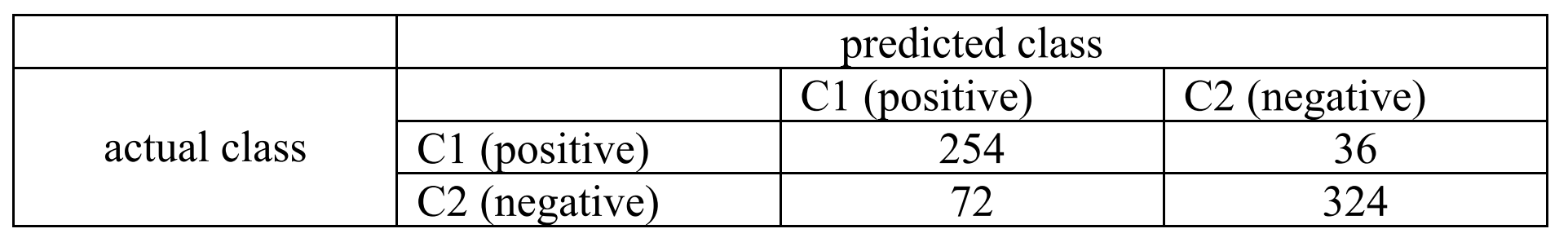
Yiduo Feng

CS 699

Assignment 4

Date: 6/2/2022

**Problem 1 (20 points)** Consider the following confusion matrix.



Compute *sensitivity*, *specificity*, *precision*, *accuracy*, *F-meassure*, *F*2, and MCC measures. You have to show all your calculations.

sensitivity = TP/P = 254/(254+36) = 0.8758621

specificity = TN/N = 324/(72+324) = 0.8181818

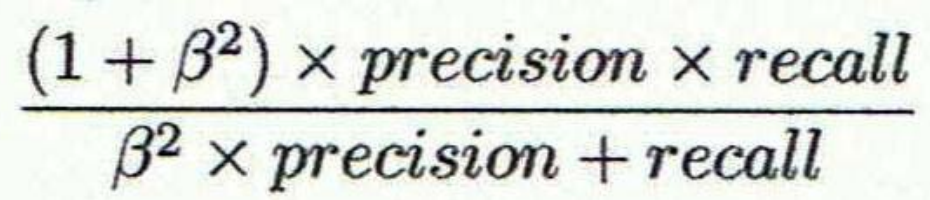
Precision = TP/(TP+FP) = 254/(254+72) = 0.7791411

Accuracy = (TP + TN)/All = (254+324)/(254+36+72+324) = 0.8425656

Recall = TP/(TP+FN) = 254/(254+36) = 0.8758621

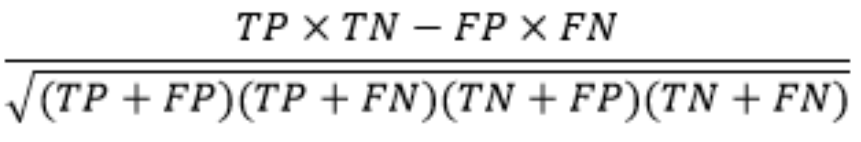
1. measure = (2\*Precision\*recall)/(Precision+recall)

= (2\*0.7791411\*0.8758621)/(0.7791411+0.8758621) = 0.8246753

F2 = 

= ((1+2^2)\*0.7791411\*0.8758621)/((2^2)\*0.7791411+0.8758621)

= 0.8546434

MCC measures = 

= (254\*324 - 72\*36)/sqrt((254+72)\*(254+36)\*(324+72)\*(324+36))

= 0.6865521

**Problem 2 (20 points)** Suppose you built two classifier models *M*1 and *M*2 from the same training dataset and tested them on the same test dataset using 10-fold cross-validation. The error rates obtained over 10 iterations (in each iteration the same training and test partitions were used for both *M*1 and *M*2) are given in the table below. Determine whether there is a significant difference between the two models using the statistical method discussed that we discussed in the class (this method is also discussed in Section 8.5.5, pp 372-373 of the textbook). Use a significance level of 1%. If there is a significant difference, which one is better?

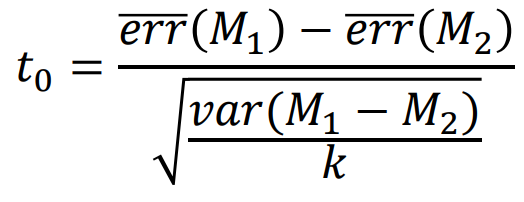
|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | M1 | M2 | M1-M2 |
| 1 | 0.12 | 0.08 | 0.04 |
| 2 | 0.12 | 0.1 | 0.02 |
| 3 | 0.15 | 0.22 | -0.07 |
| 4 | 0.15 | 0.1 | 0.05 |
| 5 | 0.03 | 0.07 | -0.04 |
| 6 | 0.17 | 0.11 | 0.06 |
| 7 | 0.2 | 0.1 | 0.1 |
| 8 | 0.14 | 0.11 | 0.03 |
| 9 | 0.1 | 0.17 | -0.07 |
| 10 | 0.14 | 0.11 | 0.03 |

Mean\_M1 = (0.12+0.12+0.15...+0.14)/10 = 0.132

Mean\_M2 = (0.08+0.1+0.22...+0.11)/10 = 0.117

Mean\_M1-2 = (0.04+0.03+...+0.03)/10 = 0.015

Var(M1-M2) = = 0.003228



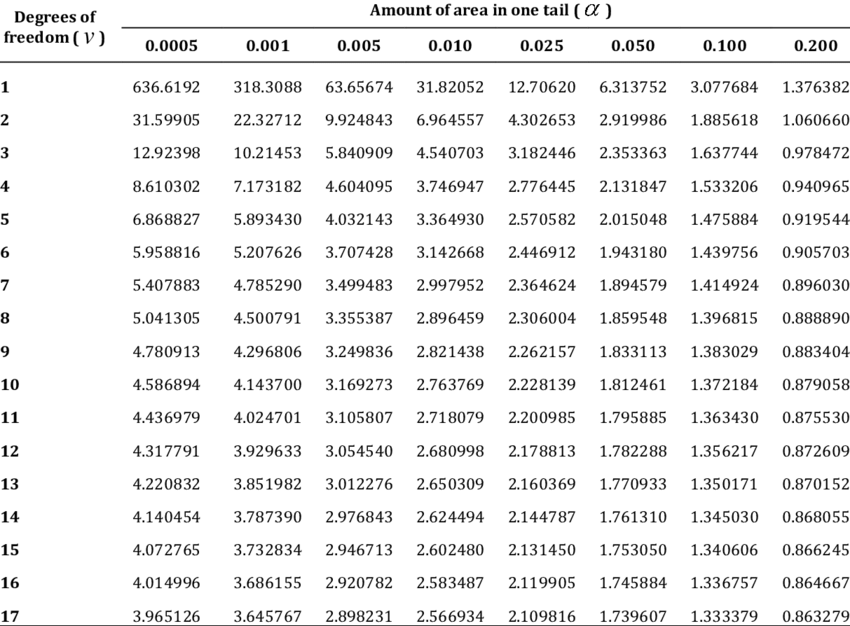
T0 = (0.132-0.117)/sqrt(0.003228/10) = 0.8348808

Alpha = 0.01

So alpha/2 = 0.005

K-1 = 9

According to the graph



T0.005,9 = 3.249836 > t0

So conclude that any difference is by chance.

**Problem 3 (20 points).** The following table shows a test result of a classifier on a dataset. Tuple\_id Actual Class Probability

1. . For each row, compute *TP*, *FP*, *TN*, *FN*, *TPR*, and *FPR*.

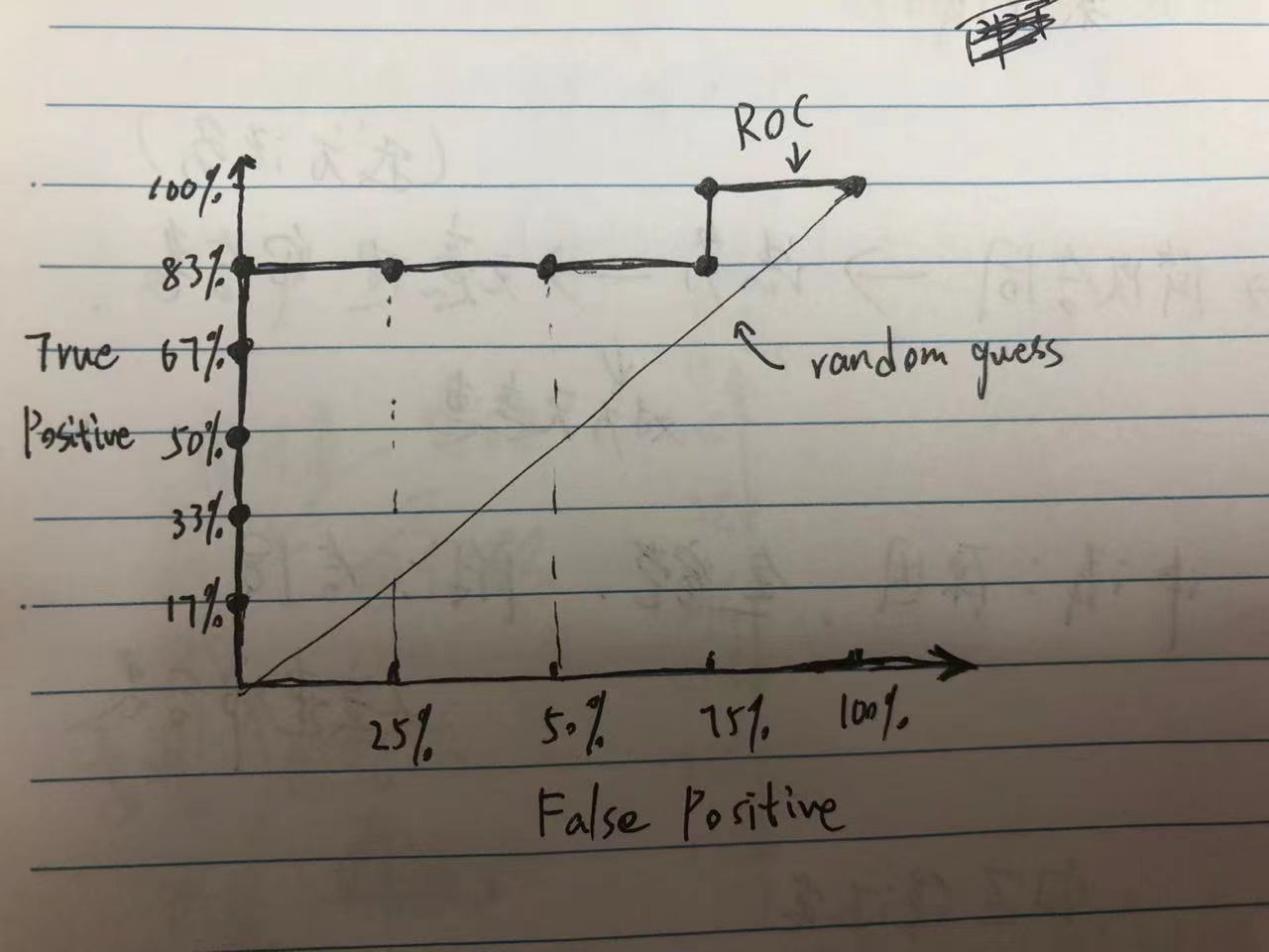
Rank the probability and calculate TP, FP, TN, FN, TPR, and FPR*.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tuple\_id | Actual Class | Probability | TP | FP | TN | FN | TPR | FPR |
| 10 | P | 0.96 | 1 | 0 | 4 | 5 | 0.166666667 | 0 |
| 1 | P | 0.92 | 2 | 0 | 4 | 4 | 0.333333333 | 0 |
| 4 | P | 0.92 | 3 | 0 | 4 | 3 | 0.5 | 0 |
| 6 | P | 0.89 | 4 | 0 | 4 | 2 | 0.666666667 | 0 |
| 5 | P | 0.83 | 5 | 0 | 4 | 1 | 0.833333333 | 0 |
| 9 | N | 0.82 | 5 | 1 | 3 | 1 | 0.833333333 | 0.25 |
| 7 | N | 0.79 | 5 | 2 | 2 | 1 | 0.833333333 | 0.5 |
| 3 | N | 0.76 | 5 | 3 | 1 | 1 | 0.833333333 | 0.75 |
| 8 | P | 0.73 | 6 | 3 | 1 | 0 | 1 | 0.75 |
| 2 | N | 0.7 | 6 | 4 | 0 | 0 | 1 | 1 |

If recover by the rank of tuple\_id, the table will be

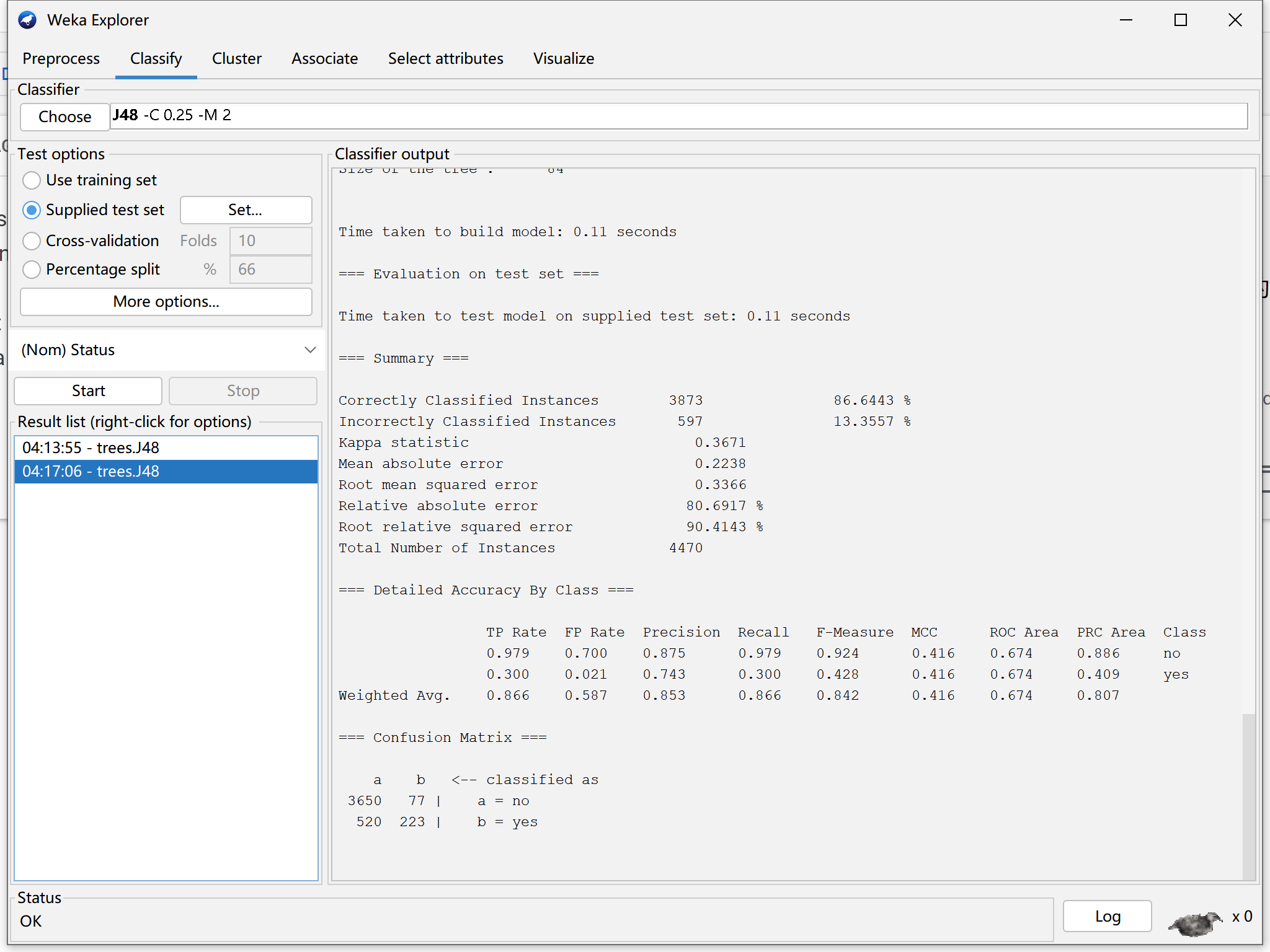
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tuple\_id | Actual Class | Probability | TP | FP | TN | FN | TPR | FPR |
| 1 | P | 0.92 | 2 | 0 | 4 | 4 | 0.333333333 | 0 |
| 2 | N | 0.7 | 6 | 4 | 0 | 0 | 1 | 1 |
| 3 | N | 0.76 | 5 | 3 | 1 | 1 | 0.833333333 | 0.75 |
| 4 | P | 0.92 | 3 | 0 | 4 | 3 | 0.5 | 0 |
| 5 | P | 0.83 | 5 | 0 | 4 | 1 | 0.833333333 | 0 |
| 6 | P | 0.89 | 4 | 0 | 4 | 2 | 0.666666667 | 0 |
| 7 | N | 0.79 | 5 | 2 | 2 | 1 | 0.833333333 | 0.5 |
| 8 | P | 0.73 | 6 | 3 | 1 | 0 | 1 | 0.75 |
| 9 | N | 0.82 | 5 | 1 | 3 | 1 | 0.833333333 | 0.25 |
| 10 | P | 0.96 | 1 | 0 | 4 | 5 | 0.166666667 | 0 |

1. . Plot the ROC curve for the dataset. You must draw the curve yourself (i.e., don’t use Weka, R, or other software to generate the curve).



**Problem 4 (20 points).** This problem is a small experiment of handling an unbalanced dataset for classification. Use *a3\_p4\_train.arff* and *a3\_p4\_test.arff* files and use J48 on Weka.

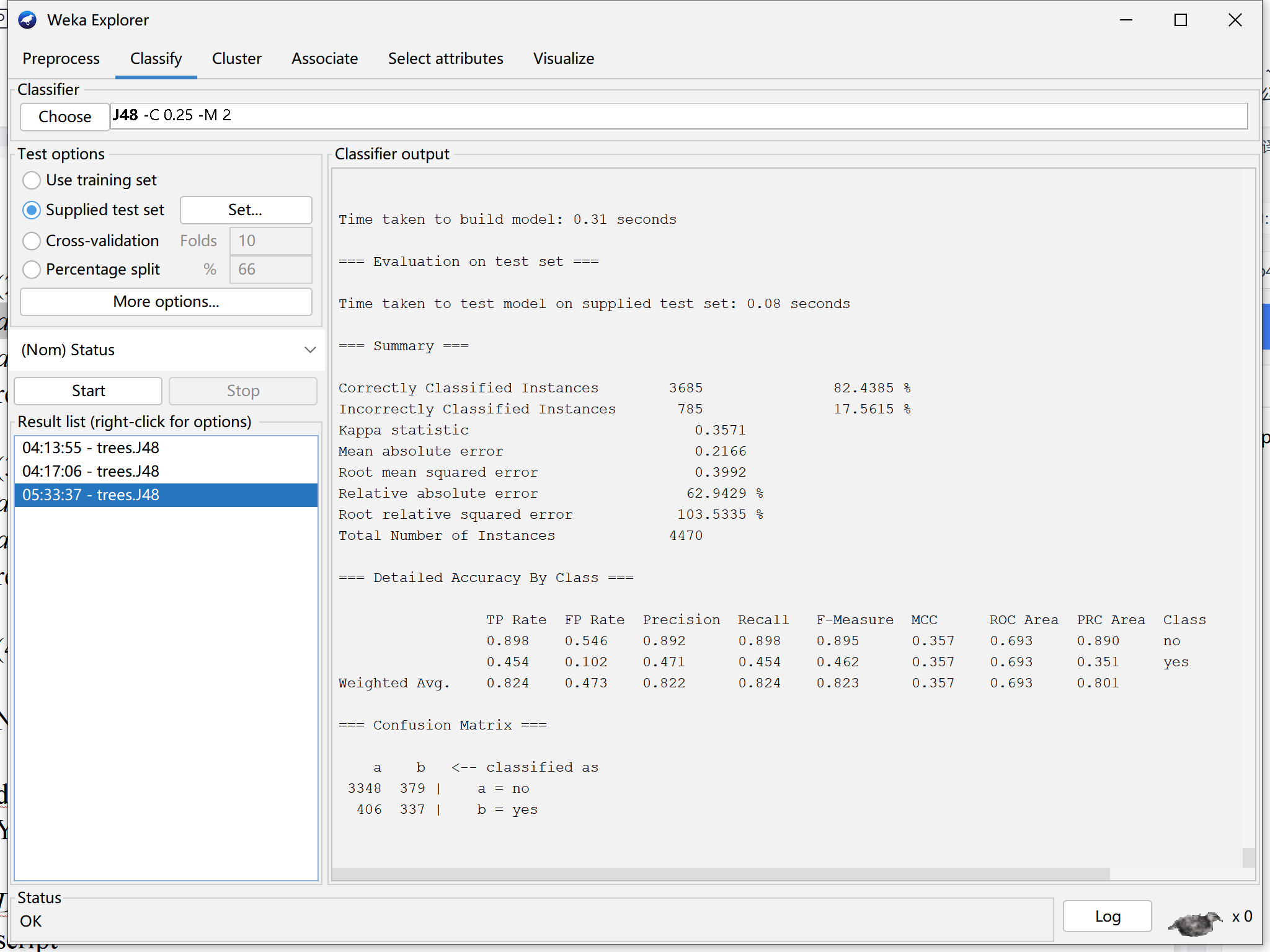
1. . Build a decision tree model from *a3\_p4\_train.arff* using J48 and test it on *a3\_p4\_test.arff*. Include the resulting confusion matrix in your submission.



(2). Create an undersampled training dataset from *a3\_p4\_train.arff* and name it

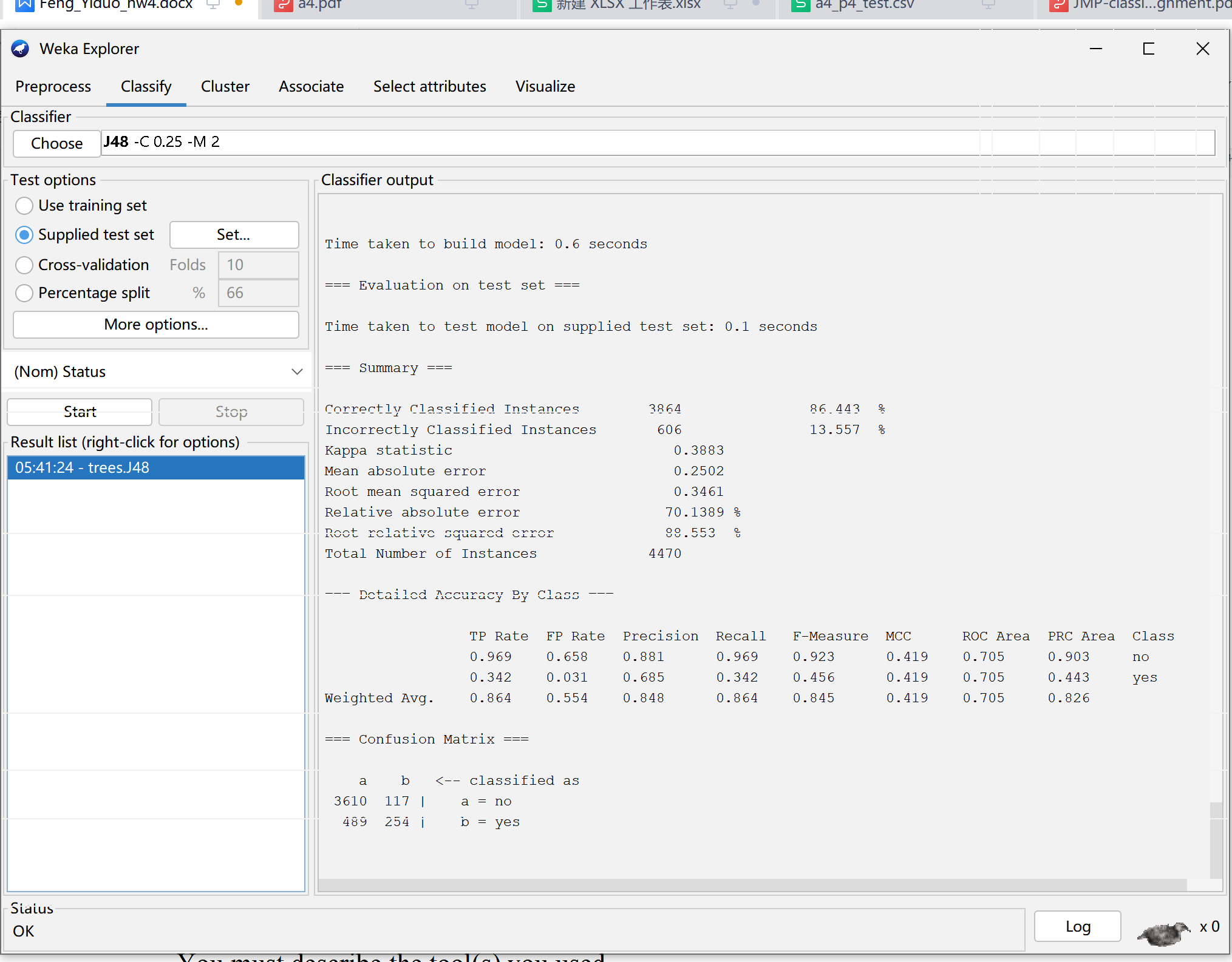
*a3\_p4\_train\_undersampled.arff*. Build a decision tree model from

*a3\_p4\_train\_undersampled.arff* using J48 and test it on *a3\_p4\_test.arff*. Include the resulting confusion matrix in your submission.



(3). Create an oversampled training dataset from *a3\_p4\_train.arff* and name it

*a3\_p4\_train\_oversampled.arff*. Build a decision tree model from *a3\_p4\_train\_oversampled.arff* using J48 and test it on *a3\_p4\_test.arff*. Include the resulting confusion matrix in your submission.

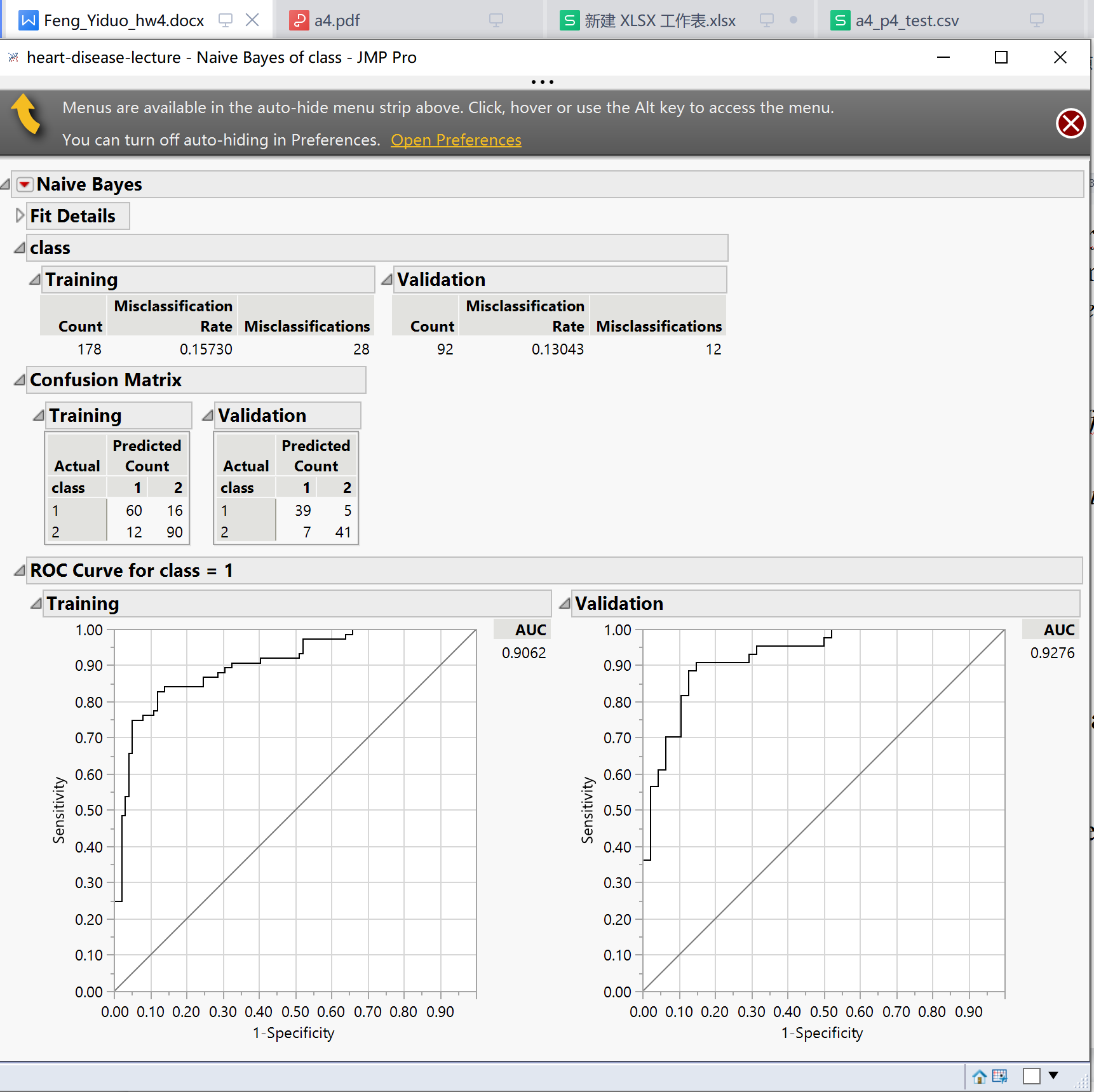


(4). What conclusion can you draw from this experiment?

Oversampling repeats proportional data, which does not actually introduce more data into the model. Overemphasizing proportional data will amplify the impact of proportional noise on the model. Undersampling discards a lot of data, and like oversampling, there is an overfitting problem. Both have slight changes to the accuracy, but the Confusion Matrix will be more balanced.

**Problem 5 (20 points).** Use JMP Pro to build and test five classifier models – Naïve Bayes, KNN, Partition (decision tree), Boosted Tree, and Neural Network – following the instruction in *JMP classification-assignment.pdf* file.

(1) Naïve Bayes



Class1:

TP Rate = TP/P = 39/44 = 0.8863636

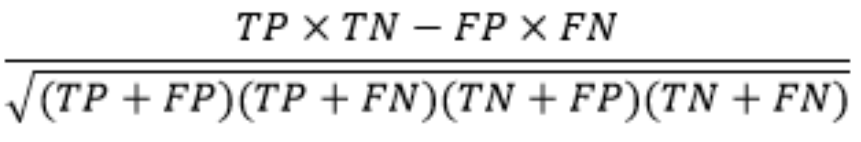
FP Rate = FP/N = 7/48 = 0.1458333

Precision =TP/(TP+FP) = 39/46 = 0.8478261

Recall = TP/(TP+FN) = 39/44 = 0.8863636

F-Measure = (2\*Precision\*recall)/(Precision+recall)

= (2\*0.8478261\*0.8863636)/(0.8478261+0.8863636) = 0.8666667

MCC = 

= (39\*41- 7\*5)/sqrt((39+7)\*(39+5)\*(41+7)\*(41+5)) = 0.73983

Class1:

TP Rate = TP/P = 39/44 = 0.8863636

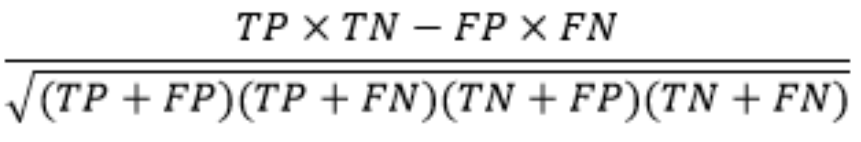
FP Rate = FP/N = 7/48 = 0.1458333

Precision =TP/(TP+FP) = 39/46 = 0.8478261

Recall = TP/(TP+FN) = 39/44 = 0.8863636

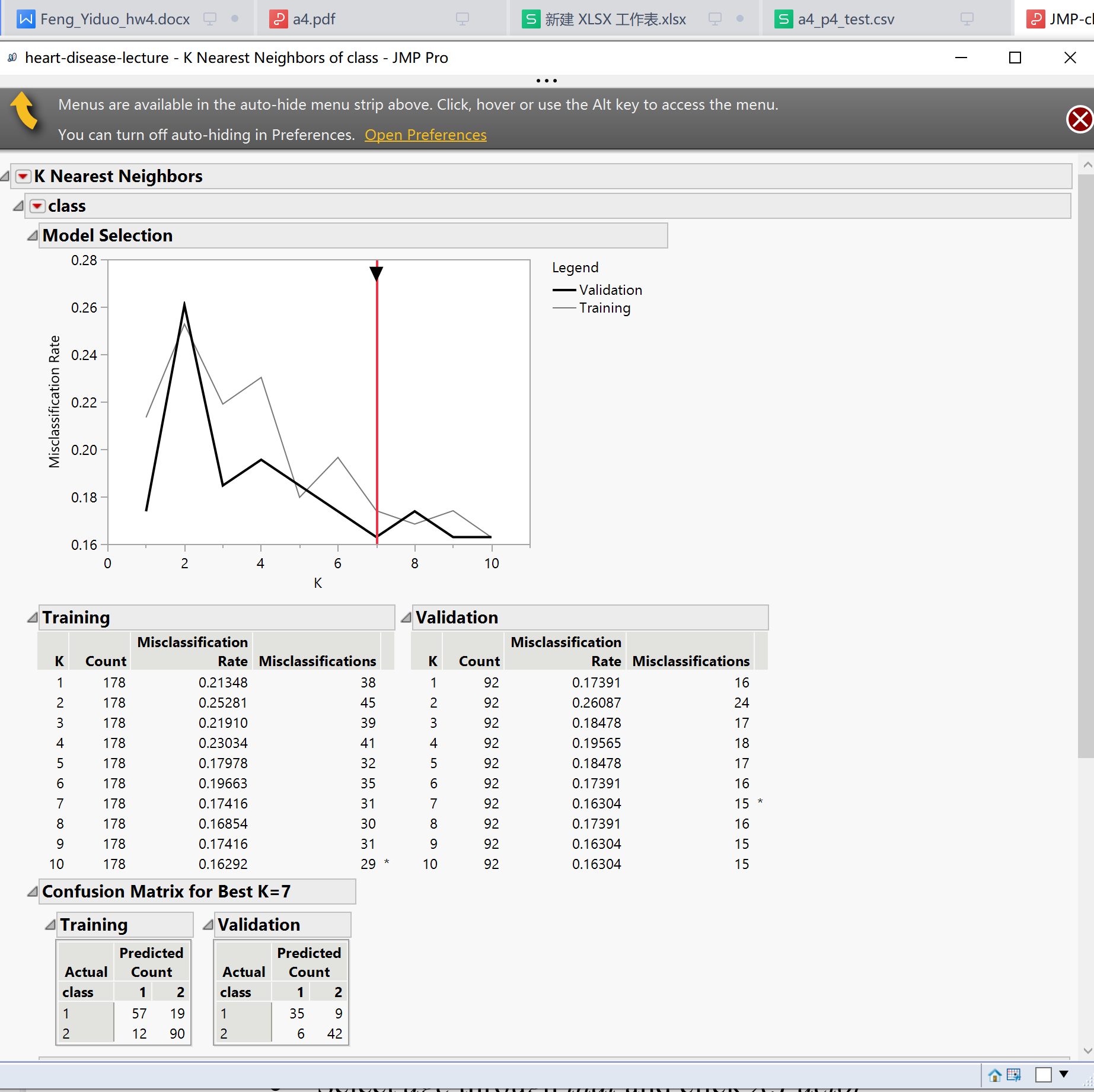
F-Measure = (2\*Precision\*recall)/(Precision+recall)

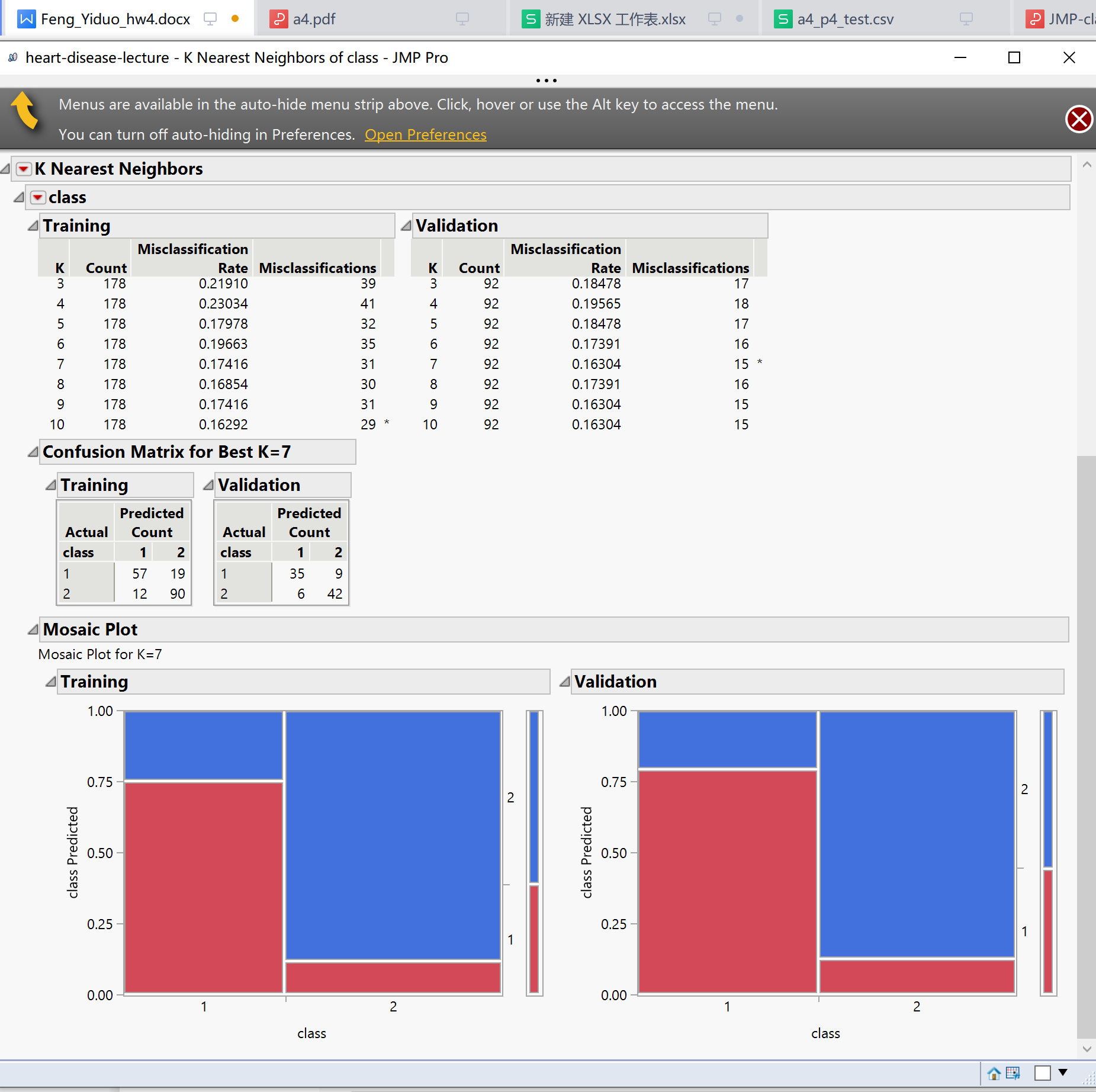
= (2\*0.8478261\*0.8863636)/(0.8478261+0.8863636) = 0.8666667

MCC = 

= (39\*41- 7\*5)/sqrt((39+7)\*(39+5)\*(41+7)\*(41+5)) = 0.73983

1. KNN



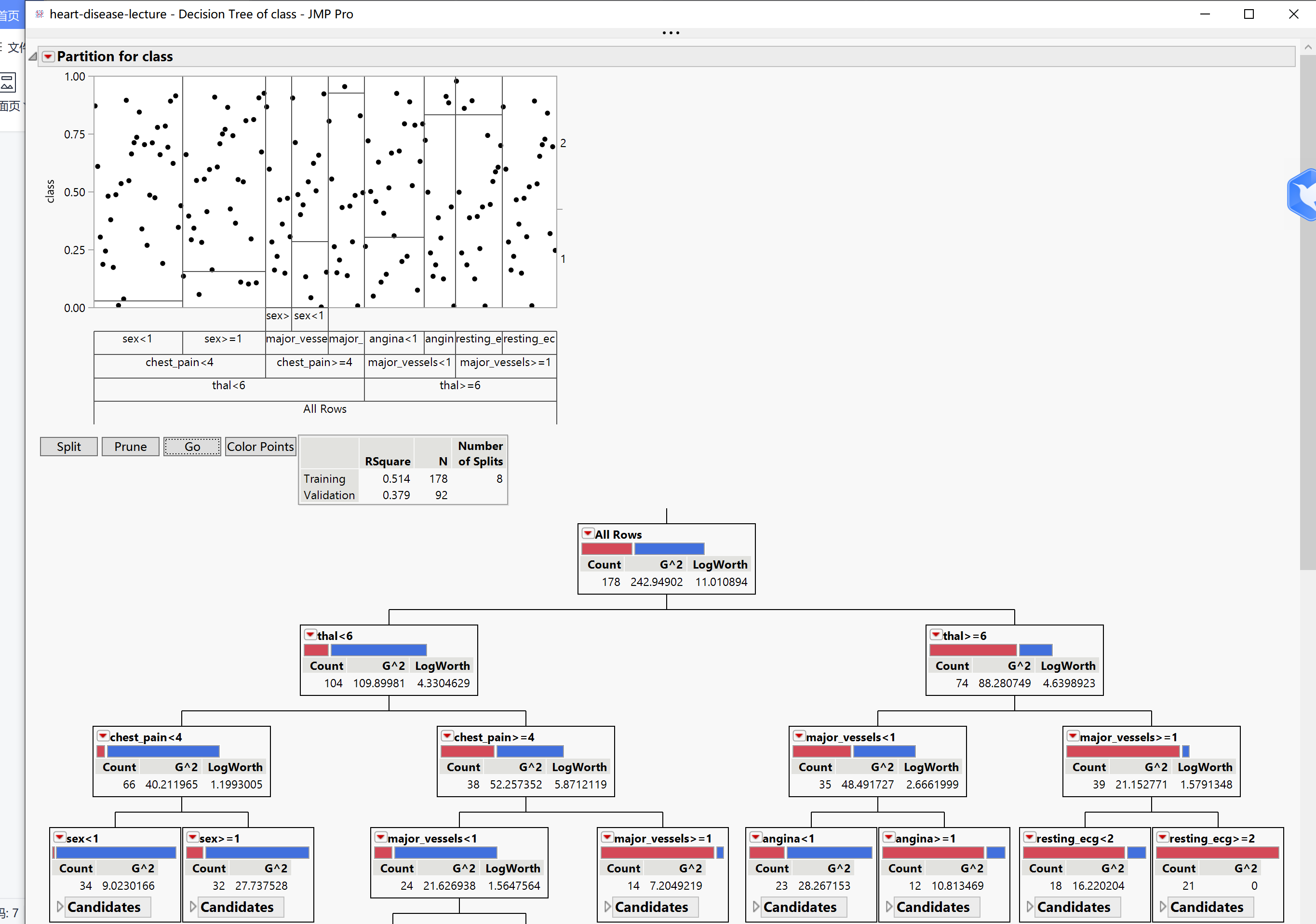


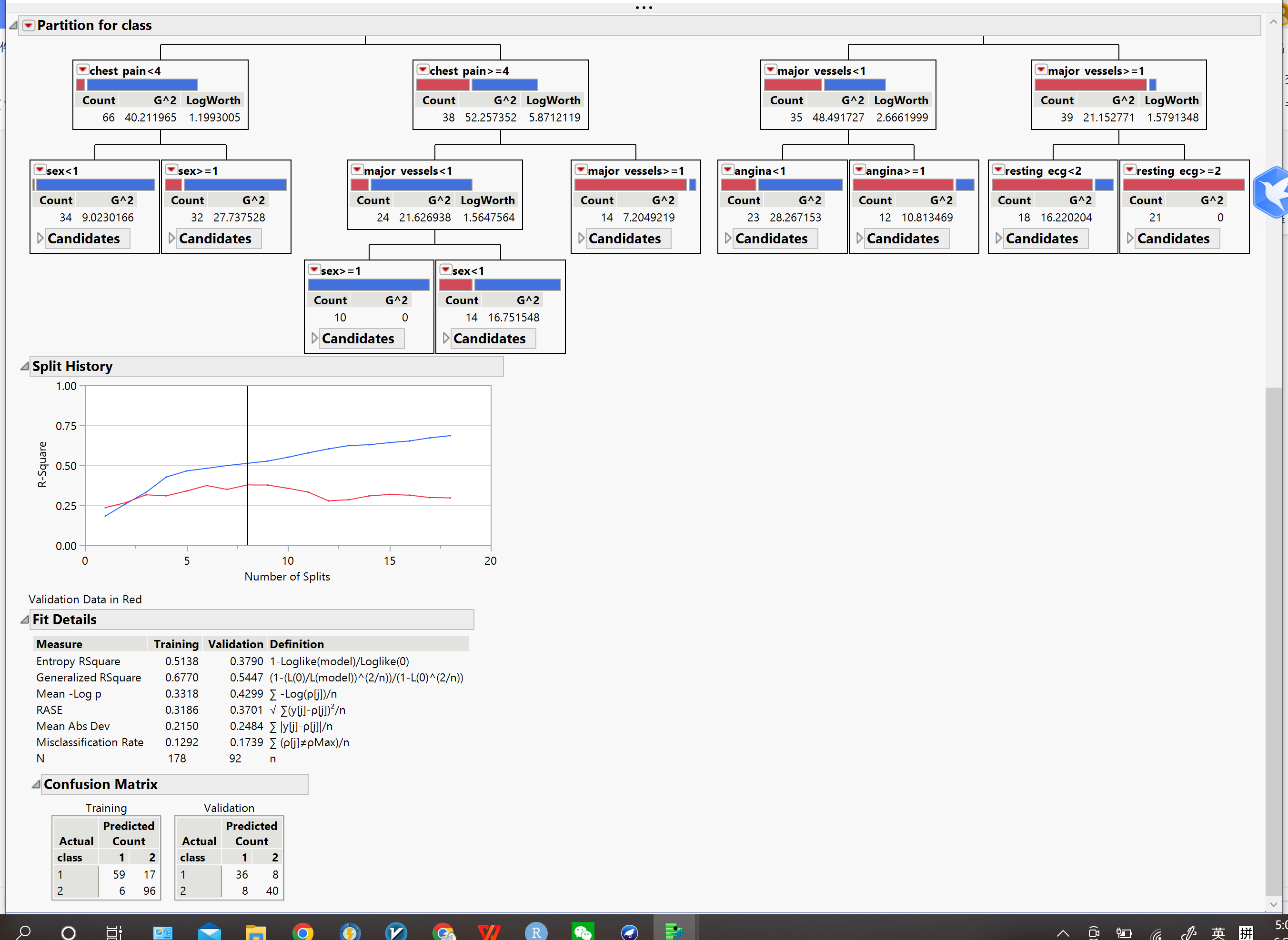
Best k is7

Same as the formula in (1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | Class |
| 0.795454545 | 0.96969697 | 0.353535354 | 0.795454545 | 0.48951049 | -0.284536687 | 1 |
| 0.03030303 | 0.204545455 | 0.181818182 | 0.03030303 | 0.051948052 | 0.284536687 | 2 |

1. . Partition Model (decision tree)

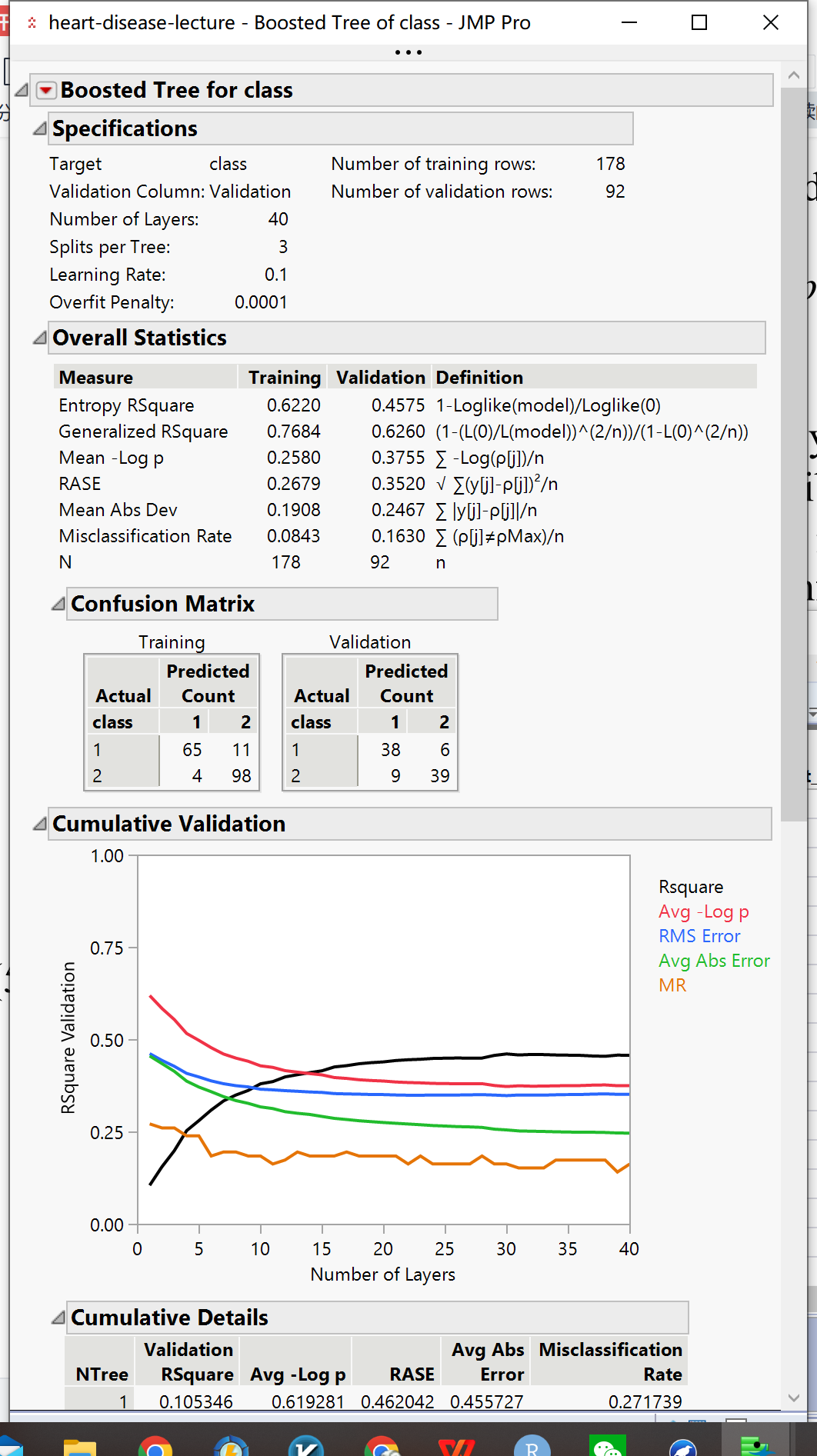


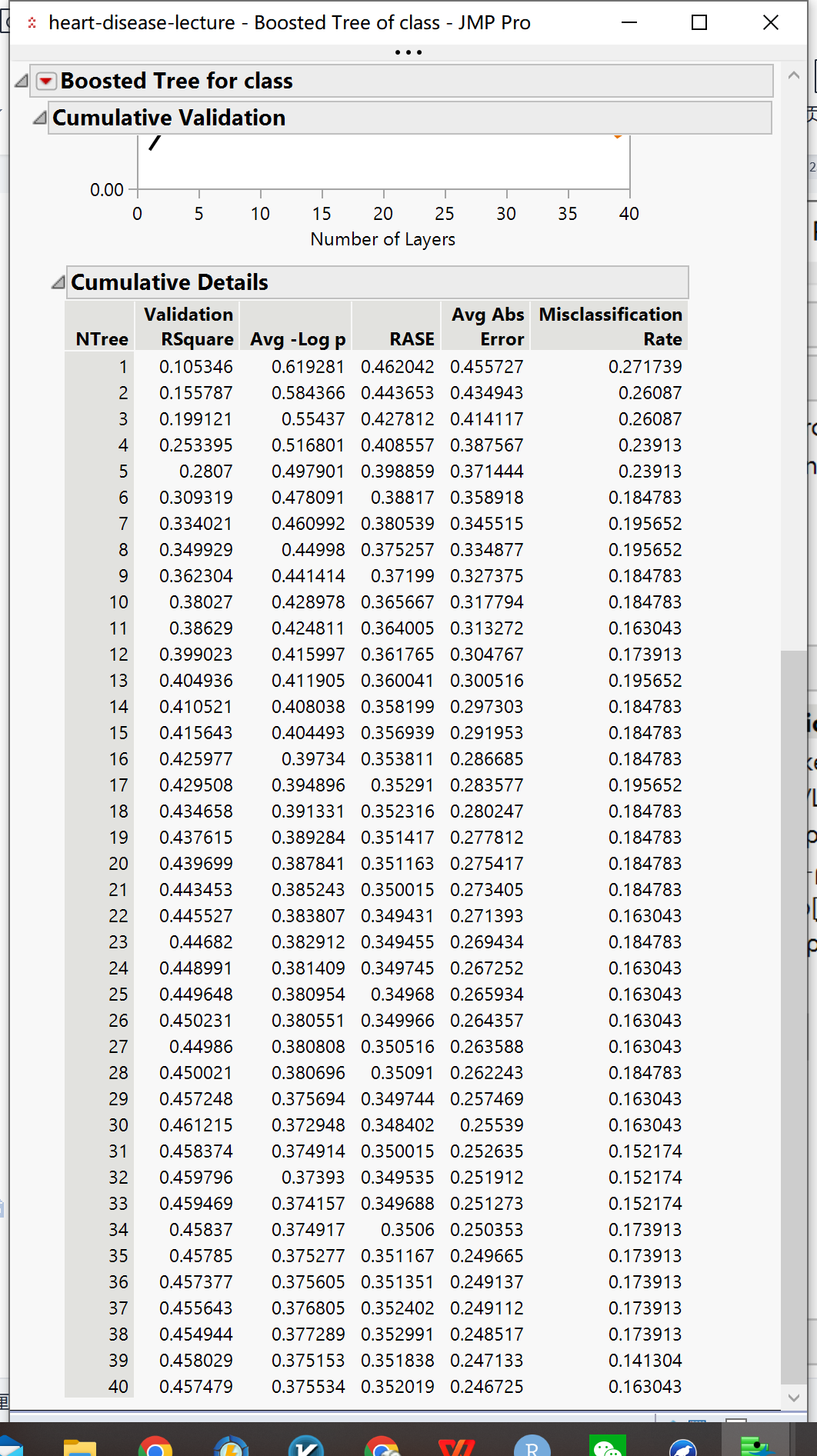


Same as the formula in (1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | Class |
| 0.818181818 | 0.166666667 | 0.818181818 | 0.818181818 | 0.818181818 | 0.651515152 | 1 |
| 0.833333333 | 0.181818182 | 0.833333333 | 0.833333333 | 0.833333333 | -0.651515152 | 2 |

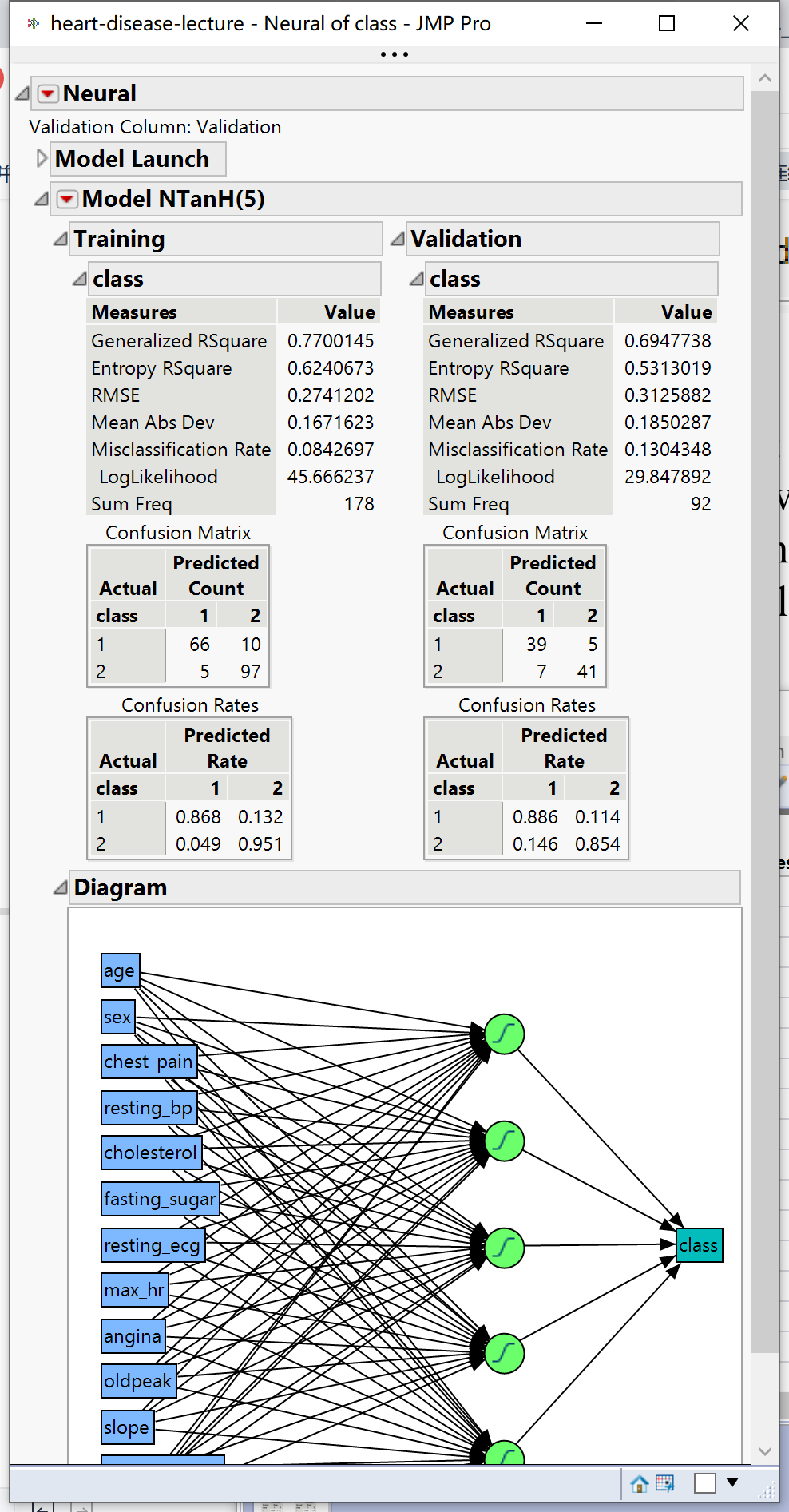
1. . Boosted Tree

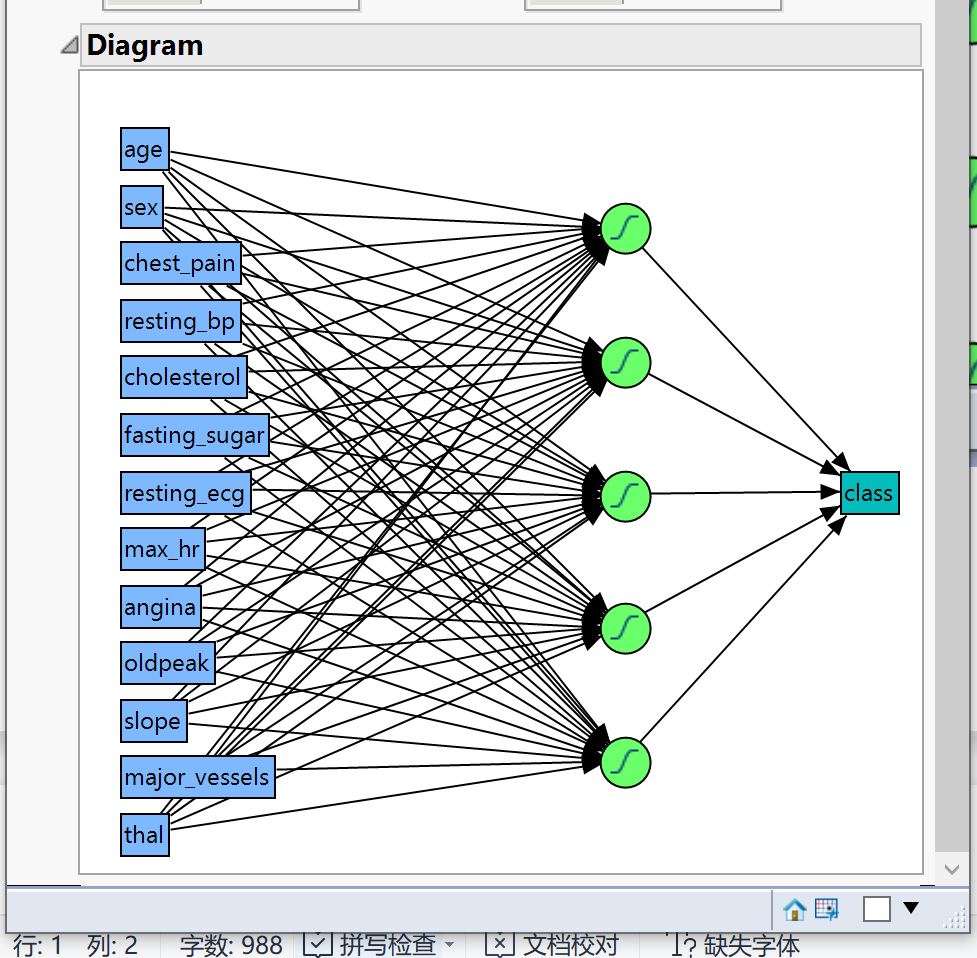




|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | Class |
| 0.863636364 | 0.1875 | 0.808510638 | 0.863636364 | 0.835164835 | 0.675656664 | 1 |
| 0.8125 | 0.136363636 | 0.866666667 | 0.8125 | 0.838709677 | -0.675656664 | 2 |

(5). Neural Network





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | Class |
| 0.886363636 | 0.145833333 | 0.847826087 | 0.886363636 | 0.866666667 | 0.739830038 | 1 |
| 0.854166667 | 0.113636364 | 0.891304348 | 0.854166667 | 0.872340426 | -0.739830038 | 2 |