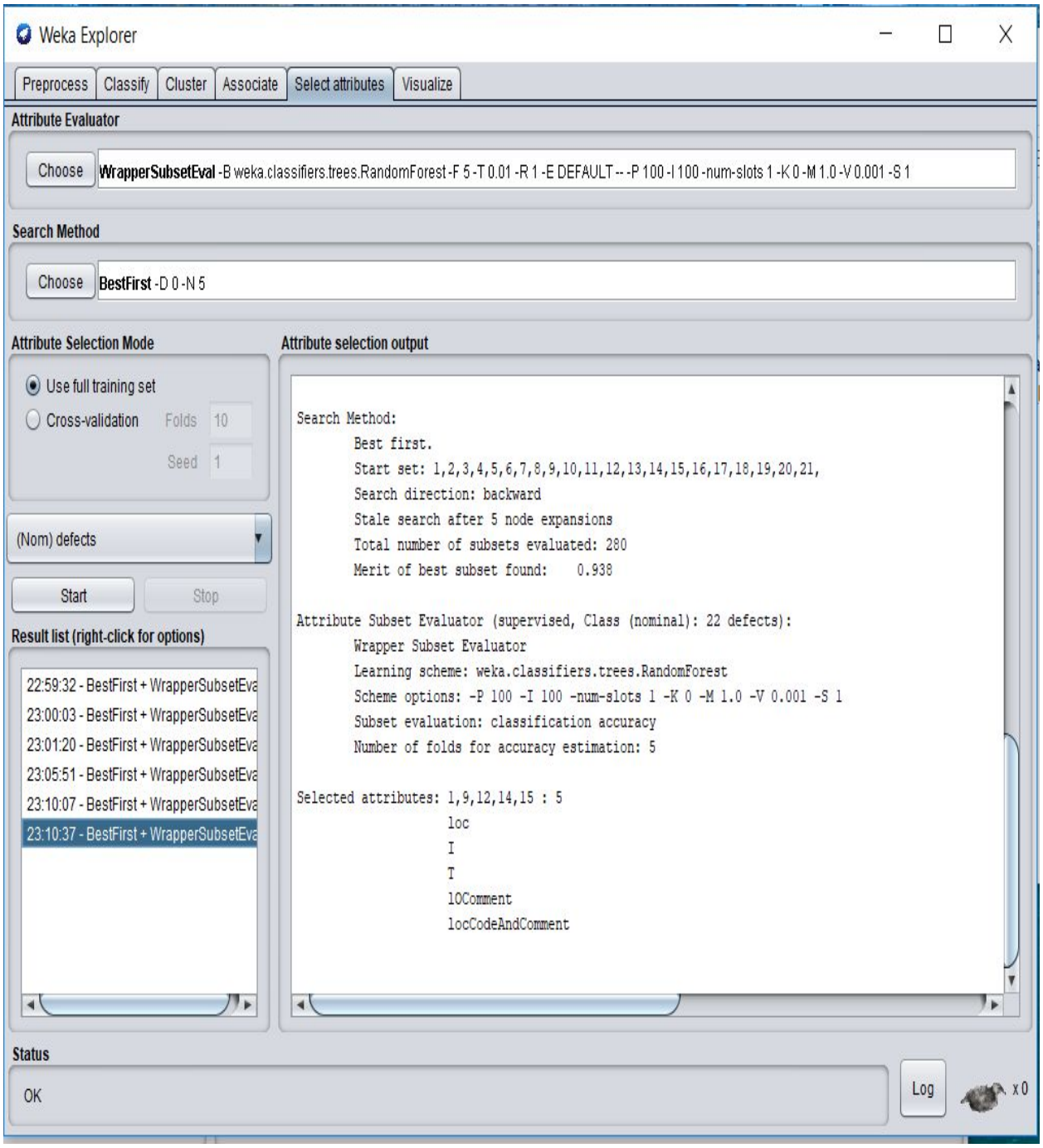


Attributes Selected: lOComment, locCodeAndComment, total_Op, total_Opnd

J48 Classifier

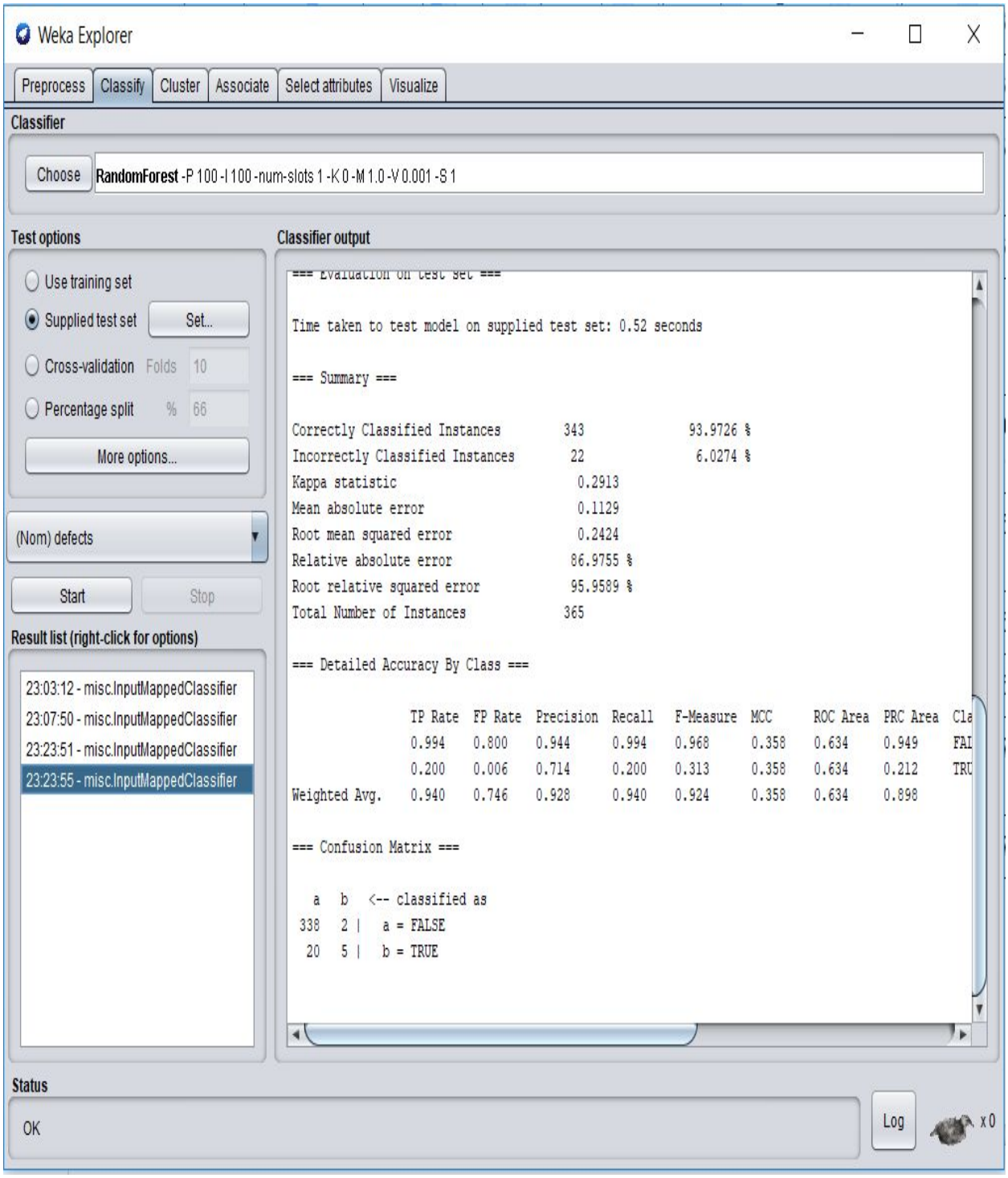


Random Forest

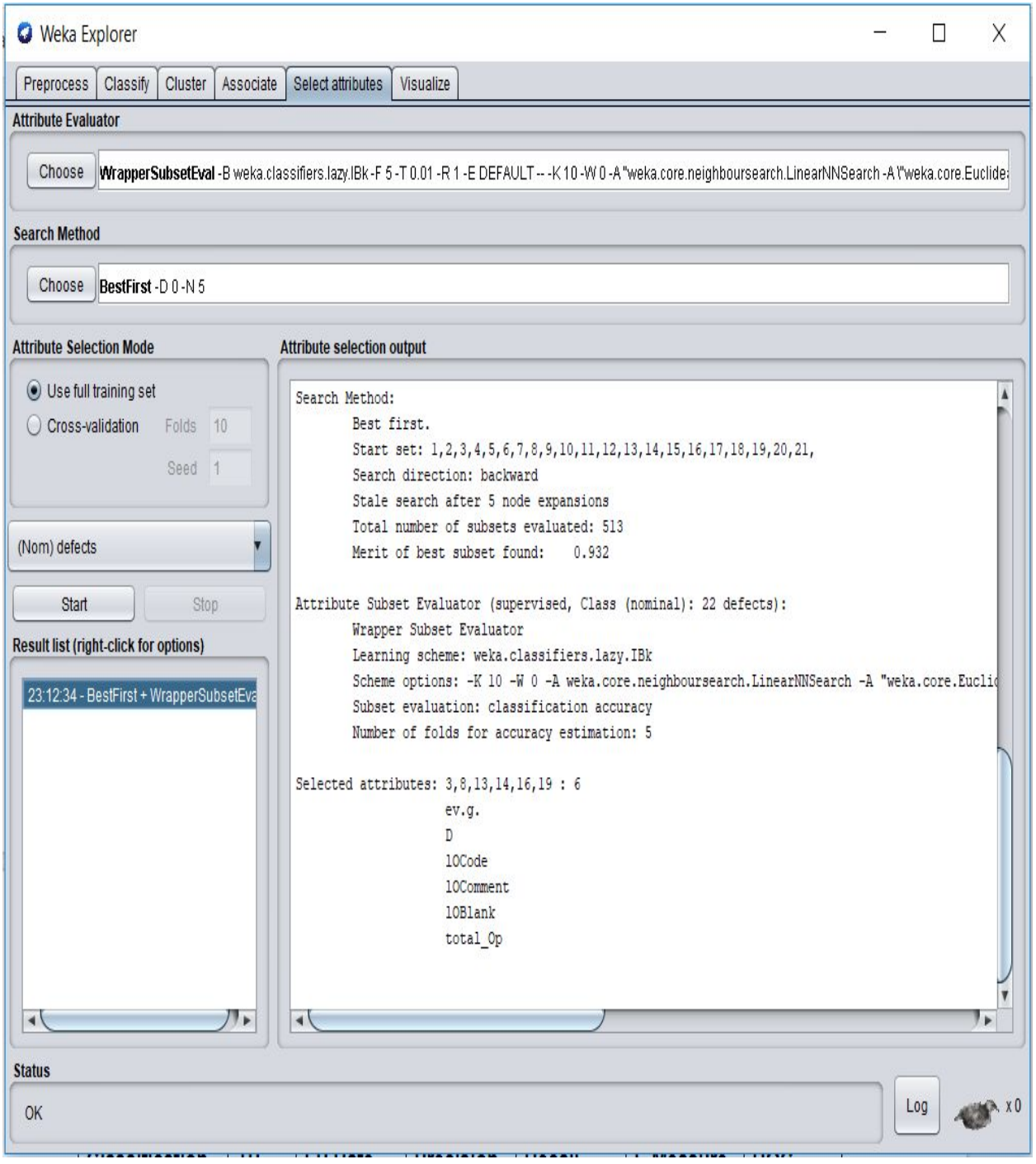


Attributes Selected: lOComment, locCodeAndComment, loc, I and T

Random Forest Classifier

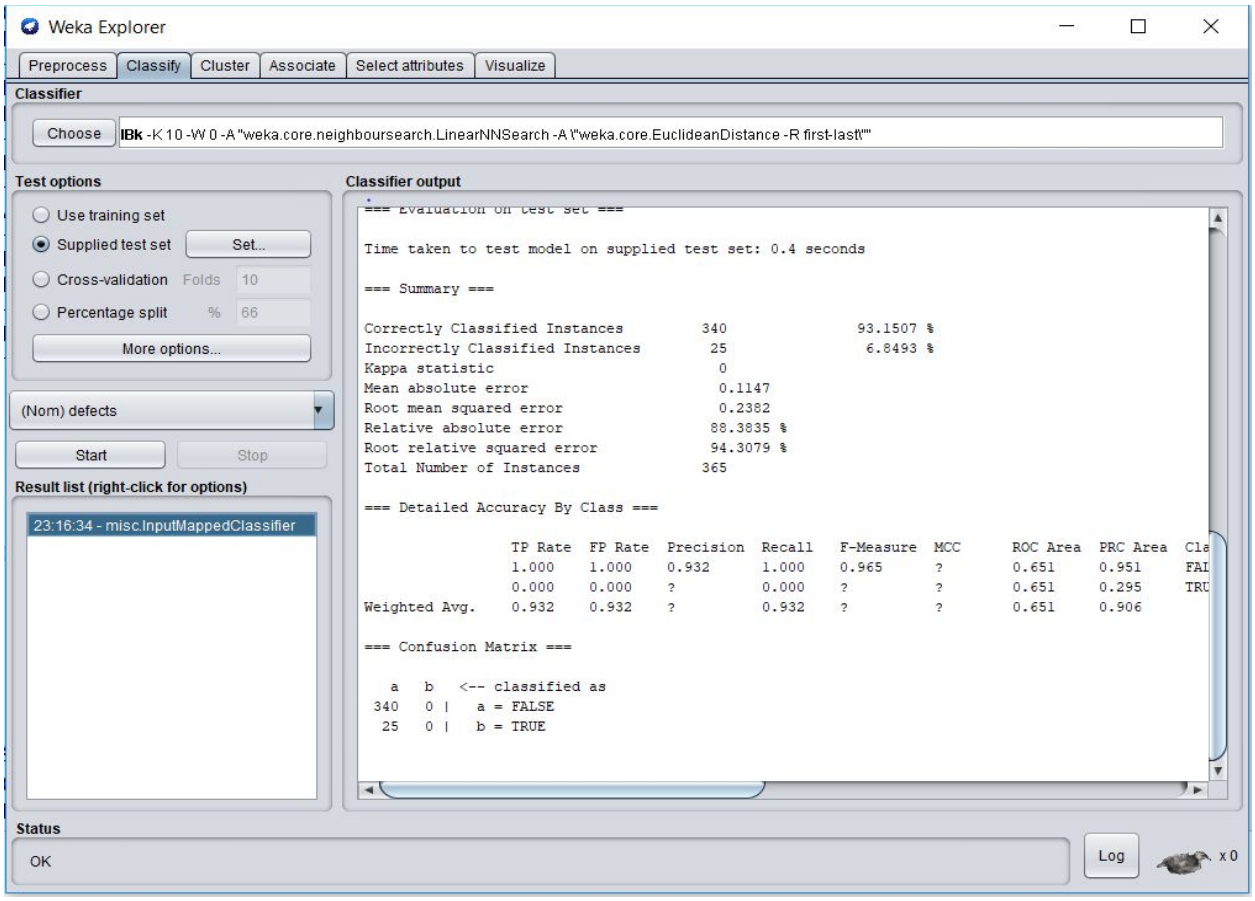


Instance-Based KNN (K = 10)



Attributes Selected: ev.g., D, loCode, lOComment, lOBlank and total_Op

Instance-Based KNN (K = 10) Classifier



Classification Model	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Accuracy
Multilayer Perceptron	0.942	0.745	0.937	0.942	0.926	0.654	94.25
Naive Bayes	0.94	0.783	0.932	0.94	0.92	0.651	93.97
J48	0.94	0.746	0.928	0.94	0.924	0.652	93.97
Random Forest	0.94	0.746	0.928	0.94	0924	0.634	93.97
IBK (K=10)	0.932	0.932	ud	0.932	ud	0.651	93.15

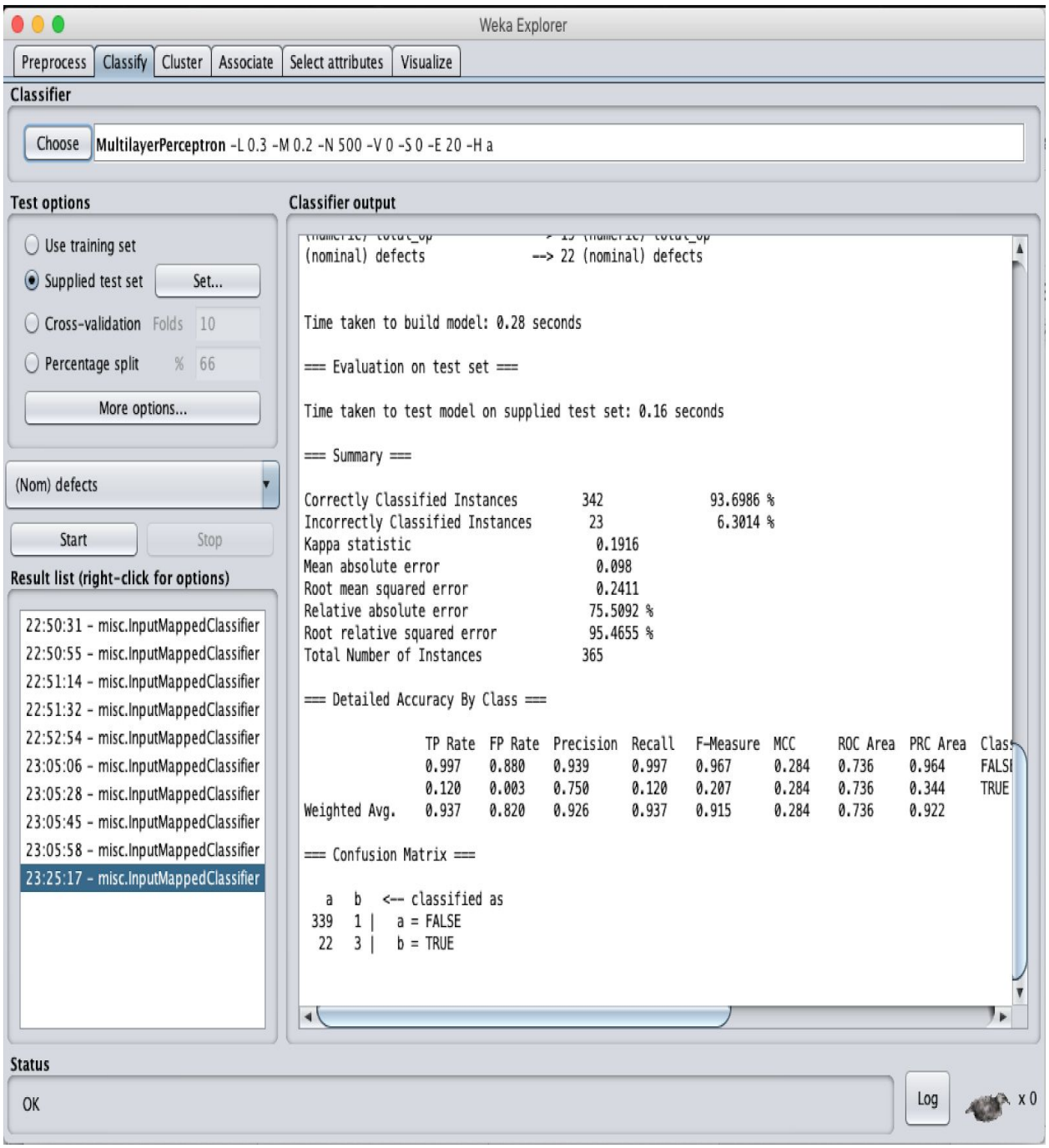
The value of Precision and F-Measure is undefined (ud) since the value of TP and FP for IBK classifier is 0.

Note:- Highlighted model gives the best results for this Attribute selection Algorithm.

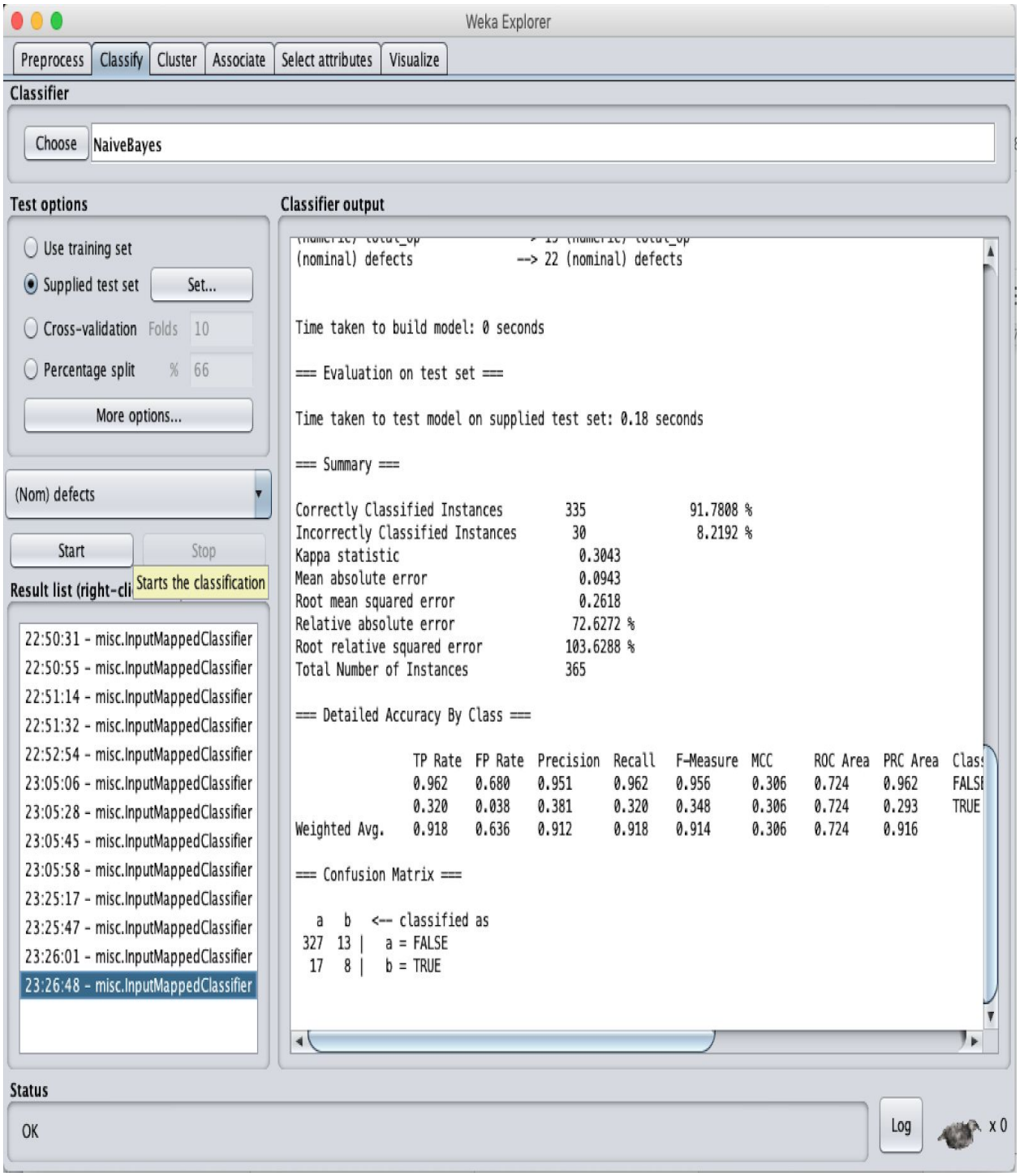
AttributeSelectedByJudgement

Attributes selected: I, lOComment, locCodeandComent, Unique_Op, Total_Op

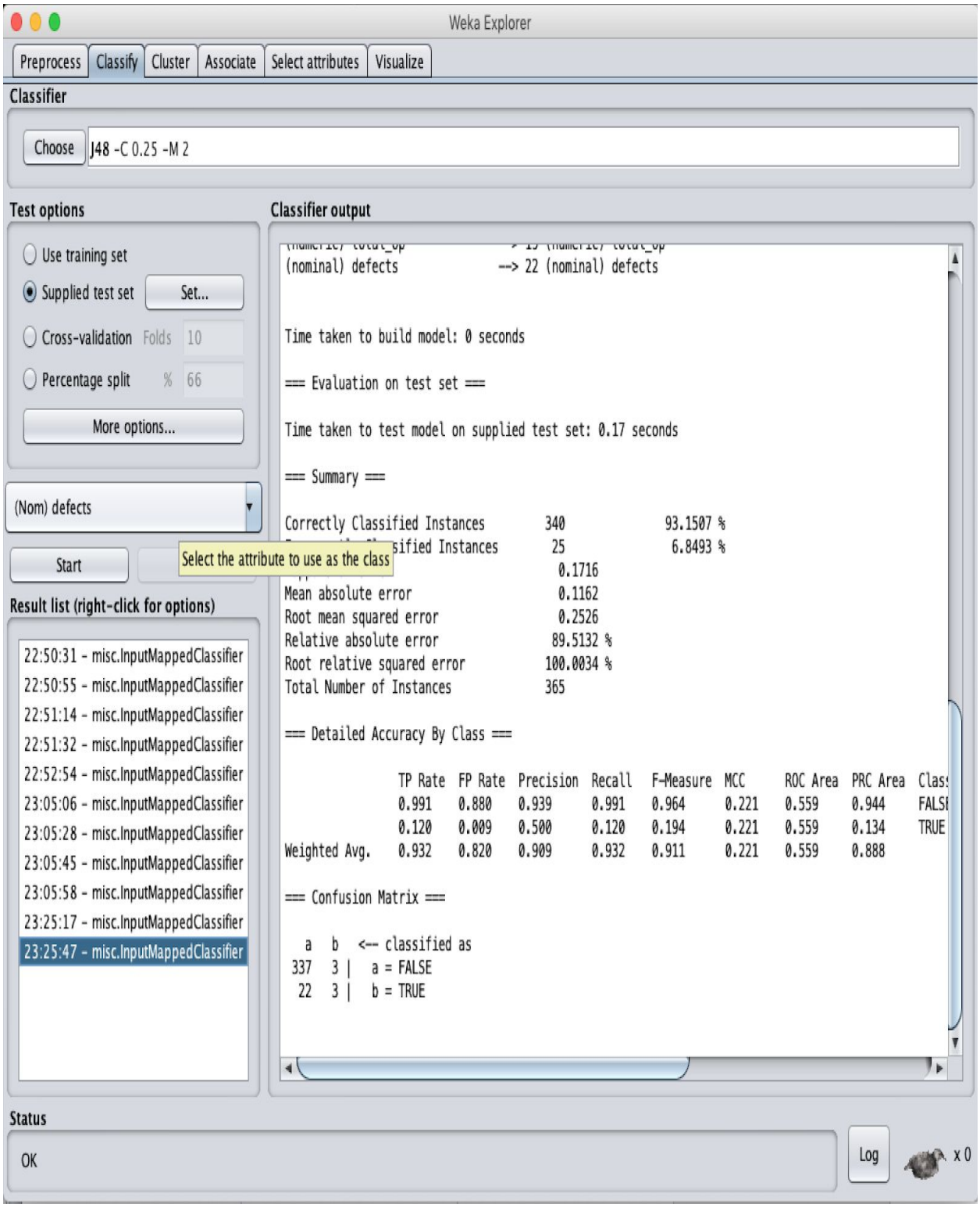
Multilayer Perceptron



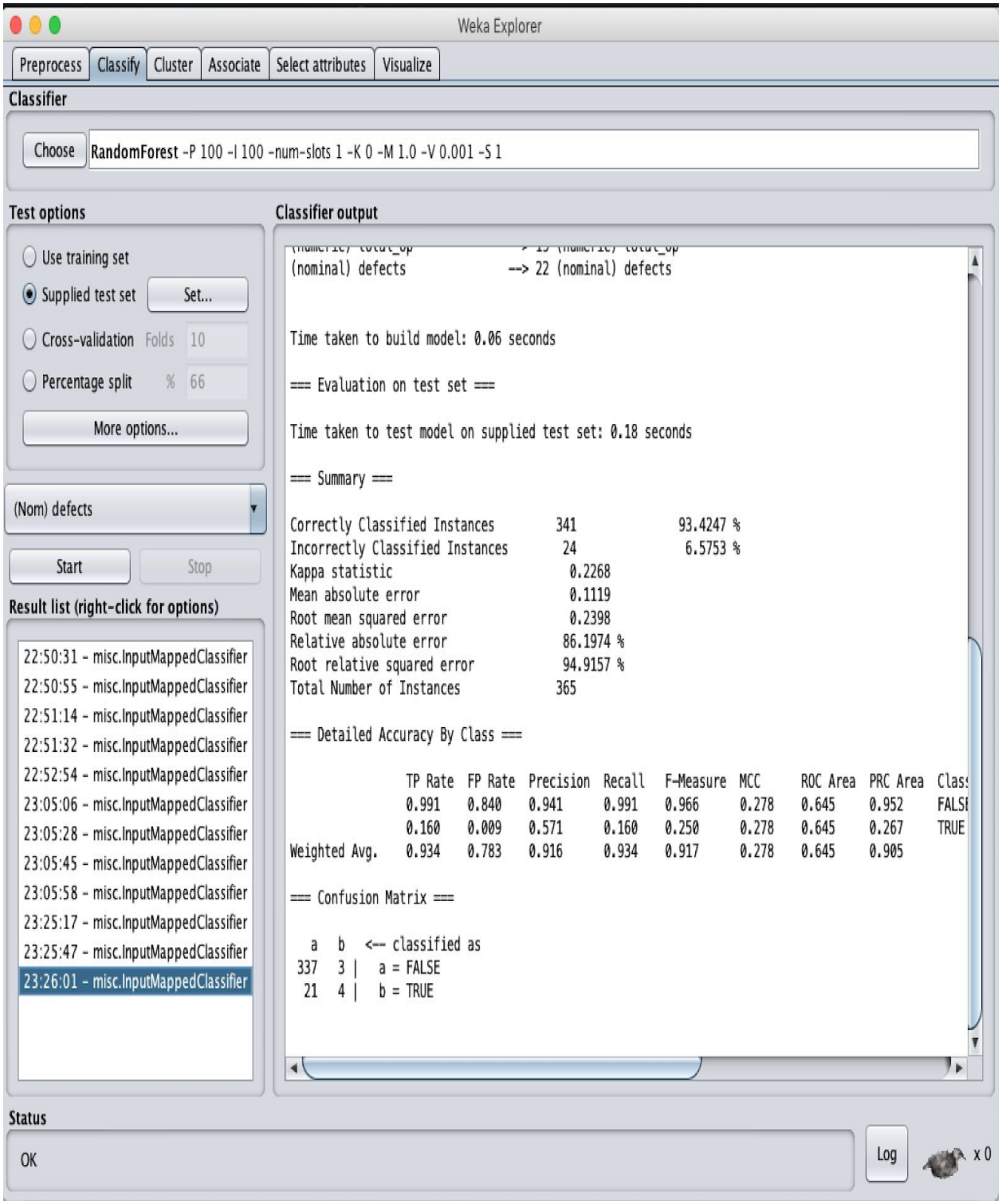
Naïve Bayes


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) defects
Start
Stop
Starts the classification
Result list (right-cli
22:50:31 - misc.InputMappedClassifier
22:50:55 - misc.InputMappedClassifier
22:51:14 - misc.InputMappedClassifier
22:51:32 - misc.InputMappedClassifier
22:52:54 - misc.InputMappedClassifier
23:05:06 - misc.InputMappedClassifier
23:05:28 - misc.InputMappedClassifier
23:05:45 - misc.InputMappedClassifier
23:05:58 - misc.InputMappedClassifier
23:25:17 - misc.InputMappedClassifier
23:25:47 - misc.InputMappedClassifier
23:26:01 - misc.InputMappedClassifier
23:26:48 - misc.InputMappedClassifier
Classifier output
(nominal) defects --> 22 (nominal) defects
Time taken to build model: 0 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.18 seconds
=== Summary ===
Correctly Classified Instances 335 91.7808 %
Incorrectly Classified Instances 30 8.2192 %
Kappa statistic 0.3043
Mean absolute error 0.0943
Root mean squared error 0.2618
Relative absolute error 72.6272 %
Root relative squared error 103.6288 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.962 0.680 0.951 0.962 0.956 0.306 0.724 0.962 FALSE
0.320 0.038 0.381 0.320 0.348 0.306 0.724 0.293 TRUE
Weighted Avg. 0.918 0.636 0.912 0.918 0.914 0.306 0.724 0.916
=== Confusion Matrix ===
a b <-- classified as
327 13 | a = FALSE
17 8 | b = TRUE
Status
OK
Log
x 0

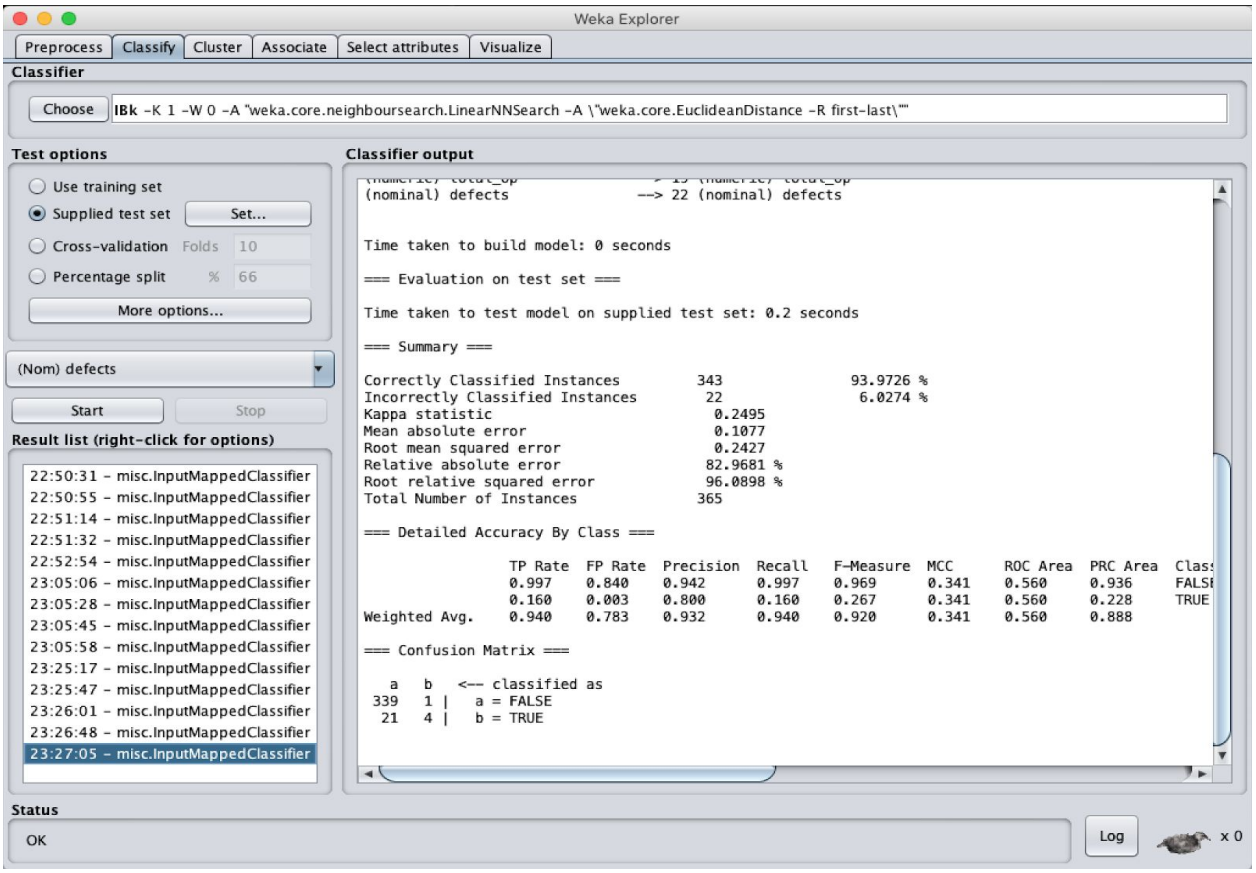
J48


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
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) defects
Start
Select the attribute to use as the class
Result list (right-click for options)
22:50:31 - misc.InputMappedClassifier
22:50:55 - misc.InputMappedClassifier
22:51:14 - misc.InputMappedClassifier
22:51:32 - misc.InputMappedClassifier
22:52:54 - misc.InputMappedClassifier
23:05:06 - misc.InputMappedClassifier
23:05:28 - misc.InputMappedClassifier
23:05:45 - misc.InputMappedClassifier
23:05:58 - misc.InputMappedClassifier
23:25:17 - misc.InputMappedClassifier
23:25:47 - misc.InputMappedClassifier
Classifier output
(nominal) defects --> 22 (nominal) defects
Time taken to build model: 0 seconds
=== Evaluation on test set ===
Time taken to test model on supplied test set: 0.17 seconds
=== Summary ===
Correctly Classified Instances 340 93.1507 %
25 6.8493 %
0.1716
Mean absolute error 0.1162
Root mean squared error 0.2526
Relative absolute error 89.5132 %
Root relative squared error 100.0034 %
Total Number of Instances 365
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area
0.991 0.880 0.939 0.991 0.964 0.221 0.559 0.944 FALSE
0.120 0.009 0.500 0.120 0.194 0.221 0.559 0.134 TRUE
Weighted Avg. 0.932 0.820 0.909 0.932 0.911 0.221 0.559 0.888
=== Confusion Matrix ===
a b <-- classified as
337 3 | a = FALSE
22 3 | b = TRUE
Status
OK
Log
x 0

Random Forest



Instance-Based KNN (K = 10)



Classification Model	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Accuracy
Multilayer Perceptron	0.937	0.82	0.926	0.937	0.915	0.736	93.69
Naive Bayes	0.918	0.636	0.912	0.918	0.914	0.724	91.79
J48	0.932	0.820	0.909	0.932	0.911	0.559	93.15
Random Forest	0.934	0.783	0.916	0.934	0.917	0.645	93.43
IBK (K=10)	0.940	0.783	0.932	0.940	0.920	0.560	93.97

Note:- Highlighted model gives the best results for this Attribute selection Algorithm.

Best Model and Justification

Table of all the best models for each attribute selection algorithm

Attribute Selection Algorithm	Classifier Algorithm	TP	FP	Precision	Recall	F	ROC Area	Accuracy
Gain Ratio	Random Forest	0.942	0.671	0.932	0.942	0.931	0.666	94.25
SymetricalUncertSubsetAttributeEval	*IBK (K=10)*	*0.942*	*0.782*	*0.946*	*0.942*	*0.922*	*0.662*	*94.25*
CfsSubsetEval	Multilayer Perceptron	0.937	0.783	0.923	0.937	0.919	0.671	93.69
WrapperSubsetEval	Multilayer Perceptron	0.942	0.745	0.937	0.942	0.926	0.654	94.25
AttributeSelectedByJudgement	Random Forest	0.934	0.783	0.916	0.934	0.917	0.645	93.43

As seen in the above table 3 models have the same accuracy of 94.25%.

- IBK(k=10) using SymetricalUncertAttributeEval as attribute selection algorithm.
- Multilayer Perceptron using WrapperSubsetEval as attribute selection algorithm.
- Random Forest using InfoGain Ratio as attribute selection algorithm.

Of all the 5 models we can consider **Random Forest using InfoGain Ratio** to be the **best model** since it has the least FP rate and the highest ROC area.

Conclusion

From this project we mainly understood the following:

- Gian Ratio is a better attribute selection algorithm when compared to Information Gain while building Decision Tree classification models.
- We also learned that Neural Network algorithms can be used as classifiers since they have a high classification accuracy.
- The Instance Base KNN Algorithm has a high accuracy for this particular dataset however, it requires a lot of trial and error to determine the optimal value of K.
- After consideration of all the above factors and different performance measures we conclude that the Random Forest Classification algorithm coupled with the Gain Ratio Attribute Selection Algorithm gives us the best classification model for this particular dataset.

Work done by each team member

Kevin Rodrigues: Binning of data, Building and Testing of Models, Documentation.

Paritosh Shirodkar: Splitting of data into training and testing set, Building and Testing of Models, Documentation.

References

- Han, J., Kamber, M., Pei, J., “Data mining: concepts and techniques,” 3rd Ed., Morgan Kaufmann, 2012
- http://www.aaai.org/Papers/KDD/1995/KDD95-049.pdf
- http://weka.sourceforge.net/doc.dev/weka/attributeSelection/SymmetricalUncertAttributeEval.html
- http://weka.sourceforge.net/doc.dev/weka/attributeSelection/CfsSubsetEval.html
- https://skymind.ai/wiki/multilayer-perceptron
- https://hackerbits.com/data/c4-5-data-mining-algorithm/
- http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics
- http://sciencewise.info/resource/Ibk_algorithm/Ibk_algorithm_by_Wikipedia
- http://www.ques10.com/p/166/write-short-note-on-bayesian-classification/

Additional Material

All the resources used for this project including the original dataset, R-Script used for binning the dataset, training, and test set after splitting, research papers referred, etc can be found on our GitHub repositories:

https://github.com/paritoshshirodkar/PC1-Software-Defect-Classification

https://github.com/kevinrodrigues13/PC1-Software-Defect-Classification