CS699 Data Mining
Fall 2020

Classification of Trump Administration's response to the COVID-19 outbreak

Jaehack Jeong
Ghazaleh Jabbariarfaei

Table of Contents

Statement of Data Mining Goal ---------- 3

Detailed Description of the Dataset ---------- 3

Attribute Selection and Algorithms for the Project ---------- 5

Data Mining Procedure ---------- 5

- *Preprocessing ---------- 5*
- *Splitting Dataset ---------- 6*
- *Attribute Selection Method ---------- 8*

Data Mining Result and Evaluation ---------- 12

- *Performance Comparison ---------- 63*
- *Discussion and Conclusion ---------- 70*

Statement of Data Mining Goal

The purpose of the project is developing successful classifier models that forecast whether or not the survey participants approve of the Trump administration's response to the outbreak of COVID-19 based on the background information of the participants and their answers to the survey questions.

The survey was conducted by ABC News in May 2020, which is the time that had been only a few months since the outbreak of virus. At that time, most of the states were experiencing the economic shut-down and chaos due to the lack of information of COVID-19.

Detailed Description of the Dataset

Title: ABC News/Ipsos Poll: 2020 Coronavirus Wave 8
https://ropercenter.cornell.edu/ipoll/study/31117363

The survey asks 6 questions about the Trump Administration's response to the COVID-19 outbreak along with its impacts, which are presented as below.

- **Q1: Do you approve or disapprove of the way Donald Trump is handling the response to the coronavirus? (Class Attribute)**
    - Approve or Disapprove
- **Q2: How concerned are you that you or someone you know will be infected with the coronavirus?**
    - Very concerned, Somewhat concerned, Not so concerned, Not concerned at all
- **Q3: Which of the following statements comes closest to your point of view?**
    - Opening the country now is worth it because it will keep economic damage to a minimum vs. Opening the country now is not worth it because it will mean more lives being lost
- **Q4: If a safe and effective coronavirus vaccine is developed, how likely would you be to get vaccinated?**
    - Very likely, Somewhat likely, Not so likely, Not likely at all
- **Q5a: Work Situation**
    - I am still working from my regular workplace, outside the home, Because of the coronavirus I am working from home instead of from my regular workplace, I normally work from home, I am not currently employed

- **Q5b: Whether lost job due to COVID-19 outbreak**
    - Yes or No

In addition to the questions, the dataset also contains background information of participants. The attributes are described as below:

1. id = participant's id
2. xspanish = participant's language (either English or Spanish)
3. complete_status = indicating whether participants fully completed the survey
4. ppage = age of participants
5. ppeduc = education of participants (from the 1st grade to doctorate degree)
6. ppeducat = education levels of participants (less than high school, high school, some college, bachelor's degree or higher)
7. ppgender = Gender of the participant (Male/Female)
8. ppethm = Race of the participant
9. pphhsize = household size of participants
10. pphouse = a brief description of house of participants
11. ppincimp = participant's income
12. ppmarit = marital status of participants
13. ppmsacat = indicating whether participants live in metro area or non-metro area
14. ppreg4 = region of participants
15. pprent = participants house rent
16. ppstaten = state where participants live
17. Q1
18. Q2
19. Q3
20. Q4
21. Q5a
22. Q5b
23. QPID = political party that participants support
24. POLIT200 = indicating whether participants are registered to vote at their current address
25. ABCAGE = The participants age range
26. Weights_pid = The study dataset(s) contain weight factors that should be employed in any data analysis. Typically, weights are used in an attempt to assure that the survey sample more accurately represents the population.

Attribute Selection and Algorithms for the Project

Attribute selection methods implemented in Weka:

1. 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. This evaluator gives us subsets of features that are highly correlated with the class (Q1) while having low intercorrelation.
2. GainRatioAttributeEval - Evaluates the worth of an attribute by measuring the gain ratio with respect to the class.
3. InfoGainAttributeEval - Evaluates the worth of an attribute by measuring the information gain with respect to the class.
4. CorrelationAttributeEval - Evaluates the worth of an attribute by measuring the correlation (Pearson's) between it and the class. Nominal attributes are considered on a value by value basis by treating each value as an indicator. An overall correlation for a nominal attribute is arrived at via a weighted average.

Classification algorithms used in this project:

1. Naive Bayes
2. J48
3. RandomForest
4. MultilayerPerceptron
5. IBK

Data Mining Procedure

Preprocessing

The original dataset is a csv file. In order to load the file in Weka, the following changes are needed.

- In the “ppeduc” column of the data set, there were two types of value “Bachelor’s” and “Master’s” that needed to be transformed into “Bachelors”, “Masters” respectively, since the apostrophe bothers it.
- In the “Q1” column, which is the class attribute, there are two ‘Skipped’ answers. We have removed the entire row corresponding to these values.
- We have also applied 3 filters to our data set

1) Filter -> NumericToNominal: to convert any categorical attribute that may appear as Numeric back to Nominal.
2) Filter -> StringToNominal with attributeRange 9 (pphhsize)
3) Filter -> StringToNominal with attributeRange 25 (ABCAGE)

- Finally, we have removed three attributes from our list of attributes ID, weights_pid and complete_status. We did not need the IDs of the participants in our project, and Complete_Status had only one value therefore was non-effective.

Splitting Dataset

After preprocessing the data set, we used R to split the dataset into the train and test datasets while remaining the distribution of the class attribute in the original dataset in order to prevent any imbalance issue.

The distribution of the class label in the original dataset

Q1	
Approved	Disapproved
237	293

Screenshot of the codes used for this process

```r
#load RWeka library to read the dataset
# load caret library to split the dataset
library(RWeka)
library(caret)

#read preprocessed dataset
data <- read.arff("preprocessed_survery.arff")

# make the class label as factor(categorical)
factor(data$Q1)

# the distribution of the class label
table(data$Q1)

# set the seed
set.seed(3456)

# creating the indexes for training dataset with respect to class distirbution
# split the dataset into 70% Training and 30% Test
trainIndex <- createDataPartition(data$Q1, p = .7,
                                  list = FALSE,  times = 1)

Train <- data[ trainIndex,]
Valid <- data[-trainIndex,]

# distribution of class in Train dataset
table(Train$Q1)
# distribution of class in Test dataset
table(Valid$Q1)

# save two datasets into arff files
write.arff(Train, file = "Initial_Training_set.arff")
write.arff(Valid, file = "Initial_Testing_set.arff")
```

The distribution of class label in the train dataset

Q1_Train	
Approved	Disapproved
166	206

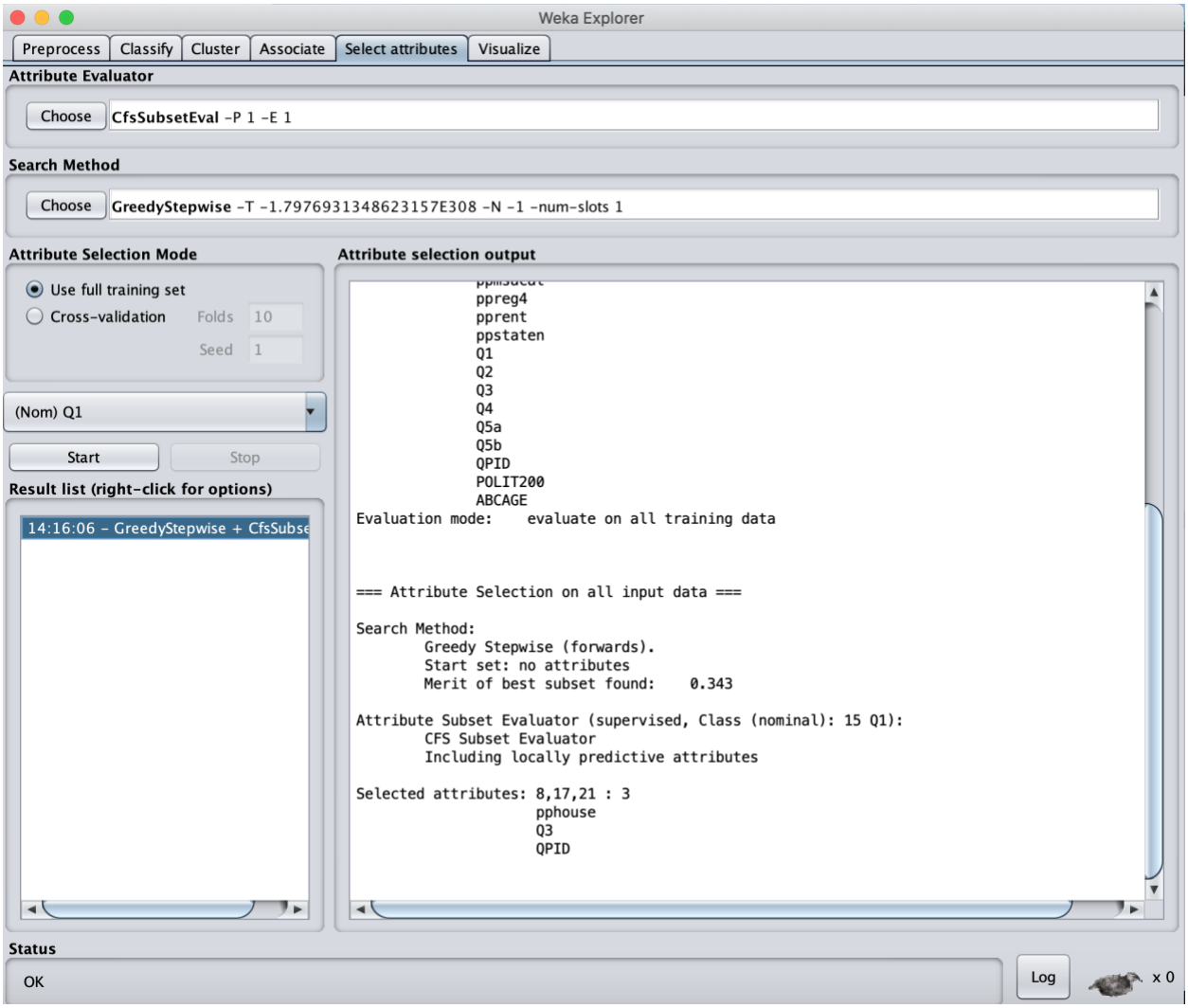
The distribution of class label in the test dataset

Q1_Test	
Approved	Disapproved
71	87

Attribute Selection Methods

With the newly created training and test dataset, we can begin with feature selection. We choose 4 attribute selection methods to select the best attributes from our 22 explanatory attributes.

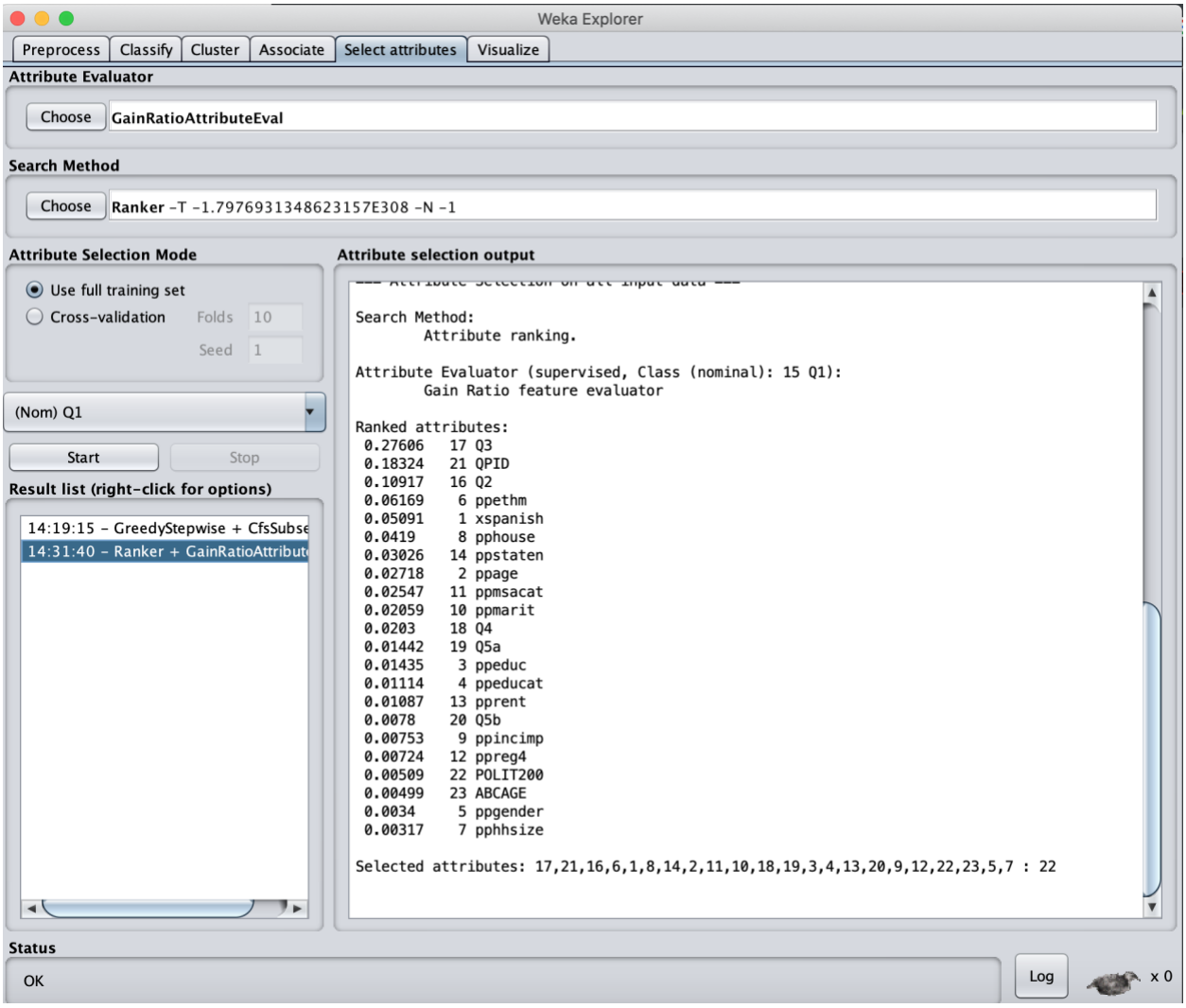
Attribute Evaluator #1: CfsSubsetEval
Search Method: GreedyStepwise
Attribute selection Mode: Use full training set



The result of this evaluator is this set of attributes: pphouse, Q3, QPID. For our first attribute sets we are using attributes selected by this method.

Attribute set 1: **[pphouse, Q3, QPID, Q1]**

Attribute Evaluator #2: GainRatioAttributeEval
Search Method: Ranker
Attribute selection Mode: Use full training set

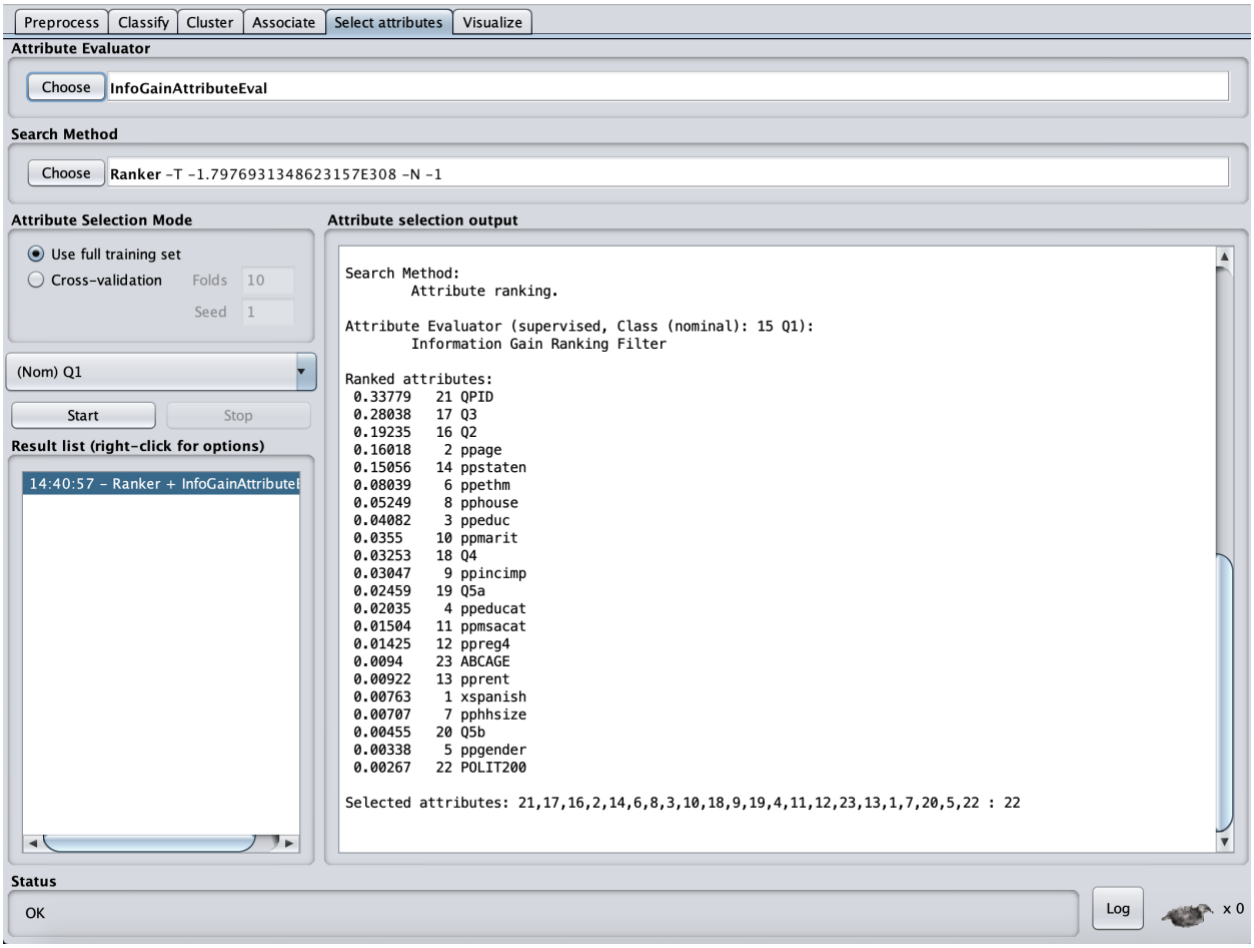


The result returns the ranking of attributes based on their gain ratio (#1 Q3, #2 QPID, #3 Q2, #4 ppethm and so on)

We have chosen the attributes with higher Gain Ratio for our second attribute set. Q3, QPID, and Q2 have the highest gain ratio. After Q2, the gain ratio drops drastically so we exclude the rest of the attributes.

Attribute set 2: **[Q3, QPID, Q2, Q1]**

Attribute Evaluator #3: InfoGainAttributeEval
Search Method: Ranker
Attribute selection Mode: Use full training set

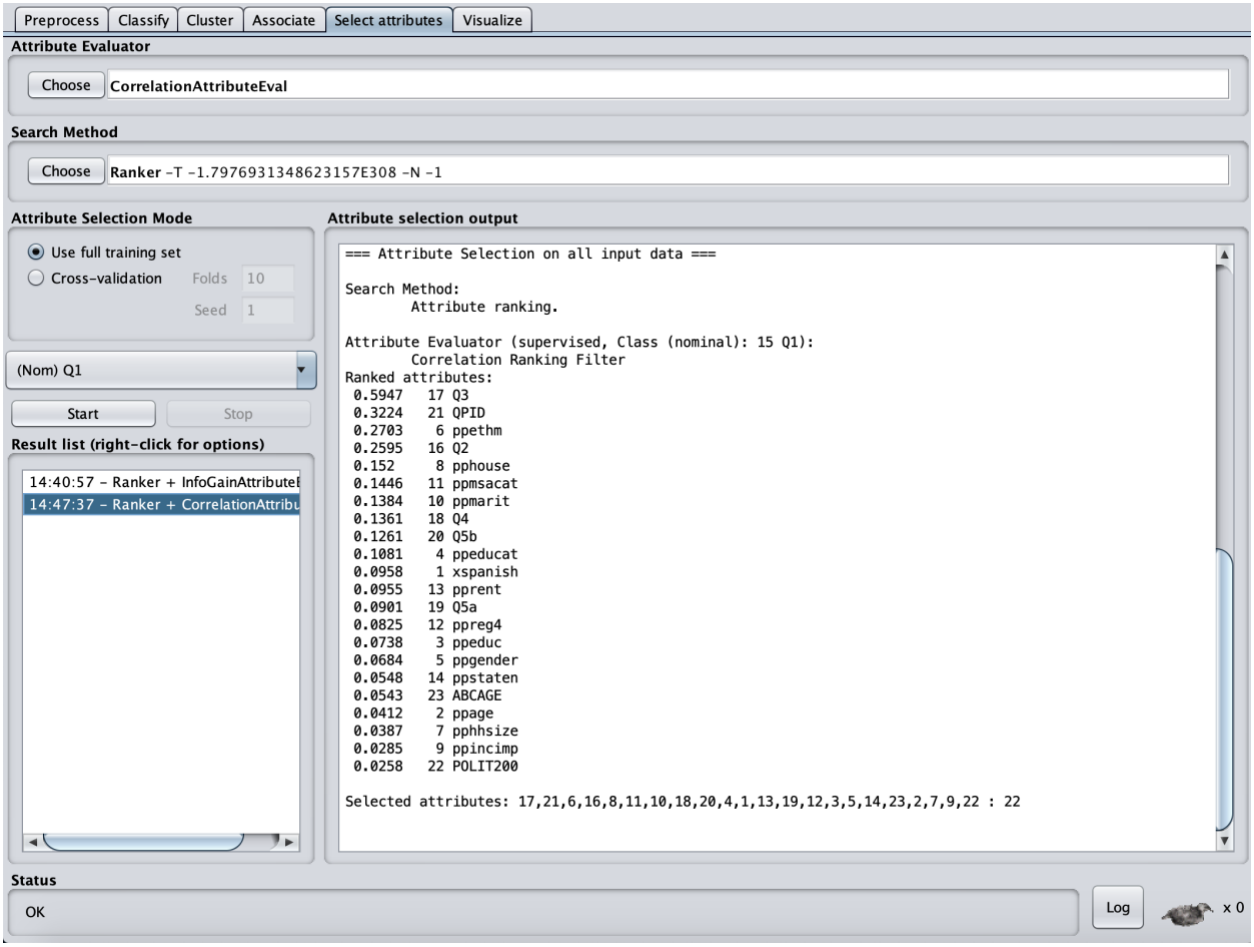


The result returns the ranking of attributes based on the information gain (#1 QPID, #2 Q3, #3 Q2, #4 ppage, #5 ppstaten and so on)

We have chosen the attributes with higher information gain for our third attribute set. QPID, Q3, Q2, ppage, and ppstaten have the highest information gain. After ppstaten, the information gain drops drastically so we exclude the rest of the attributes.

Attribute set 3: **[QPID, Q3, Q2, ppage, ppstaten, Q1]**

Attribute Evaluator #4: CorrelationAttributeEval
Search Method: Ranker
Attribute selection Mode: Use full training set



The result returns the ranking of attributes based on the correlation (#1 Q3, #2 QPID, #3 ppethm, #4 Q2 and so on)

As we can see the attributes with higher correlation to the class label are: Q3, QPID, ppethm, Q2, pphouse, ppmsacat, ppmarit, and Q4

Attribute set 4: **[Q3, QPID, ppethm, Q2, pphouse, ppmsacat, ppmarit, Q4, Q1]**

The final attribute set was picked by ourselves. We thought these attributes were interesting to perform our classification models on and compare the result with the other attribute sets selected by the methods shown above.

Attribute set 5: **[Q3, QPID, Q2, Q4, Q5a, Q5b, ppethm, ppmarit, ppeducat, ppgender, ABCAGE, Q1]**

Data Mining Result and Evaluation

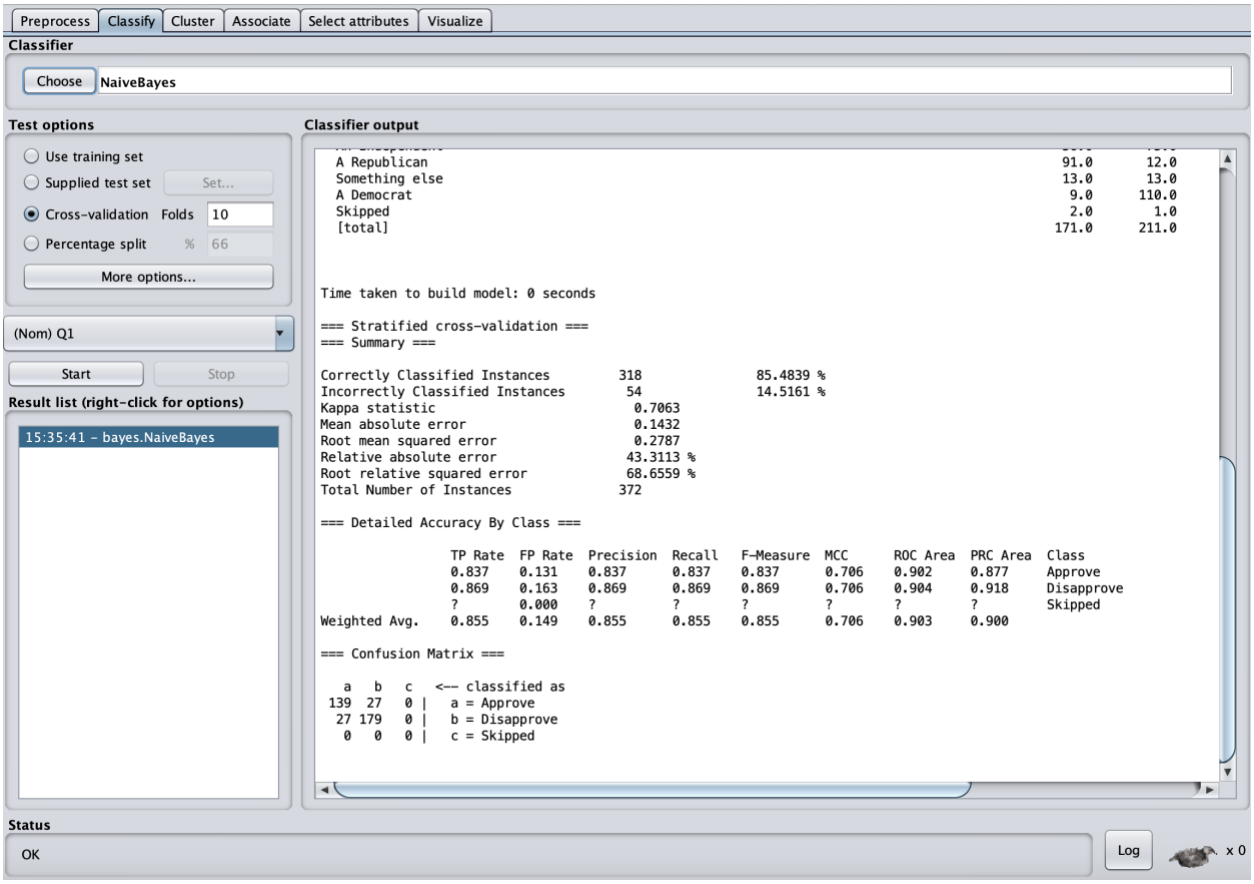
a. Attribute Set 1 [pphouse, Q3, QPID, Q1] - by the CfsSubsetEval

1. Naive Bayes

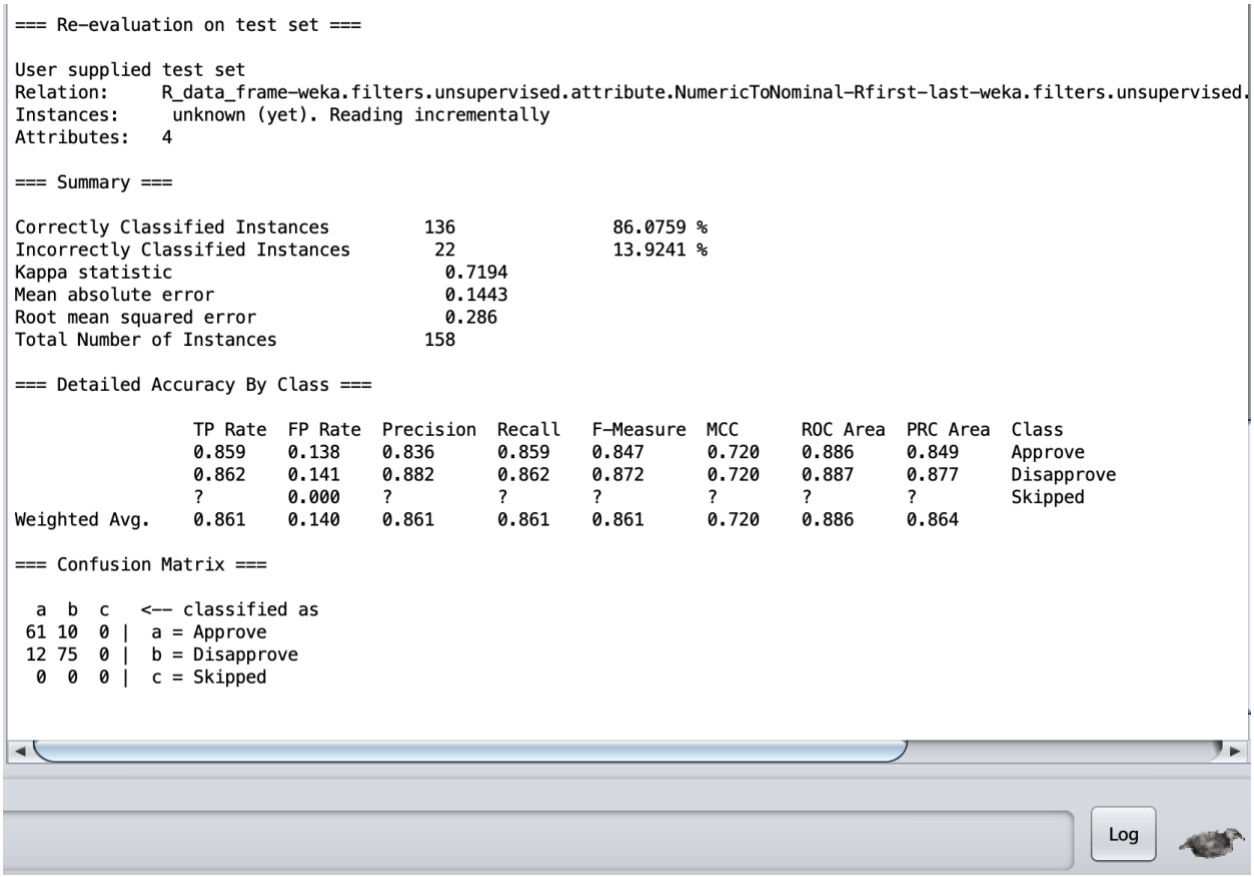
- Step 1: Open “Initial_Training_set.arff”, remove all attributes except pphouse, Q3, QPID, Q1 and save it as “set1_training”

- Step 2: Open “Initial_Testing_set.arff”, remove all attributes except pphouse, Q3, QPID, Q1 and save it as “set1_testing”

- Step 3: To assess how good Naive Bayse is, we used Cross-validation to build our model on our Train dataset. Open “set1_training”, Classify -> Choose NaiveBayes with Cross-validation (10 Folds), Q1 as class attribute -> Start



- Step 4: In order to test our model, we re-evaluate our model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set1_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

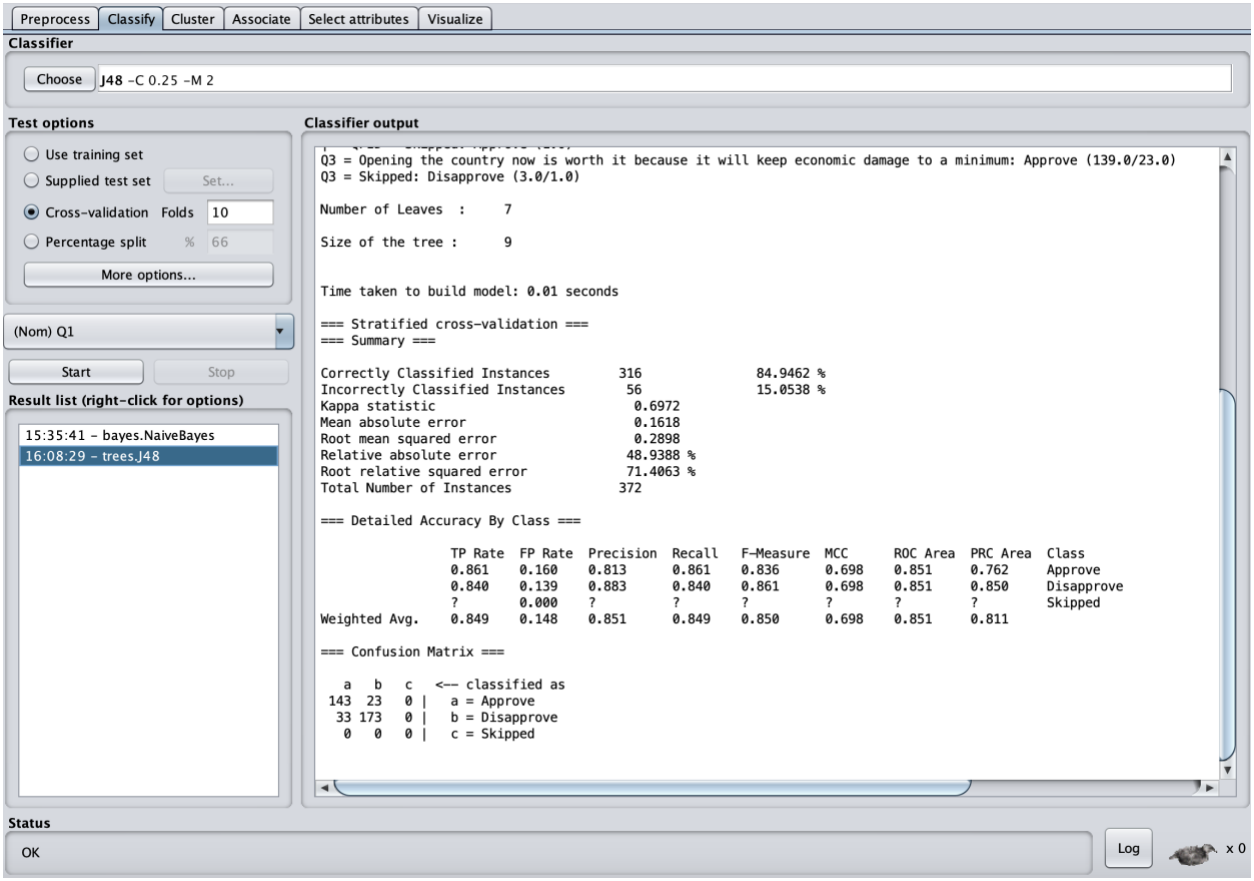


Naive Bayes - Cross Validation (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
85.48%	83.7%	13.1%	90.2%	Approve
	86.9%	16.3%	90.4%	Disapprove

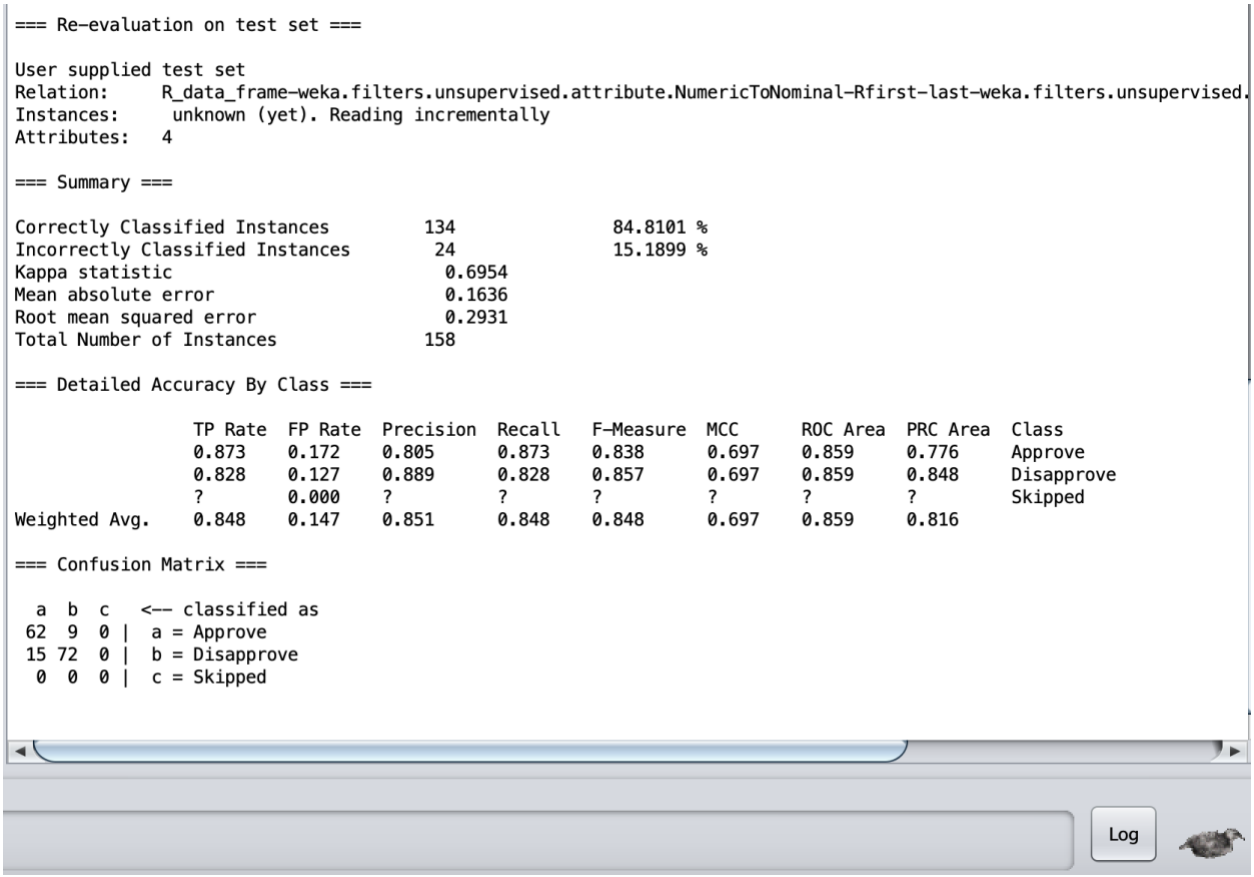
Naive Bayes - Test set (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
86.08%	0.859	0.138	0.886	Approve
	0.862	0.141	0.887	Disapprove

2. J48

- Step 1: First to assess how good J48 is, use Cross-validation to build our model. Choose J48 and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set1_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

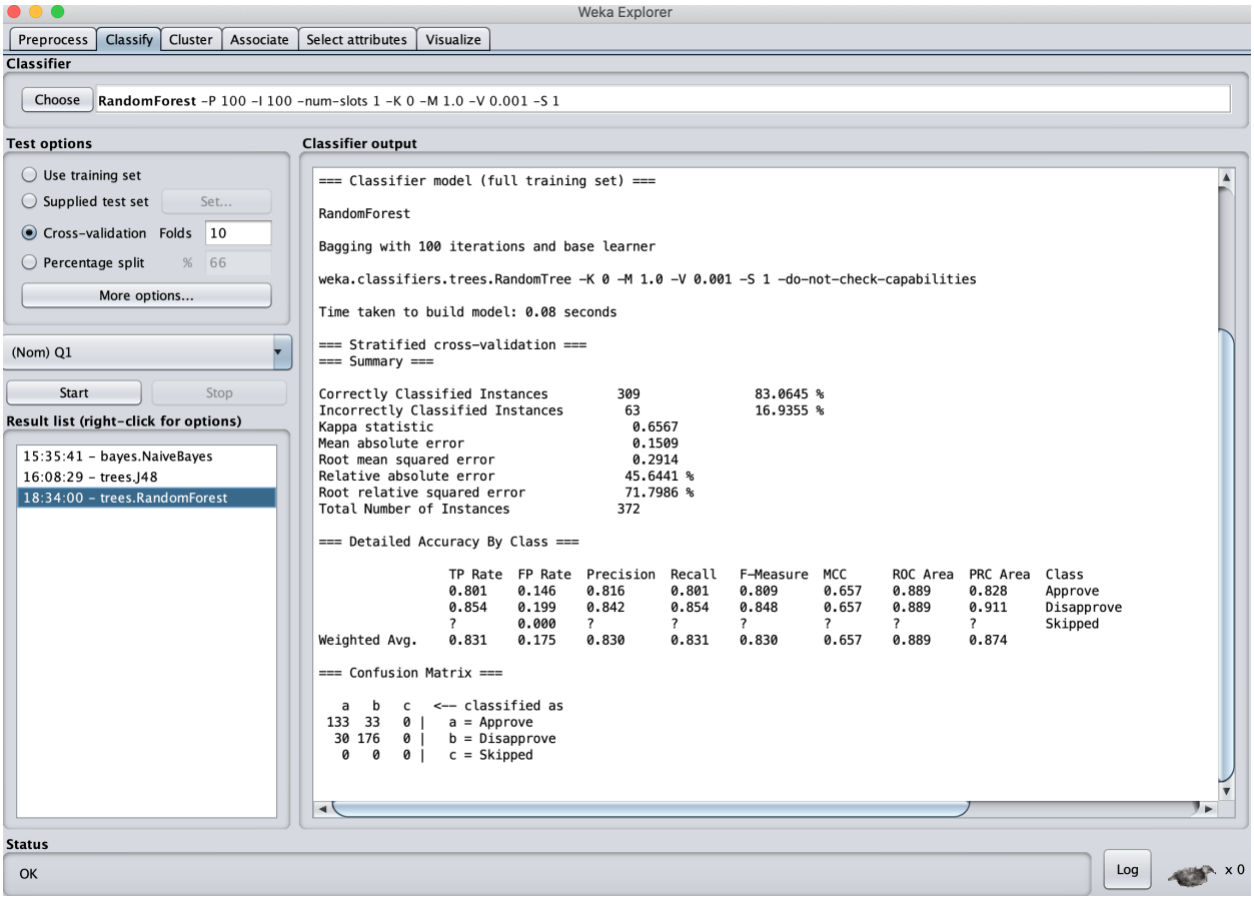


J48 - Cross Validation (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.95%	86.1%	16.0%	85.1%	Approve
	84.0%	13.9%	85.1%	Disapprove

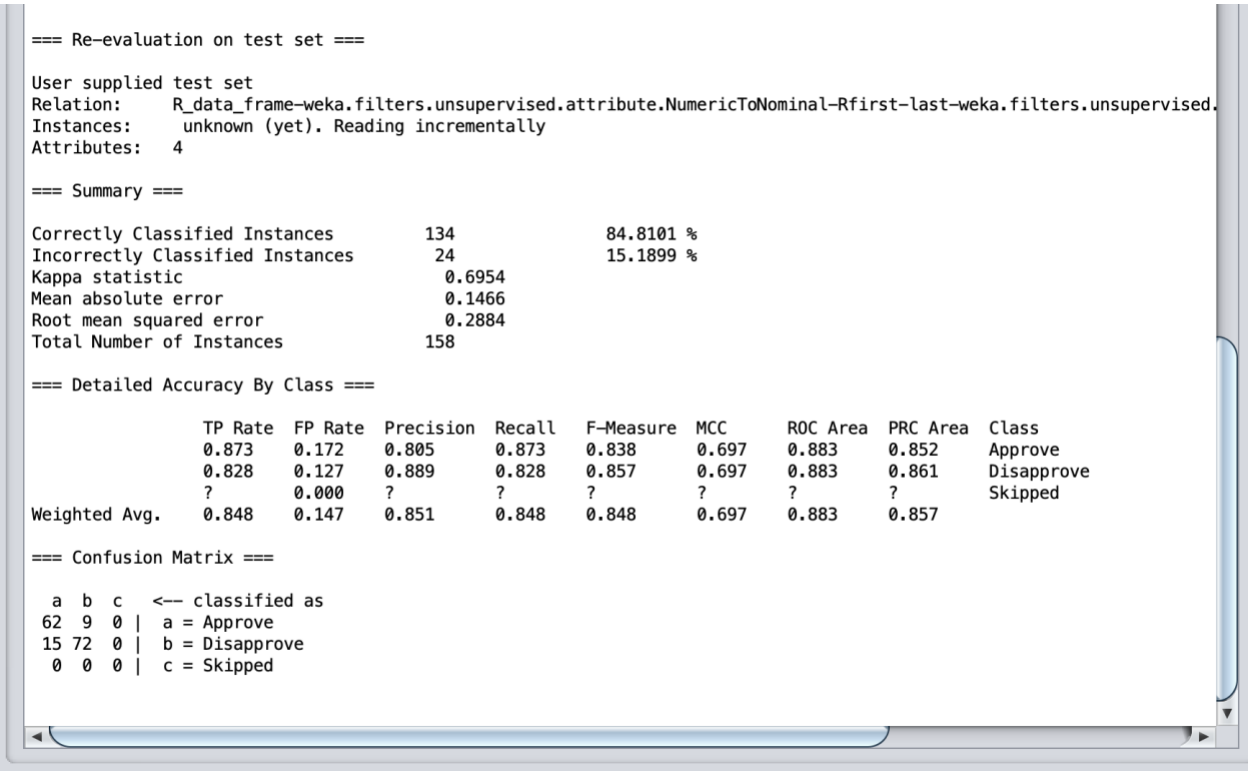
J48 - Test set (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	87.3%	17.2%	85.9%	Approve
	82.8%	12.7%	85.9%	Disapprove

3. RandomForest

- Step 1: Choose RandomForest and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set1_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



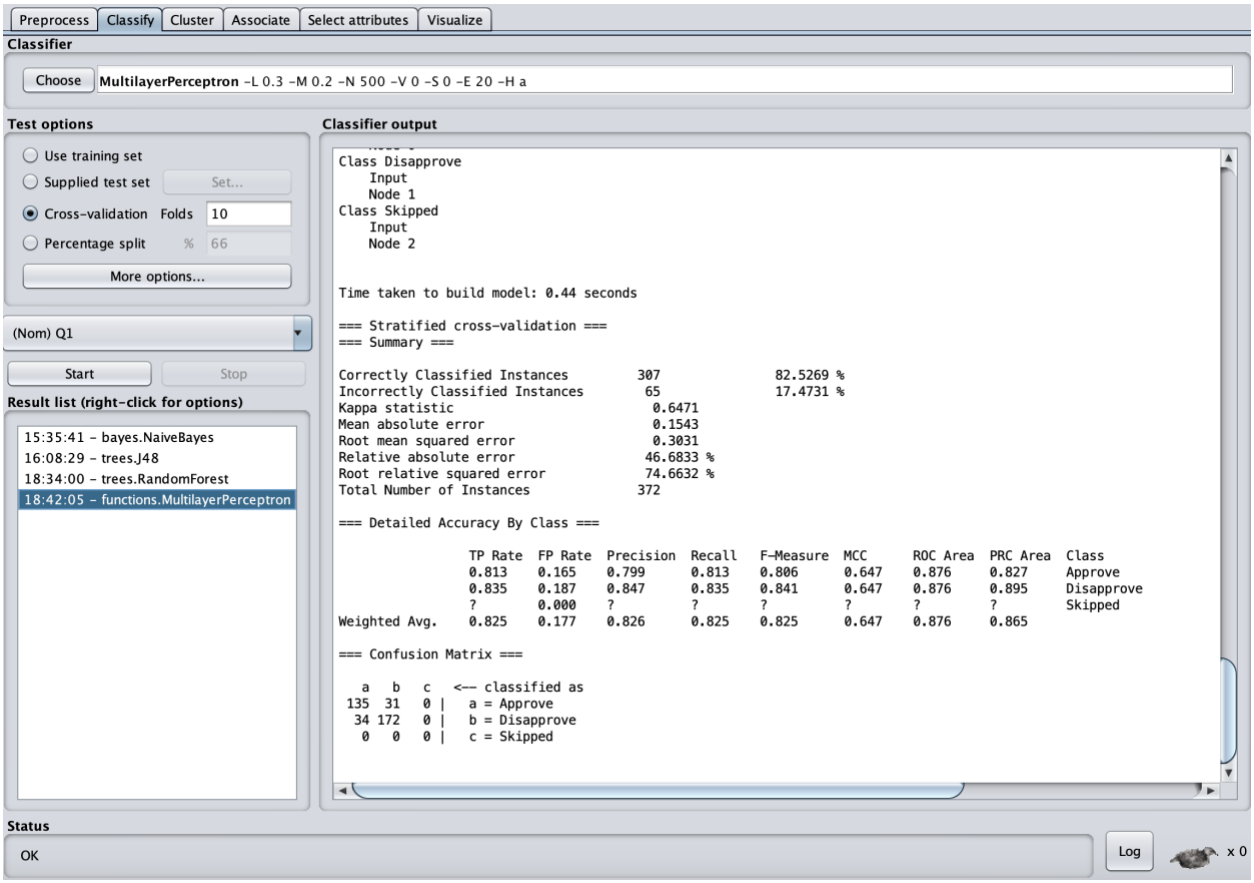
```
=== Re-evaluation on test set ===

User supplied test set
Relation:     R_data_frame-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.
Instances:     unknown (yet). Reading incrementally
Attributes:   4

=== Summary ===

Correctly Classified Instances         134               84.8101 %
Incorrectly Classified Instances        24               15.1899 %
Kappa statistic                          0.6954
Mean absolute error                      0.1466
Root mean squared error                  0.2884
Total Number of Instances              158

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.873    0.172    0.805      0.873    0.838      0.697    0.883     0.852     Approve
                 0.828    0.127    0.889      0.828    0.857      0.697    0.883     0.861     Disapprove
                 ?        0.000    ?          ?        ?          ?        ?         ?         Skipped
Weighted Avg.    0.848    0.147    0.851      0.848    0.848      0.697    0.883     0.857

=== Confusion Matrix ===

  a  b  c   <-- classified as
 62  9  0 |  a = Approve
 15 72  0 |  b = Disapprove
  0  0  0 |  c = Skipped
```

RandomForest - Cross Validation (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
83.06%	80.1%	14.6%	88.9%	Approve
	85.4%	19.9%	88.9%	Disapprove

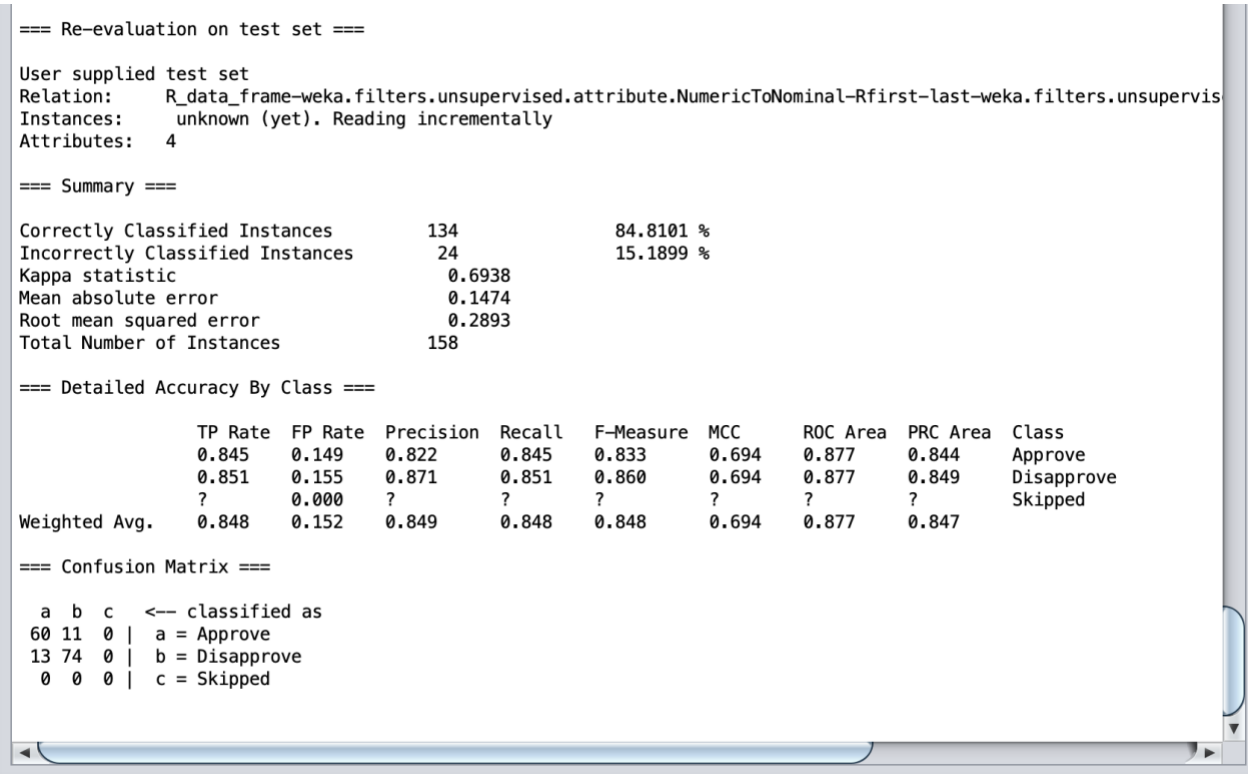
RandomForest - Test set (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	87.3%	17.2%	88.3%	Approve
	82.8%	12.7%	88.3%	Disapprove

4. MultilayerPerceptron

- Step 1: Choose MultilayerPerceptron and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set1_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



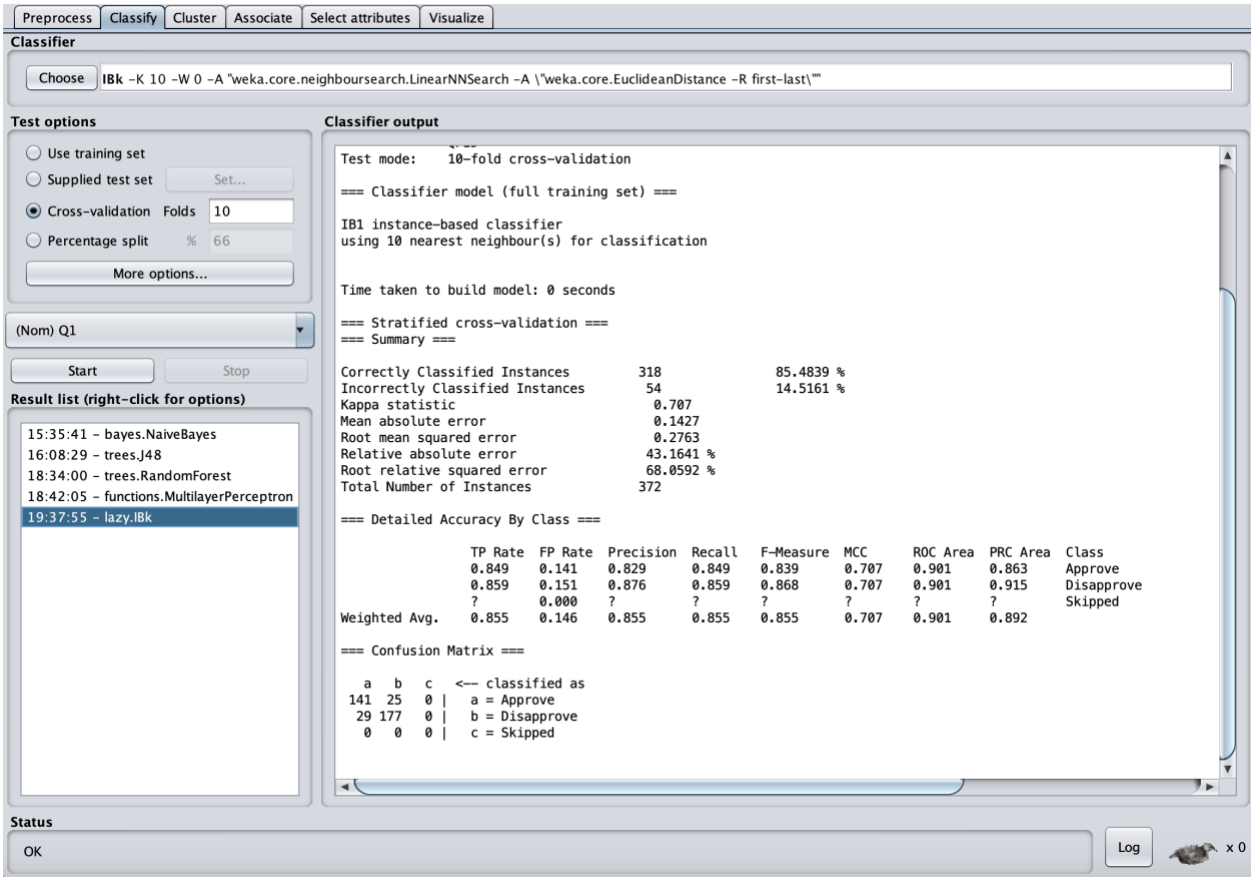
```
=== Re-evaluation on test set ===

User supplied test set
Relation:     R_data_frame-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervis
Instances:     unknown (yet). Reading incrementally
Attributes:   4

=== Summary ===

Correctly Classified Instances         134               84.8101 %
Incorrectly Classified Instances        24               15.1899 %
Kappa statistic                          0.6938
Mean absolute error                      0.1474
Root mean squared error                  0.2893
Total Number of Instances              158

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.845    0.149    0.822      0.845    0.833      0.694    0.877     0.844     Approve
                 0.851    0.155    0.871      0.851    0.860      0.694    0.877     0.849     Disapprove
                 ?        0.000    ?          ?        ?          ?        ?         ?         Skipped
Weighted Avg.    0.848    0.152    0.849      0.848    0.848      0.694    0.877     0.847

=== Confusion Matrix ===

  a  b  c   <-- classified as
 60 11  0 |  a = Approve
 13 74  0 |  b = Disapprove
  0  0  0 |  c = Skipped
```

MultiLayerPerceptron - Cross Validation (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
82.53%	81.3%	16.5%	87.6%	Approve
	83.5%	18.7%	87.6%	Disapprove

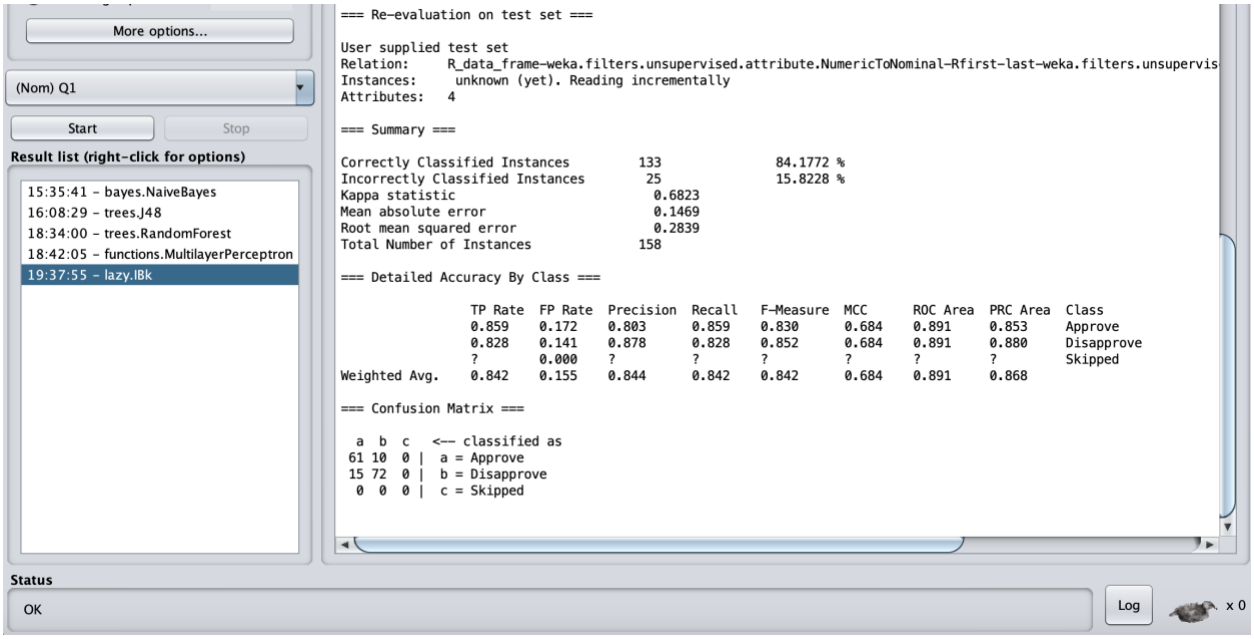
MultiLayerPerceptron - Test set (Set 1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	84.5%	14.9%	87.7%	Approve
	85.1%	15.5%	87.7%	Disapprove

5. K-nearest Neighbor (IBK) with K = 10

- Step 1: Choose IBK and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set1_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



IBK - Cross Validation (Set1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
85.48%	84.9%	14.1%	90.1%	Approve
	85.9%	15.1%	90.1%	Disapprove

IBK - Test set (Set1)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.18%	85.9%	17.2%	89.1%	Approve
	82.8%	14.1%	89.1%	Disapprove

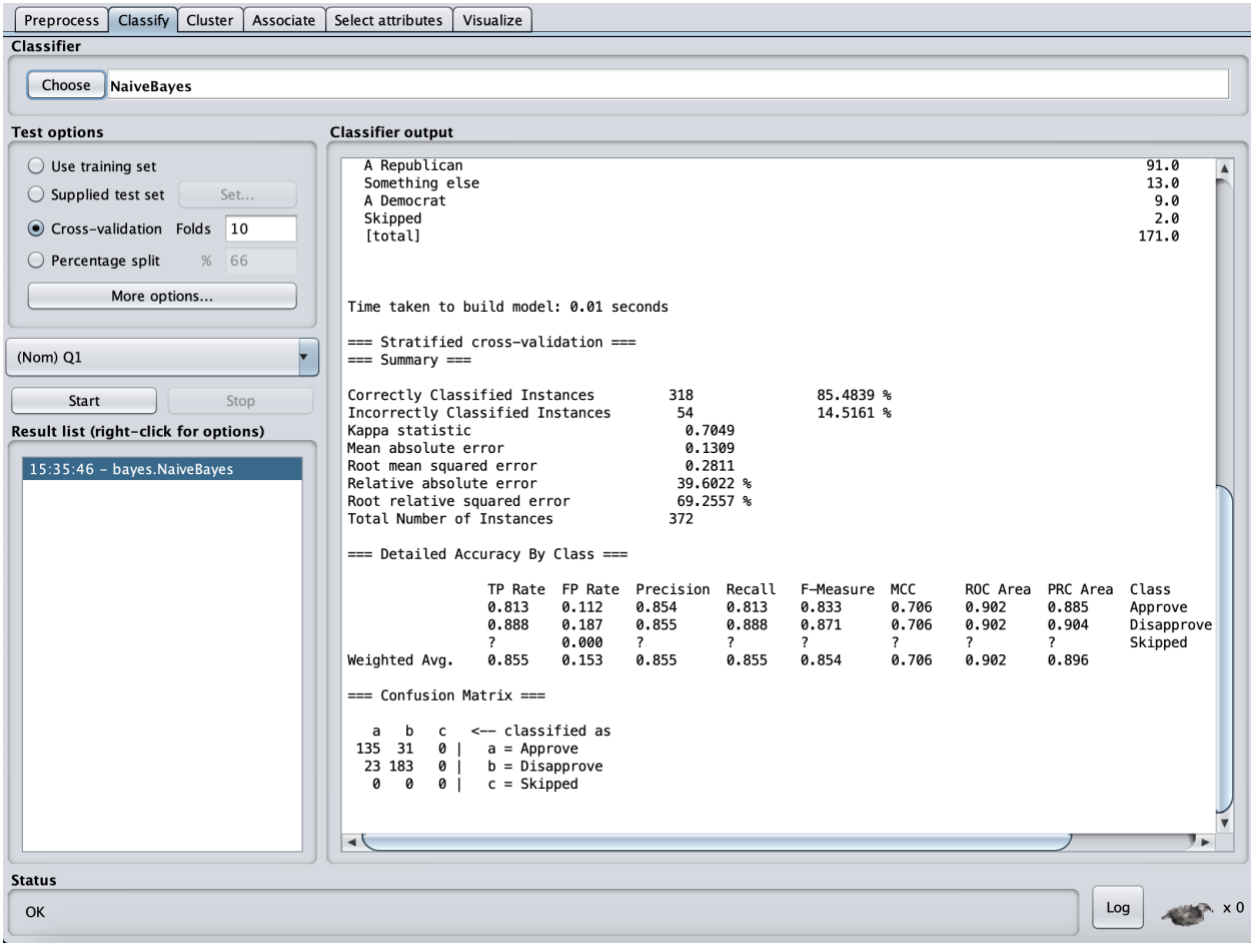
b. *Attribute Set 2: [Q3, QPID, Q2, Q1] - by the GainRatioAttributeEval*

1. Naive Bayes

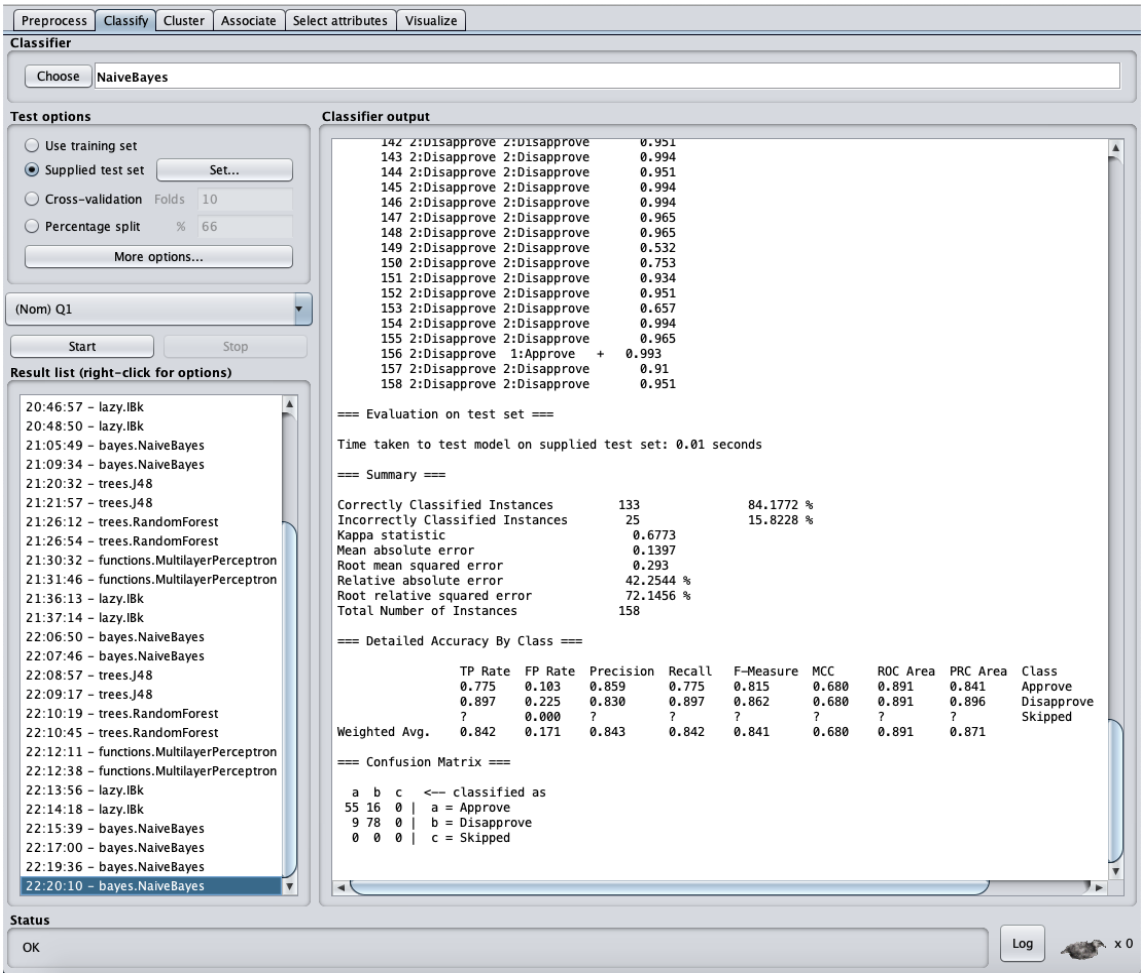
- Step 1: Open “Initial_Training_set.arff”, remove all attributes except Q3, QPID, Q2, Q1 and save it as “set2_training”

- Step 2: Open “Initial_Testing_set.arff”, remove all attributes except Q3, QPID, Q2, Q1 and save it as “set2_testing”

- Step 3: Open “set2_training”, Classify -> Choose NaiveBayes with Cross-validation (10 Folds), Q1 as class attribute -> Start



- Step 4: In order to test our model, we re-evaluate our model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set2_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

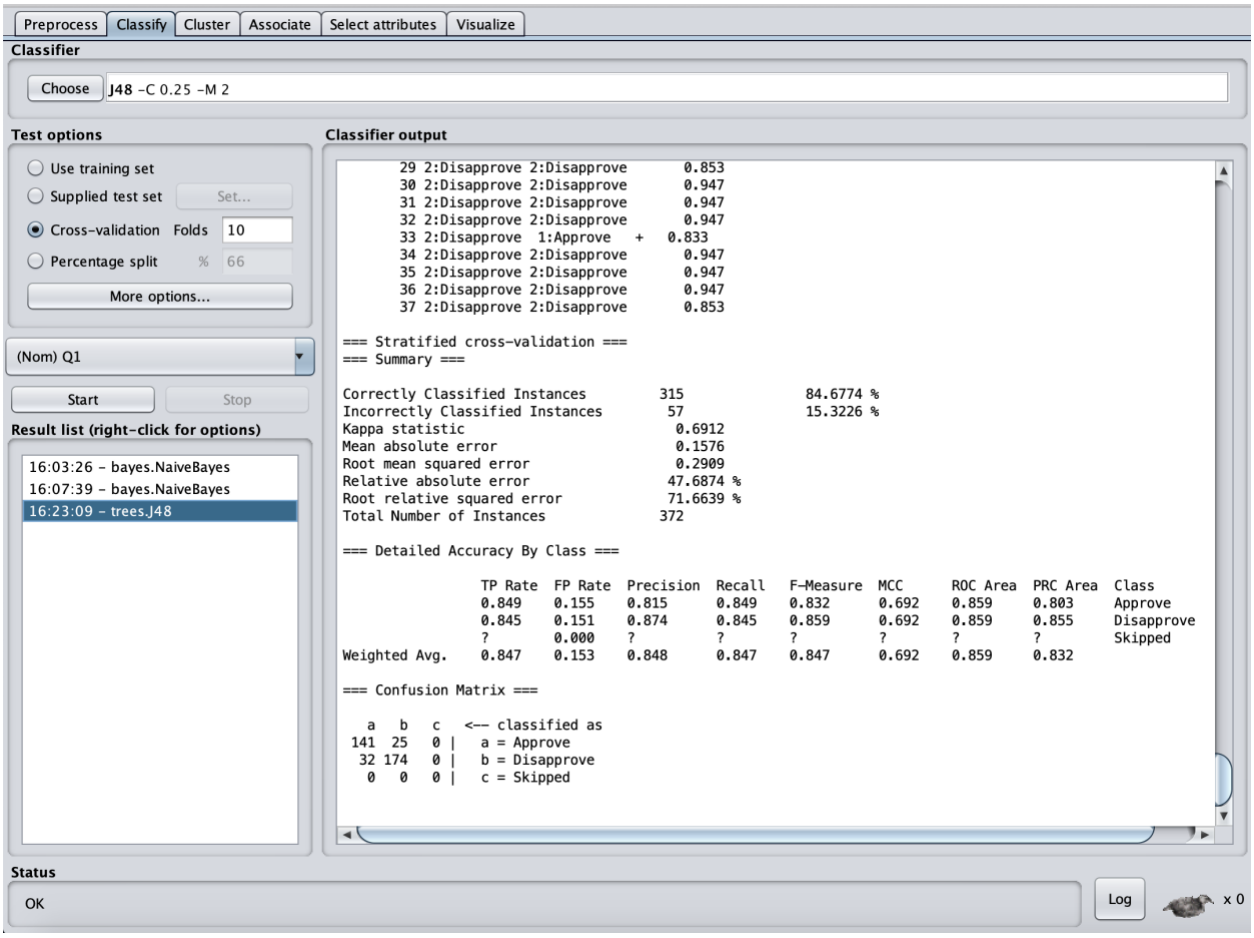


Naive Bayes - Cross Validation (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
85.48%	81.3%	11.2%	90.2%	Approve
	88.8%	18.7%	90.2%	Disapprove

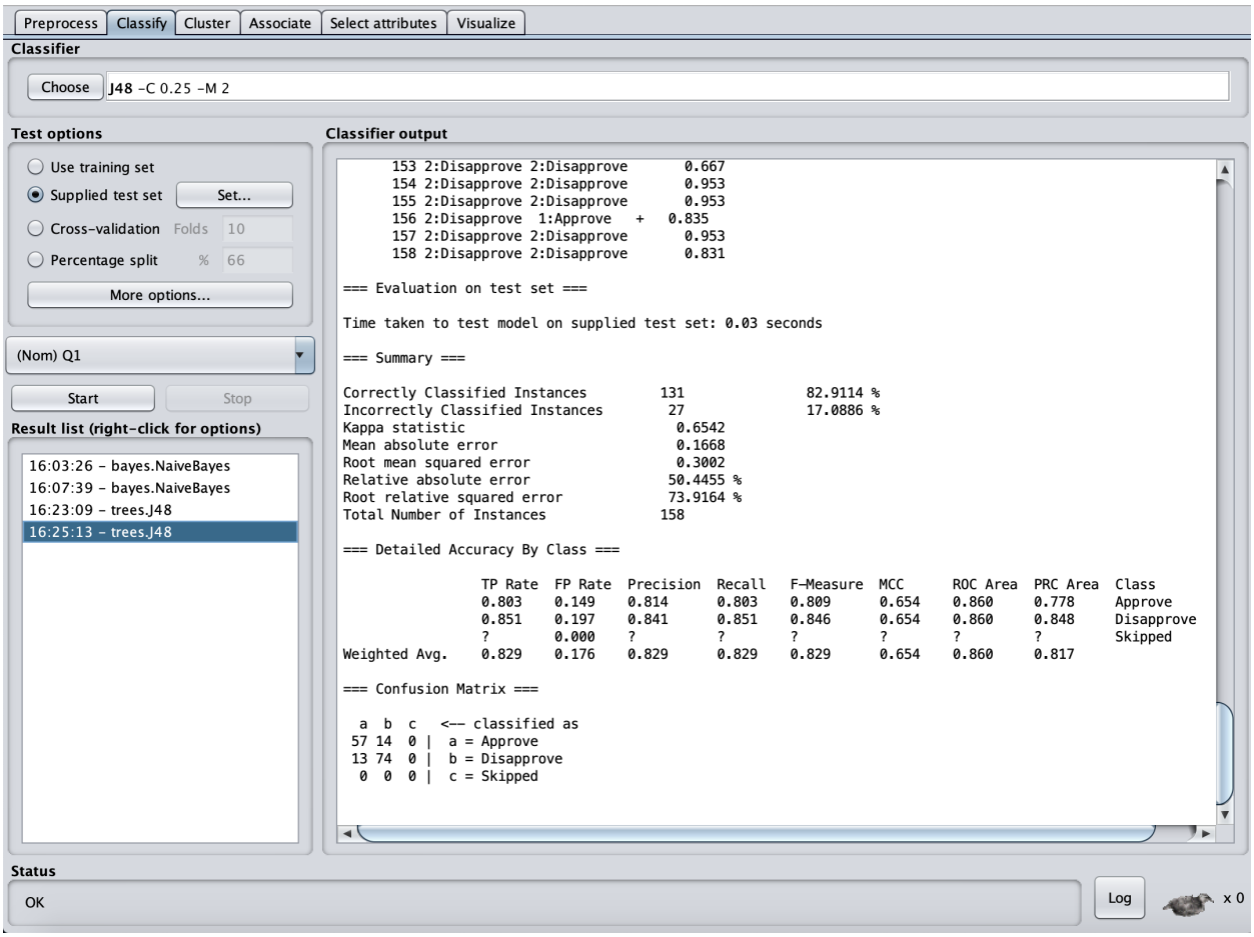
Naive Bayes - Test set (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.18%	77.5%	10.3%	89.1%	Approve
	89.7%	22.5%	89.1%	Disapprove

2. J48

- Step 1: Choose J48 and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open set2_testing with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

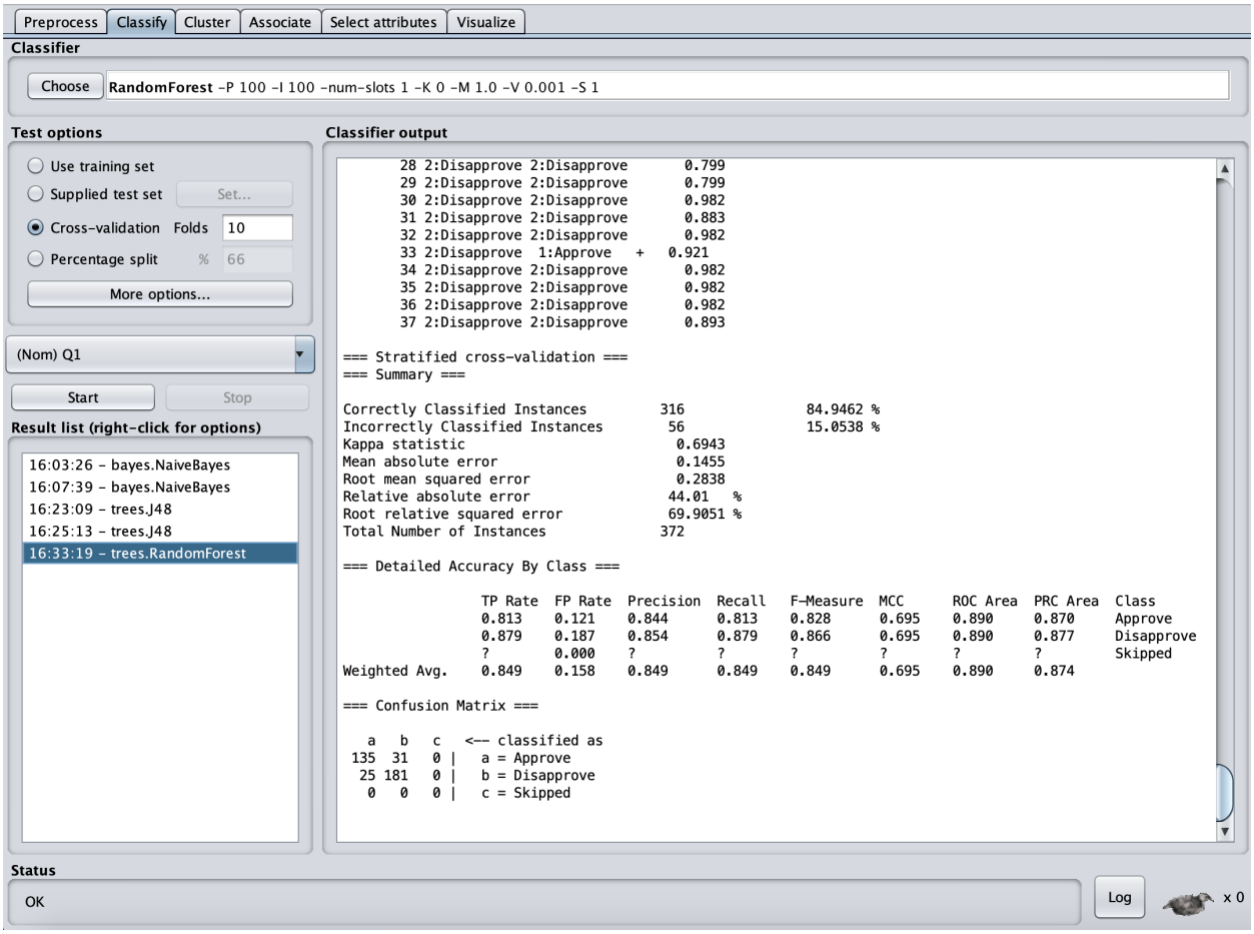


J48 - Cross Validation (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.68%	84.9%	15.5%	85.9%	Approve
	84.5%	15.1%	85.9%	Disapprove

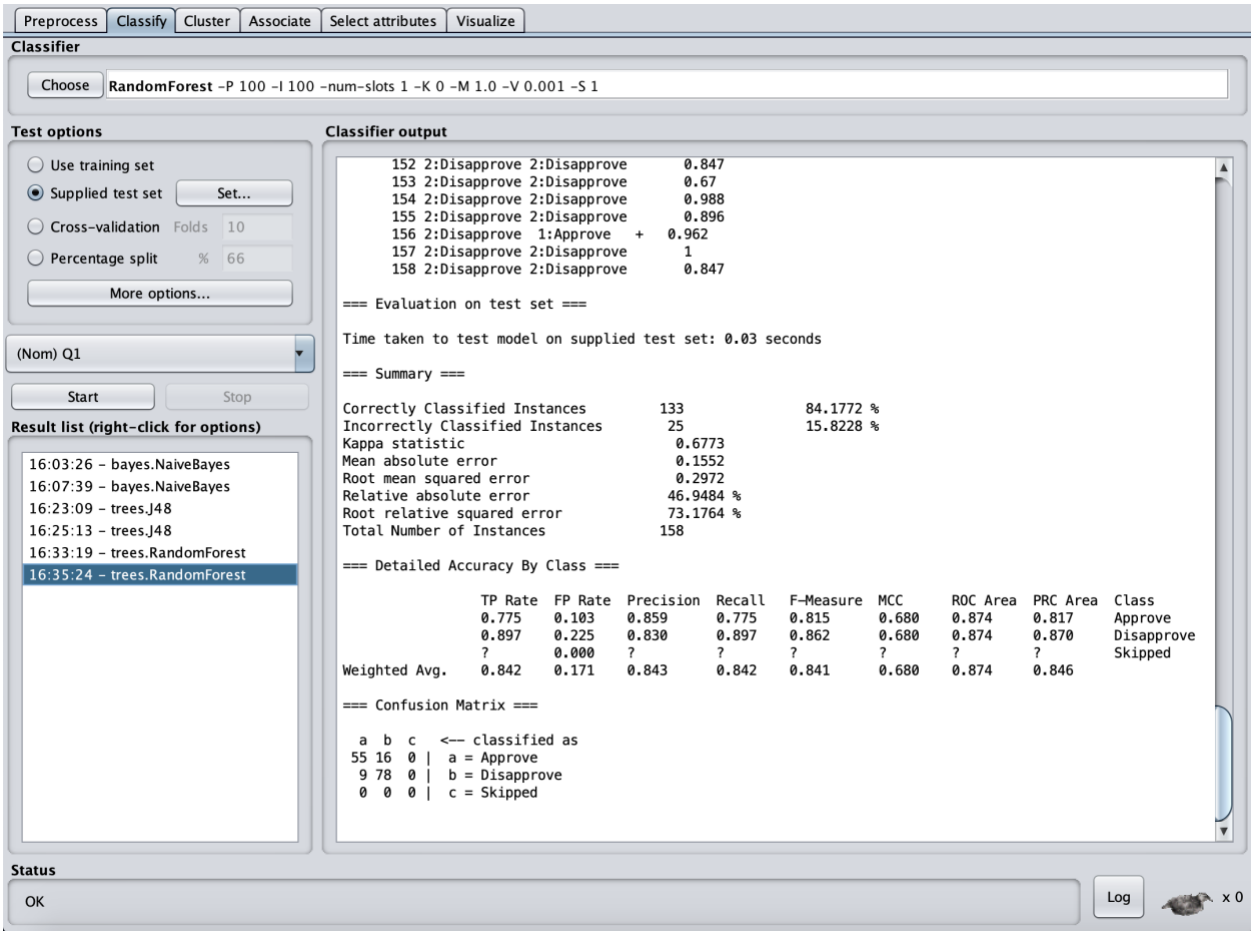
J48 - Test set (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
82.91%	80.3%	14.9%	86.0%	Approve
	85.1%	19.7%	86.0%	Disapprove

3. Random Forest

- Step 1: Choose RandomForest and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open set2_testing with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

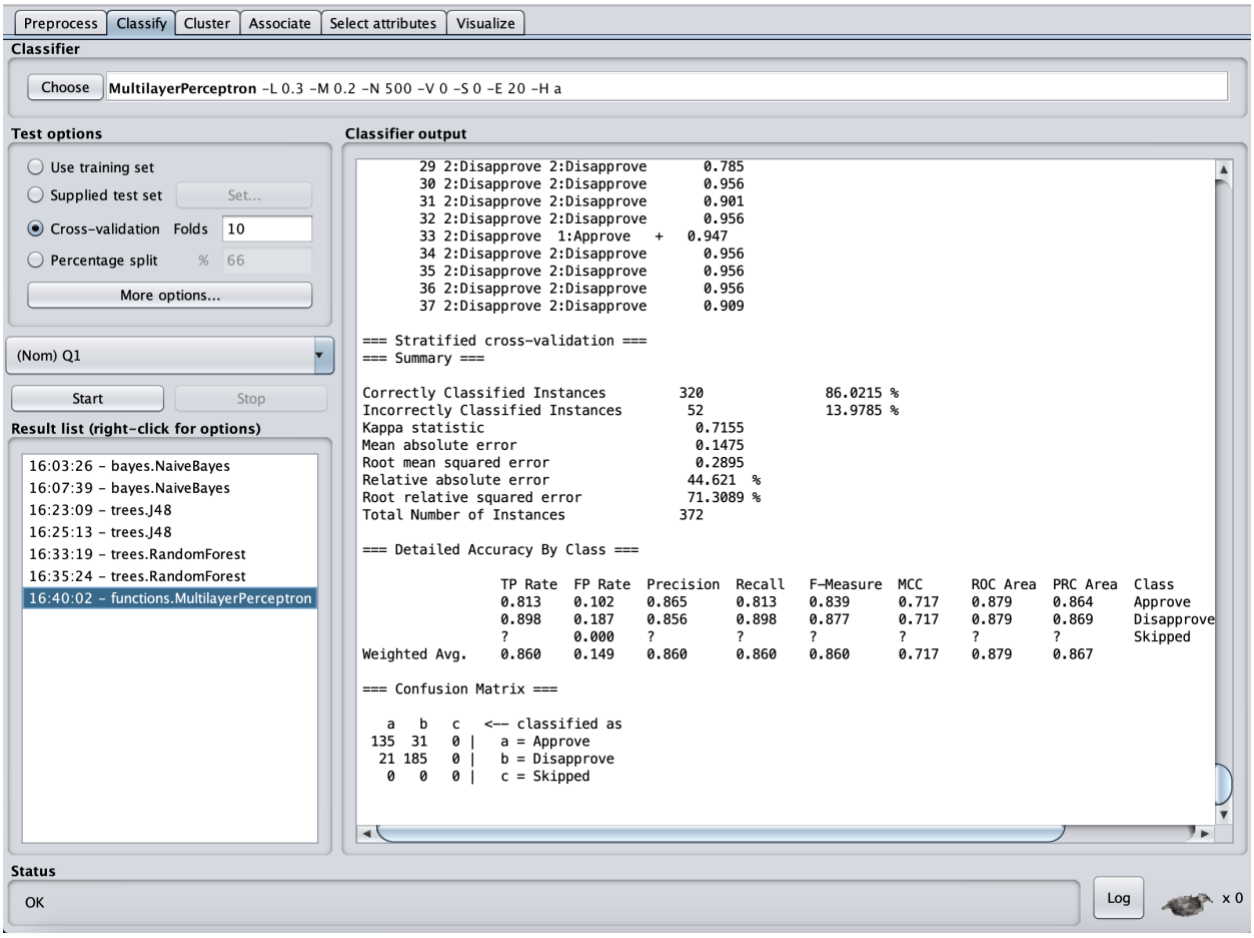


Random Forest - Cross Validation (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.95%	81.3%	12.1%	89.0%	Approve
	87.9%	18.7%	89.0%	Disapprove

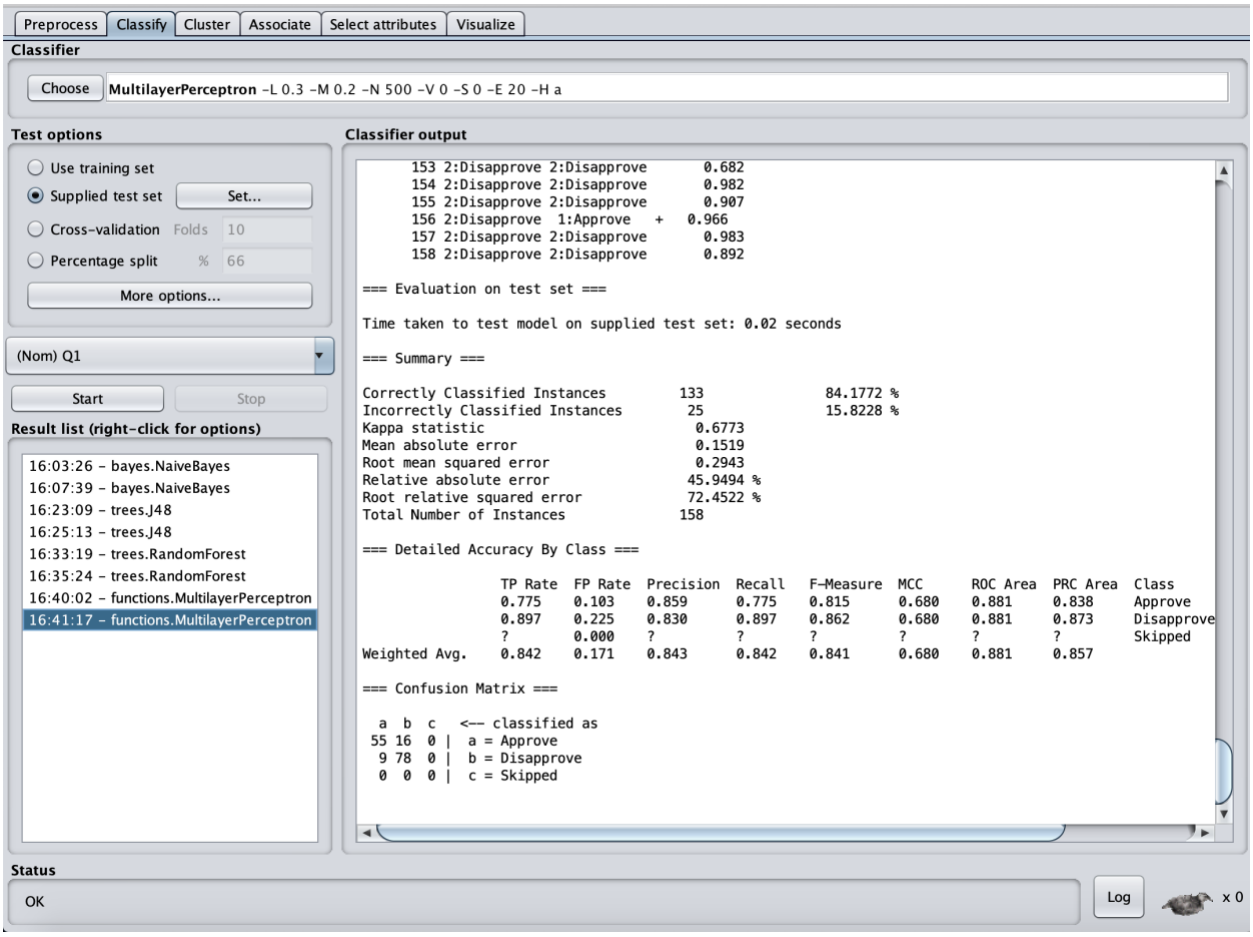
Random Forest - Test set (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.18%	77.5%	10.3%	87.4%	Approve
	89.7%	22.5%	87.4%	Disapprove

4. MultilayerPerceptron

- Step 1: Choose MultilayerPerceptron and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open set2_testing with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

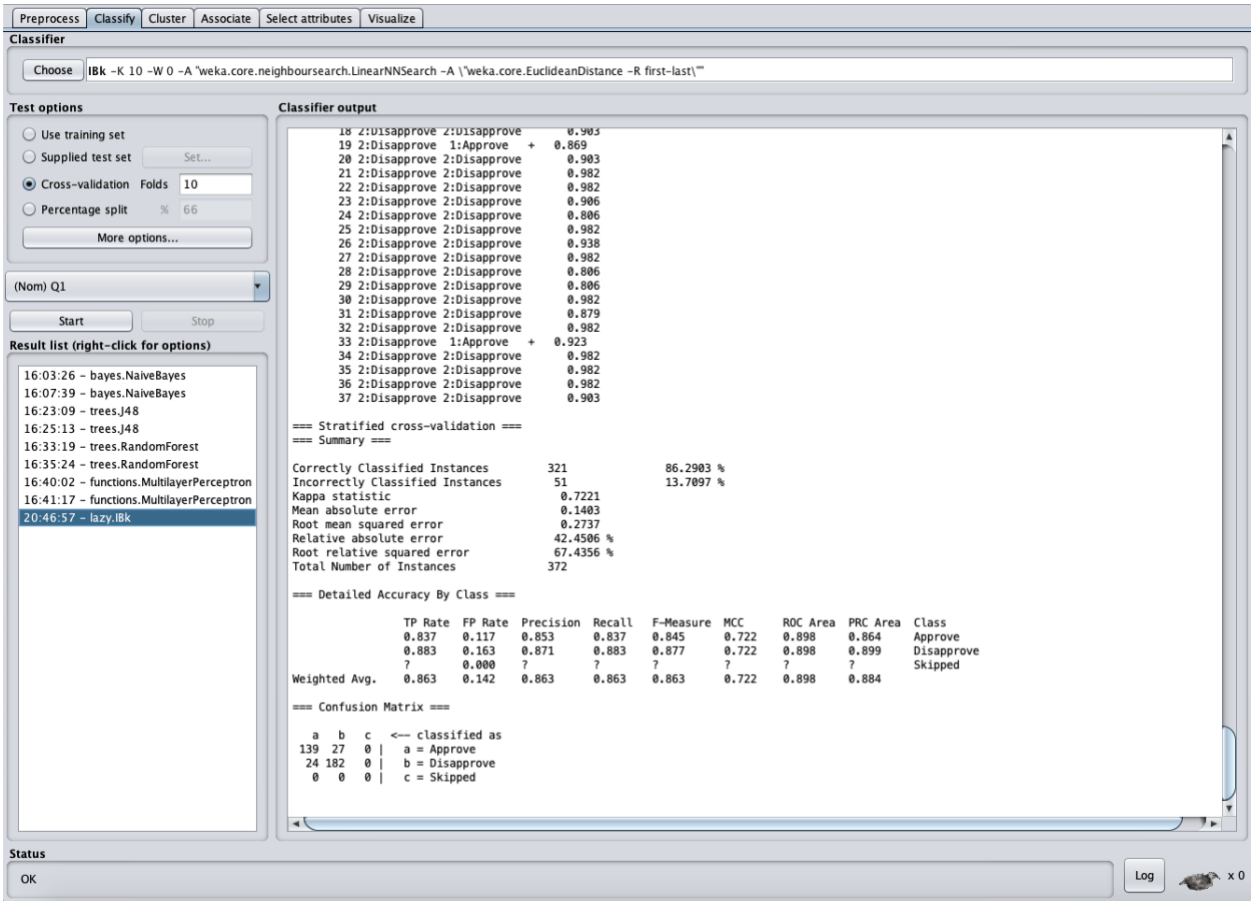


MultilayerPerceptron - Cross Validation (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
86.02%	81.3%	10.2%	87.9%	Approve
	89.8%	18.7%	87.9%	Disapprove

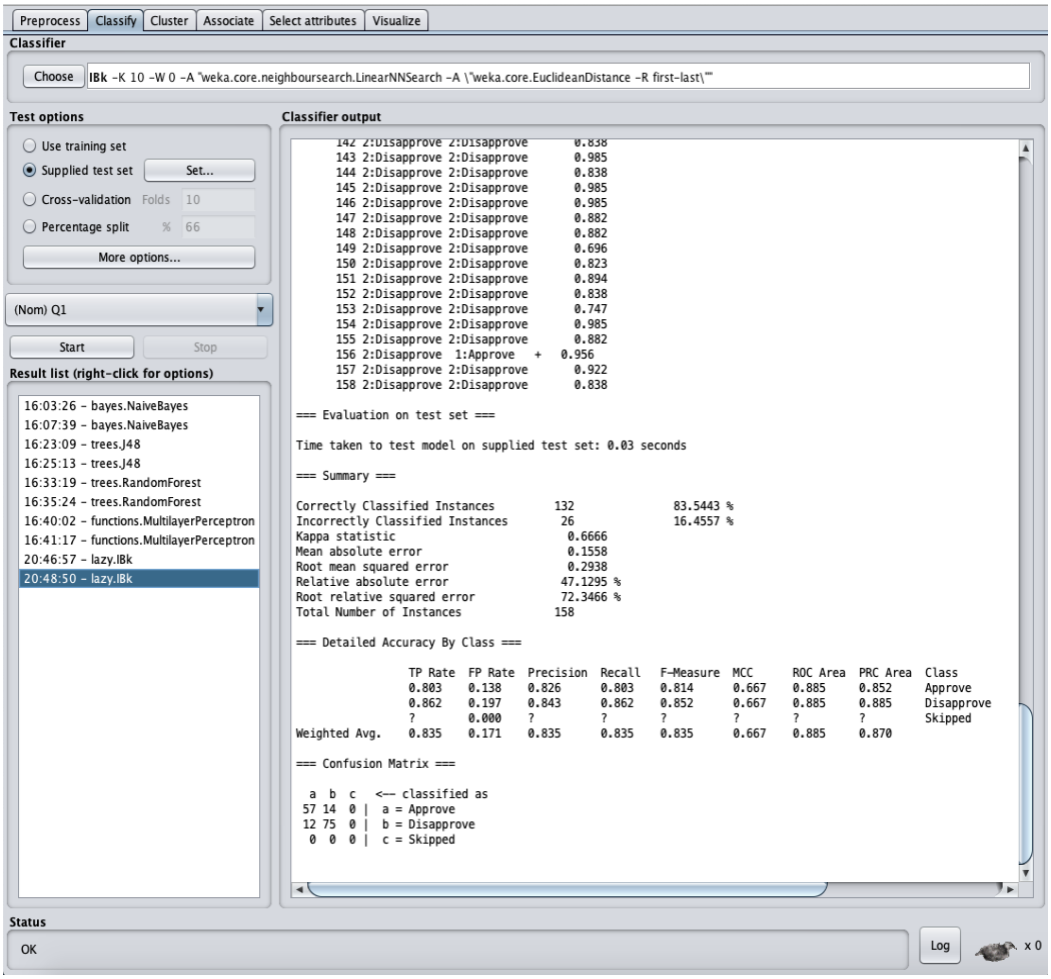
MultilayerPerceptron - Test set (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.18%	77.5%	10.3%	88.1%	Approve
	89.7%	22.5%	88.1%	Disapprove

5. K-nearest Neighbor (IBK) with K = 10

- Step 1: Choose IBK and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open set2_testing with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



IBK - Cross Validation (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
86.29%	83.7%	11.7%	89.8%	Approve
	88.3%	16.3%	89.8%	Disapprove

IBK - Test set (Set 2)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
83.54%	80.3%	13.8%	88.5%	Approve
	86.2%	19.7%	88.5%	Disapprove

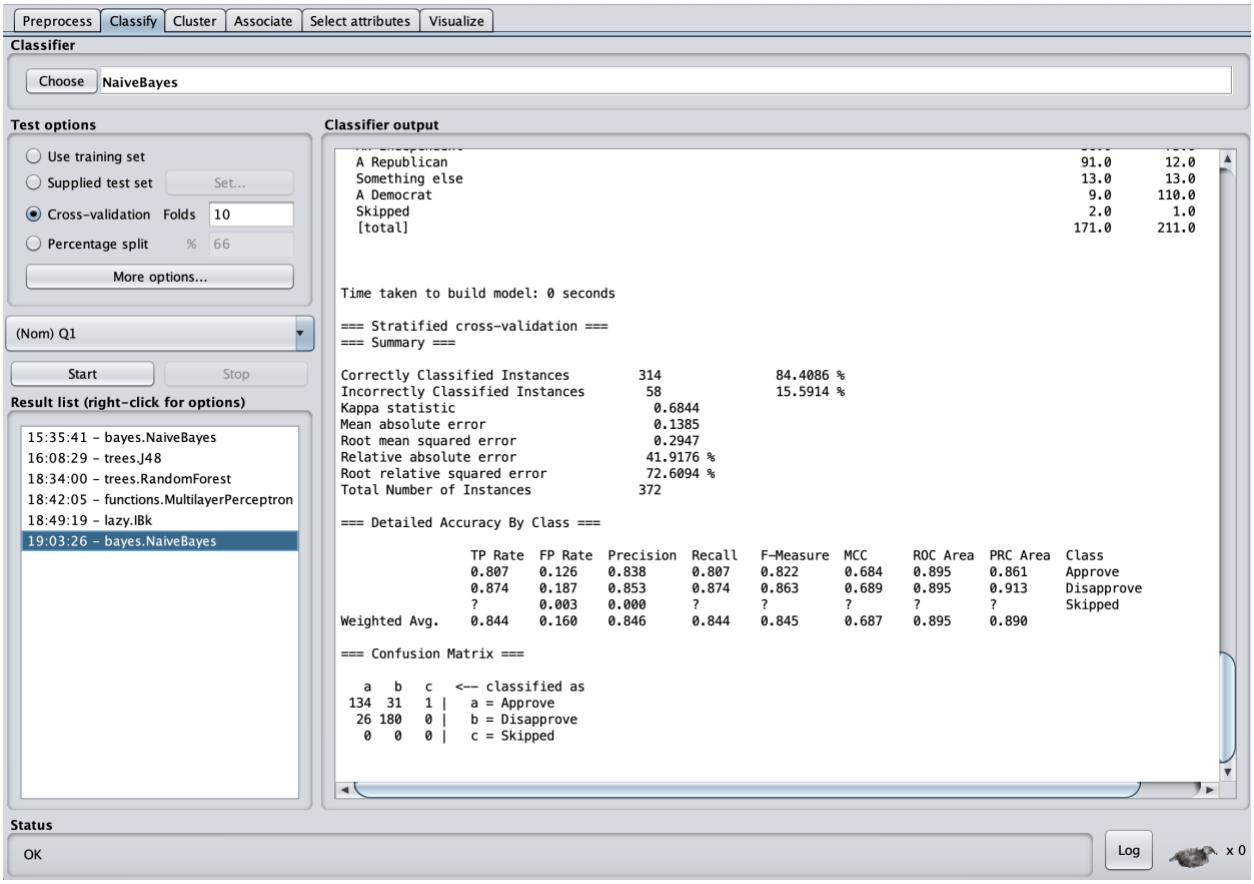
c. Attribute Set 3: [QPID, Q3, Q2, ppage, ppstaten, Q1] - by the InfoGainAttributeEval

1. Naive Bayse:

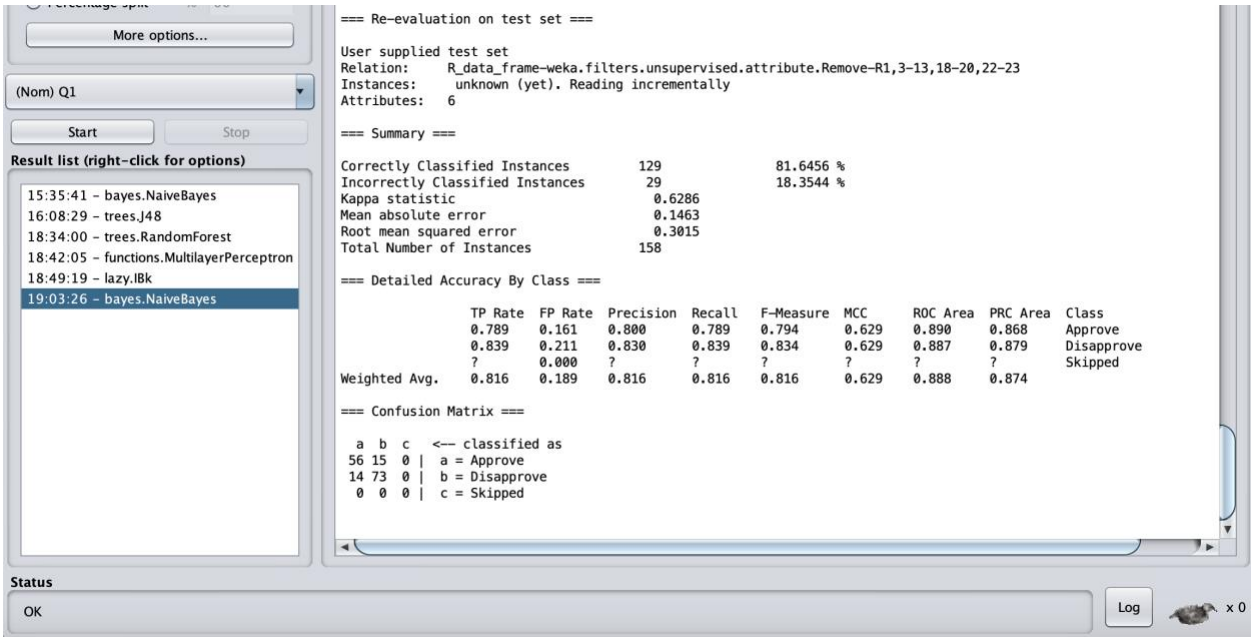
- Step 1: Open “Initial_Training_set.arff”, remove all attributes except Q3, QPID, Q2, ppage, ppstaten, and Q1. Save it as “set3_training”

- Step 2: Open “Initial_Testing_set.arff”, remove all attributes except Q3, QPID, Q2, ppage, ppstaten, and Q1. Save it as “set3_testing”

- Step 3: Open “set3_training”, Classify -> Choose NaiveBayes with Cross-validation (10 Folds), Q1 as class attribute -> Start



- Step 4: In order to test our model, we re-evaluate our model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set3_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

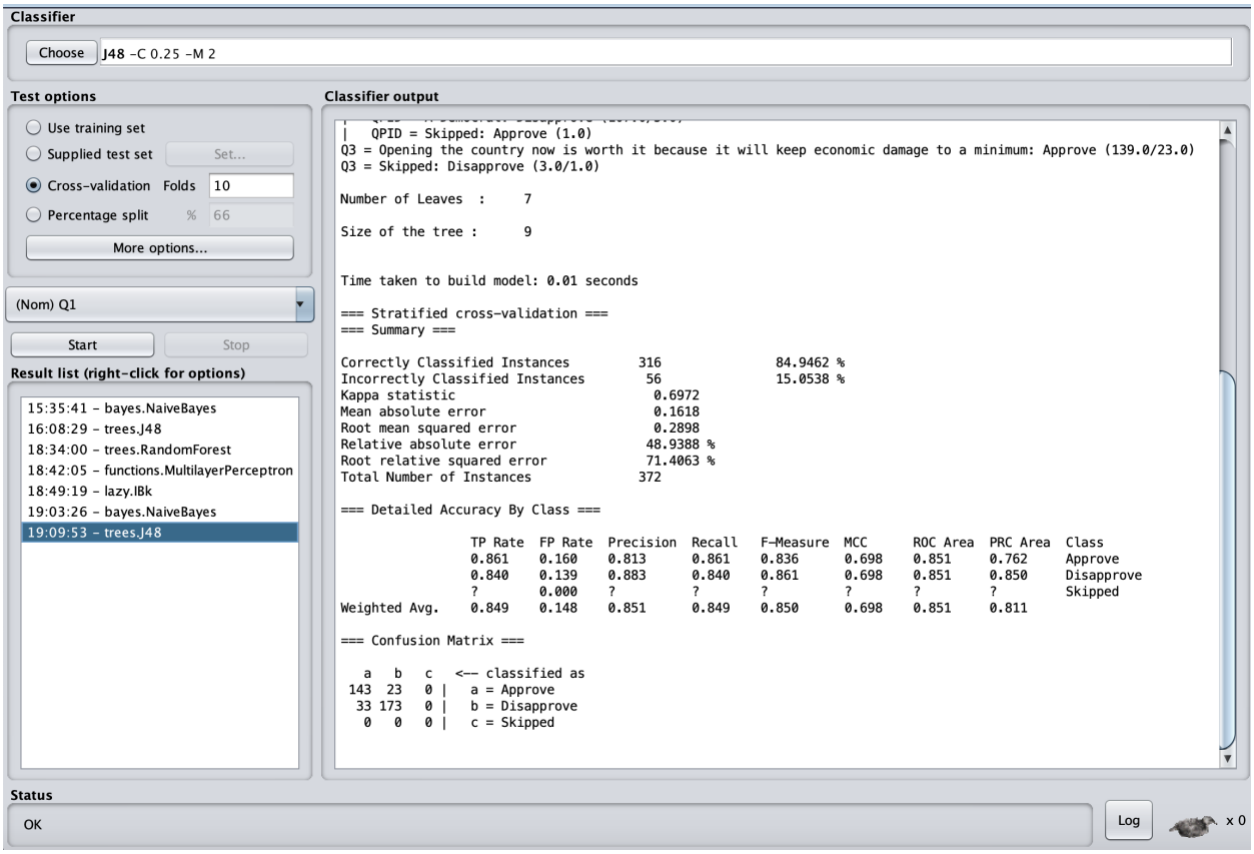


Naive Bayes - Cross Validation (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.41%	80.7%	12.6%	89.5%	Approve
	87.4%	18.7%	89.5%	Disapprove

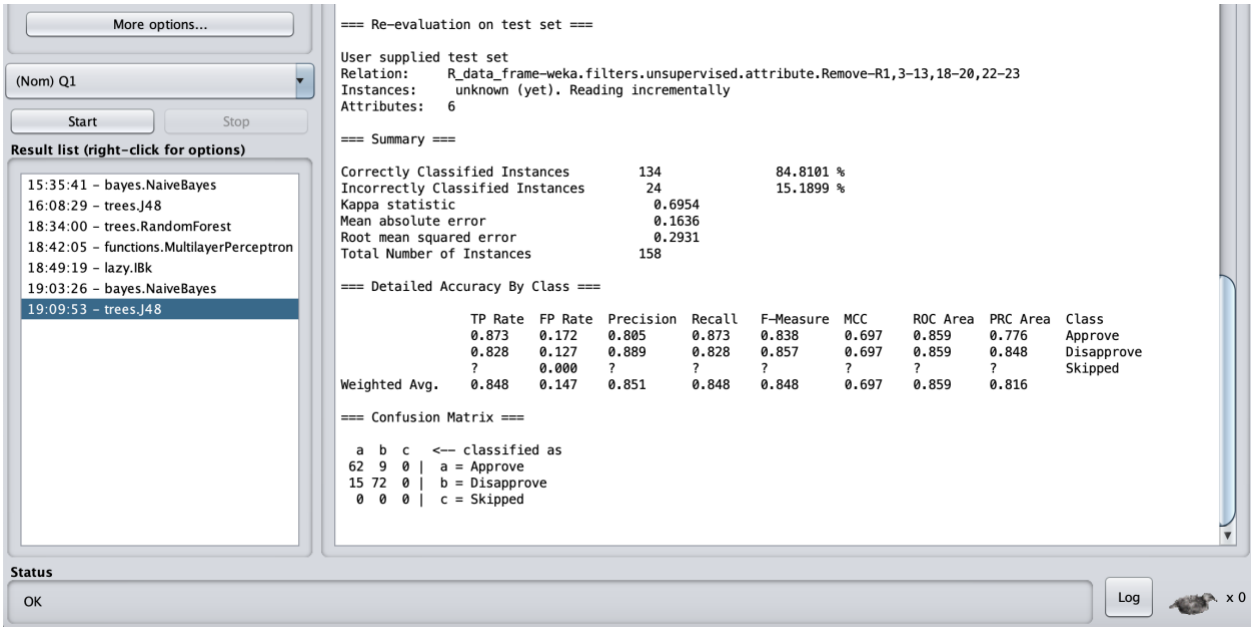
Naive Bayes - Test set (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
81.65%	78.9%	16.1%	89.0%	Approve
	83.9%	21.1%	88.7%	Disapprove

2. J48

- Step 1: First to assess how good J48 is, use Cross-validation to build our model. Choose J48 and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set3_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

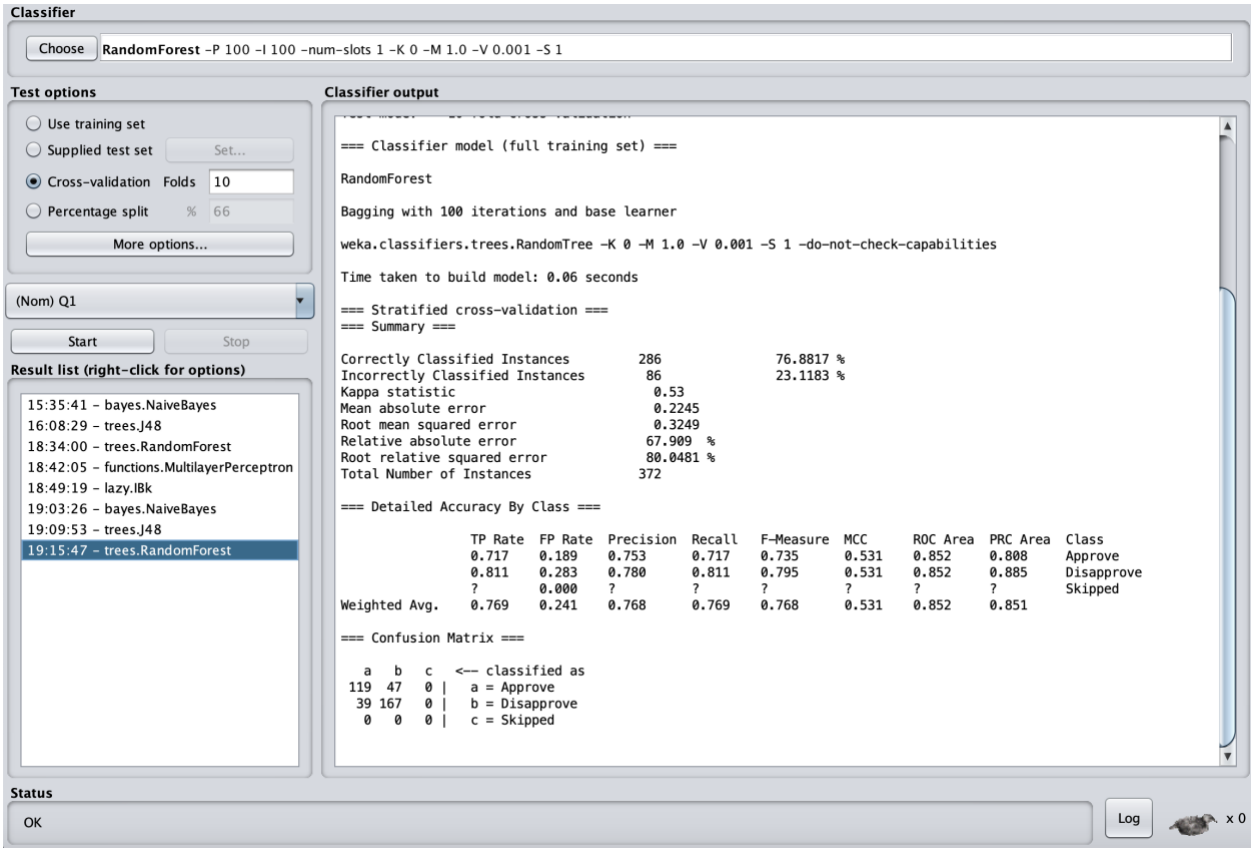


J48 - Cross Validation (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.95%	86.1%	16.0%	85.1%	Approve
	84.0%	13.9%	85.1%	Disapprove

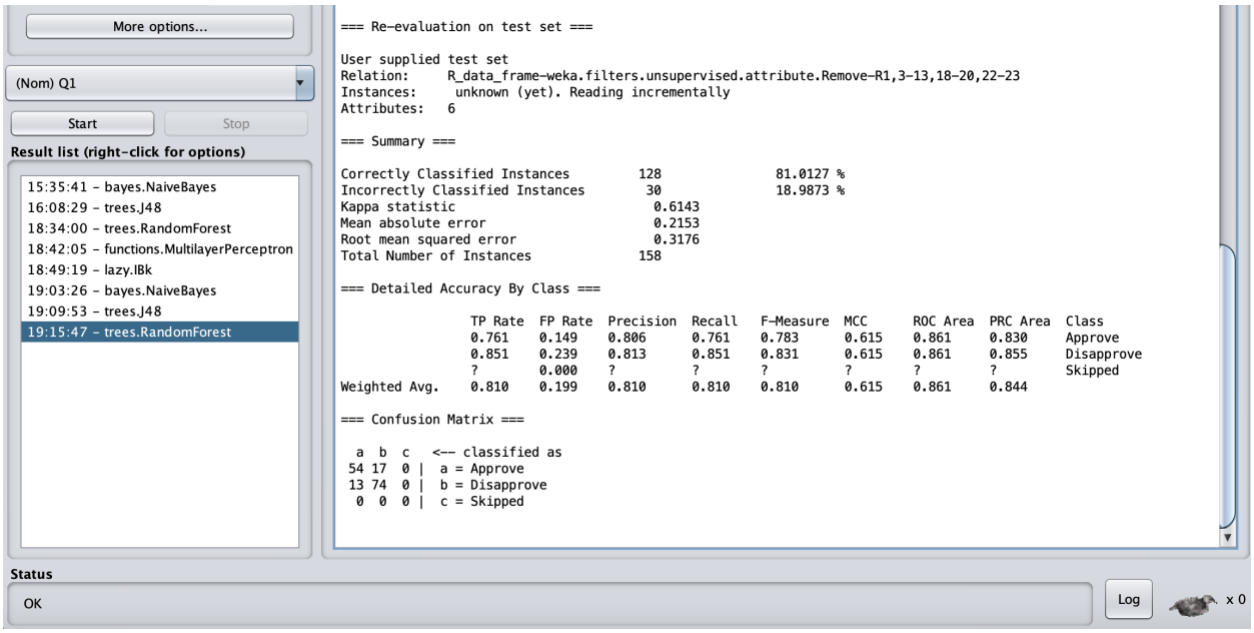
J48 - Test set (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	87.3%	17.2%	85.9%	Approve
	82.8%	12.7%	85.9%	Disapprove

3. Random Forest

- Step 1: Choose RandomForest and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set3_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

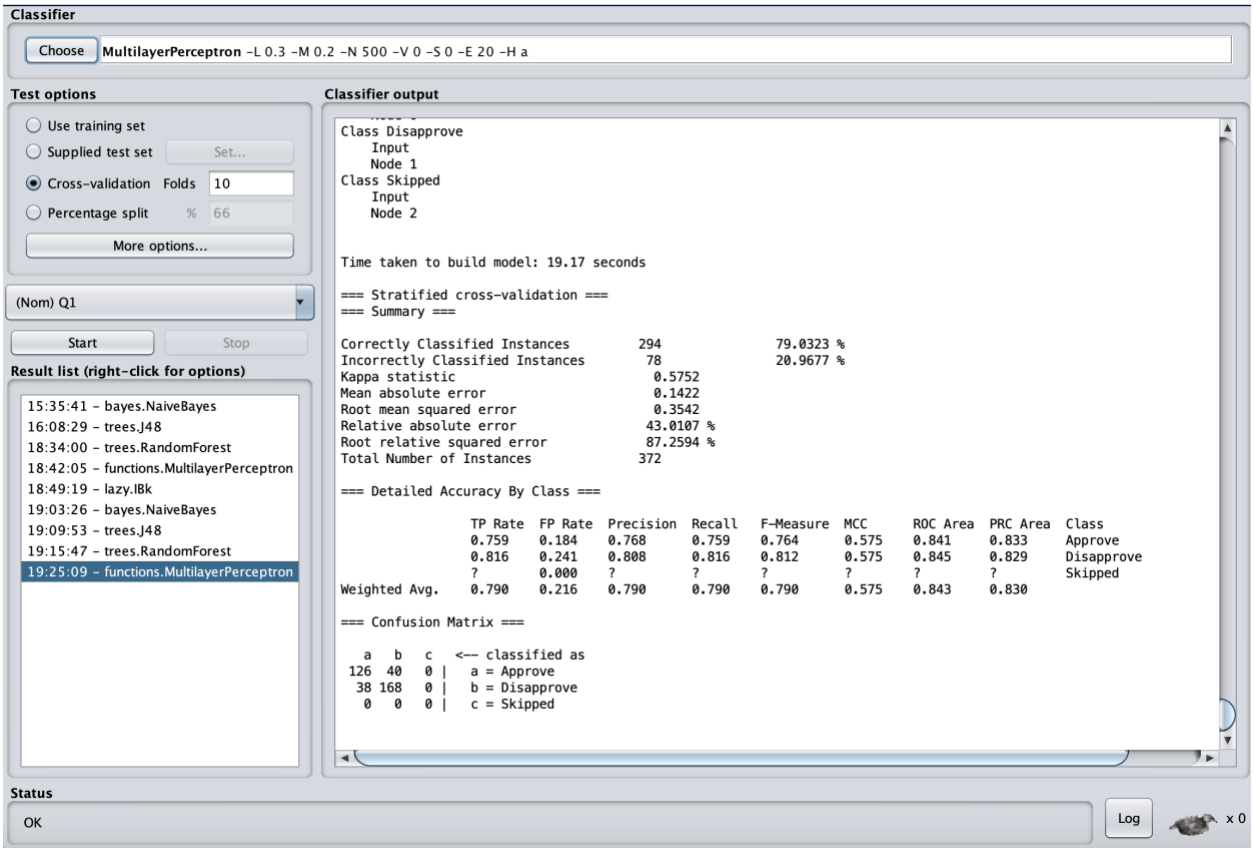


RandomForest - Cross Validation (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
76.88%	71.7%	18.9%	85.2%	Approve
	81.1%	28.3%	85.2%	Disapprove

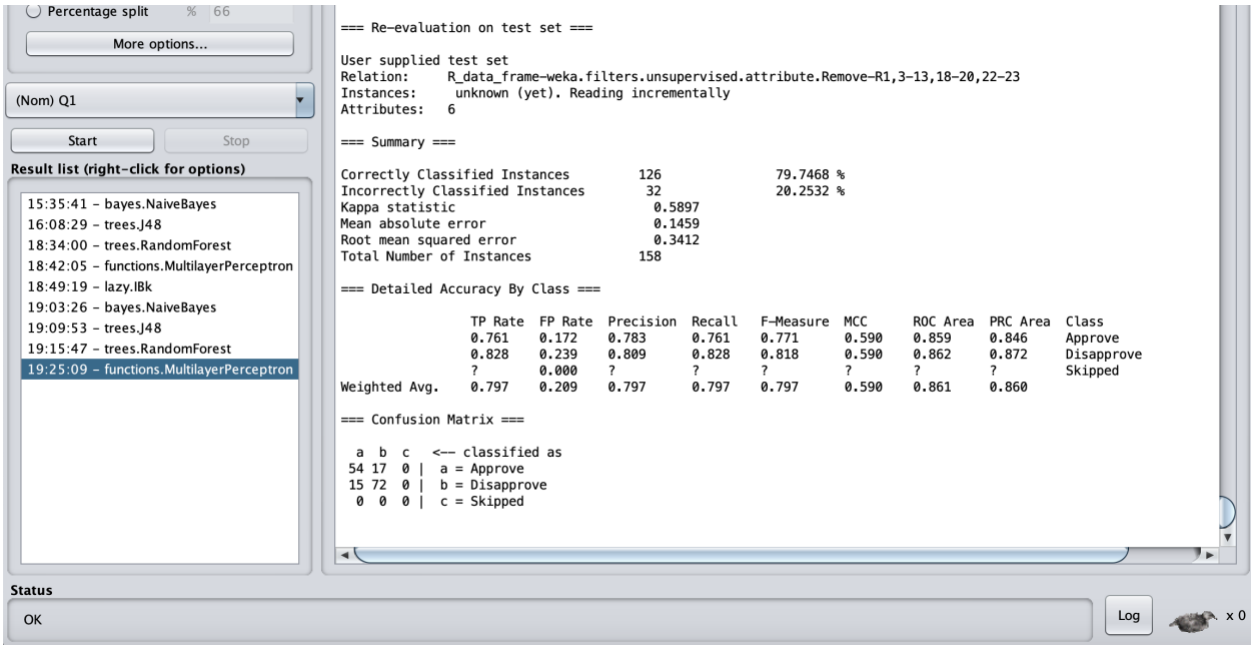
RandomForest - Test set (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
81.01%	76.1%	14.9%	86.1%	Approve
	85.1%	23.9%	86.1%	Disapprove

4. MultiLayerPerceptron

- Step 1: Choose MultilayerPerceptron and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set3_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

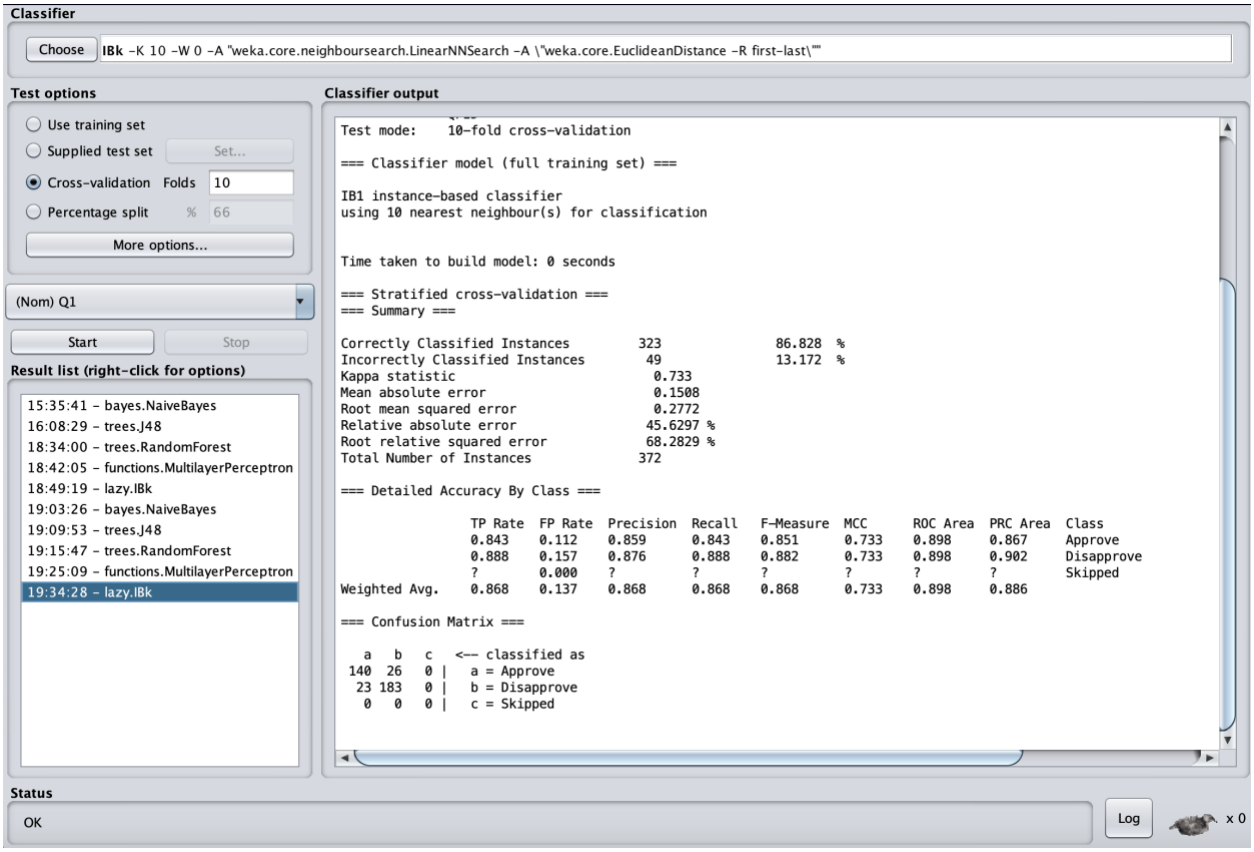


MultiLayerPerceptron - Cross Validation (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
79.03%	75.9%	18.4%	84.1%	Approve
	81.6%	24.1%	84.5%	Disapprove

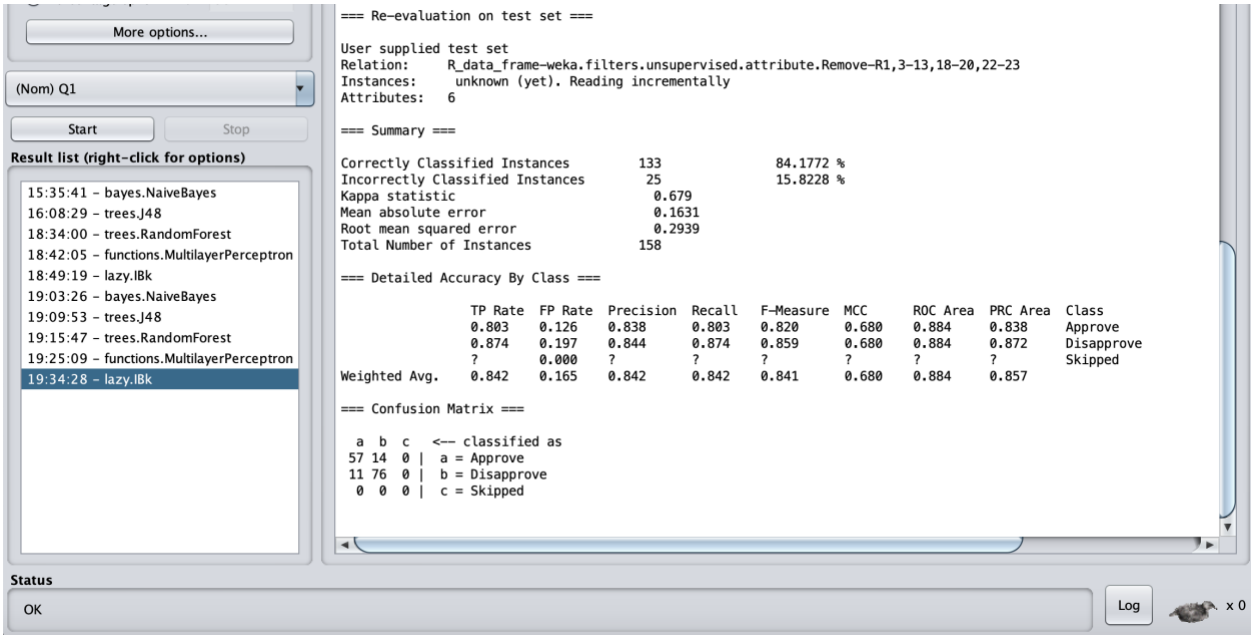
MultiLayerPerceptron - Test set (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
79.75%	76.1%	17.2%	85.9%	Approve
	82.8%	23.9%	86.2%	Disapprove

5. K-nearest Neighbor (IBK) with K = 10

- Step 1: Choose IBK and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set3_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



IBK - Cross Validation (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
86.83%	84.3%	11.2%	89.8%	Approve
	88.8%	15.7%	89.8%	Disapprove

IBK - Test set (Set 3)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.18%	80.3%	12.6%	88.4%	Approve
	87.4%	19.7%	88.4%	Disapprove

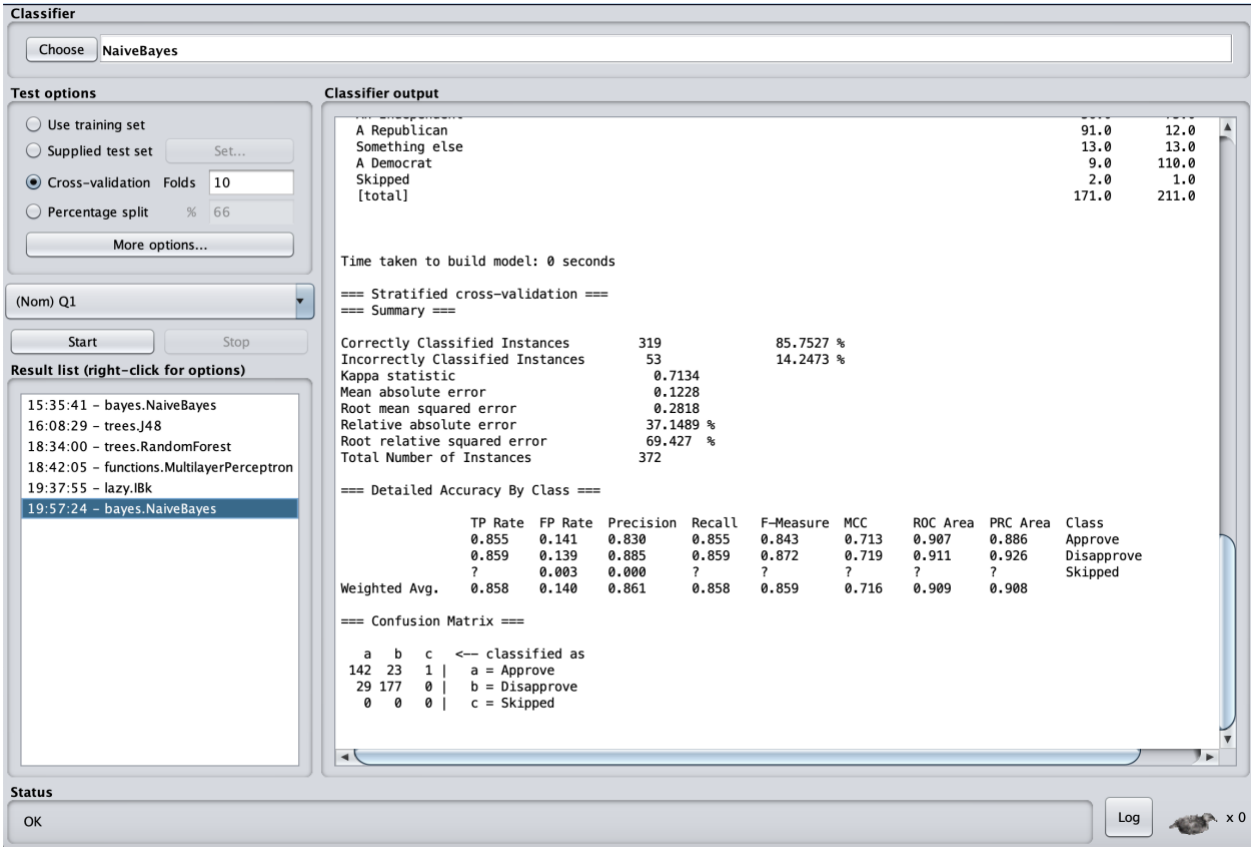
d. Attribute set 4: [Q3, QPID, ppethm, Q2, pphouse, ppmsacat, ppmarit, Q4, Q1] - by CorrelationAttributeEval

1. Naive Bayes

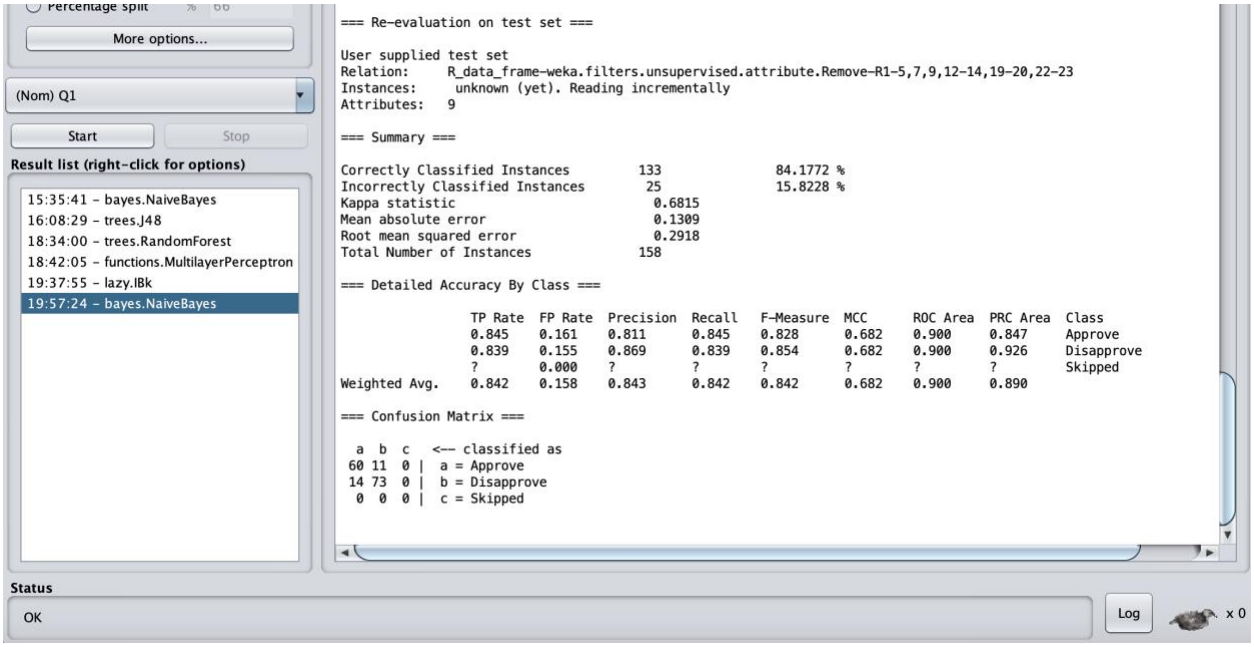
- Step 1: Open “Initial_Training_set.arff”, remove all attributes except Q3, QPID, ppethm, Q2, pphouse, ppmsacat, ppmarit, Q4, and Q1. Save it as “set4_training”

- Step 2: Open “Initial_Testing_set.arff”, remove all attributes except Q3, QPID, ppethm, Q2, pphouse, ppmsacat, ppmarit, Q4, and Q1. Save it as “set4_testing”

- Step 3: Open “set4_training”, Classify -> Choose NaiveBayes with Cross-validation (10 Folds), Q1 as class attribute -> Start



- Step 4: In order to test our model, we re-evaluate our model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set4_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

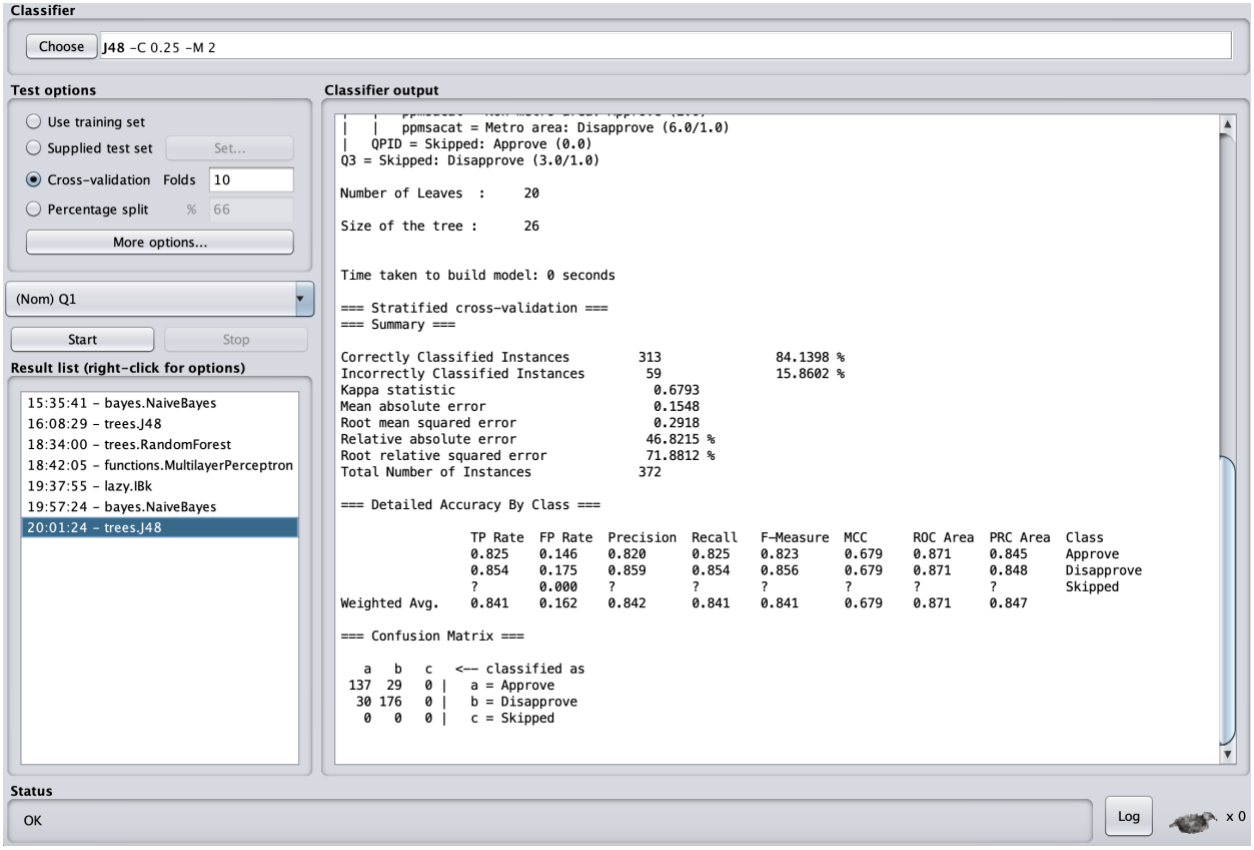


Naive Bayes - Cross Validation (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
85.75%	85.5%	14.1%	90.7%	Approve
	85.9%	13.9%	91.1%	Disapprove

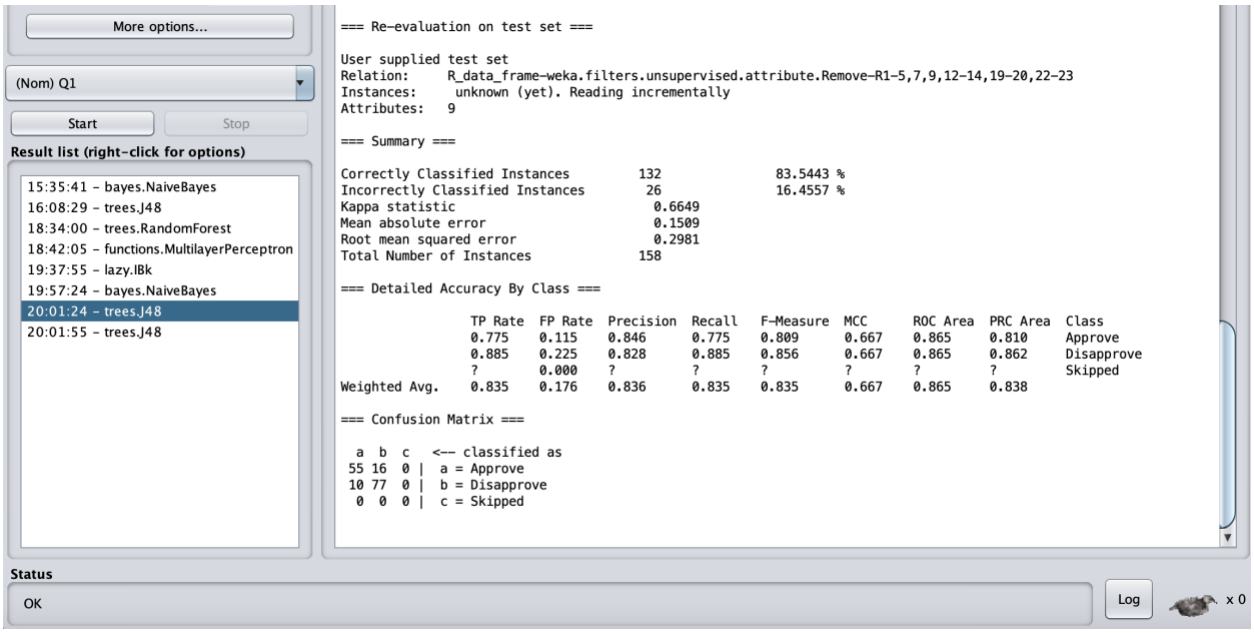
Naive Bayes - Test set (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.18%	84.5%	16.1%	90.0%	Approve
	83.9%	15.5%	90.0%	Disapprove

2. J48

- Step 1: First to assess how good J48 is, use Cross-validation to build our model. Choose J48 and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set4_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

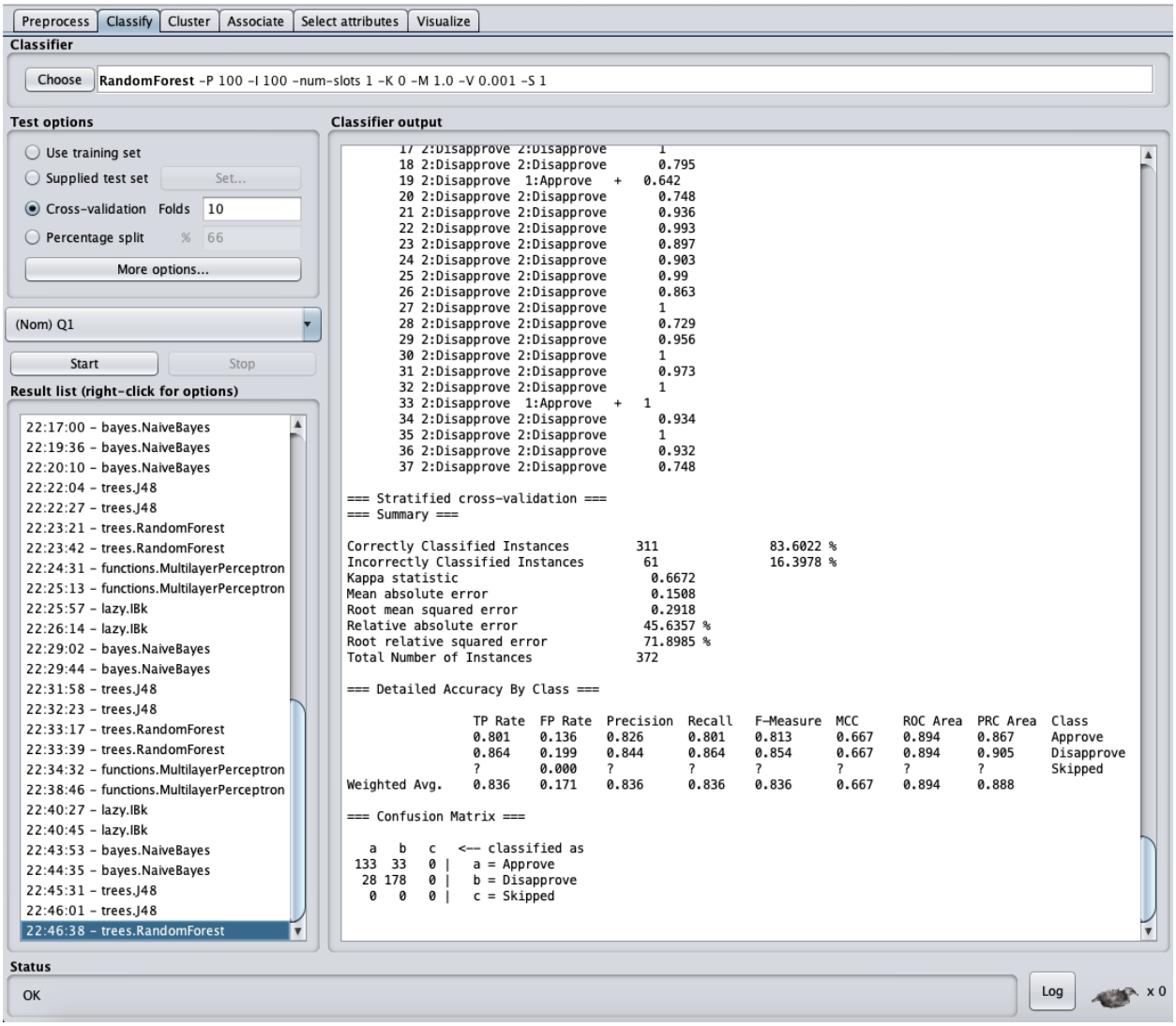


J48 - Cross Validation (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.14%	82.5%	14.6%	87.1%	Approve
	85.4%	17.5%	87.1%	Disapprove

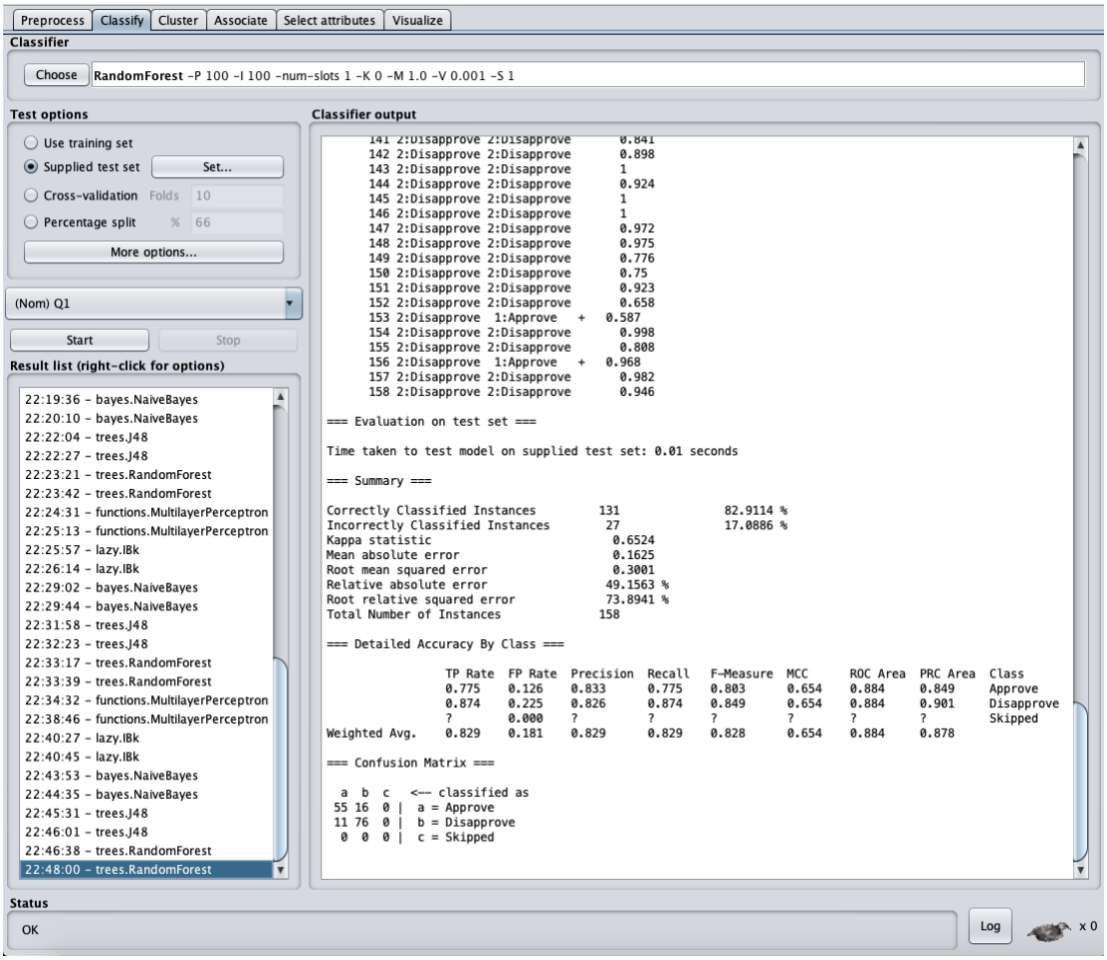
J48 - Test set (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
83.54%	77.5%	11.5%	86.5%	Approve
	88.5%	22.5%	86.5%	Disapprove

3. Random Forest

- Step 1: Choose RandomForest and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set4_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

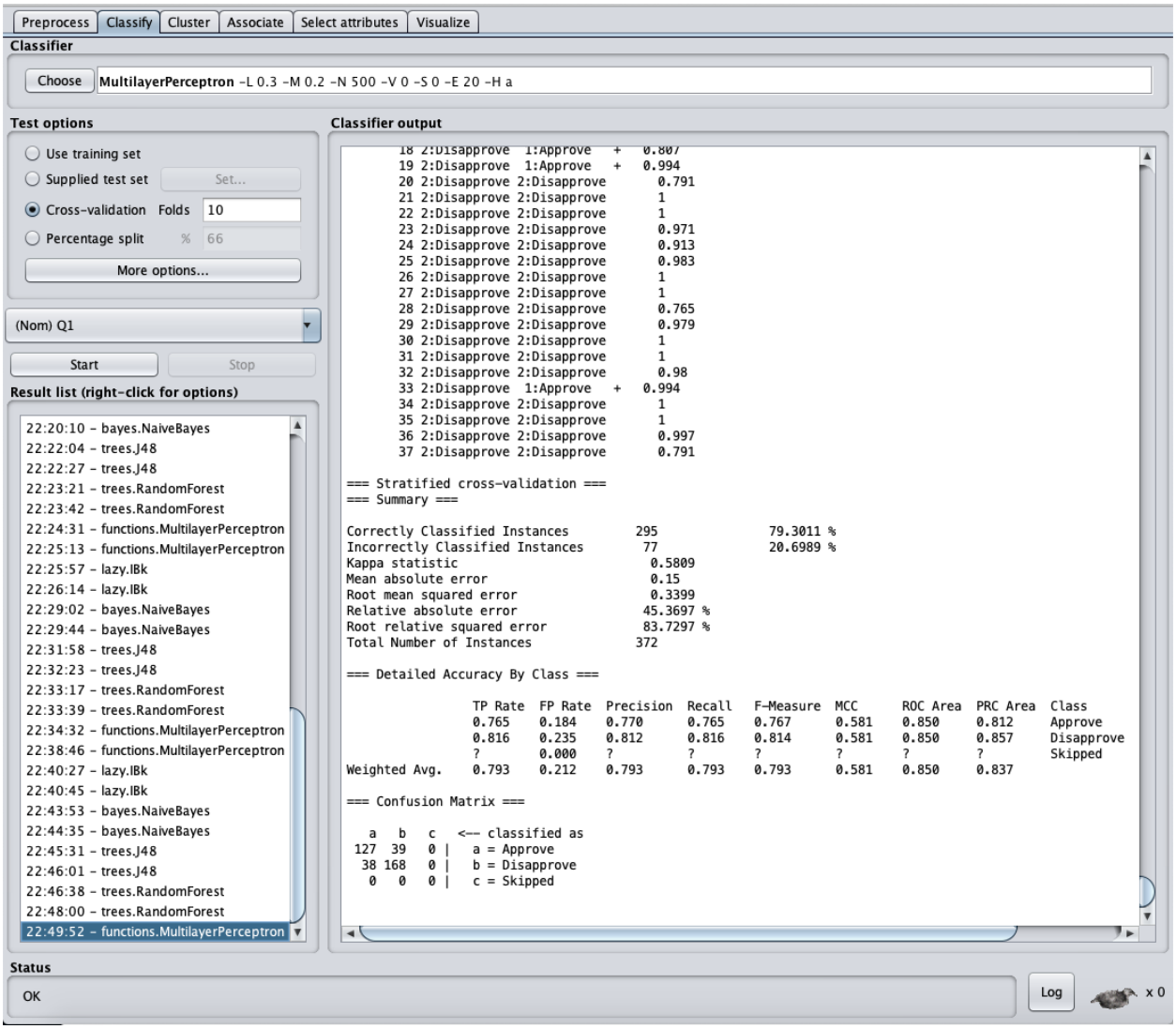


Random Forest - Cross Validation (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
83.60%	80.1%	13.6%	89.4%	Approve
	86.4%	19.9%	89.4%	Disapprove

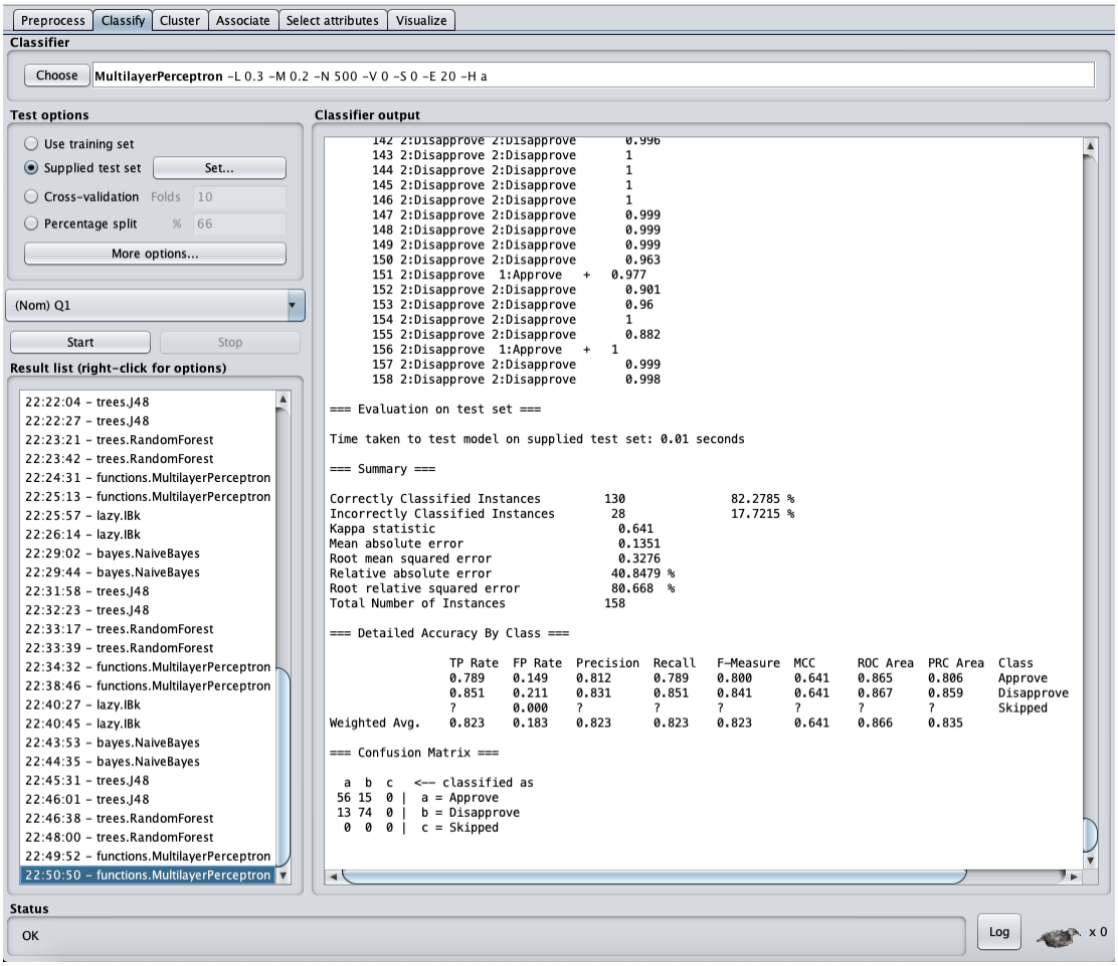
Random Forest - Test set (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
82.91%	77.5%	12.6%	88.4%	Approve
	87.4%	22.5%	88.4%	Disapprove

4. MultiLayerPerceptron

- Step 1: Choose MultilayerPerceptron and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set4_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

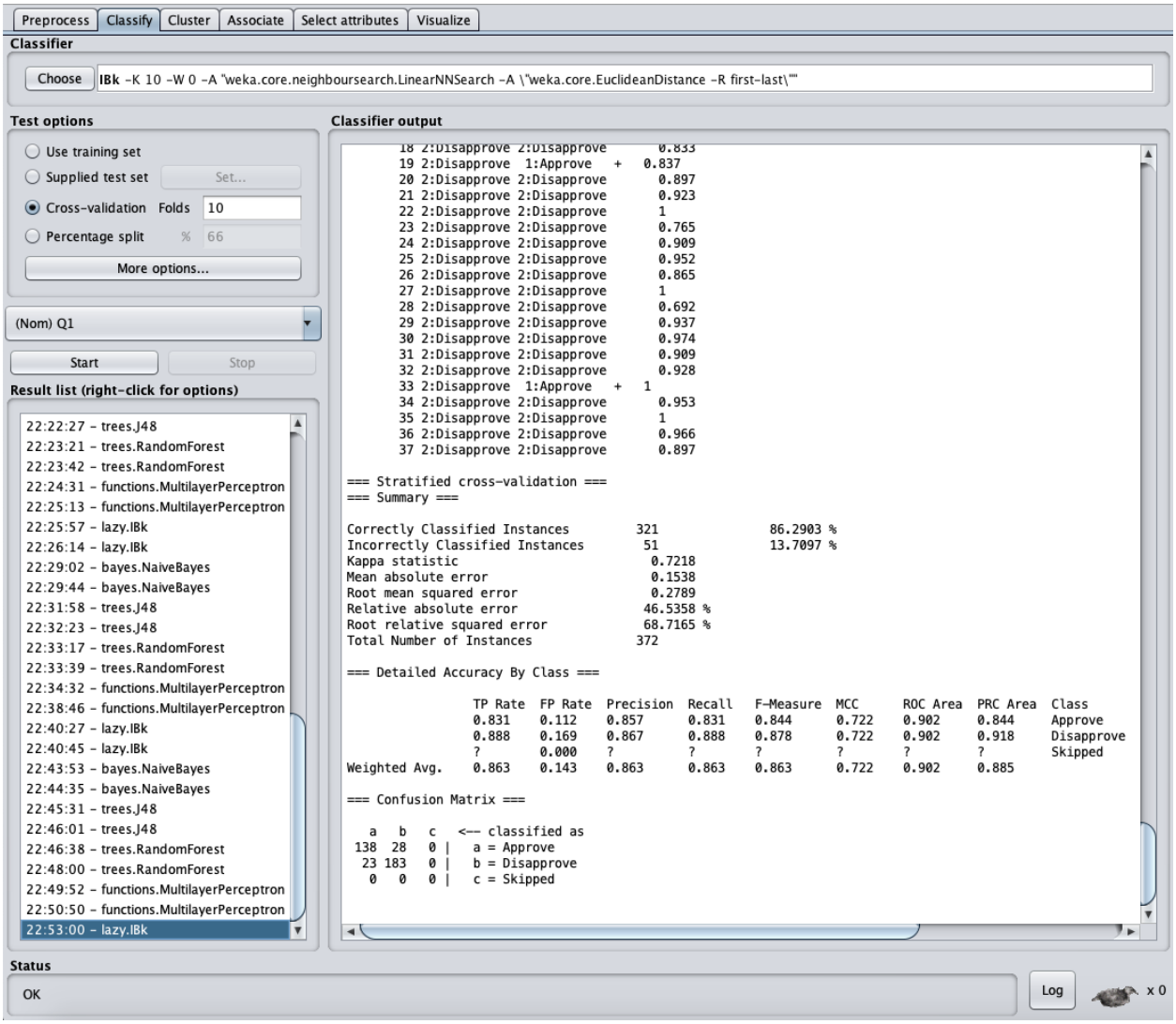


MultiLayerPerceptron - Cross Validation (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
79.30%	76.5%	18.4%	85.0%	Approve
	81.6%	23.5%	85.0%	Disapprove

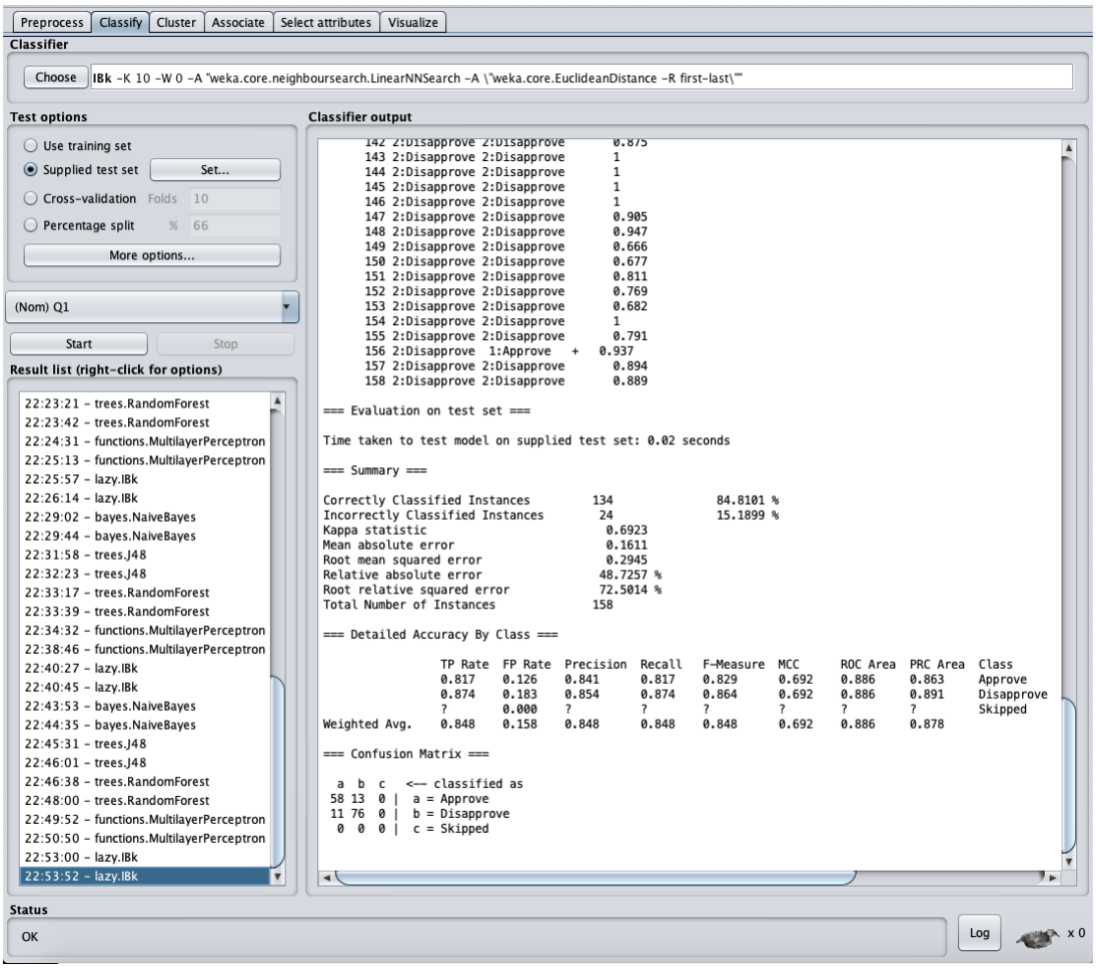
MultiLayerPerceptron - Test set (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
82.28%	78.9%	14.9%	86.5%	Approve
	85.1%	21.1%	86.7%	Disapprove

5. K-nearest Neighbor (IBK) with K = 10

- Step 1: Choose IBK and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set4_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



IBK - Cross Validation (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
86.29%	83.1%	11.2%	90.2%	Approve
	88.8%	16.9%	90.2%	Disapprove

IBK - Test set (Set 4)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	81.7%	12.6%	88.6%	Approve
	87.4%	18.3%	88.6%	Disapprove

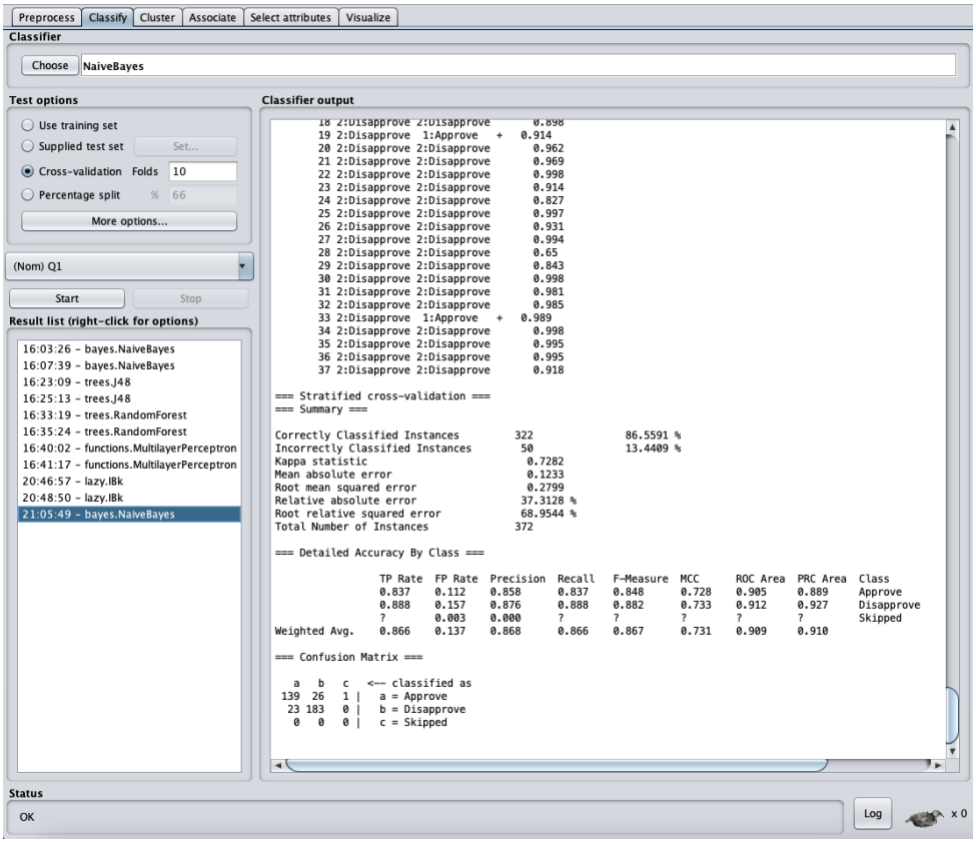
e. Attribute set 5: [Q3, QPID, Q2, Q4, Q5a, Q5b, ppethm, ppmarit, ppeducat, ppgender, ABCAGE, Q1] - by us

1. Naive Bayes:

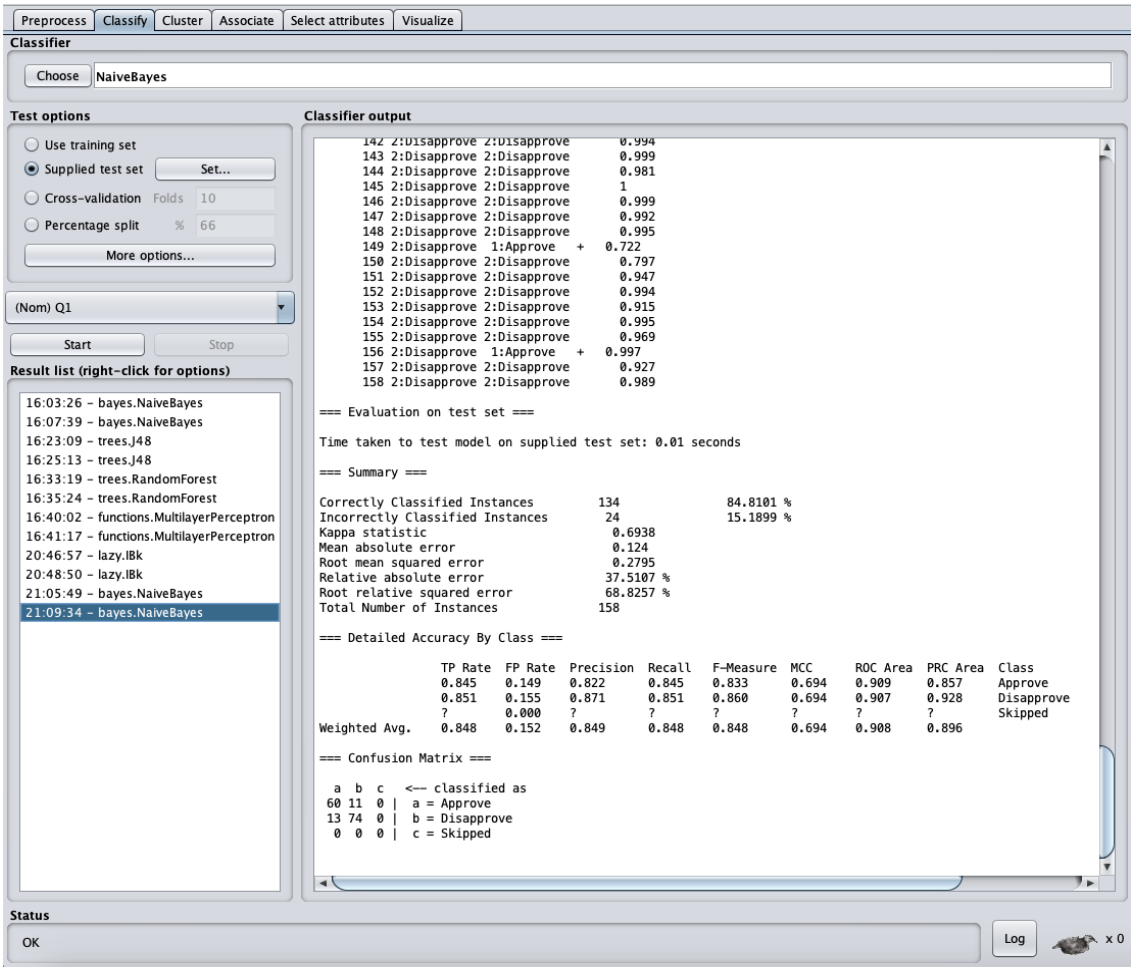
- Step 1: Open “Initial_Training_set.arff”, remove all attributes except Q3, QPID, Q2, Q4, Q5a, Q5b, ppethm, ppmarit, ppeducat, ppgender, ABCAGE and Q1. Save it as “set5_training”

- Step 2: Open “Initial_Testing_set.arff”, remove all attributes except Q3, QPID, Q2, Q4, Q5a, Q5b, ppethm, ppmarit, ppeducat, ppgender, ABCAGE and Q1. Save it as “set4_testing”

- Step 3: Open “set5_training”, Classify -> Choose NaiveBayes with Cross-validation (10 Folds), Q1 as class attribute -> Start



- Step 4: In order to test our model, we re-evaluate our model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set5_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

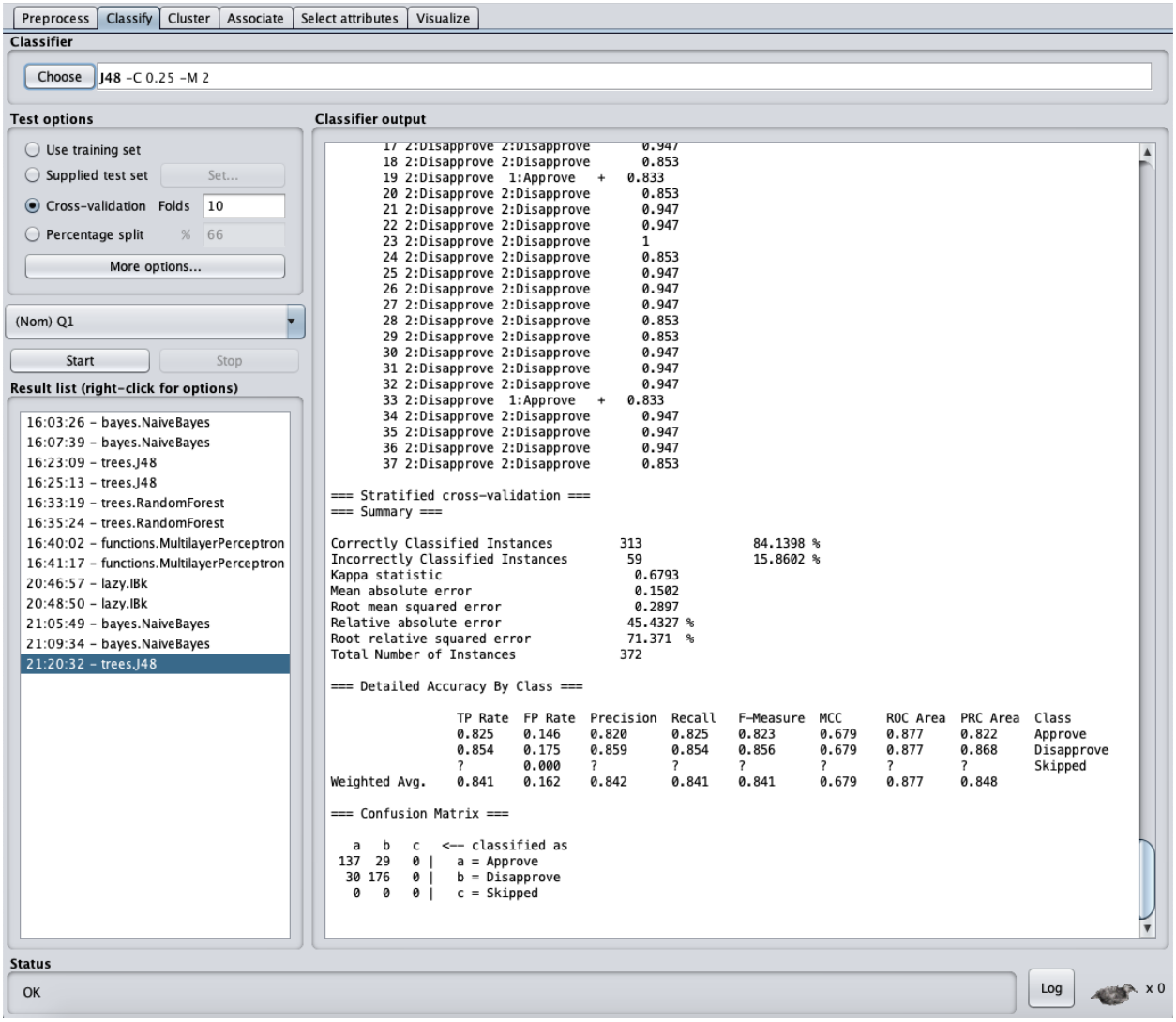


Naive Bayes - Cross Validation (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
86.56%	83.7%	11.2%	90.5%	Approve
	88.8%	15.7%	91.2%	Disapprove

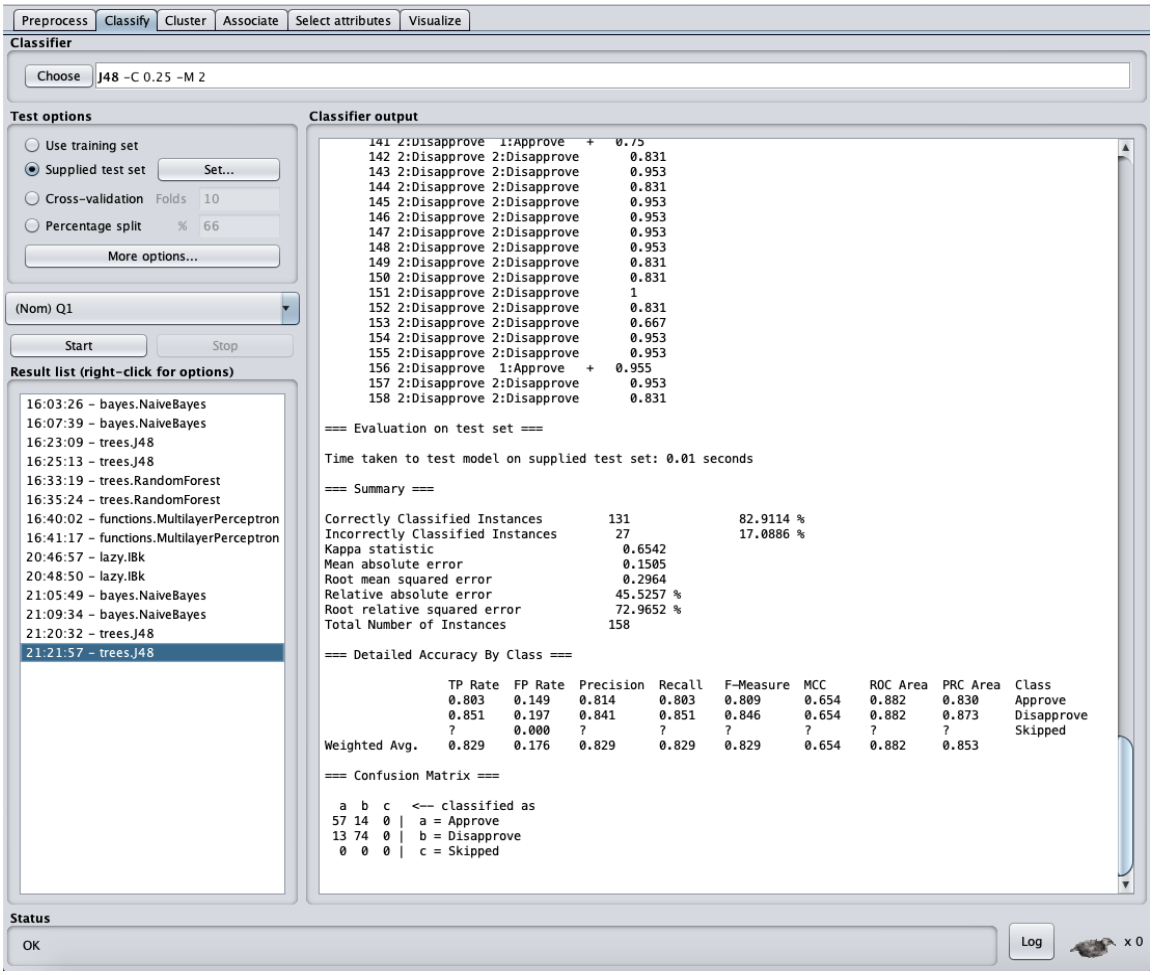
Naive Bayes - Test set (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	84.5%	14.9%	90.9%	Approve
	85.1%	15.5%	90.7%	Disapprove

2. J48

- Step 1: First to assess how good J48 is, use Cross-validation to build our model. Choose J48 and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set5_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

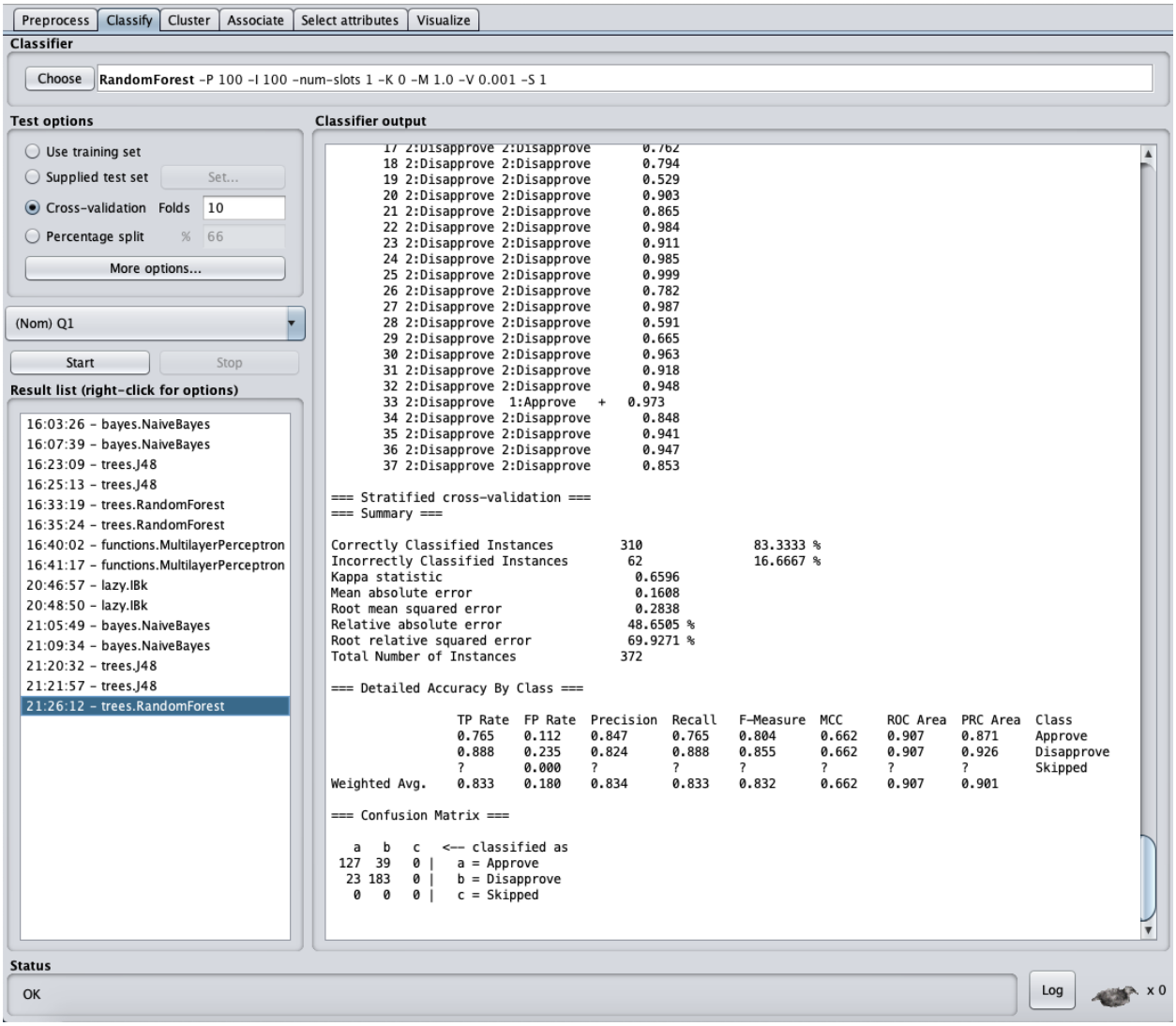


J48 - Cross Validation (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.14%	82.5%	14.6%	87.7%	Approve
	85.4%	17.5%	87.7%	Disapprove

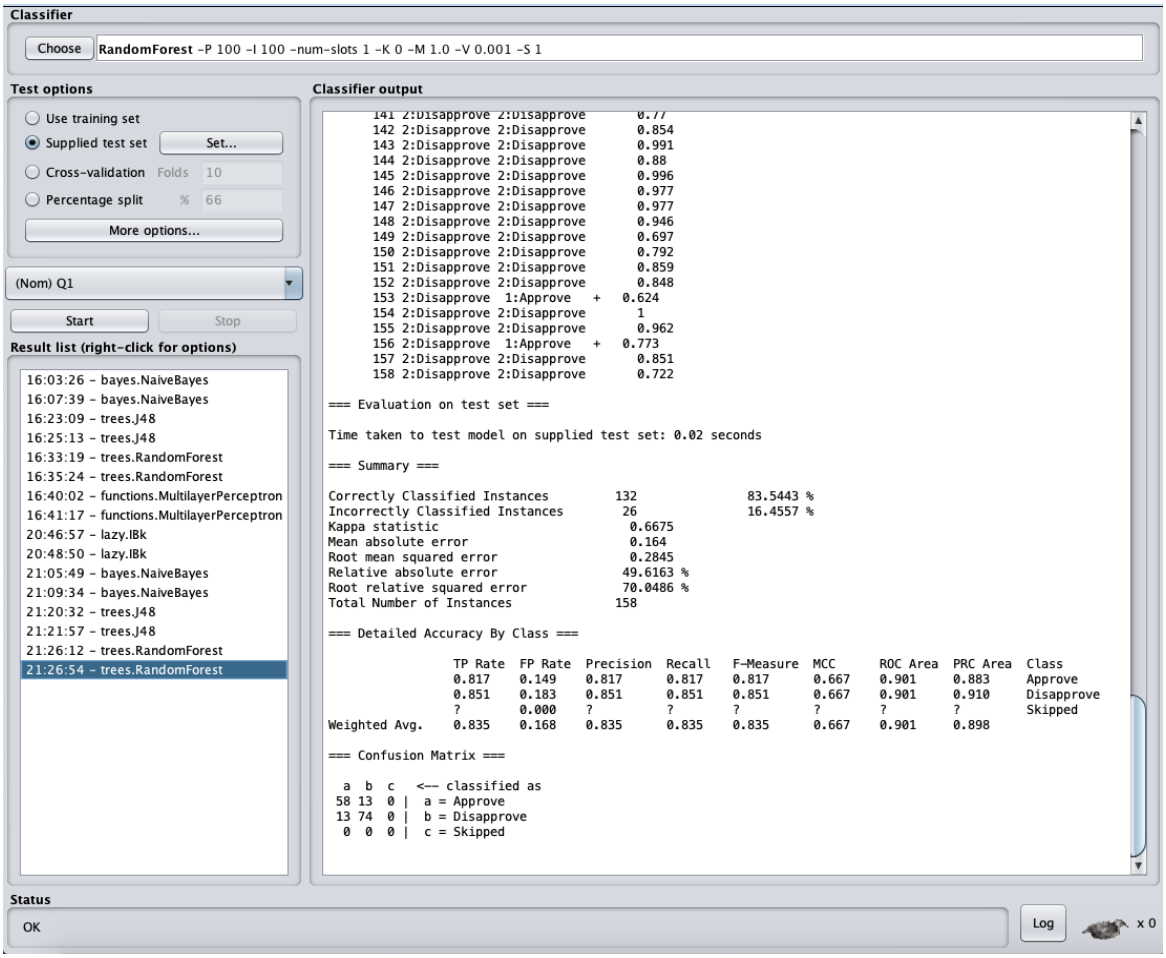
J48 - Test set (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
82.91%	80.3%	14.9%	88.2%	Approve
	85.1%	19.7%	88.2%	Disapprove

3. Random Forest

- Step 1: Choose RandomForest and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set5_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

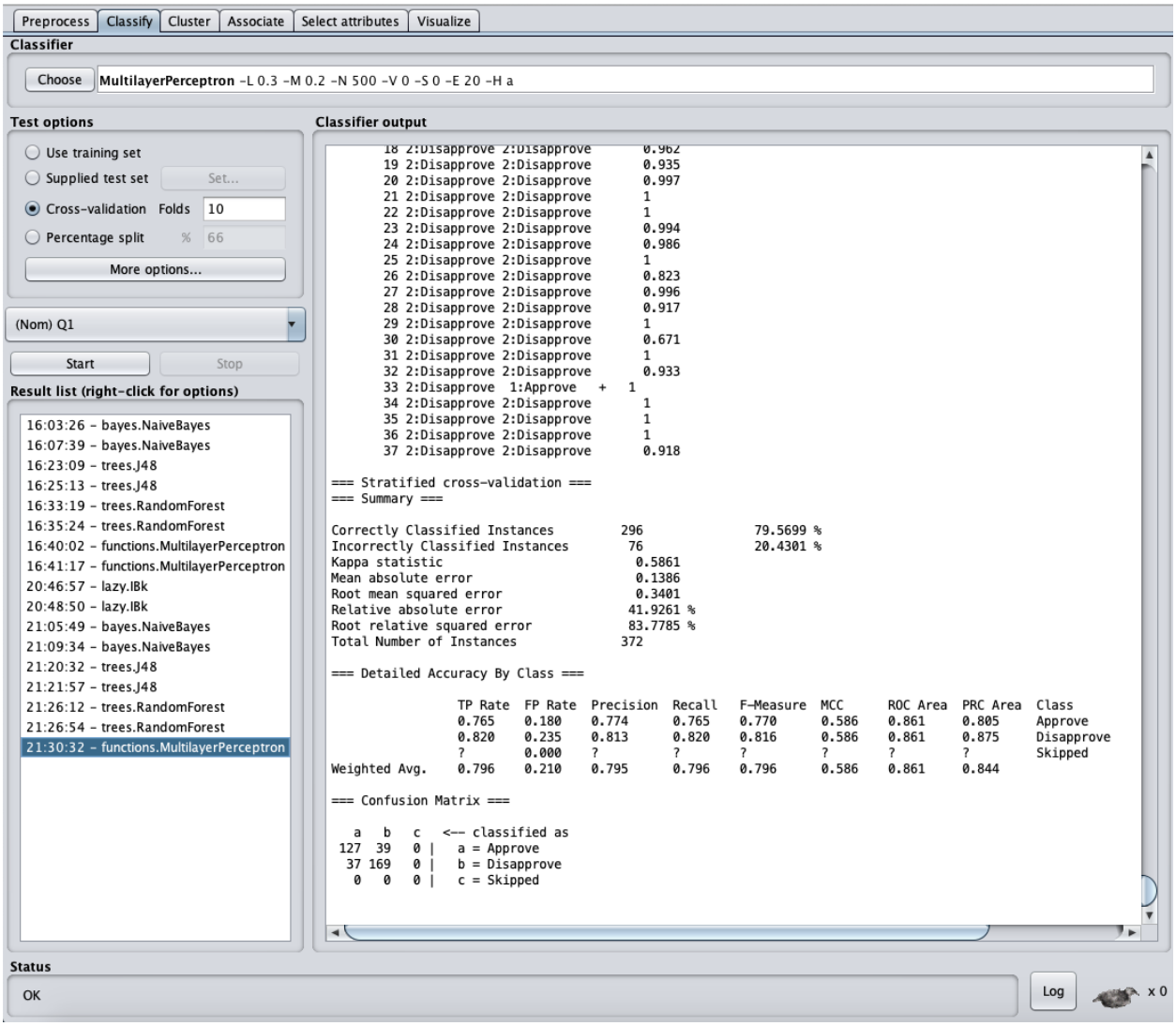


RandomForest - Cross Validation (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
83.33%	76.5%	11.2%	90.7%	Approve
	88.8%	23.5%	90.7%	Disapprove

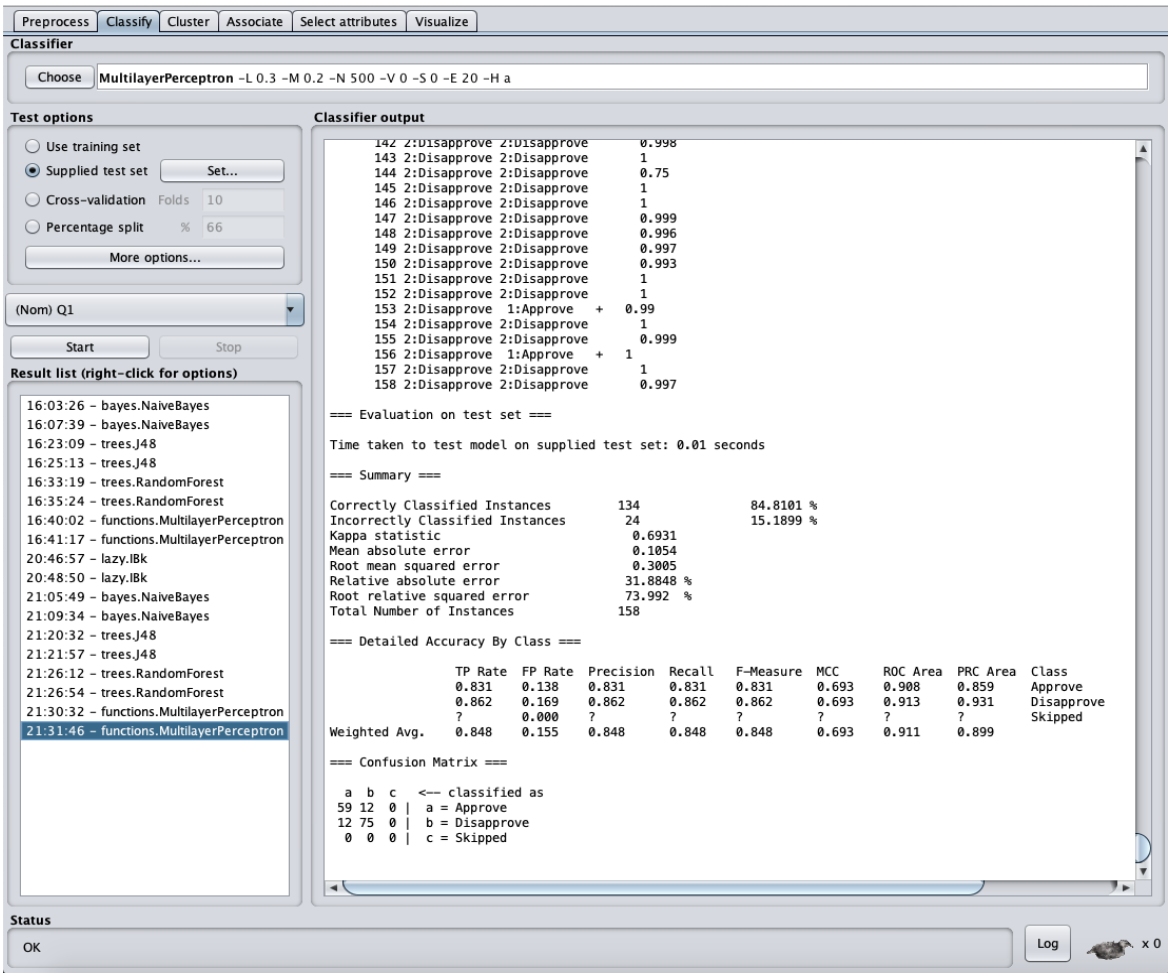
RandomForest - Test set (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
83.54%	81.7%	14.9%	90.1%	Approve
	85.1%	18.3%	90.1%	Disapprove

4. MultiLayerPerceptron

- Step 1: Choose MultilayerPerceptron and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set5_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start

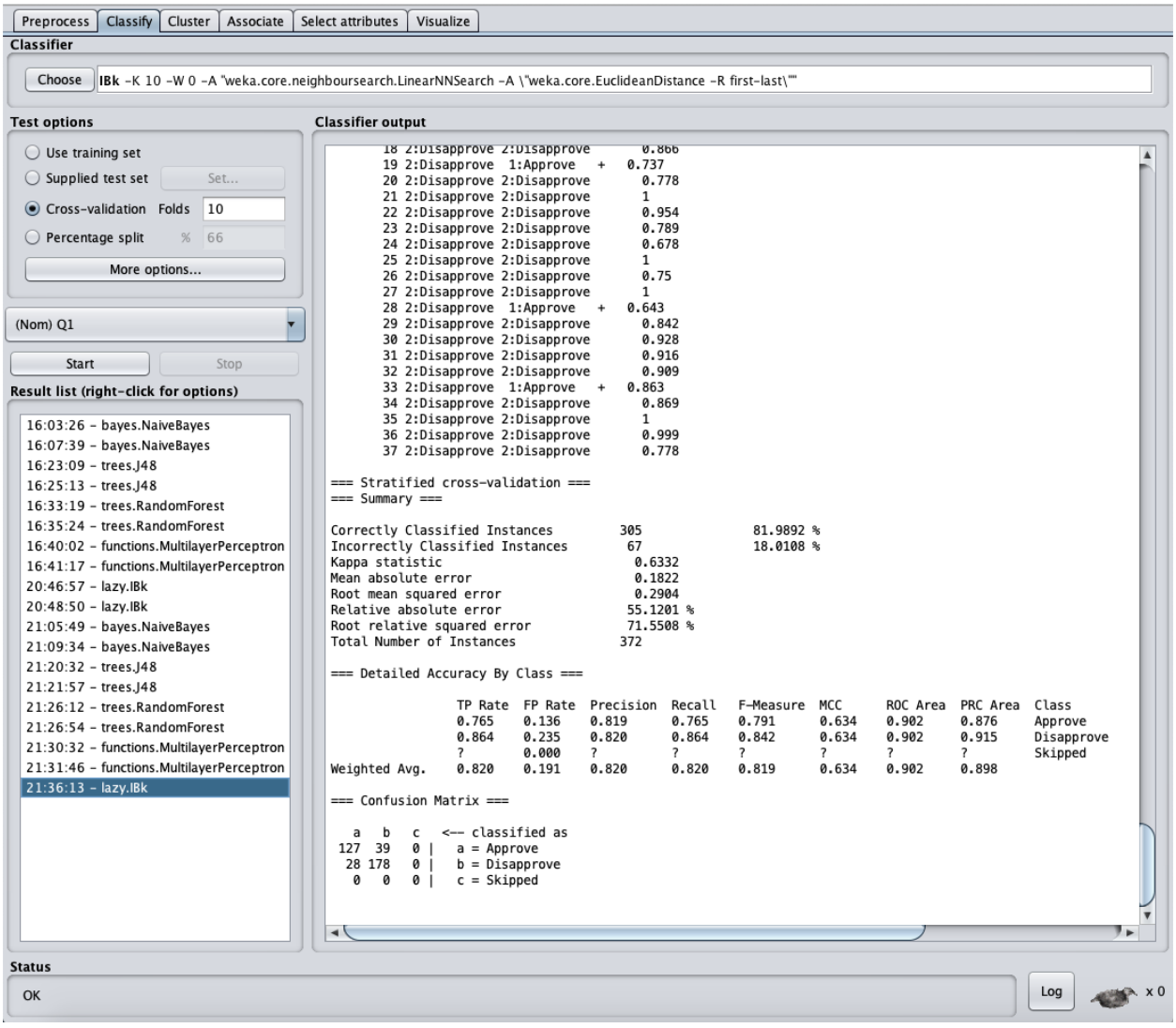


MultilayerPerceptron - Cross Validation (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
79.57%	76.5%	18.0%	86.1%	Approve
	82.0%	23.5%	86.1%	Disapprove

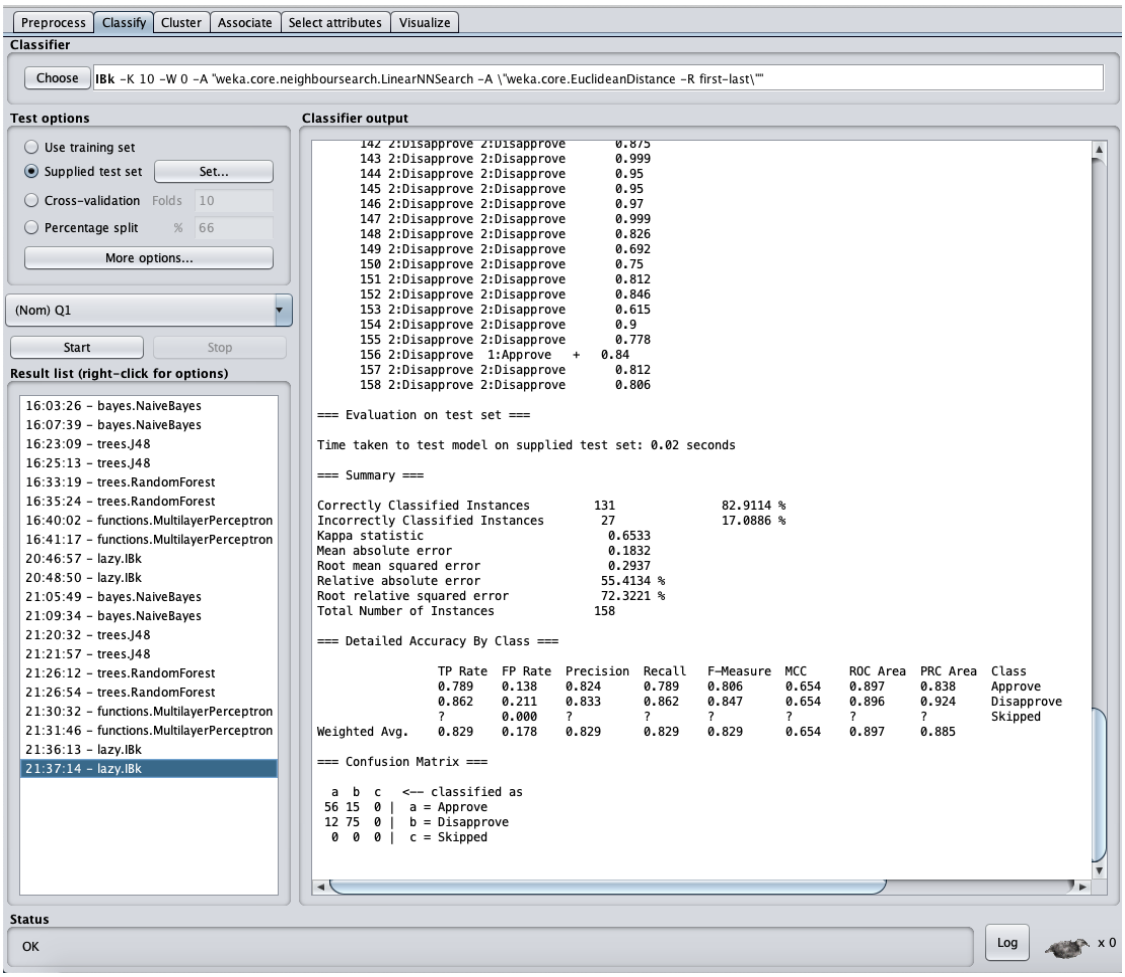
MultilayerPerceptron - Test set (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
84.81%	83.1%	13.8%	90.8%	Approve
	86.2%	16.9%	91.3%	Disapprove

5. K-nearest Neighbor (IBK) with K = 10

- Step 1: Choose IBK and select Cross-validation with 10 Folds and Q1 as Class attribute -> Start



- Step 2: re-evaluate the model with the test dataset. Select ‘Supplied test set’ and click ‘Set’ -> Open file, open “set5_testing” with Q1 as Class and close -> More options -> choose PlainText as Output prediction and OK -> Start



IBK - Cross Validation (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
81.99%	76.5%	13.6%	90.2%	Approve
	86.4%	23.5%	90.2%	Disapprove

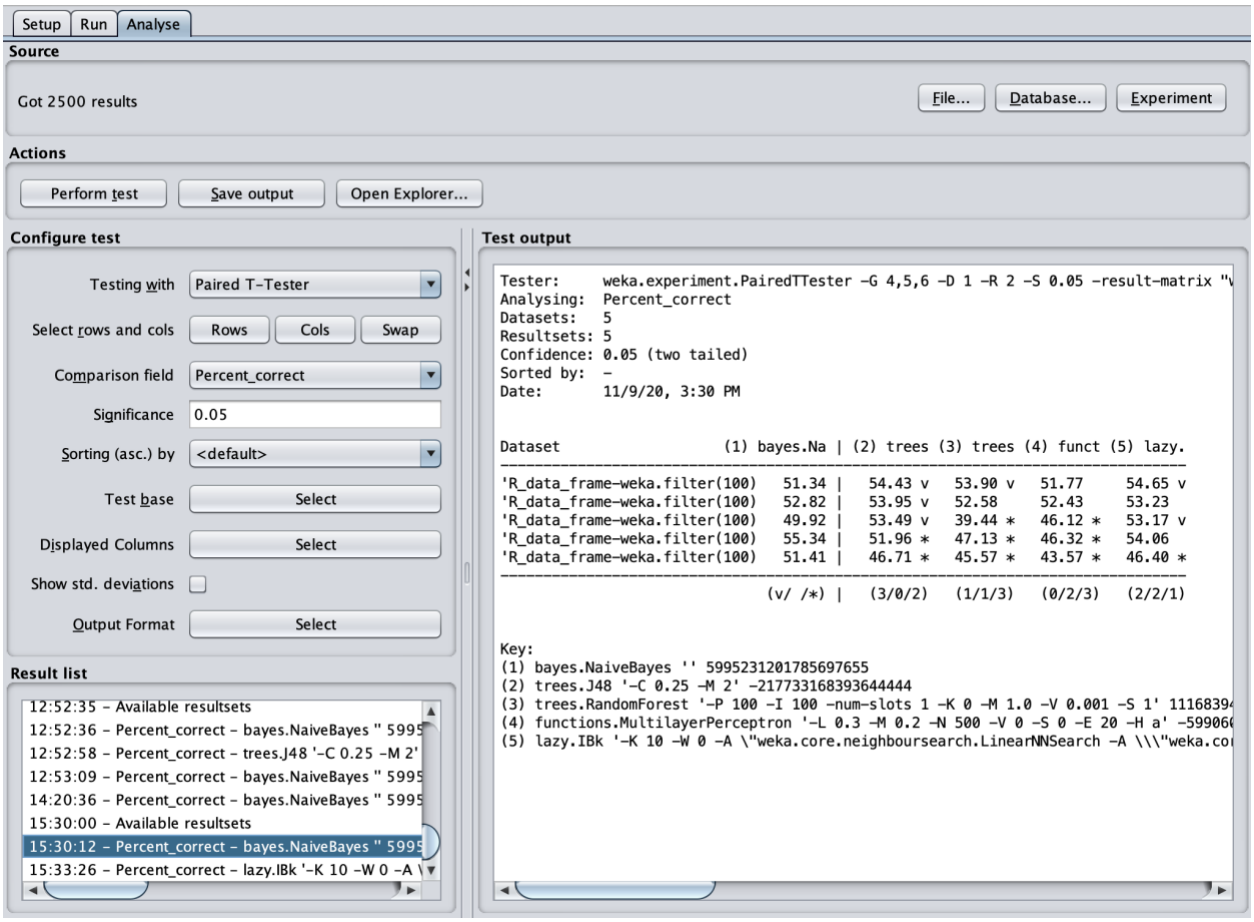
IBK - Test set (Set 5)				
Accuracy	TP Rate	FP Rate	ROC Area	Class
82.91%	78.9%	13.8%	89.7%	Approve
	86.2%	21.1%	89.6%	Disapprove

Performance Comparisons

To compare and rank the classifiers, we performed t-test comparison on all 5 attribute sets for 4 comparison fields - accuracy, FP Rate,TP Rate, ROC curve. First, the accuracy comparison is conducted as below:

- Step 1: Open ‘Experimenter’ -> Click ‘New’ -> Click ‘Add new’ under Dataset -> Select all of the five training datasets that are previously created and click ‘Open’ -> Click ‘Add new’ under Algorithms -> Click ‘Choose’ and select NaiveBayes and click OK. Do the same for the other four algorithms (make sure k = 10 for IBk algorithm)
- Step 2: Go to ‘Run’ tab and click “Start”. It will take an hour to finish the process with 0 error
- Step 3: Go to ‘Analyse’ tab and select ‘Paired T-Tester’ for ‘Testing with’, ‘Percent_correct’ for ‘Comparison field’ and significance of 0.05. Click the ‘Select’ next to ‘Test base’, select NaiveBayes algorithm and click ‘Perform test’ under ‘Actions’

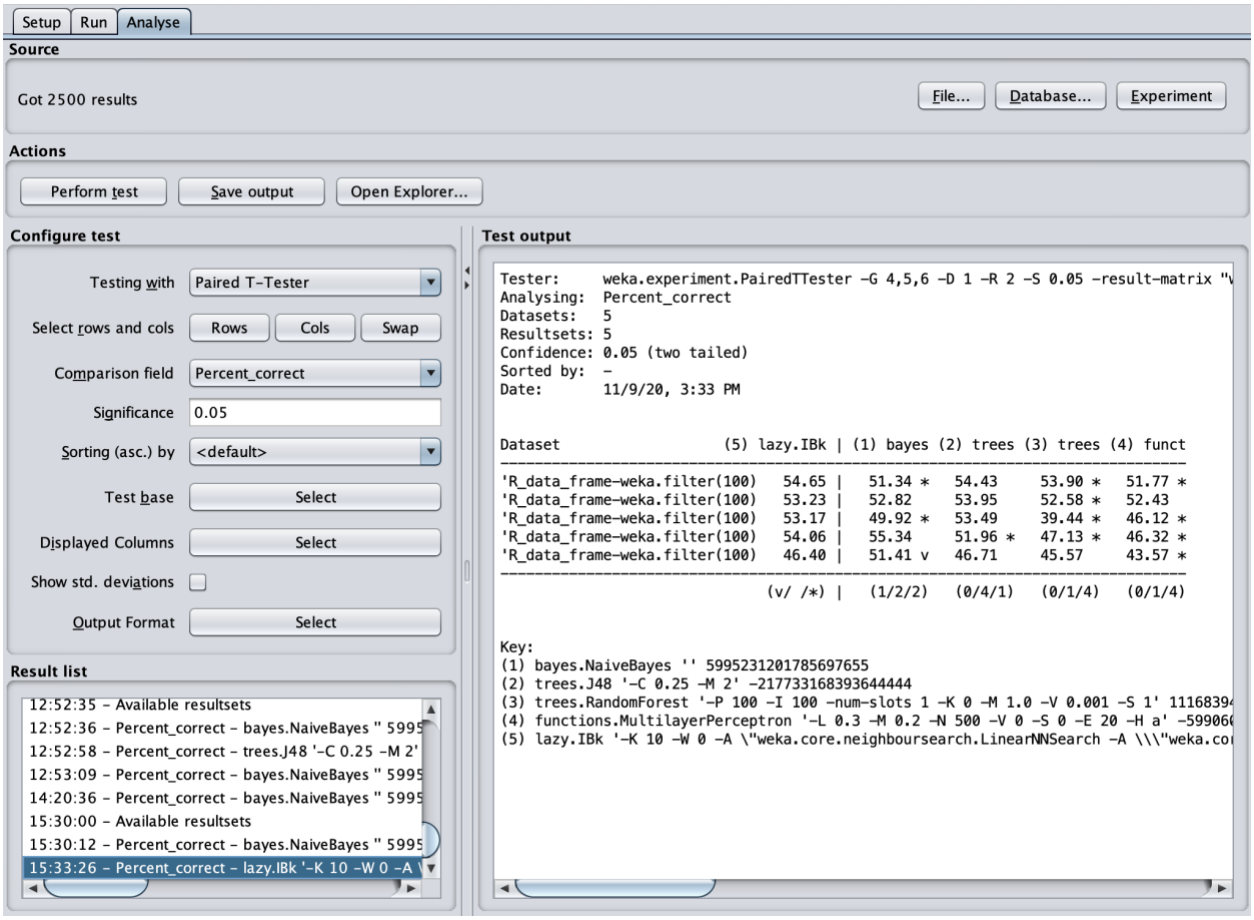
Below is the result of the t-test based on accuracy.



The test was performed with significance of 0.05 and Naive Bayes as the Test base.

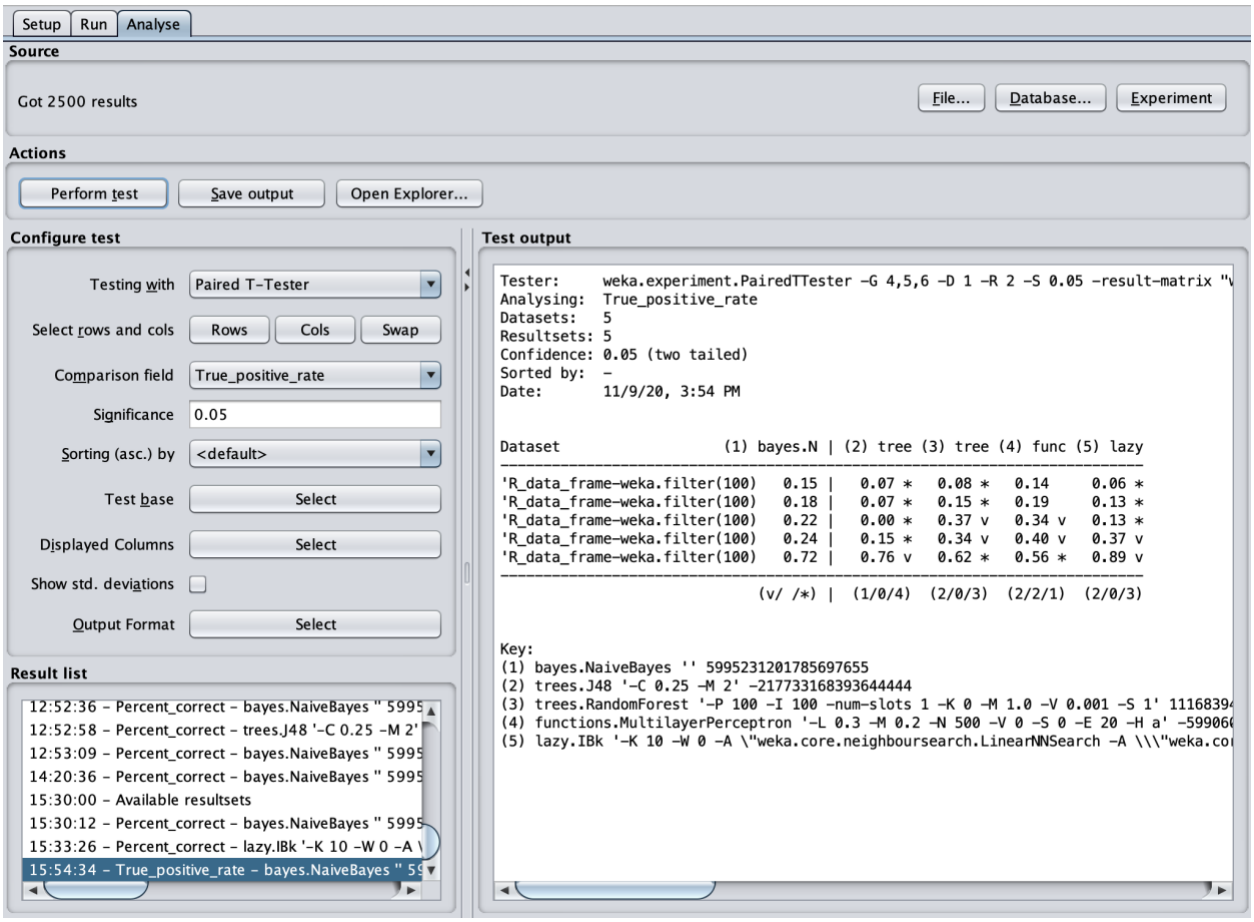
- J48 performed 3 times better than Naive Bayse and 2 times worse.
- Random Forest performed better once, 3 times worse, and 1 time equal to Naive Bayes.
- MultiLayer Perceptron performed 3 times worse than Naive Bayes, and none better.
- K-nearest Neighbor performed 2 times better and once worse.

To determine the rank between K-nearest Neighbor and J48, we redid the test with K-nearest Neighbor as Test base.



J48 performed once worse than K-Nearest Neighbor and 4 times equally the same. Therefore, the final ranking based on accuracy from high to low is **KNN, J48, Naive Bayes, Random Forest, Multilayer Perceptron.**

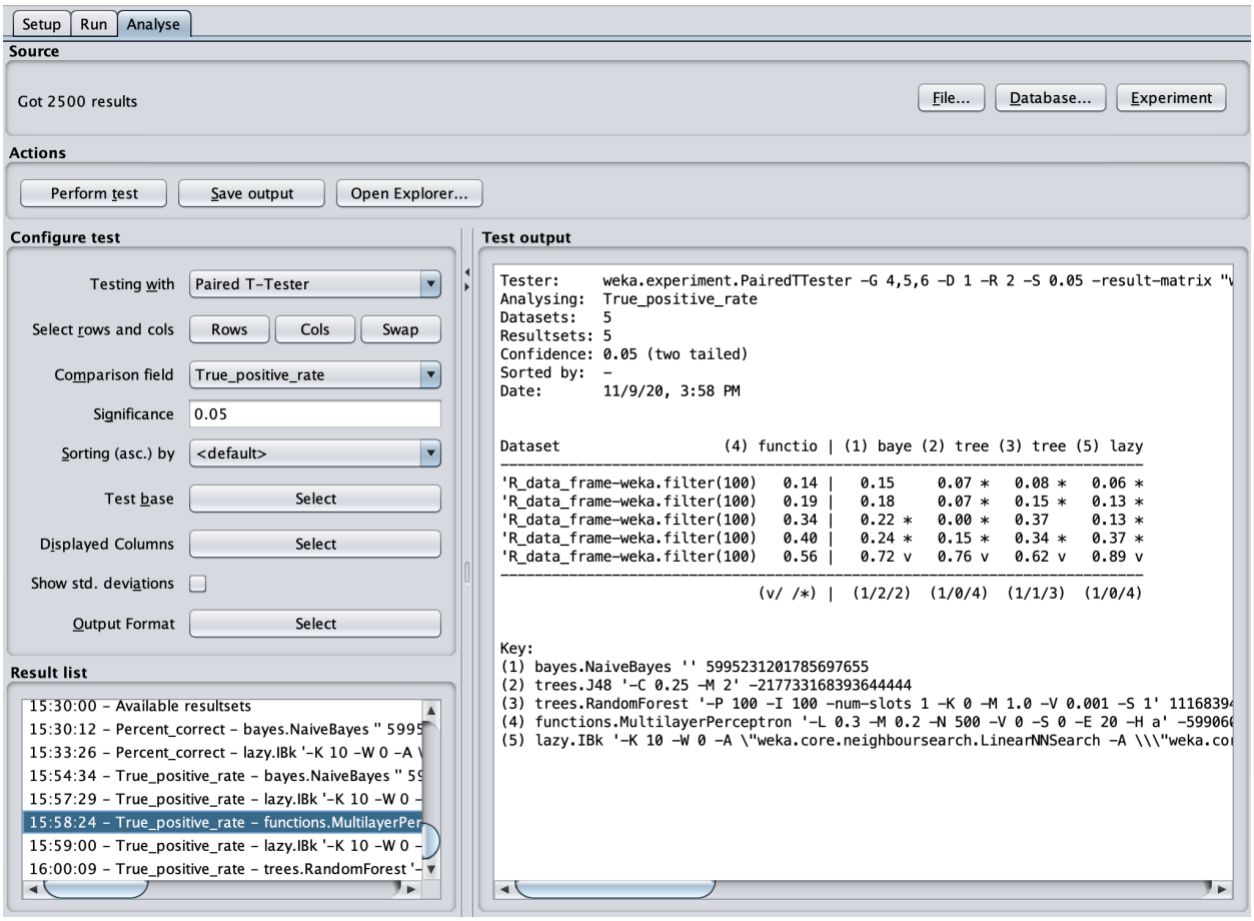
Second, TP Rate as the comparison field is performed as below.



From TP Rate perspective,

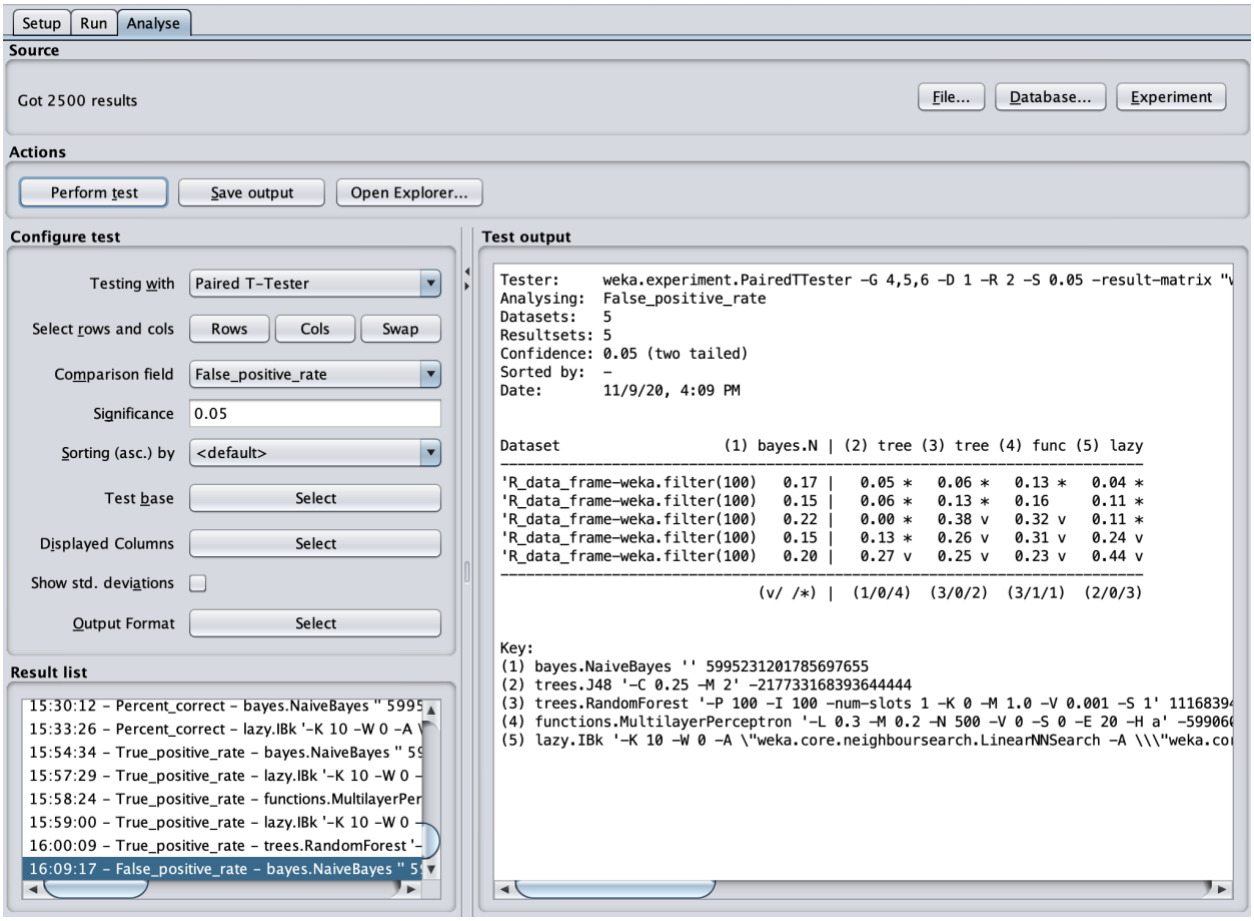
- J48 performed once better than Naive Bayse, and 4 times worse.
- Random Forest performed twice better, and 3 times worse.
- MultiLayer Perceptron performed 2 times better, 2 times equally, and once worse than Naive Bayes.
- KNN performed twice better, and 3 times worse than Naive Bayes.

If we change our Test base to Multilayer Perceptron, the result returns as below.



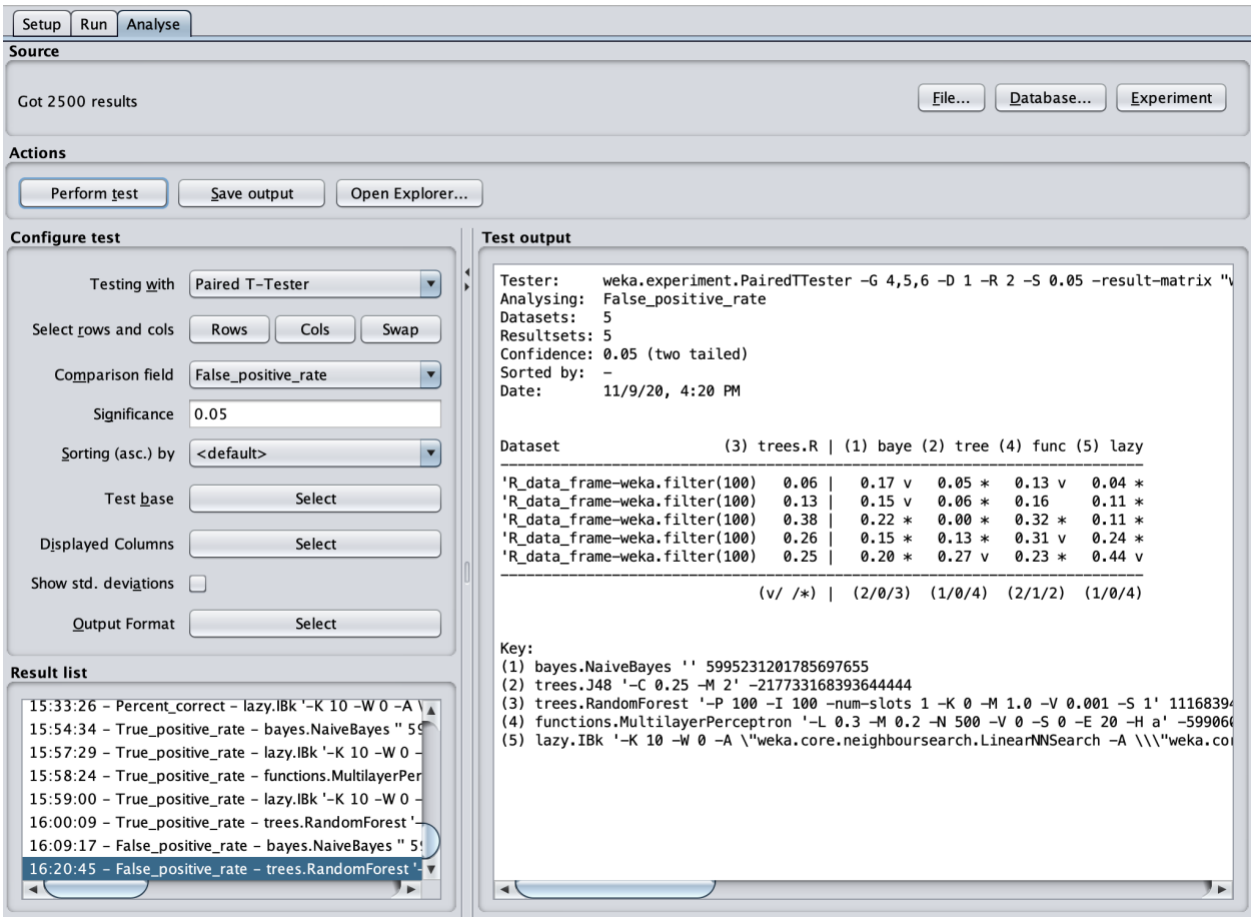
The final ranking based on the true positive rate from high to low is **Multilayer Perceptron, Naive Bayes, Random Forest, KNN, and J48.**

Third, the comparison field of FP Rate implemented as below.



As shown above, Random Forest and Multilayer Perceptron have higher false positive rate than that of Naive Bayes 3 times, and KNN's false positive rate is higher than Naïve Bayes' twice. Meanwhile, J48 performed better 4 times.

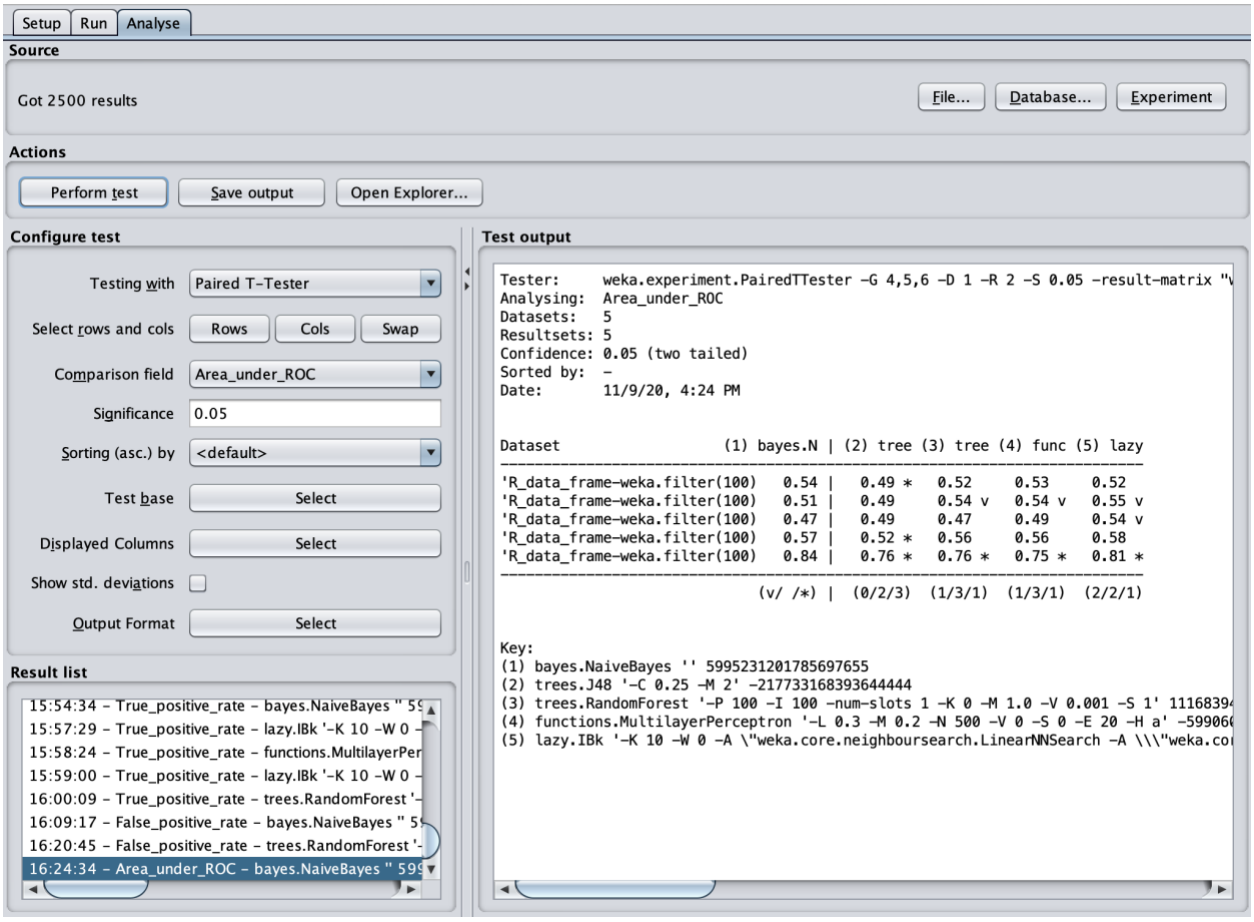
To compare MultiLayer Perceptron and Random Forest, we have to change the Test base to Random Forest.



As we can see, MultiLayer Perceptron has performed better twice, twice worse than, and once equally to Random Forest.

In sum, the final ranking based on the false positive rate from best to worst (best indicating the classifier with lowest FP rate) is **J48, KNN, Naïve Bayes, Random Forest, Multilayer Perceptron.**

Lastly, we want to compare the performances based on the area under the ROC curve.



With Naive Bayes as our Test base,

- J48 performed 3 times worse, never better.
- Random Forest and MultiLayer Perceptron both performed equally 3 times, once better and once worse than Naive Bayes.
- KNN performed twice better, once worse, and once equal to Naive Bayes.

Therefore, based on the area under the ROC curve, we can rank the models from best to worst as follows - **KNN, Naive Bayes, MultiLayer Perceptron, Random Forest, J48.**

Discussion and Conclusion

From the various attribute selection methods, we found that the most affective factors were QPID - political party the participant's support, and the response to question 3 - whether or not to open the country. This implies that the political orientation and the opinions in regards to when to reopen the US economy played more significant roles than any other factors (ex. ethnicity, age, education level, etc.) for the assessment of the Trump Administration's response.

Based on the results of the t-test we performed, each approach has come up with different results in terms of what algorithm works best. For example, KNN performed the best accuracy and highest area under ROC, while Multilayer Perceptron was the top model for TP rate and FP rate. However, KNN was second to the worst for the TP rate, while Multilayer Perceptron was the worst for accuracy.

Even though there's no absolute winner, we conclude that **K-Nearest Neighbors** is the best classifier in this case. It was wishy-washy for TP rate, but the result may potentially be better with different k values (i.e., different number of Neighbors). Moreover, it is true that accuracy is not always the best model evaluation metric. It can convey the health of a model well only when all the classes have similar prevalence in the data. However, since we split the dataset into train/test datasets with the same distribution of the class label, we have symmetric datasets (FN & FP counts are close). Since KNN was the best for the accuracy and area under ROC, we've ultimately chosen it as the best classifier model.