# MINING ASSOCIATION:

# A CASE STUDY OF BREAST CANCER DATA

by

Asmaul Hosna
(2014-1-60-057)

and

Nasim Ahmed Sazzad
(2014-1-60-096)

Supervised by
Dr. Shamim H Ripon

A thesis presented in partial fulfillment of the Bachelor of Science in Computer Science in the Department of computer Science and Engineering

East-West University, Aftabnaghar, Dhaka-1212, Bangladesh
April 2018

# MINING ASSOCIATION:

# A CASE STUDY OF BREAST CANCER DATA

Asmaul Hosna
(2014-1-60-057)

and

Nasim Ahmed Sazzad
(2014-1-60-096)

East West University

Department of Computer Science and Engineering

Aftabnaghar, Dhaka-1212, Bangladesh

April 2018

# Declaration

We, Asmaul Hosna Mumu and Nasim Ahmed Sazzad, declare that the work presented in this thesis is the outcome of the investigation performed by us under the supervision of DR Shamim H Ripon, Associate Professor, Department of Computer Science and engineering, East West University. We confirm that:

No part of this thesis/project has been or is being submitted elsewhere for the award of any degree or diploma.

Where we have quoted from the work of others, the source is always given.

We have acknowledge all main sources of help.

Signature

......................

**Asmaul Hosna**<br>
**2014-1-60-057**

Signature

........................

**Nasim Ahmed Sazzad**<br>
**2014-1-60-096**

# Abstract

In breast cancer field early detection of breast cancer can provide potential advantages in the treatment of this diseases .Data mining algorithm can provide a great assistance in prediction of early age breast cancer that has always been an challenging research problem. The main objective of this research is to find how precisely can this data mining algorithms predict the probability of recurrence of the diseases among the patients on the basis of their clinical data.

Keywords: breast cancer, recurrence, data mining, probability.

# Acknowledgments

Our science gratitude goes to our parents for their all kindness and encouragement during our studies, without them this thesis would have been very hard to finish.

We would like to thank our supervisor DR. Shamim H Ripon for his utmost direction, constant help, support and feedbacks.

We are grateful to the Kaggle dataset providers, without the dataset it could not be easy to accomplish our work.

We also thank the researchers for their works that helped us a lot to learn new this and helped us to enrich our self.

# LETTER OF ACCAPTENCE

WE HEREBY DECLARE THAT THIS THESIS IS THE STUDENT'S OWN WORK AND BEST EFFORT OF MINE.ALL OTHER SOURCES OF INFORMATION USED HAVE BEEN ACKNOWLEDGED. THIS THESIS HAS BEEN SUBMITTED WITH MY APPROVAL.

Approved by:

Dr. Shamim H Ripon , supervisor

Associate professor
Department of Computer Science of Engineering
EAST WEST UNIVERSITY

Dr. Ahmed Wasif Reza
Associate Professor and Chairperson

Department of Computer Science of Engineering
EAST WEST UNIVERSITY

Date Approved: [April 18, 2018]

# Table of Contents

# List of Figures

# List of Tables

# List of Algorithms

**Classification Algorithm:**

1. Decision tree(j48)
2. Naïve Baye’s
3. SMO(sequential model optimization)
4. Random Forest

**Clustering Algorithm:**

1. K-means
2. EM (Expectation-Maximization)

# Chapter 1
# Introduction

The most dangerous disease in the world is cancer and one of the cancer that kills the women is breast cancer. Detecting the breast cancer manually takes lot of time and it is very difficult for the physician to classify it. Hence for easy classification, detecting the cancer thought various automatic diagnostic techniques is necessary. There are a various methods for detecting breast cancer such as Biopsy, Mammogram, MRI and Ultrasound. Breast cancer happens due to uncontrolled growth of cells and this growths of cells must be stopped as soon as possible by detecting it earlier. There are two classes of tumor: one is beginning tumor and the other is malignant tumor, in which benign tumor is noncancerous the latter is cancerous. Many researcher are still performing research for developing a proper diagnostic system for detecting the tumor as early as possible and also in an easier way, so that the treatment can be started earlier and the rate of survive ability can be increased. For developing the computerized diagnostic system machine learning algorithm plays an important rule. There are many machine learning algorithm which are used to classify the tumor easily and in effective way. This work deals with the comparative analysis of association rules and correlation matrix.

## 1.1 Background

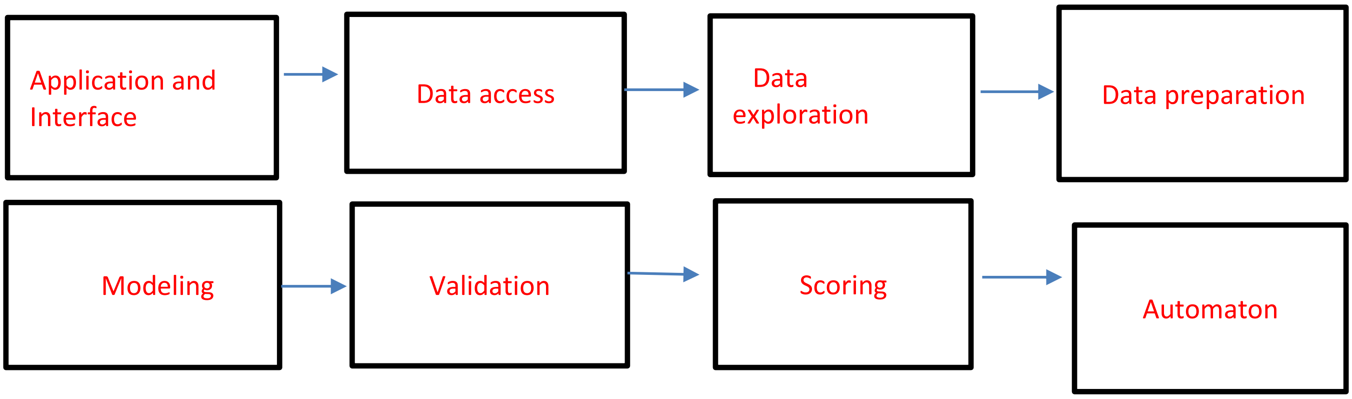
The past and current research reports on medical data using data mining techniques have been studied. All these reports are taken as a base of this paper. On the other hand all the medical term was learned for getting knowledge about the different cause and stages of breast cancer.

In our paper, we used two tools for our process; one is 'WEKA" and another one is 'Rapidminer'.

### 1.1.1 RapidMiner

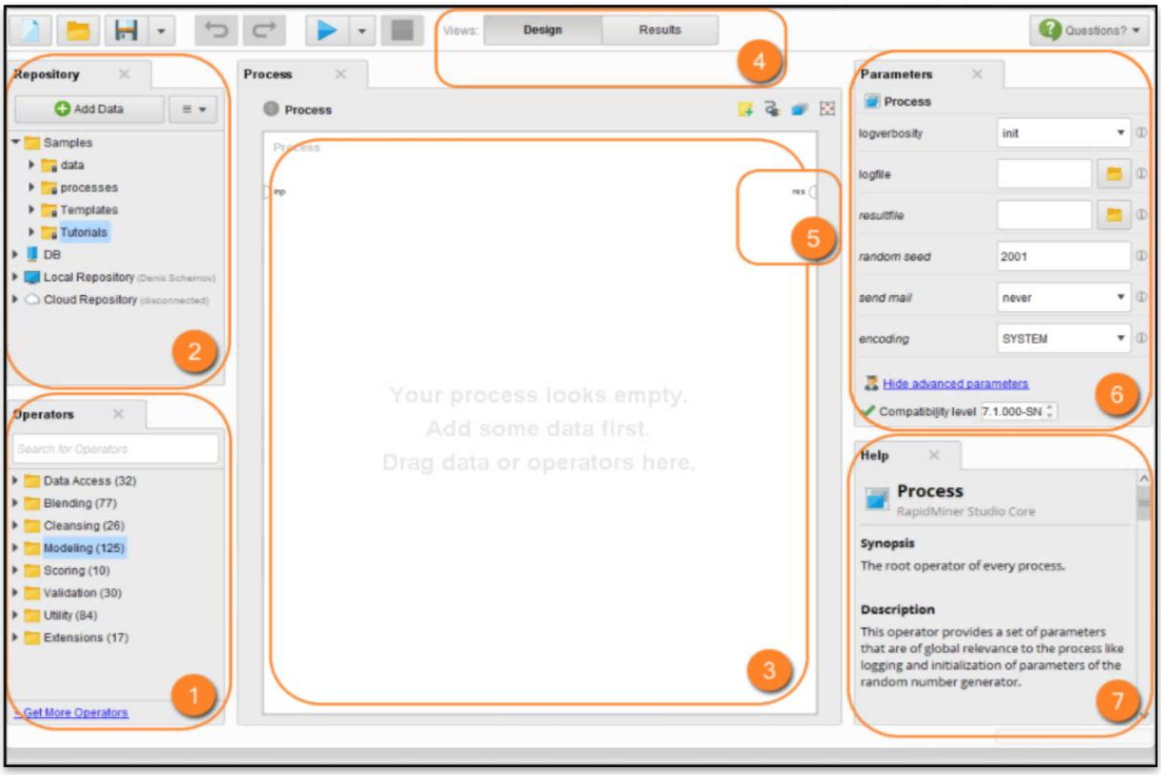
RapidMiner is a data science software platform developed by the company of the same name that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. It uses a client/server model with the server offered as either on premise, or in public or private cloud infrastructures. According to Bloor Research, RapidMiner provides 99% of an advanced analytical solution through template-based frameworks that speed delivery and reduce errors by nearly eliminating the need to write code.

The features of RapidMiner are like:



Application and Interface → Data access → Data exploration → Data preparation

Modeling → Validation → Scoring → Automaton



logverbosity: init

logfile:

resultfile:

random seed: 2001

send mail: never

encoding: SYSTEM

Compatibility level 7.1.000-SN

**Process**
RapidMiner Studio Core

**Synopsis**
The root operator of every process.

**Description**
This operator provides a set of parameters that are of global relevance to the process like logging and initialization of parameters of the random number generator.

Fig 1.Shows the user interfaces of Rapidminer.

### 1.1.2 Weka:

Weka is data mining software that uses a collection of machine learning algorithms. These algorithms can be applied directly to the data or called from the Java code.

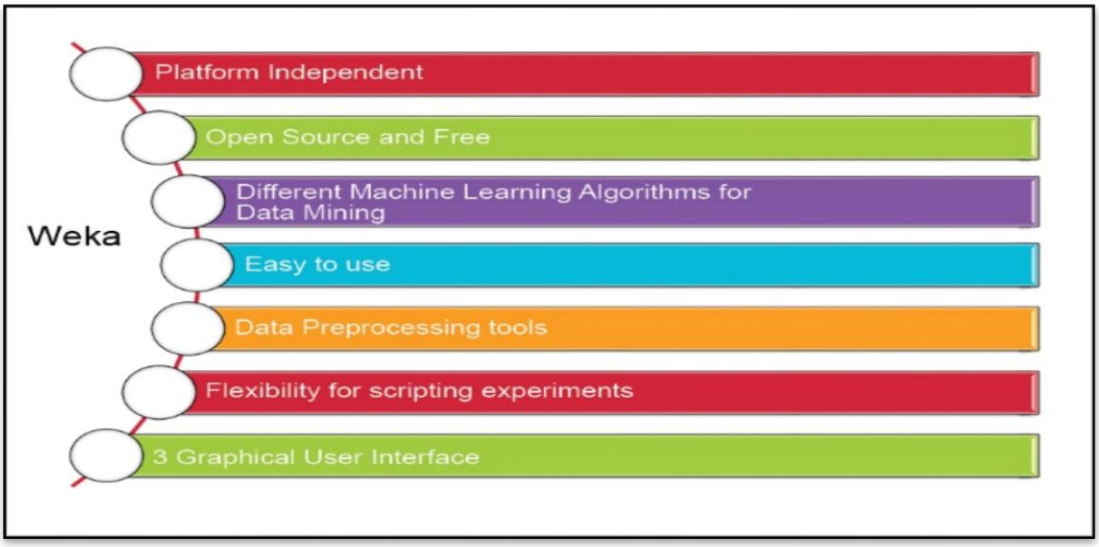
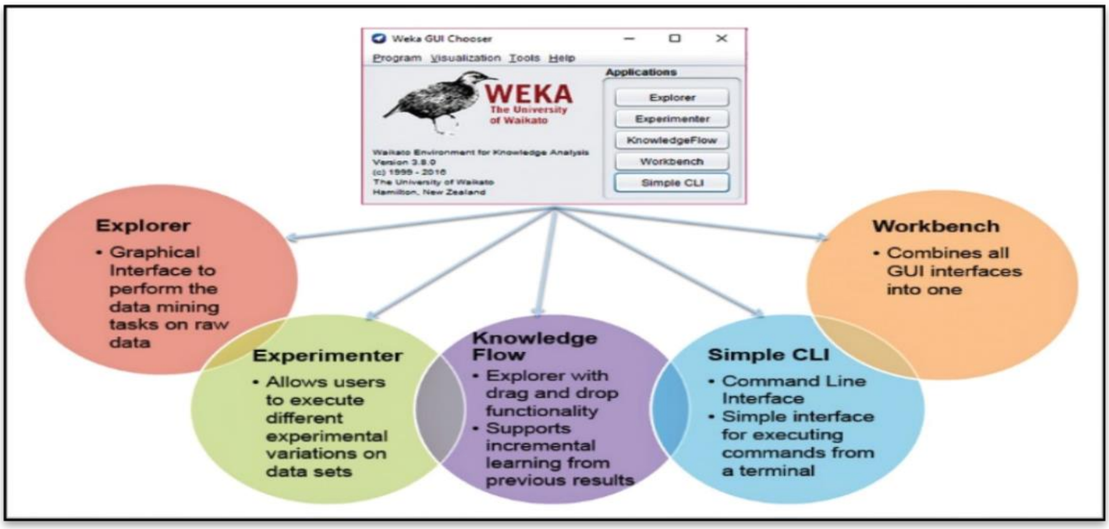

Weka
Platform Independent
Open Source and Free
Different Machine Learning Algorithms for Data Mining
Easy to use
Data Preprocessing tools
Flexibility for scripting experiments
3 Graphical User Interface

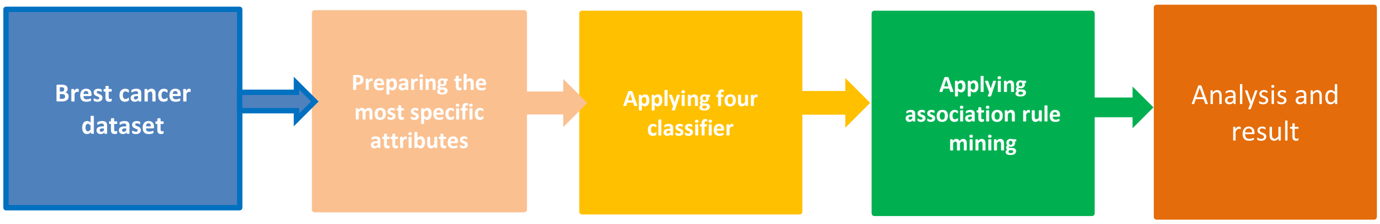
Fig 2.The features of Weka


Weka GUI Chooser
Program
Visualization
Tools
Help
WEKA
The University of Waikato
Applications
Explorer
Experimenter
KnowledgeFlow
Workbench
Simple CLI
Explorer
• Graphical Interface to perform the data mining tasks on raw data
Experimenter
• Allows users to execute different experimental variations on data sets
Knowledge Flow
• Explorer with drag and drop functionality
• Supports incremental learning from previous results
Simple CLI
• Command Line Interface
• Simple interface for executing commands from a terminal
Workbench
• Combines all GUI interfaces into one

Waikato Environment for Knowledge Analysis
Version 3.8.0
(c) 1999 - 2016
The University of Waikato
Hamilton, New Zealand

Fig 3: Interface of Weka

# Chapter 2
# Methodology


Brest cancer dataset
Preparing the most specific attributes
Applying four classifier
Applying association rule mining
Analysis and result

The proposed method of breast cancer detection consists of two main parts: select attribute/s and applying machine learning algorithms.

## 2.1 Training dataset description:

“Clinical_data_breast_cancer.csv” was collected from the most popular site ‘kaggle’.The dataset contains clinical data and various breast cancer classifications from 105 breast cancer patients.

Variables: Complete TCGA ID', 'Gender', 'Age at Initial Pathologic Diagnosis', 'ER Status', 'PR Status', 'HER2 Final Status', 'Tumor', 'Tumor--T1 Coded', 'Node', 'Node-Coded', 'Metastasis', 'Metastasis-Coded', 'AJCC Stage', 'Converted Stage', 'Survival Data Form', 'Vital Status', 'Days to Date of Last Contact', 'Days to date of Death', 'OS event', 'OS Time', 'PAM50 mRNA', 'SigClust Unsupervised mRNA', 'SigClust Intrinsic mRNA', 'miRNA Clusters', 'methylation Clusters', 'RPPA Clusters', 'CN Clusters', 'Integrated Clusters (with PAM50)', 'Integrated Clusters (no exp)', 'Integrated Clusters (unsup exp)'.Among all this attributes we took only 7 attributes for training dataset.

## More details about the dataset:

| S/N | Training attributes | Values |
|---|---|---|
| 1 | Age | 38-88 |
| 2 | HER2 Final Status | Positive, Negative |
| 3 | Tumor | T1,T2,T3,T4 |
| 4 | Node | N0,N1,N3 |
| 5 | Metastasis | M0, M1 |
| 6 | AJCC Stage | IA, IB, IIA ,IIB, IIIA, IIIB, IIIC, IV |
| 7 | Converted stage | No-conversion, IA, IB, IIA ,IIB, IIIA, IIIB, IIIC, IV |

Table 1: Description of Dataset

## 2.2 Attribute selection:

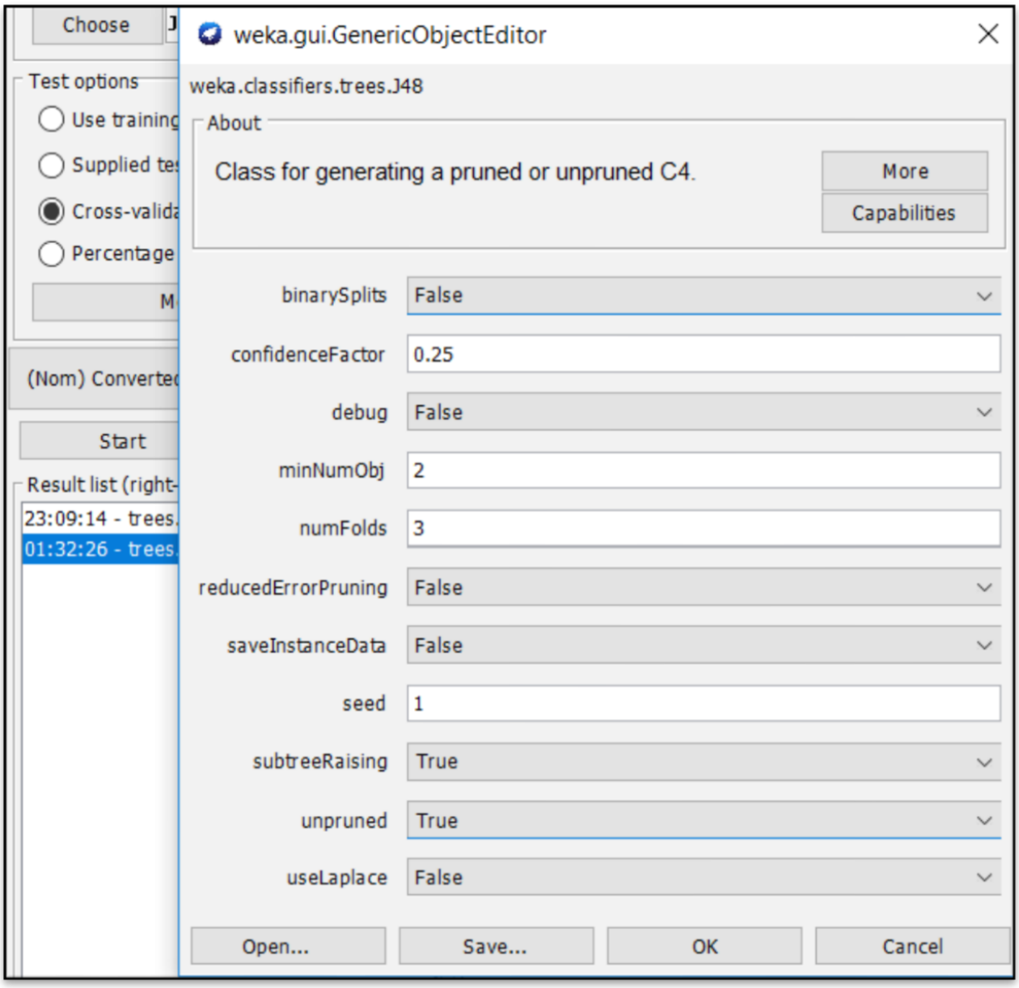
Among all attributes we had to find out the most significant attributes which would give us a perfect accuracy over all algorithms. By testing in various finally we came into a decision that metastasis is the attribute which the reason behind the highest accuracy in our dataset. Metastasis (M) category tells whether or not there is evidence that the cancer has traveled to other parts of the body.

## 2.3 Data mining algorithms used:

*Machine learning is one of the branch of computer science,which is useful to pattern recognition and computational learning. Machine learning can be used to construct algorithms which can learn and make relationship with mathematical and computational statistics. By using machine learning, the user can create new algorithms which can learn and predict the data without explicitly being programmed.* In our research we applied four classification method like – J48, SMO, Naïve Bayes and two clustering EM and K means. Our main implementation part was applying association rule over the dataset and make a relation between them by using correlation algorithm; which we applied in the last part our implementation.

### 2.3.1 Decision Tree (J48):

J48 classifier is a simple decision learning algorithm, it accepts only categorical data for building model. Sometimes it can handle both categorical and data.



weka.gui.GenericObjectEditor

weka.classifiers.trees.J48

About

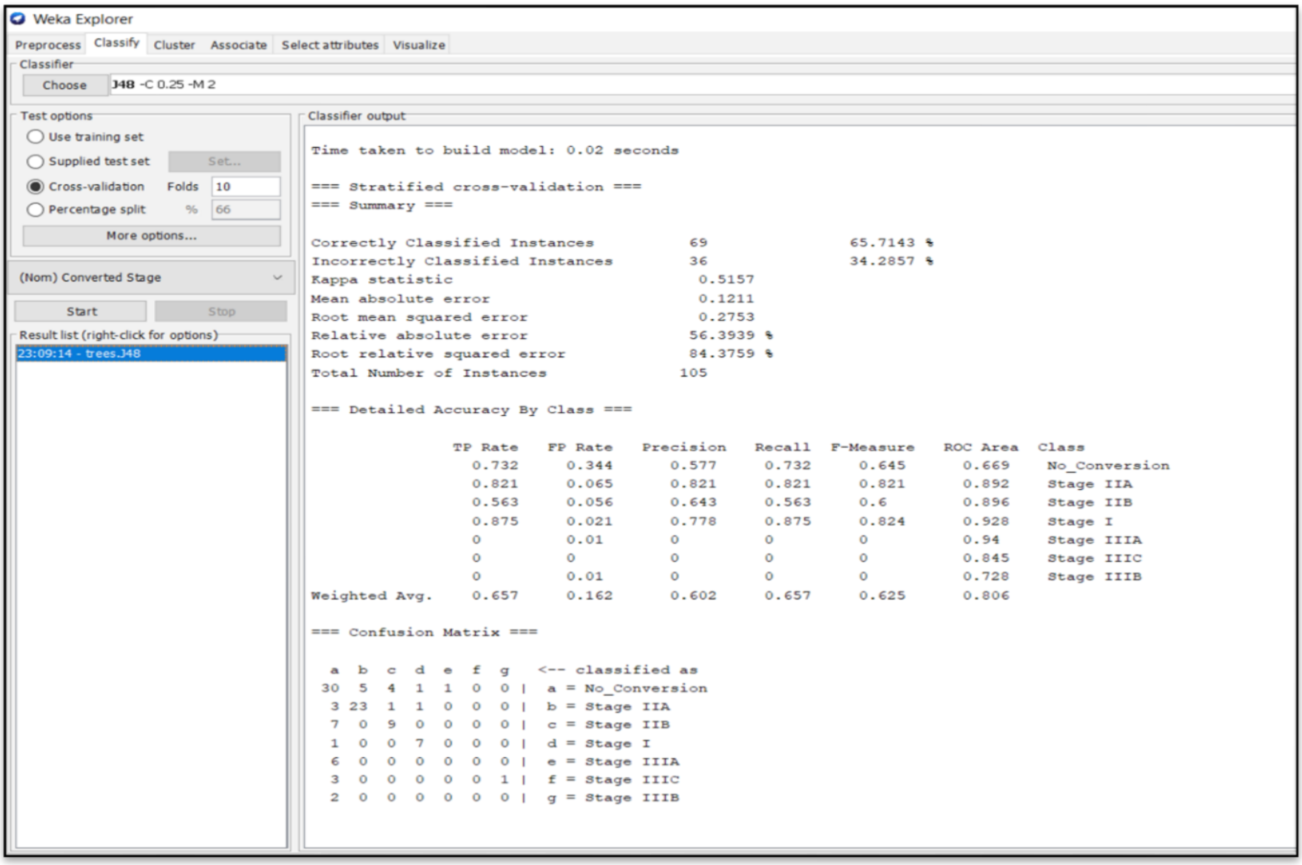
Class for generating a pruned or unpruned C4.

| | |
|---|---|
| binarySplits | False |
| confidenceFactor | 0.25 |
| debug | False |
| minNumObj | 2 |
| numFolds | 3 |
| reducedErrorPruning | False |
| saveInstanceData | False |
| seed | 1 |
| subtreeRaising | True |
| unpruned | True |
| useLaplace | False |

Fig 4: Configuration of the process J48

In fig 4 first we started with the configuration panel of J48. Where mainly “subtreesting” and “unpruned” can change the result in various issue. SubtreeRaising increases the complexity of the algorithm and it was then controlled by true/false. On the other hand unpruned option is for pruning or unpruning the tree. If the unpruned option is set true then J48 will show the bigger true than before. For our research purpose we test the data for both case. After set all the configuration we started our process with cross validation approach where we set the fold number equals to 10; where fold= 10 means dataset is divided into 10 paths for testing purpose.

After running the process we get the accuracy nearly 89% .In algorithm section there is a part called subtreeraising which increases the complexity of the algorithm. In this process we use 6 attribute to find the accuracy and find maximum 89%, then try to remove some attribute to check whether it changes the accuracy level or not. When we remove the two attribute ‘Node’ and ‘AJCC stage’ individually and tested again, they made a huge change in the accuracy.



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

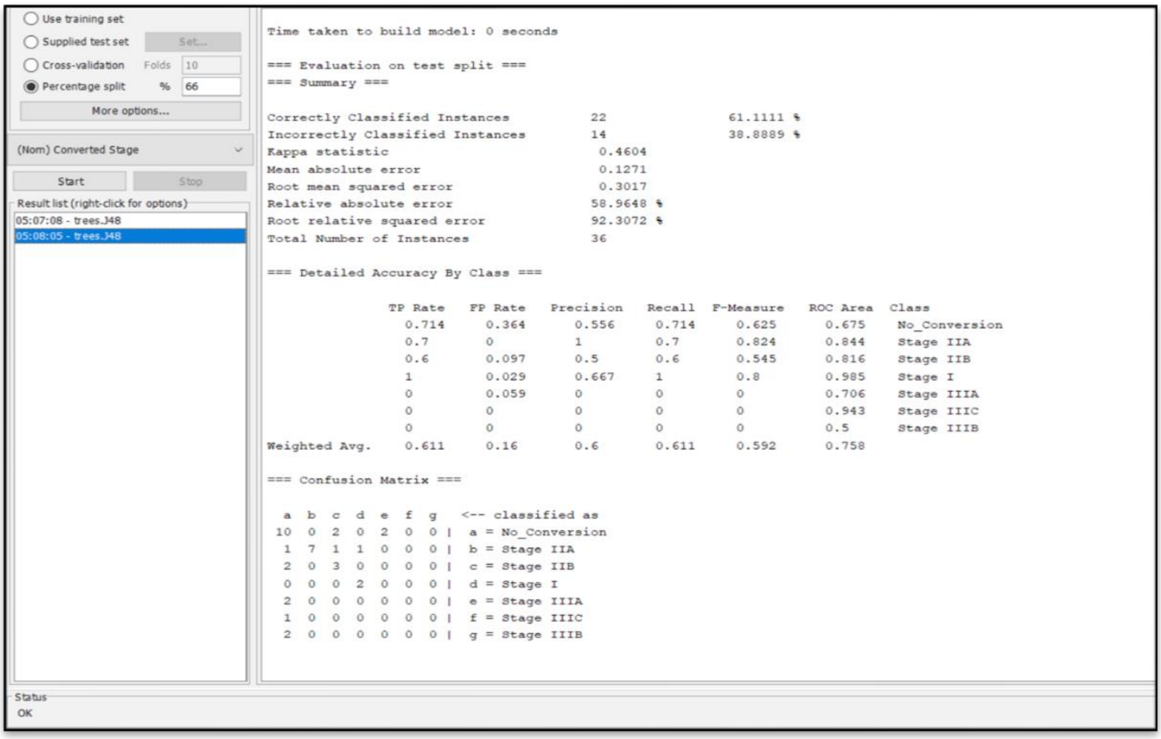
(Nom) Converted Stage

23:09:14 - trees.J48

```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances          69               65.7143 %
Incorrectly Classified Instances        36               34.2857 %
Kappa statistic                          0.5157
Mean absolute error                      0.1211
Root mean squared error                  0.2753
Relative absolute error                 56.3939 %
Root relative squared error             84.3759 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.732     0.344     0.577     0.732     0.645     0.669    No_Conversion
                 0.821     0.065     0.821     0.821     0.821     0.892    Stage IIA
                 0.563     0.056     0.643     0.563     0.6       0.896    Stage IIB
                 0.875     0.021     0.778     0.875     0.824     0.928    Stage I
                 0         0.01      0         0         0         0.94     Stage IIIA
                 0         0         0         0         0         0.845    Stage IIIC
                 0         0.01      0         0         0         0.728    Stage IIIB
Weighted Avg.    0.657     0.162     0.602     0.657     0.625     0.806

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 30  5  4  1  1  0  0 |  a = No_Conversion
  3 23  1  1  0  0  0 |  b = Stage IIA
  7  0  9  0  0  0  0 |  c = Stage IIB
  1  0  0  7  0  0  0 |  d = Stage I
  6  0  0  0  0  0  0 |  e = Stage IIIA
  3  0  0  0  0  0  1 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig 5: J48 algorithm using cross-validation

Then we used the training set and percentage split to test again for comparing whether it changes the accuracy level than cross validation process or not. Training set gave the accuracy of 78% and the percentage split 62%.So from 3 of this test we can evaluate that training test showed the best result.



○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

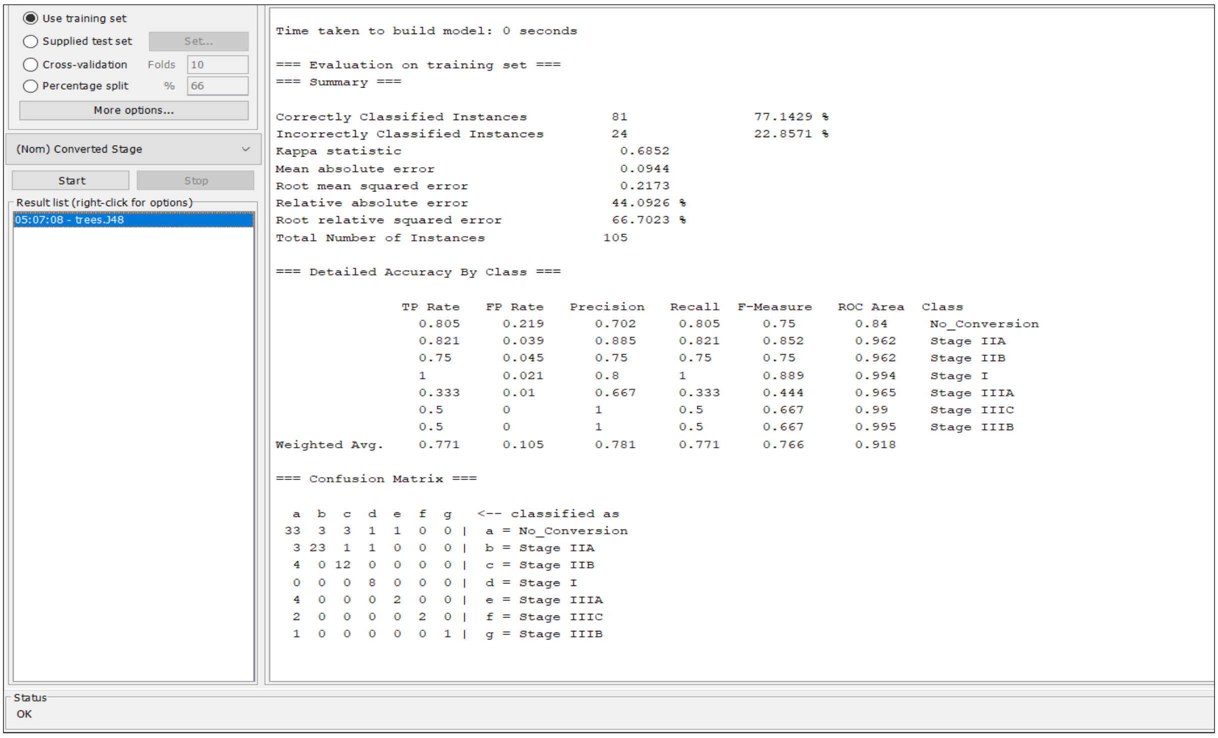
(Nom) Converted Stage

Result list (right-click for options)
05:07:08 - trees.J48
05:08:05 - trees.J48

Status
OK

```
Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          22               61.1111 %
Incorrectly Classified Instances        14               38.8889 %
Kappa statistic                          0.4604
Mean absolute error                      0.1271
Root mean squared error                  0.3017
Relative absolute error                 58.9648 %
Root relative squared error             92.3072 %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.714     0.364     0.556     0.714     0.625      0.675     No_Conversion
                 0.7       0         1         0.7       0.824      0.844     Stage IIA
                 0.6       0.097     0.5       0.6       0.545      0.816     Stage IIB
                 1         0.029     0.667     1         0.8        0.985     Stage I
                 0         0.059     0         0         0          0.706     Stage IIIA
                 0         0         0         0         0          0.943     Stage IIIC
                 0         0         0         0         0          0.5       Stage IIIB
Weighted Avg.    0.611     0.16      0.6       0.611     0.592      0.758

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 10  0  2  0  2  0  0 |  a = No_Conversion
  1  7  1  1  0  0  0 |  b = Stage IIA
  2  0  3  0  0  0  0 |  c = Stage IIB
  0  0  0  2  0  0  0 |  d = Stage I
  2  0  0  0  0  0  0 |  e = Stage IIIA
  1  0  0  0  0  0  0 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig 6: J48 algorithm using percentage split



● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) Converted Stage

Result list (right-click for options)
05:07:08 - trees.J48

Status
OK

```
Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          81               77.1429 %
Incorrectly Classified Instances        24               22.8571 %
Kappa statistic                          0.6852
Mean absolute error                      0.0944
Root mean squared error                  0.2173
Relative absolute error                 44.0926 %
Root relative squared error             66.7023 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.805     0.219     0.702     0.805     0.75       0.84      No_Conversion
                 0.821     0.039     0.885     0.821     0.852      0.962     Stage IIA
                 0.75      0.045     0.75      0.75      0.75       0.962     Stage IIB
                 1         0.021     0.8       1         0.889      0.994     Stage I
                 0.333     0.01      0.667     0.333     0.444      0.965     Stage IIIA
                 0.5       0         1         0.5       0.667      0.99      Stage IIIC
                 0.5       0         1         0.5       0.667      0.995     Stage IIIB
Weighted Avg.    0.771     0.105     0.781     0.771     0.766      0.918

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 33  3  3  1  1  0  0 |  a = No_Conversion
  3 23  1  1  0  0  0 |  b = Stage IIA
  4  0 12  0  0  0  0 |  c = Stage IIB
  0  0  0  8  0  0  0 |  d = Stage I
  4  0  0  0  2  0  0 |  e = Stage IIIA
  2  0  0  0  0  2  0 |  f = Stage IIIC
  1  0  0  0  0  0  1 |  g = Stage IIIB
```

Fig 7: J48 algorithm using training set

As we said we took another tool for comparing or getting the best accuracy purpose we used our dataset on Rapidminer tool for further analysis. In WEKA, we kept metastasis as a target value for our testing; keeping the same thing on mind first we set the role to “tumor” and then “metastasis” to see the difference.

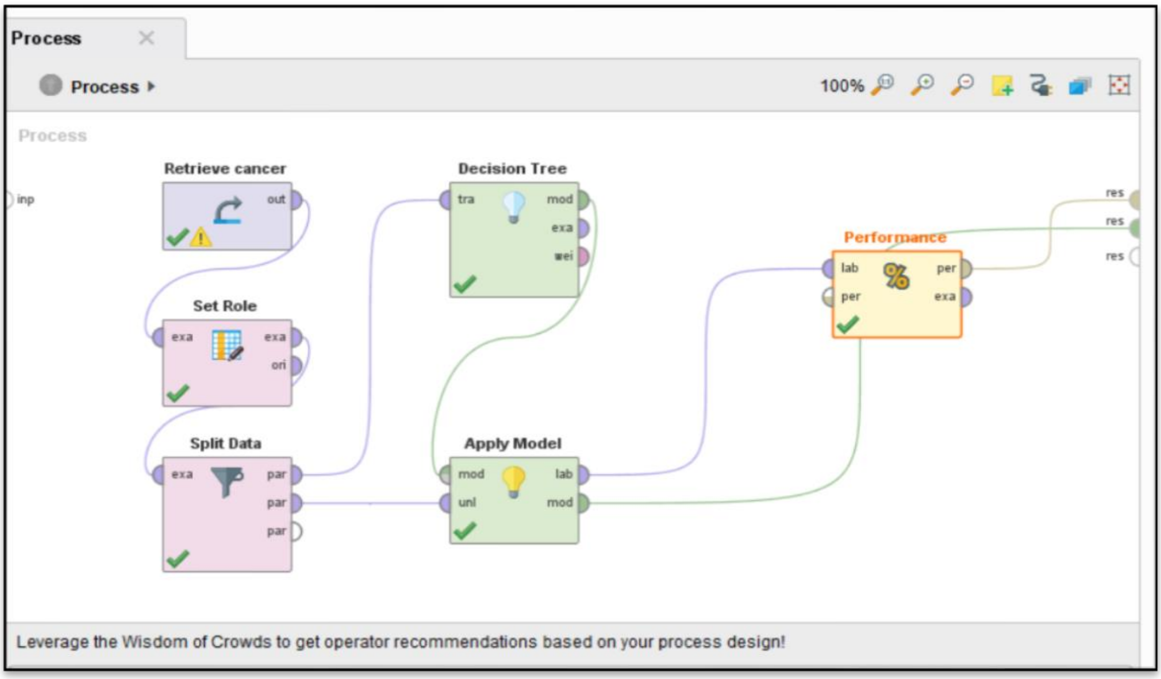


Fig 8: J48 algorithm model design on rapidminer tool

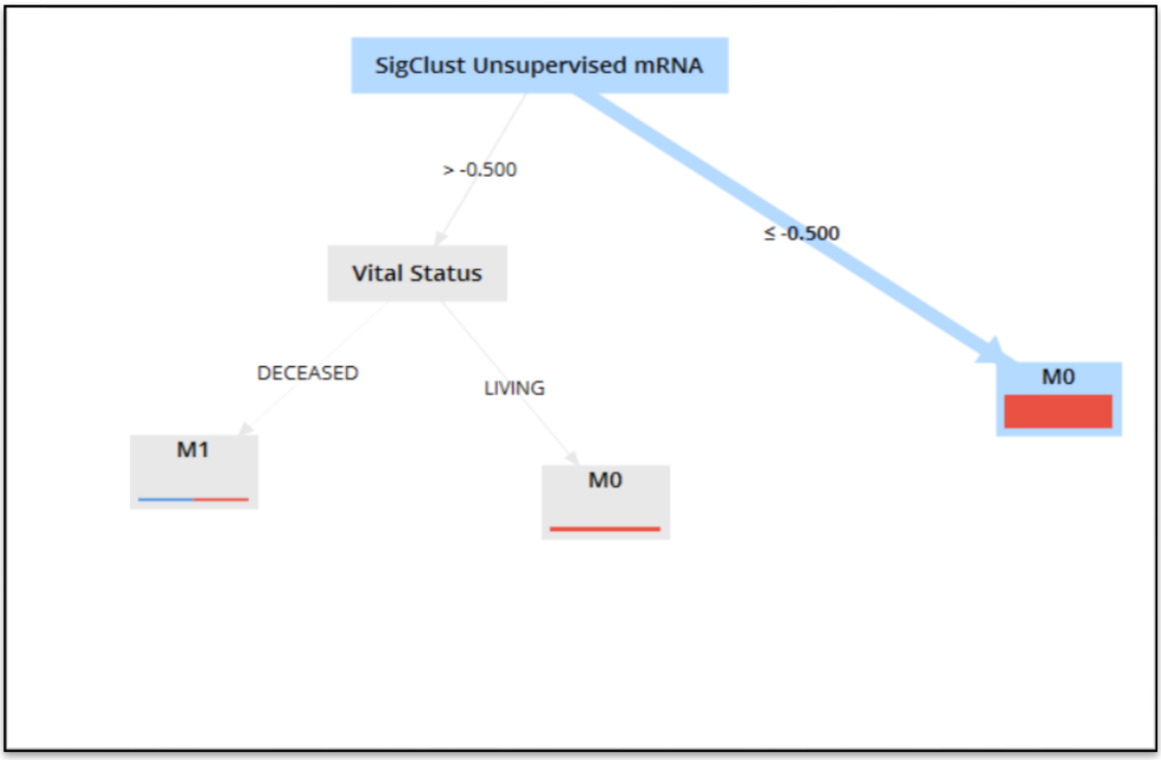
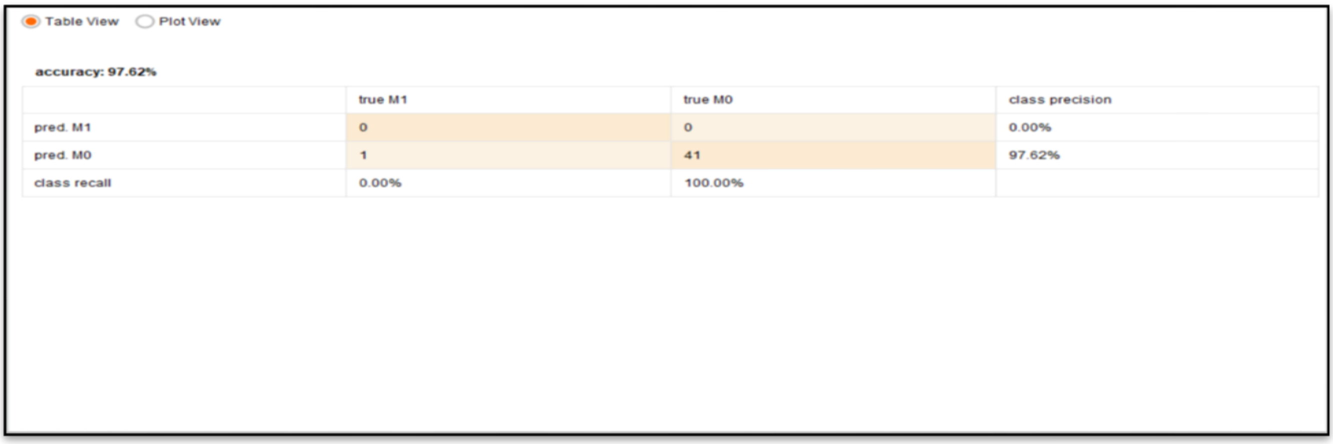


Fig 9: visualization of decision tree



accuracy: 97.62%

| | true M1 | true M0 | class precision |
|---|---|---|---|
| pred. M1 | 0 | 0 | 0.00% |
| pred. M0 | 1 | 41 | 97.62% |
| class recall | 0.00% | 100.00% | |

Fig 10:  Accuracy by taken metastasis as a target attribute

Metastasis showed the best accuracy based on sigClust unsupervised mRNA values and made a tree with five nodes.

On the hand when we set ‘tumor’ as target value, it made a big tree than the “metastasis” was formed but accuracy was less than the previous one.

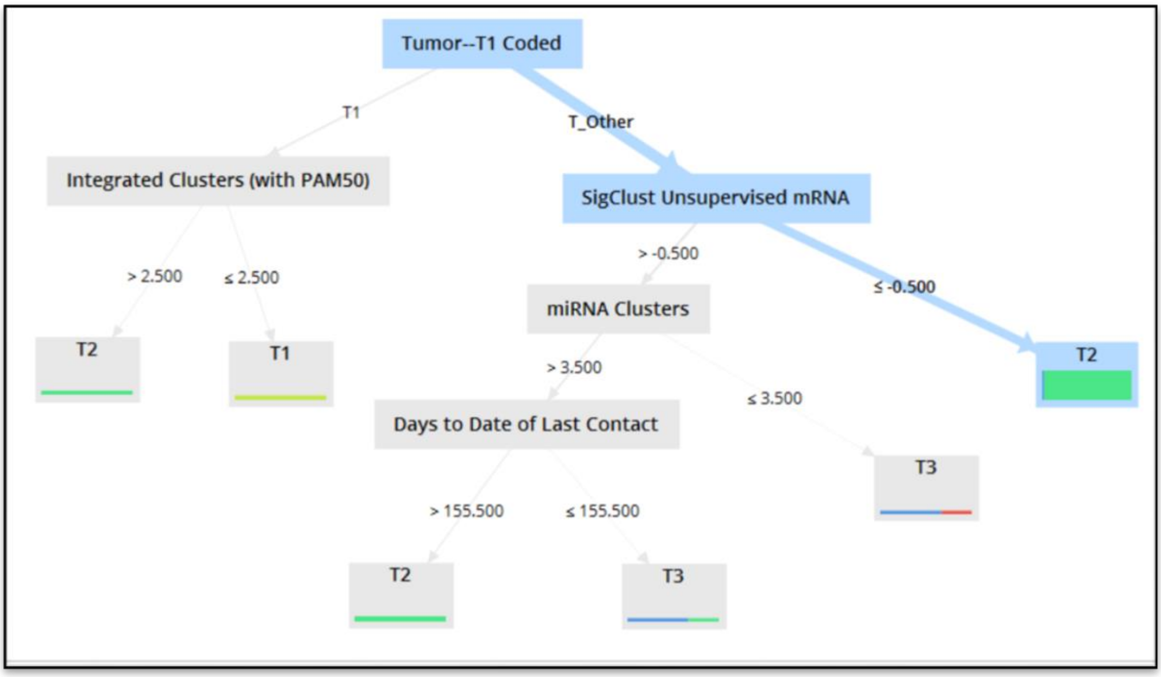
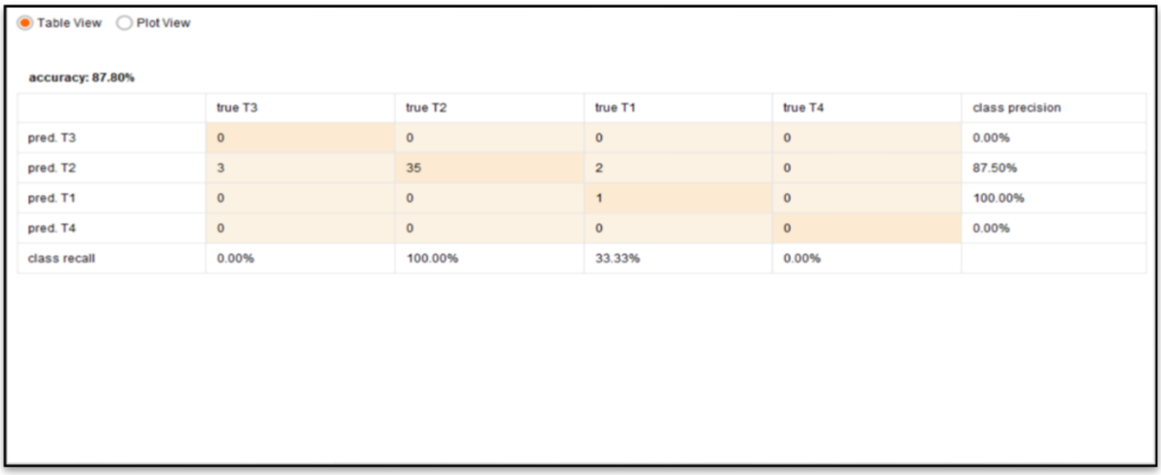


Fig 11: Visualization of tree by tumor



accuracy: 87.80%

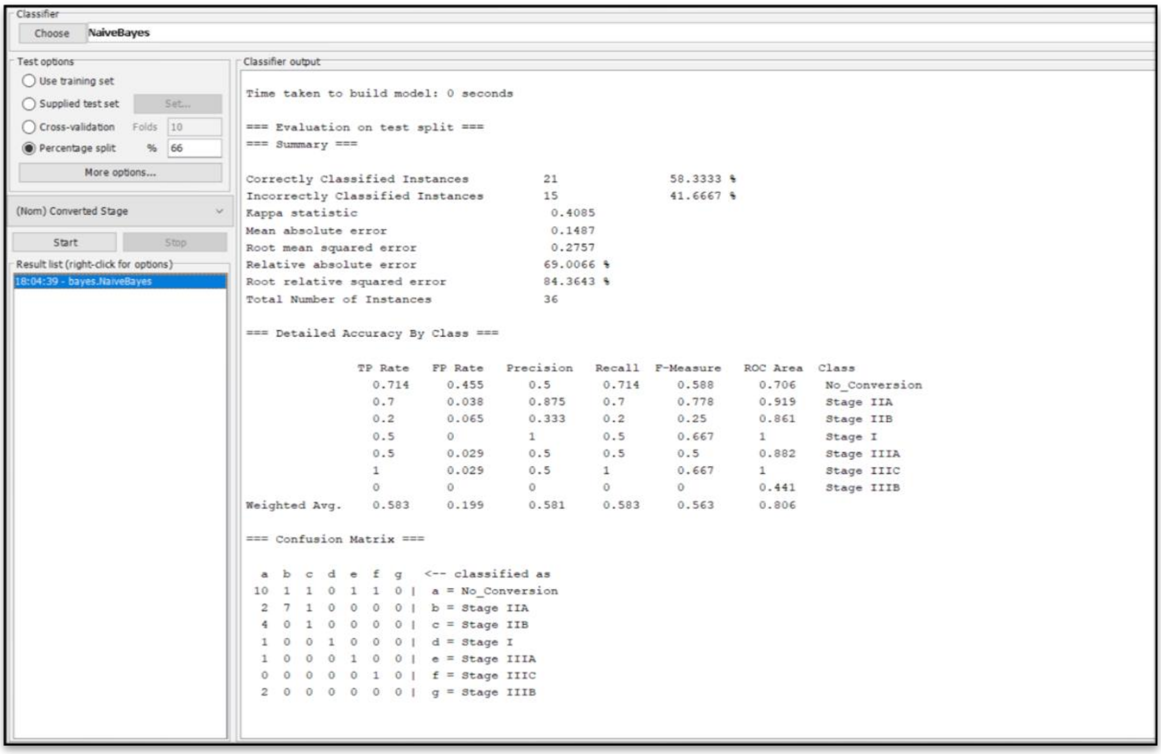
| | true T3 | true T2 | true T1 | true T4 | class precision |
|---|---|---|---|---|---|
| pred. T3 | 0 | 0 | 0 | 0 | 0.00% |
| pred. T2 | 3 | 35 | 2 | 0 | 87.50% |
| pred. T1 | 0 | 0 | 1 | 0 | 100.00% |
| pred. T4 | 0 | 0 | 0 | 0 | 0.00% |
| class recall | 0.00% | 100.00% | 33.33% | 0.00% | |

Fig 12:  Accuracy taken by tumor as a target attribute

Considering the both tools; decision tree (J48) shows the best accuracy in RapidMiner. So it can be said that it is possible to find the best accuracy from J48 by using this dataset.

### 2.3.2 Naïve Baye's classifier:

Naïve Baye's classifier is one of the method of supervised learning. It provides an efficient way of handling any number of attributes or classes which is purely based on probabilistic theory.* To find out the most probability constraints we applied naïve baye's classifier on our data set in WEKA. First we tested for cross-validation, then we tested for percentage split and next for training set



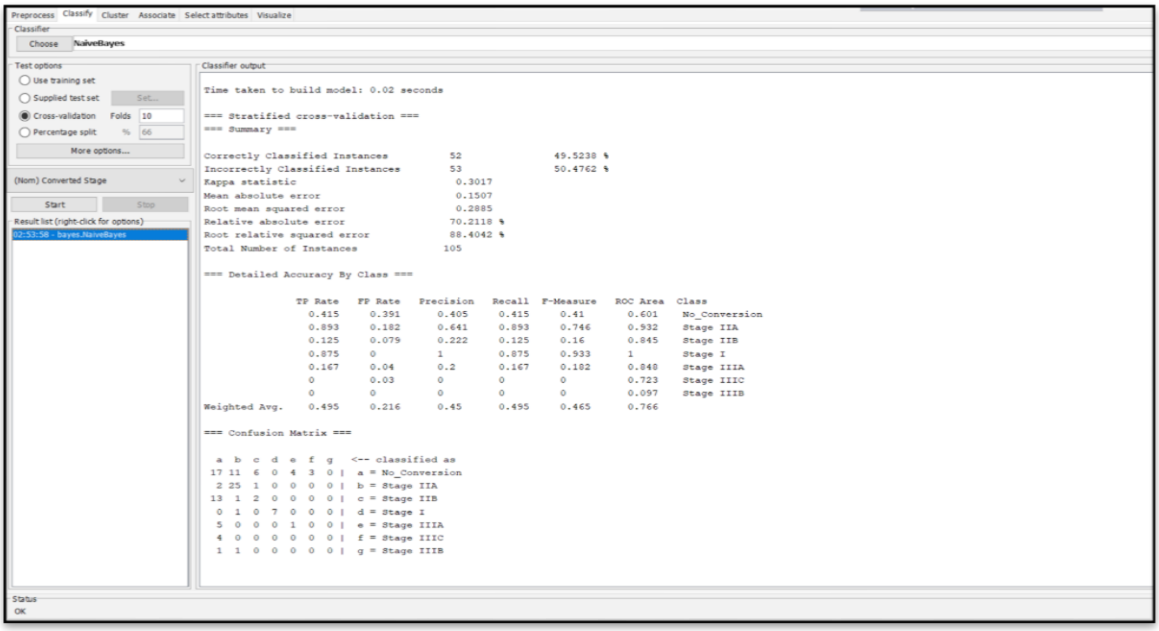
NaiveBayes

Folds 10

Percentage split % 66

```
Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          21               58.3333 %
Incorrectly Classified Instances        15               41.6667 %
Kappa statistic                          0.4085
Mean absolute error                      0.1487
Root mean squared error                  0.2757
Relative absolute error                 69.0066 %
Root relative squared error             84.3643 %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   0.714     0.455      0.5       0.714     0.588      0.706    No_Conversion
                   0.7       0.038      0.875     0.7       0.778      0.919    Stage IIA
                   0.2       0.065      0.333     0.2       0.25       0.861    Stage IIB
                   0.5       0          1         0.5       0.667      1        Stage I
                   0.5       0.029      0.5       0.5       0.5        0.882    Stage IIIA
                   1         0.029      0.5       1         0.667      1        Stage IIIC
                   0         0          0         0         0          0.441    Stage IIIB
Weighted Avg.      0.583     0.199      0.581     0.583     0.563      0.806

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 10  1  1  0  1  1  0 |  a = No_Conversion
  2  7  1  0  0  0  0 |  b = Stage IIA
  4  0  1  0  0  0  0 |  c = Stage IIB
  1  0  0  1  0  0  0 |  d = Stage I
  1  0  0  0  1  0  0 |  e = Stage IIIA
  0  0  0  0  0  1  0 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig 13: Naïve Bayes using percentage spli



Classifier: NaiveBayes

Test options:
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

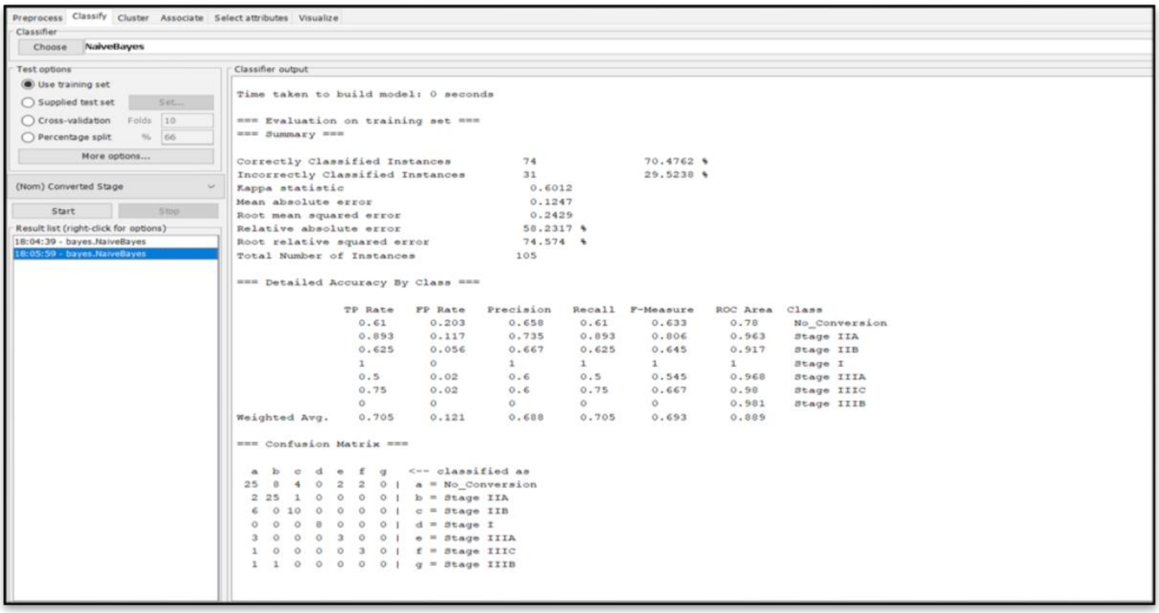
(Nom) Converted Stage

02:53:58 - bayes.NaiveBayes

```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances          52               49.5238 %
Incorrectly Classified Instances        53               50.4762 %
Kappa statistic                          0.3017
Mean absolute error                      0.1507
Root mean squared error                  0.2885
Relative absolute error                 70.2118 %
Root relative squared error             88.4042 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                 0.415    0.391    0.405      0.415   0.41       0.601     No_Conversion
                 0.893    0.182    0.641      0.893   0.746      0.932     Stage IIA
                 0.125    0.079    0.222      0.125   0.16       0.845     Stage IIB
                 0.875    0        1          0.875   0.933      1         Stage I
                 0.167    0.04     0.2        0.167   0.182      0.848     Stage IIIA
                 0        0.03     0          0       0          0.723     Stage IIIC
                 0        0        0          0       0          0.097     Stage IIIB
Weighted Avg.    0.495    0.216    0.45       0.495   0.465      0.766

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 17 11  6  0  4  3  0 |  a = No_Conversion
  2 25  1  0  0  0  0 |  b = Stage IIA
 13  1  2  0  0  0  0 |  c = Stage IIB
  0  1  0  7  0  0  0 |  d = Stage I
  5  0  0  0  1  0  0 |  e = Stage IIIA
  4  0  0  0  0  0  0 |  f = Stage IIIC
  1  1  0  0  0  0  0 |  g = Stage IIIB
```

OK

Fig 14: Naïve Bayes using cross validation



Classifier: NaiveBayes

Test options:
● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) Converted Stage

18:04:39 - bayes.NaiveBayes
18:05:59 - bayes.NaiveBayes

```
Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          74               70.4762 %
Incorrectly Classified Instances        31               29.5238 %
Kappa statistic                          0.6012
Mean absolute error                      0.1247
Root mean squared error                  0.2429
Relative absolute error                 58.2317 %
Root relative squared error             74.574  %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                 0.61     0.203    0.658      0.61    0.633      0.78      No_Conversion
                 0.893    0.117    0.735      0.893   0.806      0.963     Stage IIA
                 0.625    0.056    0.667      0.625   0.645      0.917     Stage IIB
                 1        0        1          1       1          1         Stage I
                 0.5      0.02     0.6        0.5     0.545      0.968     Stage IIIA
                 0.75     0.02     0.6        0.75    0.667      0.98      Stage IIIC
                 0        0        0          0       0          0.981     Stage IIIB
Weighted Avg.    0.705    0.121    0.688      0.705   0.693      0.889

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 25  8  4  0  2  2  0 |  a = No_Conversion
  2 25  1  0  0  0  0 |  b = Stage IIA
  6  0 10  0  0  0  0 |  c = Stage IIB
  0  0  0  8  0  0  0 |  d = Stage I
  3  0  0  0  3  0  0 |  e = Stage IIIA
  1  0  0  0  0  3  0 |  f = Stage IIIC
  1  1  0  0  0  0  0 |  g = Stage IIIB
```

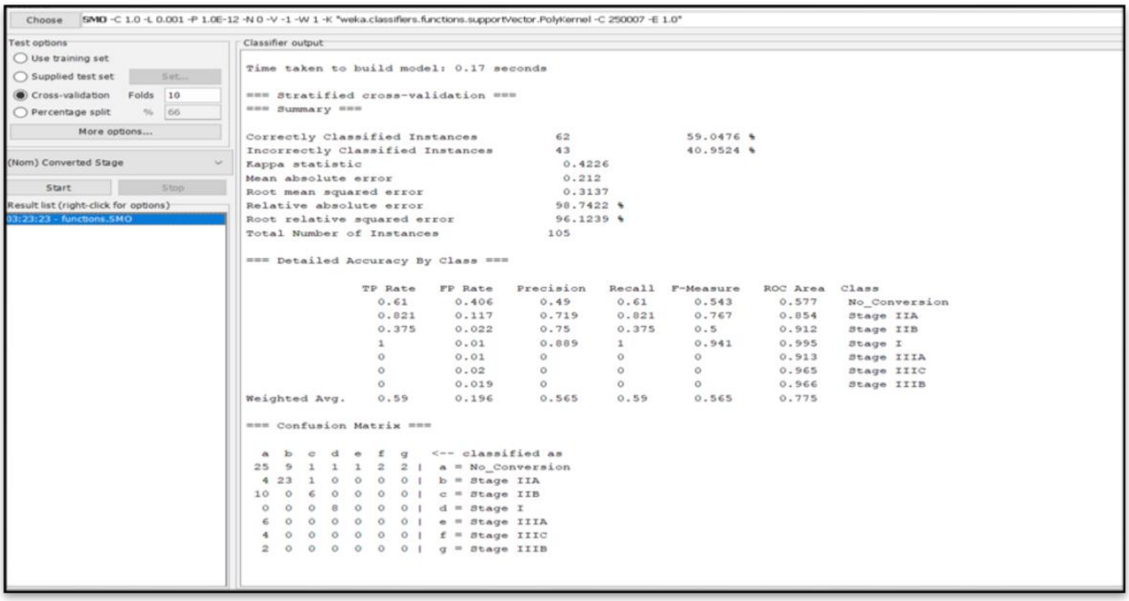
Fig 15: Naïve Bayes using training set

Overall we can recognize that training set gave the best accuracy in this algorithm.

## 2.3.3. SMO: (sequential model optimization) :

Training a support vector machine requires the solution of a very large quadratic programming (QP) optimization problem. SMO breaks this large QP problem into a series of smallest possible QP problems. These small QP problems are solved analytically, which avoids using a_time-consuming numerical QP optimization as an inner loop. The amount of memory required for SMO is linear in the training set size, which allows SMO to handle very large training sets.

Because matrix computation is avoided, SMO scales somewhere between linear and quadratic in the training set size for various test problem. As we are using different algorithm on our dataset; we used SMO for another testing purposes. Same as before testing divided into three part - cross validation, percentage split and training set. They accordingly gave the accuracy



```
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
```

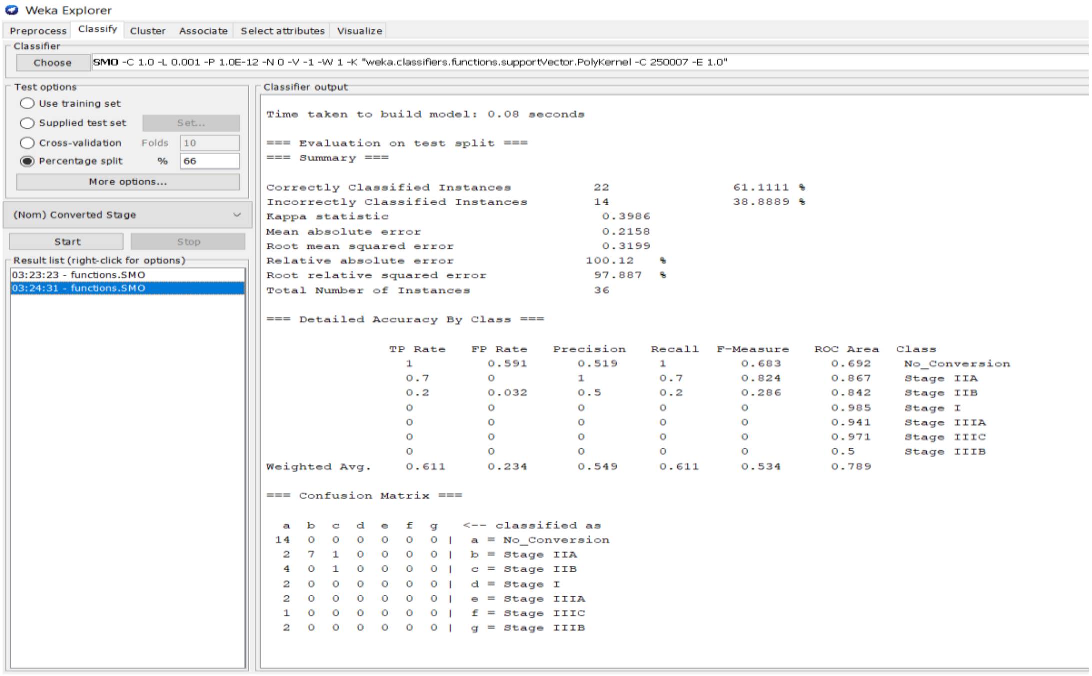
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Converted Stage

03:23:23 - functions.SMO

```
Time taken to build model: 0.17 seconds

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

Correctly Classified Instances          62               59.0476 %
Incorrectly Classified Instances        43               40.9524 %
Kappa statistic                          0.4226
Mean absolute error                      0.212
Root mean squared error                  0.3137
Relative absolute error                 98.7422 %
Root relative squared error             96.1239 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                  0.61     0.406    0.49       0.61    0.543      0.577     No_Conversion
                  0.821    0.117    0.719      0.821   0.767      0.854     Stage IIA
                  0.375    0.022    0.75       0.375   0.5        0.912     Stage IIB
                  1        0.01     0.889      1       0.941      0.995     Stage I
                  0        0.01     0          0       0          0.913     Stage IIIA
                  0        0.02     0          0       0          0.965     Stage IIIC
                  0        0.019    0          0       0          0.966     Stage IIIB
Weighted Avg.     0.59     0.196    0.565      0.59    0.565      0.775

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 25  9  1  1  1  2  2 |  a = No_Conversion
  4 23  1  0  0  0  0 |  b = Stage IIA
 10  0  6  0  0  0  0 |  c = Stage IIB
  0  0  0  8  0  0  0 |  d = Stage I
  6  0  0  0  0  0  0 |  e = Stage IIIA
  4  0  0  0  0  0  0 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig 16: Result of SMO algorithm by using cross-validation



```
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
```

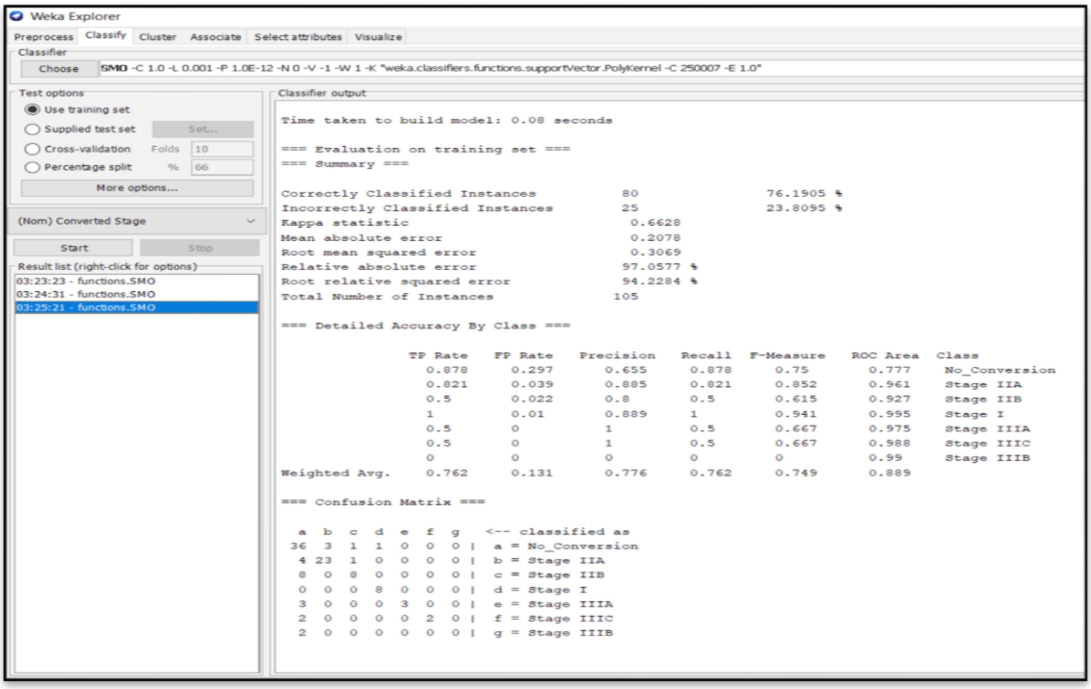
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) Converted Stage

03:23:23 - functions.SMO
03:24:31 - functions.SMO

```
Time taken to build model: 0.08 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          22               61.1111 %
Incorrectly Classified Instances        14               38.8889 %
Kappa statistic                          0.3986
Mean absolute error                      0.2158
Root mean squared error                  0.3199
Relative absolute error                100.12   %
Root relative squared error             97.887  %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   1         0.591     0.519      1         0.683      0.692    No_Conversion
                   0.7       0         1          0.7       0.824      0.867    Stage IIA
                   0.2       0.032     0.5        0.2       0.286      0.842    Stage IIB
                   0         0         0          0         0          0.985    Stage I
                   0         0         0          0         0          0.941    Stage IIIA
                   0         0         0          0         0          0.971    Stage IIIC
                   0         0         0          0         0          0.5      Stage IIIB
Weighted Avg.      0.611     0.234     0.549      0.611     0.534      0.789

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 14  0  0  0  0  0  0 |  a = No_Conversion
  2  7  1  0  0  0  0 |  b = Stage IIA
  4  0  1  0  0  0  0 |  c = Stage IIB
  2  0  0  0  0  0  0 |  d = Stage I
  2  0  0  0  0  0  0 |  e = Stage IIIA
  1  0  0  0  0  0  0 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig17:   Result of SMO algorithm by using percentage split



```
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
```

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) Converted Stage

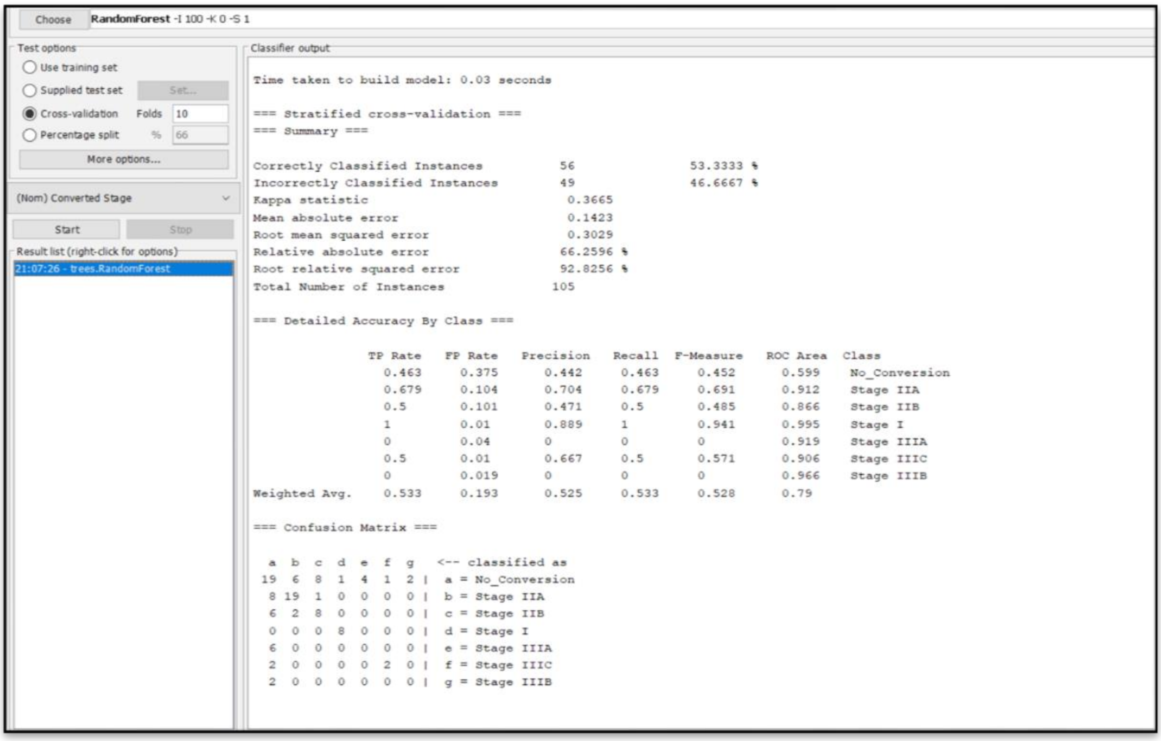
03:23:23 - functions.SMO
03:24:31 - functions.SMO
03:25:21 - functions.SMO

```
Time taken to build model: 0.08 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          80               76.1905 %
Incorrectly Classified Instances        25               23.8095 %
Kappa statistic                          0.6628
Mean absolute error                      0.2078
Root mean squared error                  0.3069
Relative absolute error                 97.0577 %
Root relative squared error             94.2284 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   0.878     0.297     0.655      0.878     0.75       0.777    No_Conversion
                   0.821     0.039     0.885      0.821     0.852      0.961    Stage IIA
                   0.5       0.022     0.8        0.5       0.615      0.927    Stage IIB
                   1         0.01      0.889      1         0.941      0.995    Stage I
                   0.5       0         1          0.5       0.667      0.975    Stage IIIA
                   0.5       0         1          0.5       0.667      0.988    Stage IIIC
                   0         0         0          0         0          0.99     Stage IIIB
Weighted Avg.      0.762     0.131     0.776      0.762     0.749      0.889

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 36  3  1  1  0  0  0 |  a = No_Conversion
  4 23  1  0  0  0  0 |  b = Stage IIA
  8  0  8  0  0  0  0 |  c = Stage IIB
  0  0  0  8  0  0  0 |  d = Stage I
  3  0  0  0  3  0  0 |  e = Stage IIIA
  2  0  0  0  0  2  0 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig 18: Result of SMO using training set

### 2.3.4 RandomForest classification:

Random forest algorithm is mostly used for both classification and regression task. The main advantages of this algorithm is that it can handle the missing values so it can be easily used for featured engineering. In WEKA we applied random forest to check how it works with our dataset; and we set out the best accuracy among all the algorithm we have used.



Choose RandomForest -I 100 -K 0 -S 1

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Converted Stage

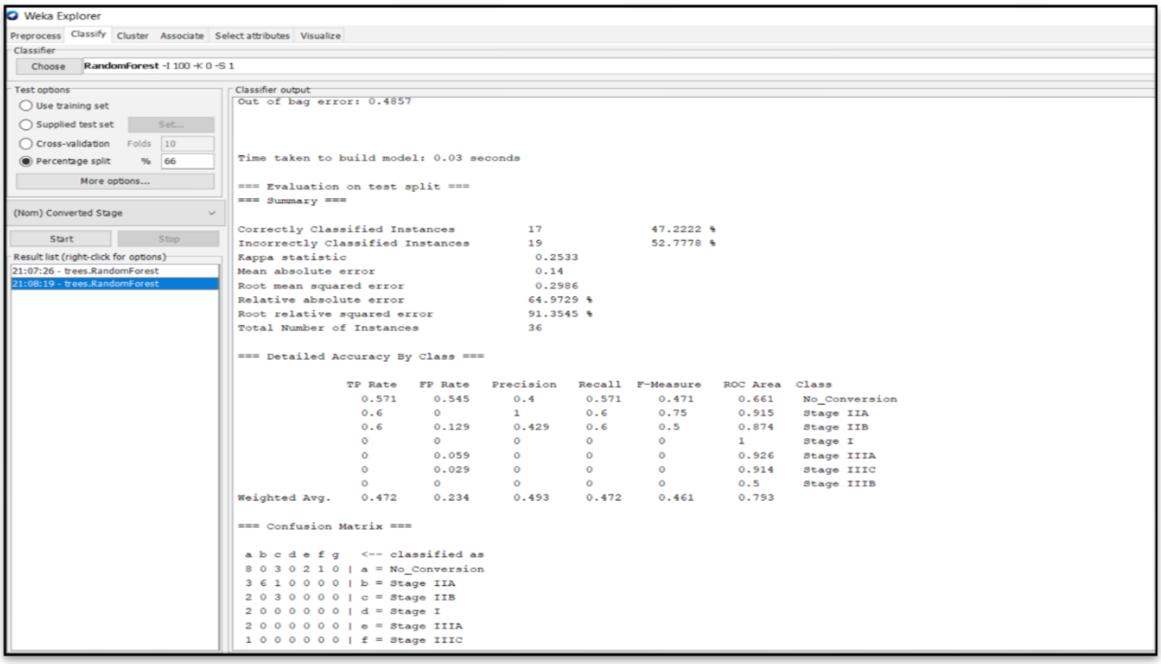
Result list (right-click for options)

21:07:26 - trees.RandomForest

Classifier output

```
Time taken to build model: 0.03 seconds

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

Correctly Classified Instances          56               53.3333 %
Incorrectly Classified Instances        49               46.6667 %
Kappa statistic                          0.3665
Mean absolute error                      0.1423
Root mean squared error                  0.3029
Relative absolute error                 66.2596 %
Root relative squared error             92.8256 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
                 0.463     0.375     0.442      0.463     0.452      0.599     No_Conversion
                 0.679     0.104     0.704      0.679     0.691      0.912     Stage IIA
                 0.5       0.101     0.471      0.5       0.485      0.866     Stage IIB
                 1         0.01      0.889      1         0.941      0.995     Stage I
                 0         0.04      0          0         0          0.919     Stage IIIA
                 0.5       0.01      0.667      0.5       0.571      0.906     Stage IIIC
                 0         0.019     0          0         0          0.966     Stage IIIB
Weighted Avg.    0.533     0.193     0.525      0.533     0.528      0.79

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 19  6  8  1  4  1  2 |  a = No_Conversion
  8 19  1  0  0  0  0 |  b = Stage IIA
  6  2  8  0  0  0  0 |  c = Stage IIB
  0  0  0  8  0  0  0 |  d = Stage I
  6  0  0  0  0  0  0 |  e = Stage IIIA
  2  0  0  0  0  2  0 |  f = Stage IIIC
  2  0  0  0  0  0  0 |  g = Stage IIIB
```

Fig 19: Result of Random forest algorithm by using cross-validation



Classifier: RandomForest -I 100 -K 0 -S 1

Test options
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

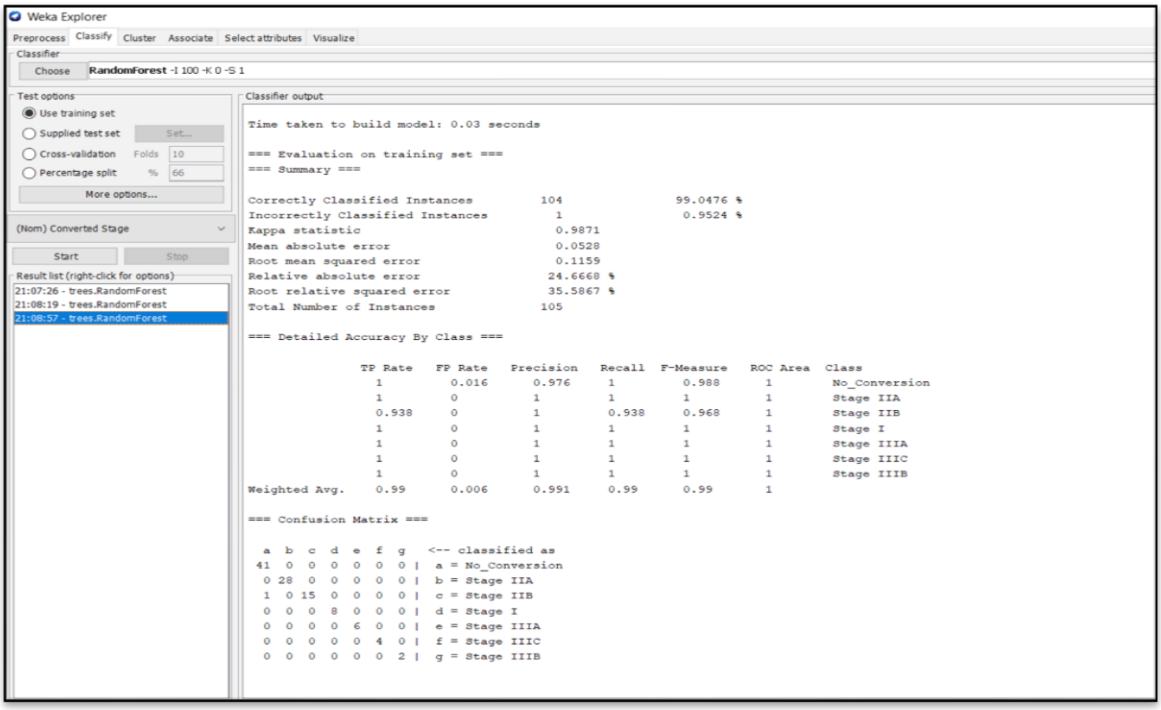
(Nom) Converted Stage

Result list (right-click for options)
21:07:26 - trees.RandomForest
21:08:19 - trees.RandomForest

Classifier output

```
Out of bag error: 0.4857



Time taken to build model: 0.03 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          17               47.2222 %
Incorrectly Classified Instances        19               52.7778 %
Kappa statistic                          0.2533
Mean absolute error                      0.14
Root mean squared error                  0.2986
Relative absolute error                 64.9729 %
Root relative squared error             91.3545 %
Total Number of Instances               36

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.571     0.545     0.4        0.571     0.471      0.661    No_Conversion
                 0.6       0         1          0.6       0.75       0.915    Stage IIA
                 0.6       0.129     0.429      0.6       0.5        0.874    Stage IIB
                 0         0         0          0         0          1        Stage I
                 0         0.059     0          0         0          0.926    Stage IIIA
                 0         0.029     0          0         0          0.914    Stage IIIC
                 0         0         0          0         0          0.5      Stage IIIB
Weighted Avg.    0.472     0.234     0.493      0.472     0.461      0.793

=== Confusion Matrix ===

 a b c d e f g   <-- classified as
 8 0 3 0 2 1 0 | a = No_Conversion
 3 6 1 0 0 0 0 | b = Stage IIA
 2 0 3 0 0 0 0 | c = Stage IIB
 2 0 0 0 0 0 0 | d = Stage I
 2 0 0 0 0 0 0 | e = Stage IIIA
 1 0 0 0 0 0 0 | f = Stage IIIC
```

Fig 20:  Result of Random forest algorithm by using percentage split



Classifier: RandomForest -I 100 -K 0 -S 1

Test options
● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) Converted Stage

Result list (right-click for options)
21:07:26 - trees.RandomForest
21:08:19 - trees.RandomForest
21:08:57 - trees.RandomForest

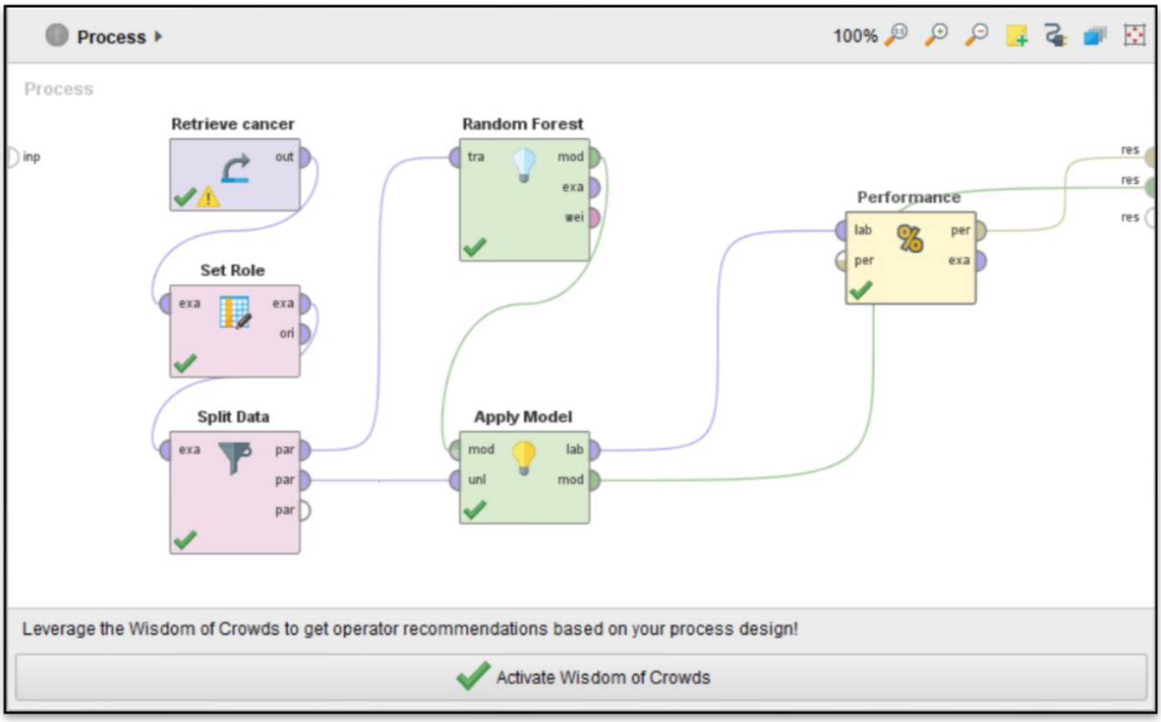
Classifier output

```
Time taken to build model: 0.03 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         104               99.0476 %
Incorrectly Classified Instances         1                0.9524 %
Kappa statistic                          0.9871
Mean absolute error                      0.0528
Root mean squared error                  0.1159
Relative absolute error                 24.6668 %
Root relative squared error             35.5867 %
Total Number of Instances              105

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.016     0.976      1         0.988      1         No_Conversion
                 1         0         1          1         1          1         Stage IIA
                 0.938     0         1          0.938     0.968      1         Stage IIB
                 1         0         1          1         1          1         Stage I
                 1         0         1          1         1          1         Stage IIIA
                 1         0         1          1         1          1         Stage IIIC
                 1         0         1          1         1          1         Stage IIIB
Weighted Avg.    0.99      0.006     0.991      0.99      0.99       1

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = No_Conversion
  0 28  0  0  0  0  0 |  b = Stage IIA
  1  0 15  0  0  0  0 |  c = Stage IIB
  0  0  0  8  0  0  0 |  d = Stage I
  0  0  0  0  6  0  0 |  e = Stage IIIA
  0  0  0  0  0  4  0 |  f = Stage IIIC
  0  0  0  0  0  0  2 |  g = Stage IIIB
```

Fig 21:  Result of Random forest algorithm by using training set

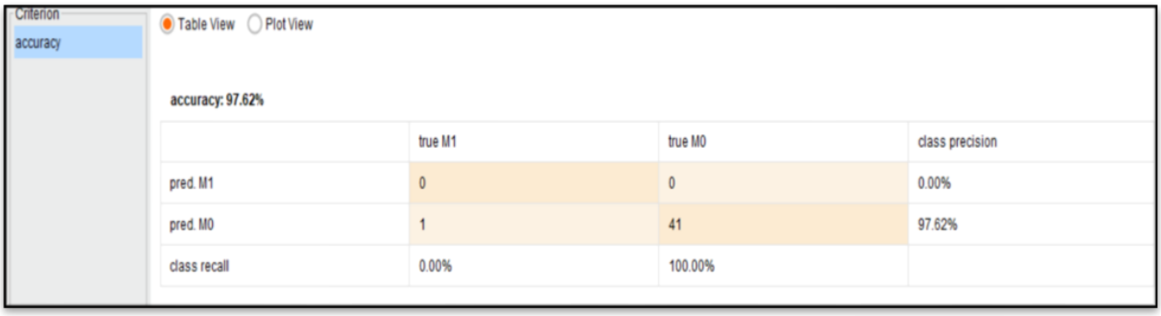
From overall results we can see that when we used training set it gave the most accuracy over the data.

Next we moved on to the RapidMiner tool to check whether this tool give any different result on same algorithm or not. And the result showed less accuracy then WEKA.



Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Fig 22: Design of random forest model in Rapidminer



accuracy: 97.62%

| | true M1 | true M0 | class precision |
|---|---|---|---|
| pred. M1 | 0 | 0 | 0.00% |
| pred. M0 | 1 | 41 | 97.62% |
| class recall | 0.00% | 100.00% | |

Fig 23: Accuracy measurement by using metastasis as a target attribute

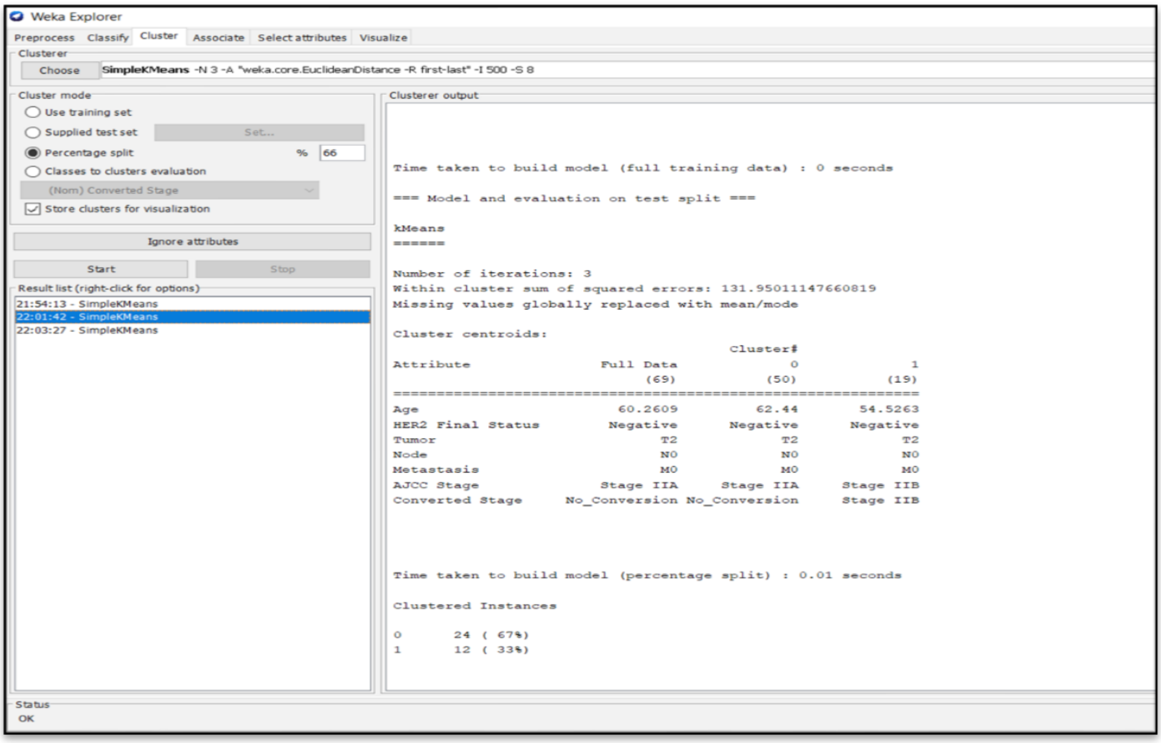
The above figure shows us that, though there was no huge difference between “WEKA” and “RapinMiner” over RandomForest algorithm; but here ‘WEKA’ set the best accuracy over this dataset.

## 2.3.4 Clustering algorithm:

*In order to predict the best predictor model we again apply two clustering process; K-means and EM. They both are iterative algorithms. EM(Expectation-maximizations) is a statistical model that depends on unobserved latent variables to estimate the parameters using maximum likelihood; where K-means clustering algorithm works by partitioning n observation into k sub classes.*

### a. The output of k-means over dataset:

We got two clusters, each cluster has two instances. The clustering produced by k-means shows 67% (24instances) in cluster 0 and 33% (12 instances) in cluster 1, in figure 23 and figure 24 shows 8% (3 instances) in cluster 0 and 92% (33 instances)



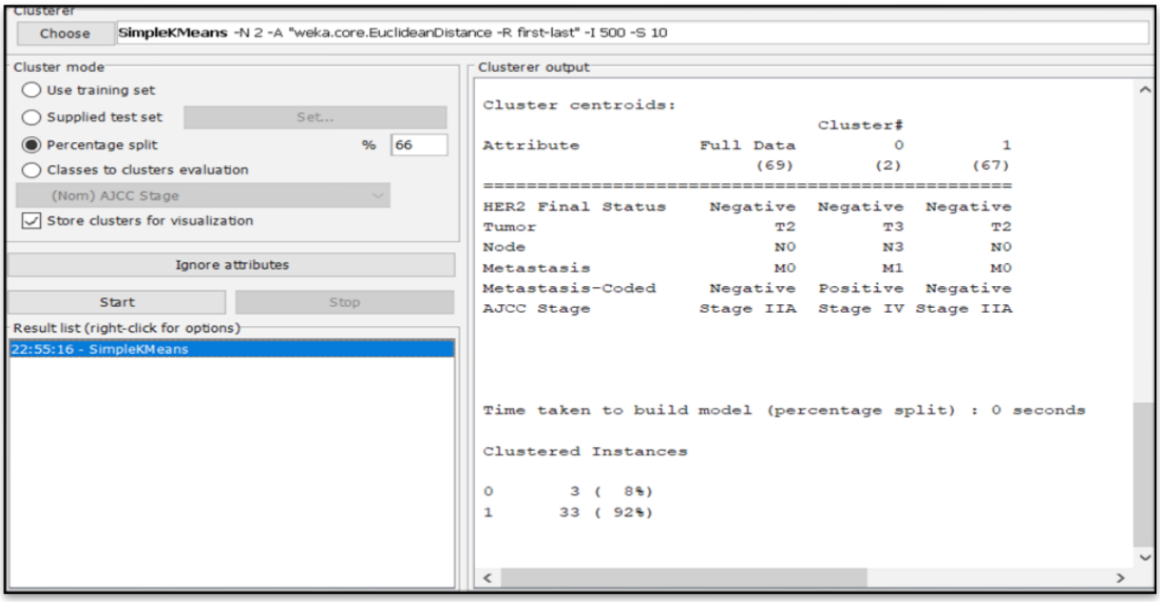
SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 8

○ Use training set
○ Supplied test set
● Percentage split % 66
○ Classes to clusters evaluation
(Nom) Converted Stage
☑ Store clusters for visualization

21:54:13 - SimpleKMeans
22:01:42 - SimpleKMeans
22:03:27 - SimpleKMeans

```
Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on test split ===

kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 131.95011147660819
Missing values globally replaced with mean/mode

Cluster centroids:
                                         Cluster#
Attribute                   Full Data           0              1
                                 (69)        (50)           (19)
================================================================
Age                           60.2609       62.44        54.5263
HER2 Final Status            Negative    Negative       Negative
Tumor                              T2          T2             T2
Node                               N0          N0             N0
Metastasis                         M0          M0             M0
AJCC Stage                  Stage IIA   Stage IIA      Stage IIB
Converted Stage         No_Conversion No_Conversion    Stage IIB




Time taken to build model (percentage split) : 0.01 seconds

Clustered Instances

0       24 ( 67%)
1       12 ( 33%)
```

Fig 24: K means (metastasis) clustering using percentage split



SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

○ Use training set
○ Supplied test set
● Percentage split % 66
○ Classes to clusters evaluation
(Nom) AJCC Stage
☑ Store clusters for visualization

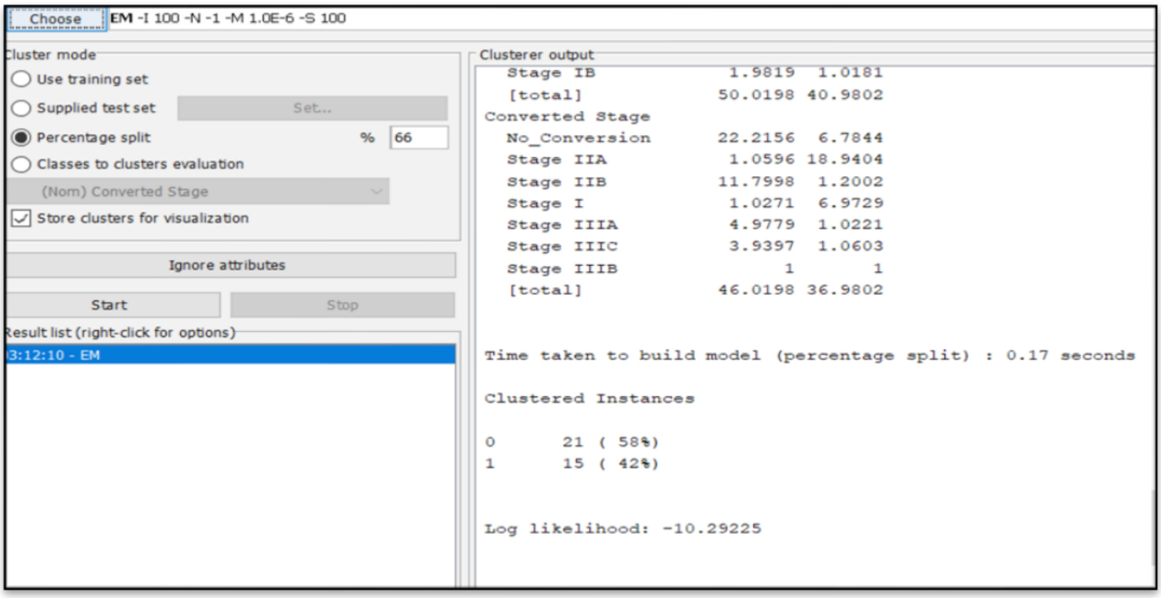
22:55:16 - SimpleKMeans

```
Cluster centroids:
                                    Cluster#
Attribute            Full Data          0          1
                          (69)        (2)       (67)
====================================================
HER2 Final Status     Negative   Negative   Negative
Tumor                       T2         T3         T2
Node                        N0         N3         N0
Metastasis                  M0         M1         M0
Metastasis-Coded      Negative   Positive   Negative
AJCC Stage           Stage IIA   Stage IV  Stage IIA




Time taken to build model (percentage split) : 0 seconds

Clustered Instances

0        3 (  8%)
1       33 ( 92%)
```

Fig 25: K means (tumor) clustering using percentage split

## b. The output of EM over dataset:

Like K means got two clusters in EM also; here they have two instances too. The clustering produced by k-means shows 58% (21instances) in cluster 0 and 42% (15 instances) in cluster 1, in figure 25 and figure 26 shows 67% (24 instances) in cluster 0 and 33% (12 instances)



EM -I 100 -N -1 -M 1.0E-6 -S 100

Cluster mode

○ Use training set

○ Supplied test set

● Percentage split % 66

○ Classes to clusters evaluation

(Nom) Converted Stage

☑ Store clusters for visualization

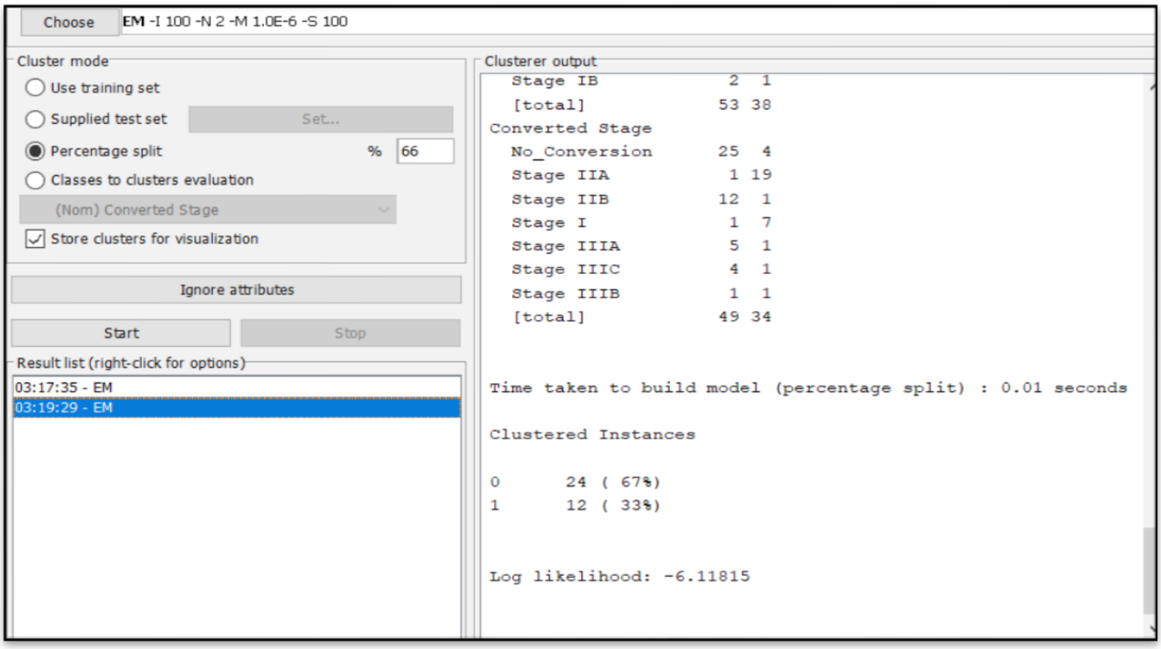
Result list (right-click for options)

3:12:10 - EM

Clusterer output

```
  Stage IB               1.9819  1.0181
  [total]               50.0198 40.9802
Converted Stage
  No_Conversion         22.2156  6.7844
  Stage IIA              1.0596 18.9404
  Stage IIB             11.7998  1.2002
  Stage I                1.0271  6.9729
  Stage IIIA             4.9779  1.0221
  Stage IIIC             3.9397  1.0603
  Stage IIIB                  1       1
  [total]               46.0198 36.9802


Time taken to build model (percentage split) : 0.17 seconds

Clustered Instances

0       21 ( 58%)
1       15 ( 42%)


Log likelihood: -10.29225
```

Fig26:  EM (tumor) clustering using percentage split



EM -I 100 -N 2 -M 1.0E-6 -S 100

Cluster mode

○ Use training set

○ Supplied test set

● Percentage split % 66

○ Classes to clusters evaluation

(Nom) Converted Stage

☑ Store clusters for visualization

Clusterer output

```
  Stage IB              2  1
  [total]              53 38
Converted Stage
  No_Conversion        25  4
  Stage IIA             1 19
  Stage IIB            12  1
  Stage I               1  7
  Stage IIIA            5  1
  Stage IIIC            4  1
  Stage IIIB            1  1
  [total]              49 34


Time taken to build model (percentage split) : 0.01 seconds

Clustered Instances

0       24 ( 67%)
1       12 ( 33%)


Log likelihood: -6.11815
```

Fig27: EM (metastasis) clustering using percentage split.

## 2.4 Association rules for breast cancer observation:

Association rules can be define as the process of finding valuable association rules and/or relationship among amount of data. Through this technique is possible to quantify the value of each feature by evaluation its frequency within the dataset, thus allowing to capture all possible rules that explain the presence of some features according to the features of another features.

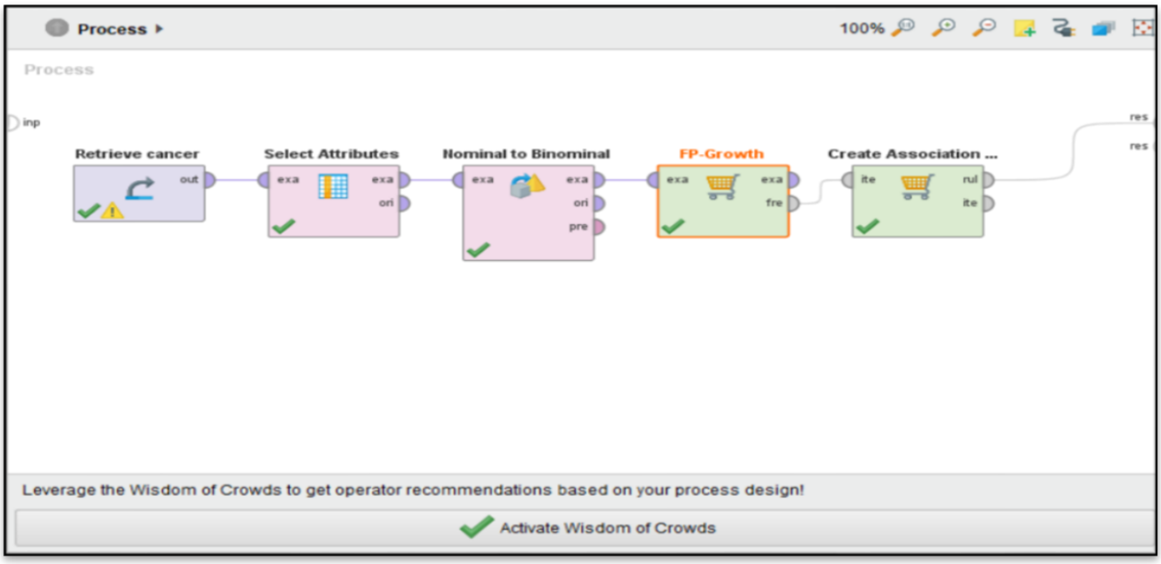
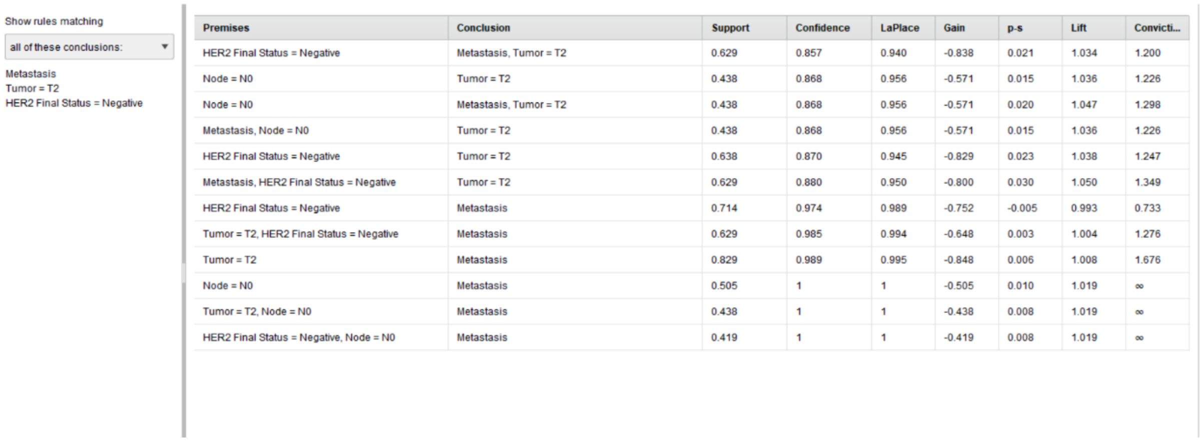


Fig 28:  Design of association rule in Rapidminer.


Show rules matching

all of these conclusions:

Metastasis
Tumor = T2
HER2 Final Status = Negative

| Premises | Conclusion | Support | Confidence | LaPlace | Gain | p-s | Lift | Convicti... |
|---|---|---|---|---|---|---|---|---|
| HER2 Final Status = Negative | Metastasis, Tumor = T2 | 0.629 | 0.857 | 0.940 | -0.838 | 0.021 | 1.034 | 1.200 |
| Node = N0 | Tumor = T2 | 0.438 | 0.868 | 0.956 | -0.571 | 0.015 | 1.036 | 1.226 |
| Node = N0 | Metastasis, Tumor = T2 | 0.438 | 0.868 | 0.956 | -0.571 | 0.020 | 1.047 | 1.298 |
| Metastasis, Node = N0 | Tumor = T2 | 0.438 | 0.868 | 0.956 | -0.571 | 0.015 | 1.036 | 1.226 |
| HER2 Final Status = Negative | Tumor = T2 | 0.638 | 0.870 | 0.945 | -0.829 | 0.023 | 1.038 | 1.247 |
| Metastasis, HER2 Final Status = Negative | Tumor = T2 | 0.629 | 0.880 | 0.950 | -0.800 | 0.030 | 1.050 | 1.349 |
| HER2 Final Status = Negative | Metastasis | 0.714 | 0.974 | 0.989 | -0.752 | -0.005 | 0.993 | 0.733 |
| Tumor = T2, HER2 Final Status = Negative | Metastasis | 0.629 | 0.985 | 0.994 | -0.648 | 0.003 | 1.004 | 1.276 |
| Tumor = T2 | Metastasis | 0.829 | 0.989 | 0.995 | -0.848 | 0.006 | 1.008 | 1.676 |
| Node = N0 | Metastasis | 0.505 | 1 | 1 | -0.505 | 0.010 | 1.019 | ∞ |
| Tumor = T2, Node = N0 | Metastasis | 0.438 | 1 | 1 | -0.438 | 0.008 | 1.019 | ∞ |
| HER2 Final Status = Negative, Node = N0 | Metastasis | 0.419 | 1 | 1 | -0.419 | 0.008 | 1.019 | ∞ |

Fig 29: Rules given by Association model

We got 12 rules (fig28) after applying association mining on our dataset. We picked ‘Metastasis’ as our target attribute as in earlier case, classification and clustering algorithm gave us the best accuracy with it. Among the 12 rules 7 rules set the relationship with metastasis. So now considered all that rules for our process.

## Relations between the attributes getting from rules:

Figure 28 shows the relationship between tumor =2, HER2 final status = negative and metastasis. We got various confidence in this three relation. This three attribute relate that this three stage of cancer can be act as a risk factor of breast cancer.

### 2.5 Co-relation analysis:

RapidMiner this is very easy to do using the Correlation Matrix operator. In order to use it however we first need to join the two Example sets that we created separately, so we'll have the words and the aspect: polarity pairs in one dataset. We took that attributes to create co-relation which we had get in the association rule for building the relation between the cause and risk factor of breast cancer.

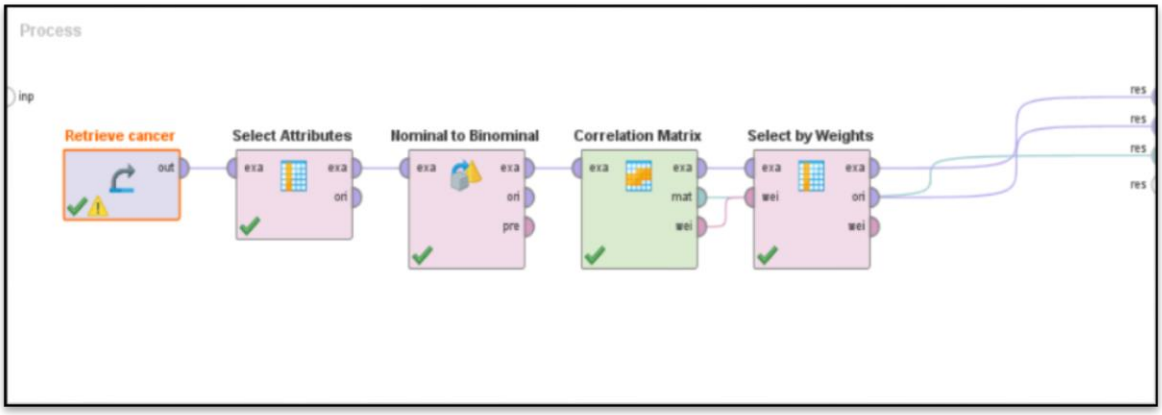
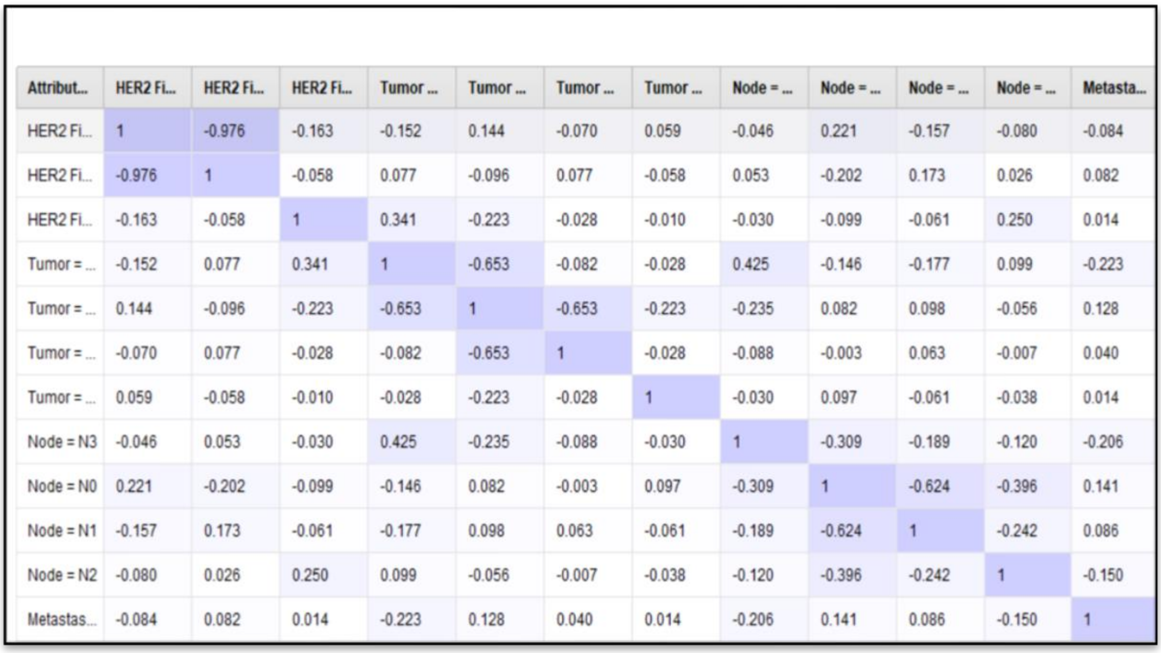


Fig 30: Model of correlation matrix in rapidminer

**Applied Correlation Matrix resembled the one below:**

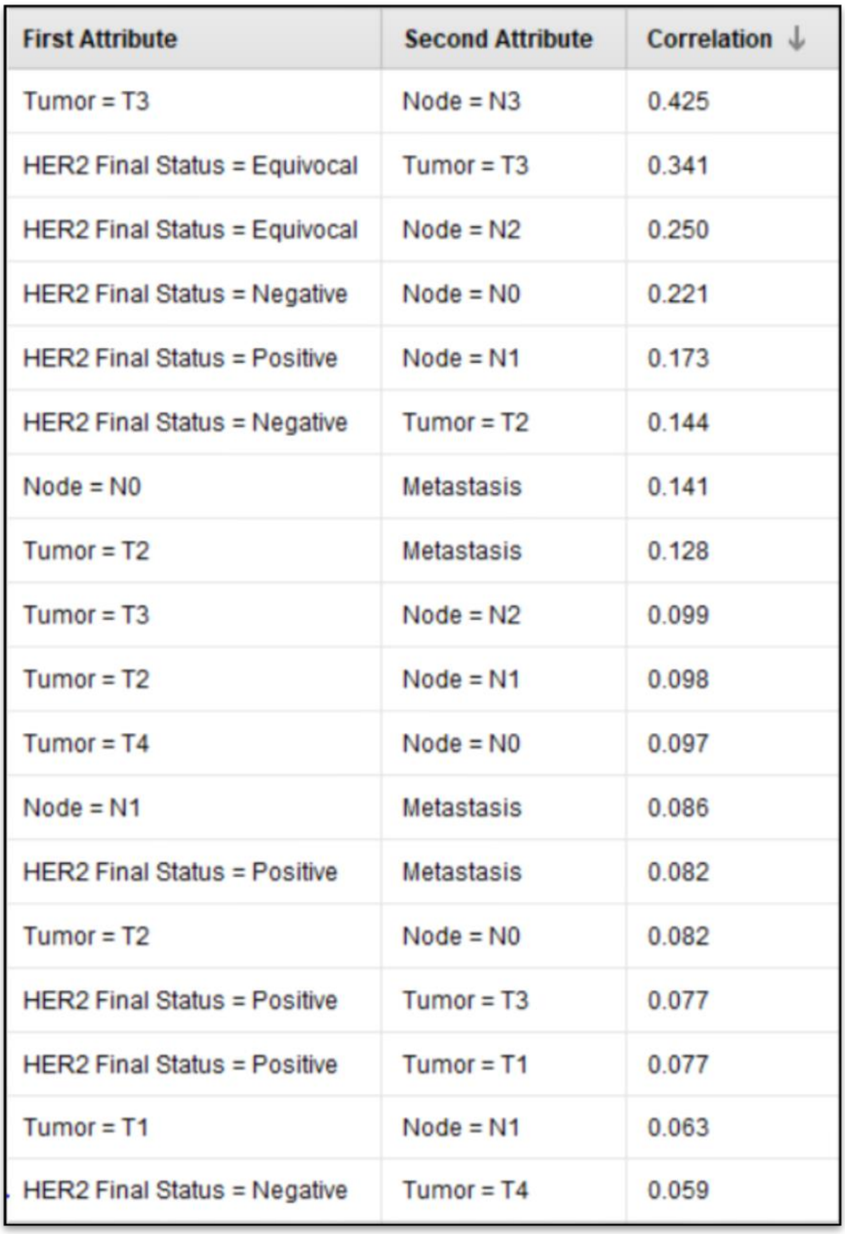


| Attribut... | HER2 Fi... | HER2 Fi... | HER2 Fi... | Tumor ... | Tumor ... | Tumor ... | Tumor ... | Node = ... | Node = ... | Node = ... | Node = ... | Metasta... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HER2 Fi... | 1 | -0.976 | -0.163 | -0.152 | 0.144 | -0.070 | 0.059 | -0.046 | 0.221 | -0.157 | -0.080 | -0.084 |
| HER2 Fi... | -0.976 | 1 | -0.058 | 0.077 | -0.096 | 0.077 | -0.058 | 0.053 | -0.202 | 0.173 | 0.026 | 0.082 |
| HER2 Fi... | -0.163 | -0.058 | 1 | 0.341 | -0.223 | -0.028 | -0.010 | -0.030 | -0.099 | -0.061 | 0.250 | 0.014 |
| Tumor = ... | -0.152 | 0.077 | 0.341 | 1 | -0.653 | -0.082 | -0.028 | 0.425 | -0.146 | -0.177 | 0.099 | -0.223 |
| Tumor = ... | 0.144 | -0.096 | -0.223 | -0.653 | 1 | -0.653 | -0.223 | -0.235 | 0.082 | 0.098 | -0.056 | 0.128 |
| Tumor = ... | -0.070 | 0.077 | -0.028 | -0.082 | -0.653 | 1 | -0.028 | -0.088 | -0.003 | 0.063 | -0.007 | 0.040 |
| Tumor = ... | 0.059 | -0.058 | -0.010 | -0.028 | -0.223 | -0.028 | 1 | -0.030 | 0.097 | -0.061 | -0.038 | 0.014 |
| Node = N3 | -0.046 | 0.053 | -0.030 | 0.425 | -0.235 | -0.088 | -0.030 | 1 | -0.309 | -0.189 | -0.120 | -0.206 |
| Node = N0 | 0.221 | -0.202 | -0.099 | -0.146 | 0.082 | -0.003 | 0.097 | -0.309 | 1 | -0.624 | -0.396 | 0.141 |
| Node = N1 | -0.157 | 0.173 | -0.061 | -0.177 | 0.098 | 0.063 | -0.061 | -0.189 | -0.624 | 1 | -0.242 | 0.086 |
| Node = N2 | -0.080 | 0.026 | 0.250 | 0.099 | -0.056 | -0.007 | -0.038 | -0.120 | -0.396 | -0.242 | 1 | -0.150 |
| Metastas... | -0.084 | 0.082 | 0.014 | -0.223 | 0.128 | 0.040 | 0.014 | -0.206 | 0.141 | 0.086 | -0.150 | 1 |

Fig 31: correlation matrix

The higher the correlation coefficient (the values in the matrix), the stronger the correlation, with 1 being the highest and -1 the lowest, i.e. an inverse correlation.

Using the matrix table we filtered and identify words extracted from reviews that correlate with a certain aspect



| First Attribute | Second Attribute | Correlation ↓ |
|---|---|---|
| Tumor = T3 | Node = N3 | 0.425 |
| HER2 Final Status = Equivocal | Tumor = T3 | 0.341 |
| HER2 Final Status = Equivocal | Node = N2 | 0.250 |
| HER2 Final Status = Negative | Node = N0 | 0.221 |
| HER2 Final Status = Positive | Node = N1 | 0.173 |
| HER2 Final Status = Negative | Tumor = T2 | 0.144 |
| Node = N0 | Metastasis | 0.141 |
| Tumor = T2 | Metastasis | 0.128 |
| Tumor = T3 | Node = N2 | 0.099 |
| Tumor = T2 | Node = N1 | 0.098 |
| Tumor = T4 | Node = N0 | 0.097 |
| Node = N1 | Metastasis | 0.086 |
| HER2 Final Status = Positive | Metastasis | 0.082 |
| Tumor = T2 | Node = N0 | 0.082 |
| HER2 Final Status = Positive | Tumor = T3 | 0.077 |
| HER2 Final Status = Positive | Tumor = T1 | 0.077 |
| Tumor = T1 | Node = N1 | 0.063 |
| HER2 Final Status = Negative | Tumor = T4 | 0.059 |

Table2: correlation attributes

# Chapter 3

# Result and Analysis

## 3.1 Performance evaluation of all classification models:

| Performance Criteria | J48 | | | Naïve Bayes | | | SMO | | | Random Forest | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unlikely | Likely | Average | Unlike | Likely | Average | Likely | Likely | Average | Likely | Likely | Average |
| FP rate | 0.021 | 0.219 | 0.039 | 0 | 0.203 | 0.117 | 0.01 | 0.297 | 0.039 | 0 | 0.16 | 0 |
| TP rate | 1 | 0.805 | 0.821 | 0 | 0.61 | 0.893 | 1 | 0.878 | 0.821 | 1 | 1 | 1 |
| Precision | 0.8 | 0.702 | 0.885 | 1 | 0.658 | 0.735 | 0.889 | 0.655 | 0.885 | 1 | 0.976 | 1 |
| ROC Area | 0.994 | 0.84 | 0.962 | 1 | 0.78 | 0.963 | 0.995 | 0.777 | 0.961 | 1 | 1 | 1 |

| | **J48** | **Naïve Bayes** | **SMO** | **Random Forest** |
|---|---|---|---|---|
| Correct Classification | 81 | 74 | 80 | 104 |
| Incorrect Classification | 24 | 31 | 25 | 1 |
| **Accuracy (%)** | 76.6 | 69.3 | 74.9 | 99.04 |
| **Error rate** | 0.094 | 0.124 | 0.207 | 0.053 |

Table 3: Analysis between Different Algorithms

From the four models developed for the prediction of breast cancer risk, we can see the difference between their accuracy, error rates, correct and incorrect classification.

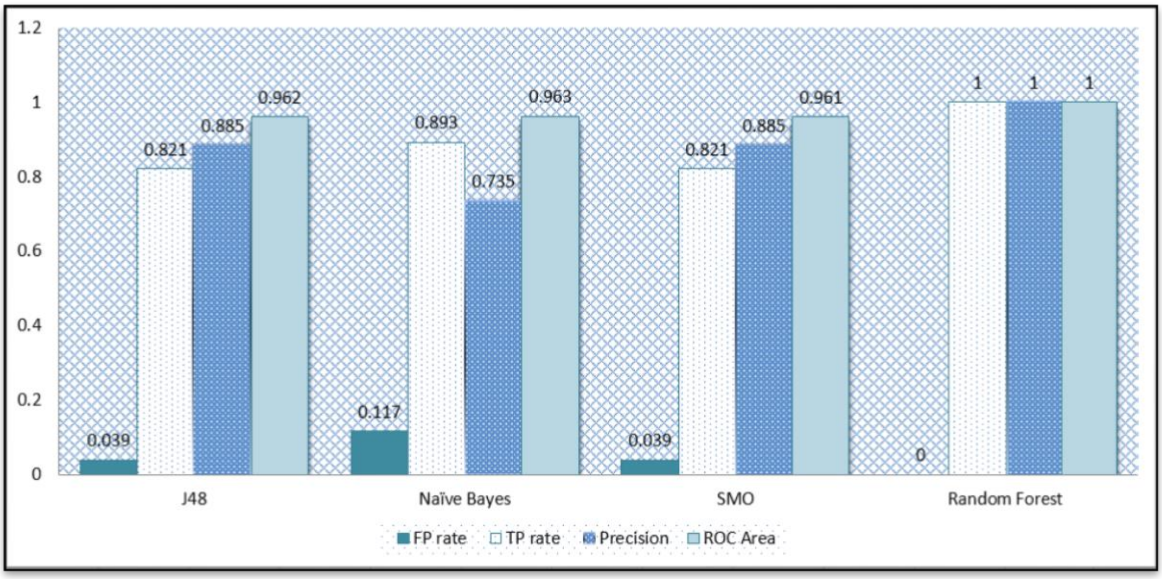


Fig 32: Performance evaluation of J48, Naïve Bayes, SMO and Random forest

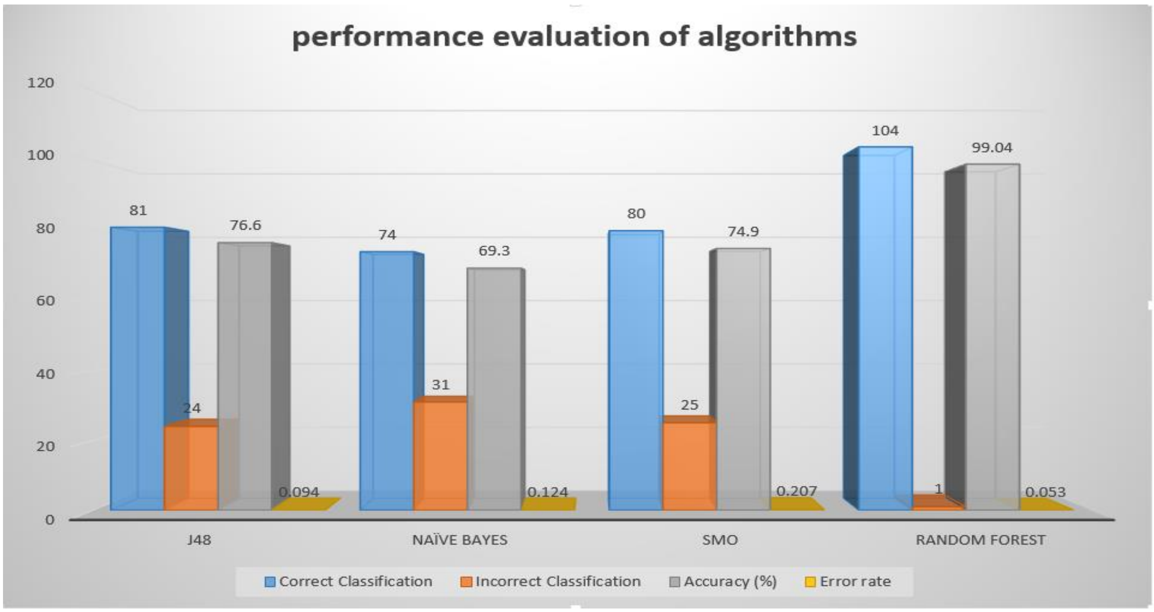


Fig 33: Performance evaluation of J48, Naïve Bayes, SMO and Random

## 3.2 Performance evaluation table of two clustering models:

| Cluster type | k-means<br>(tumor) | EM<br>(tumor) | k-means<br>(metastasis) | EM<br>(metastasis) |
|---|---|---|---|---|
| Cluster instances | | | | |
| 0 | 8% | 67% | 67% | 58% |
| 1 | 92% | 33% | 33% | 42% |

Table 4: Analysis between 2 clustering

From the table we can evaluate that according to the dataset when we set the target attribute (tumor) k means gave the best result and when we set the target attribute (metastasis) EM gave the best result.

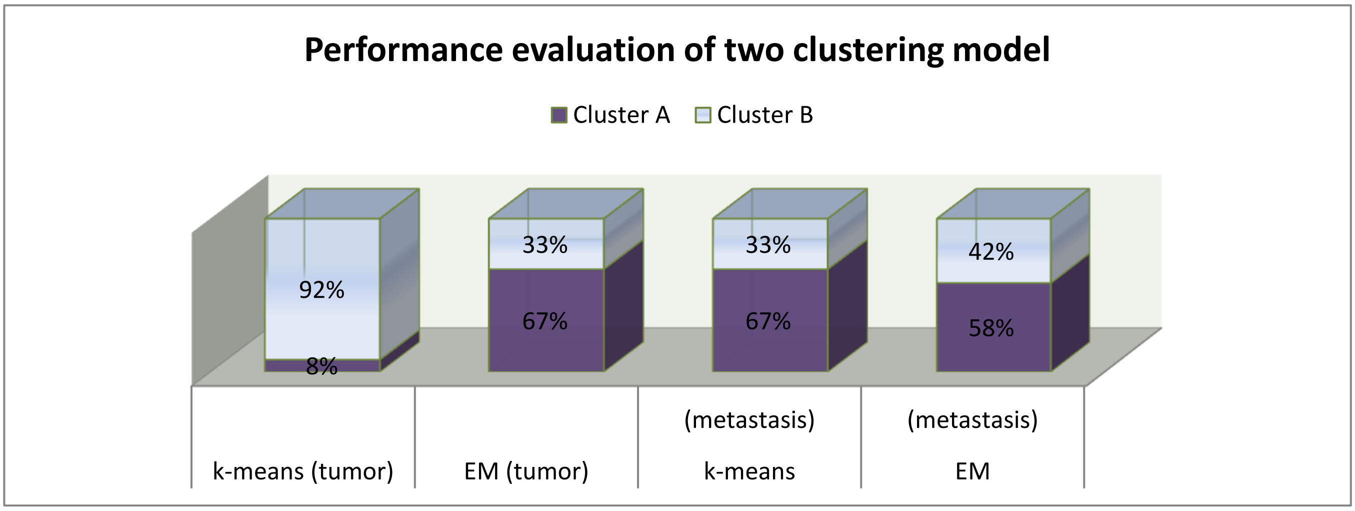


Fig 34 : performance evaluation of two clustering model

## 3.3 Performance evaluation of association rules mining:

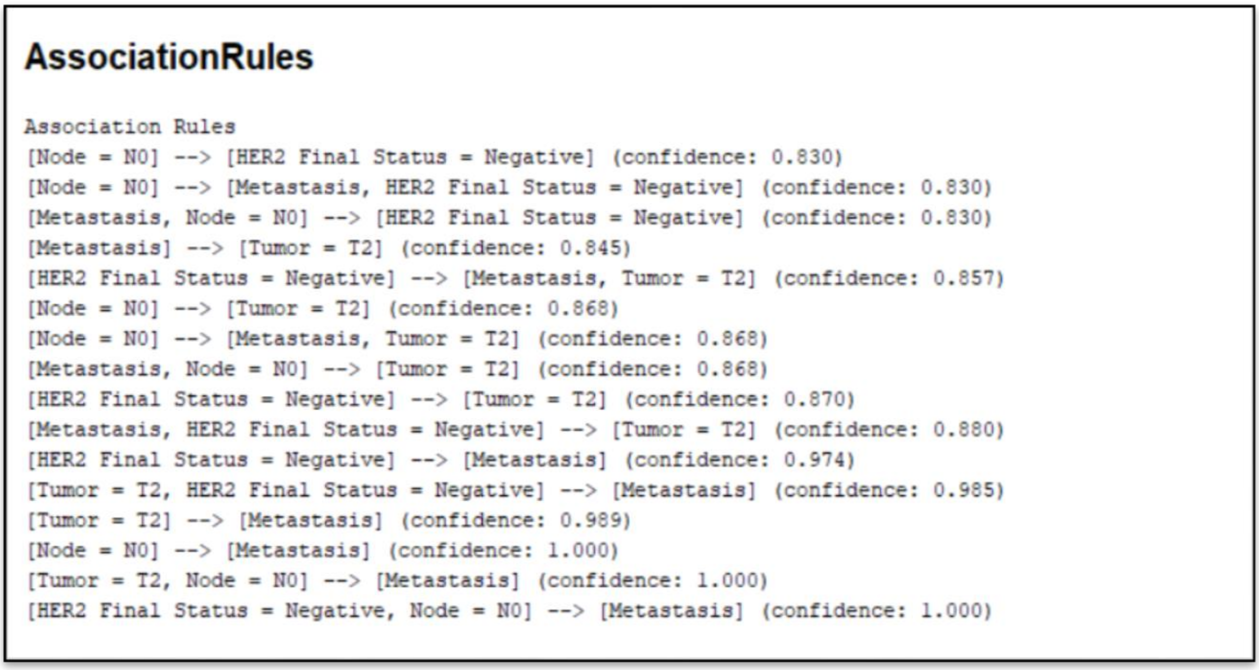
TNM (Tumor, Node, Metastasis) is another staging system researchers use to provide more details about how the cancer looks and behaves. In figure 34, we can see that we have 100% confidence when there is a relationship between Tumor=T2, Node = N0, HER2 final status = negative and Metastasis. The relation between tumor size and lymph node status was investigated in detail. Tumor diameter and lymph node status were found to act as independent but additive prognostic indicators. As tumor size increased, survival decreased regardless of lymph node status; and as lymph node involvement increased, survival status also decreased regardless of tumor size.

T0; means there isn't any evidence of the primary tumor. T1, T2, T3 and T4: These numbers are based on the size of the tumor and the extent to which it has grown into tissue. The higher the T number, the larger the tumor and/or the more it may have grown into the breast tissue.

N0 means nearby lymph nodes do not contain cancer. N1, N2, and N3: These numbers are based on the number of lymph nodes involved and how much cancer is found in them. The higher the N number, the greater the extent of the lymph node involvement.

The M (metastasis) category tells whether or not there is evidence that the cancer has traveled to other parts of the body: M0 means there is no distant metastasis; M1 means that distant metastasis is present.

As from our analysis we had the relationship between T2 ->N0->MO; it would mean that the primary breast tumor is equal to 2 centimeters across (T2), has not involved the lymph nodes (N0), and has not spread to distant parts of the body (M0). This cancer would be grouped as stage I. Which can be considered as invasive breast cancer and can be cured.



## AssociationRules

```
Association Rules
[Node = N0] --> [HER2 Final Status = Negative] (confidence: 0.830)
[Node = N0] --> [Metastasis, HER2 Final Status = Negative] (confidence: 0.830)
[Metastasis, Node = N0] --> [HER2 Final Status = Negative] (confidence: 0.830)
[Metastasis] --> [Tumor = T2] (confidence: 0.845)
[HER2 Final Status = Negative] --> [Metastasis, Tumor = T2] (confidence: 0.857)
[Node = N0] --> [Tumor = T2] (confidence: 0.868)
[Node = N0] --> [Metastasis, Tumor = T2] (confidence: 0.868)
[Metastasis, Node = N0] --> [Tumor = T2] (confidence: 0.868)
[HER2 Final Status = Negative] --> [Tumor = T2] (confidence: 0.870)
[Metastasis, HER2 Final Status = Negative] --> [Tumor = T2] (confidence: 0.880)
[HER2 Final Status = Negative] --> [Metastasis] (confidence: 0.974)
[Tumor = T2, HER2 Final Status = Negative] --> [Metastasis] (confidence: 0.985)
[Tumor = T2] --> [Metastasis] (confidence: 0.989)
[Node = N0] --> [Metastasis] (confidence: 1.000)
[Tumor = T2, Node = N0] --> [Metastasis] (confidence: 1.000)
[HER2 Final Status = Negative, Node = N0] --> [Metastasis] (confidence: 1.000)
```

Fig 35: linked list of the association rules.

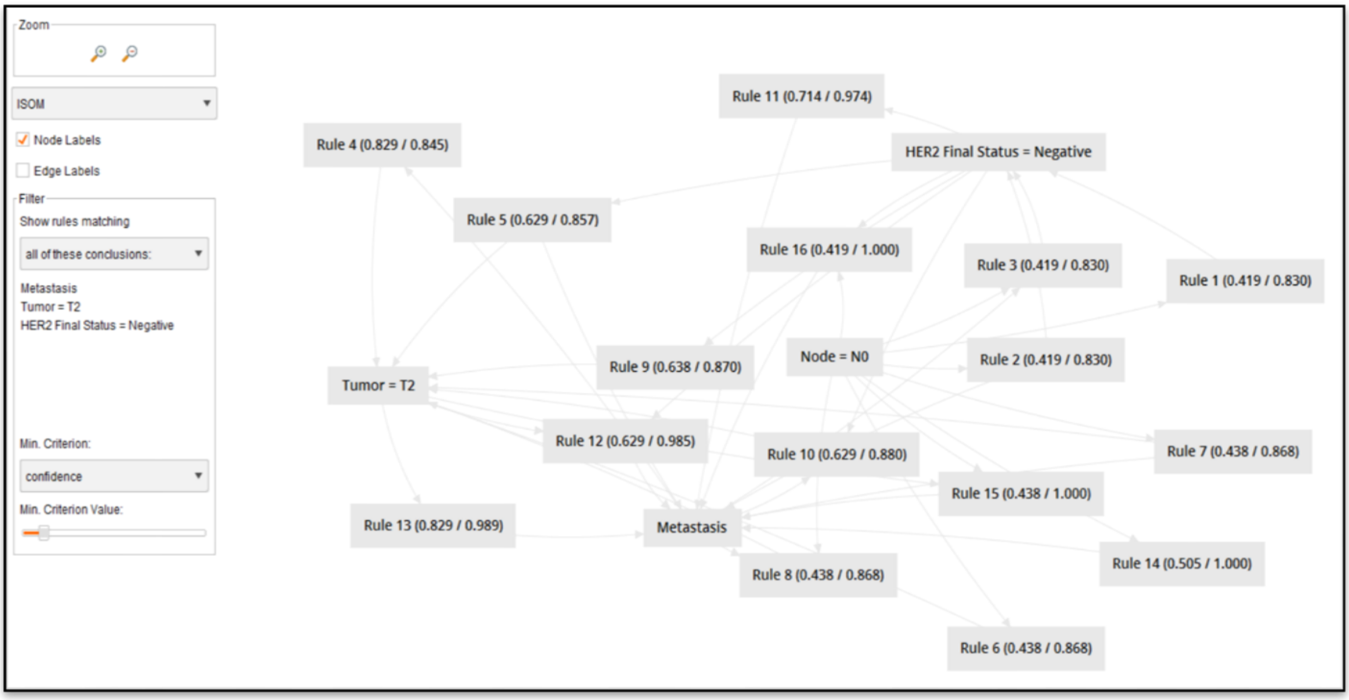


Figure 36: Tree of the association rules we get from result.

If we set the result of association rule in correlation we can get the following scatter graph of the correlation matrix. Which specifically showed us the exact target attributes relationship to identify the stage of breast cancer from our result.

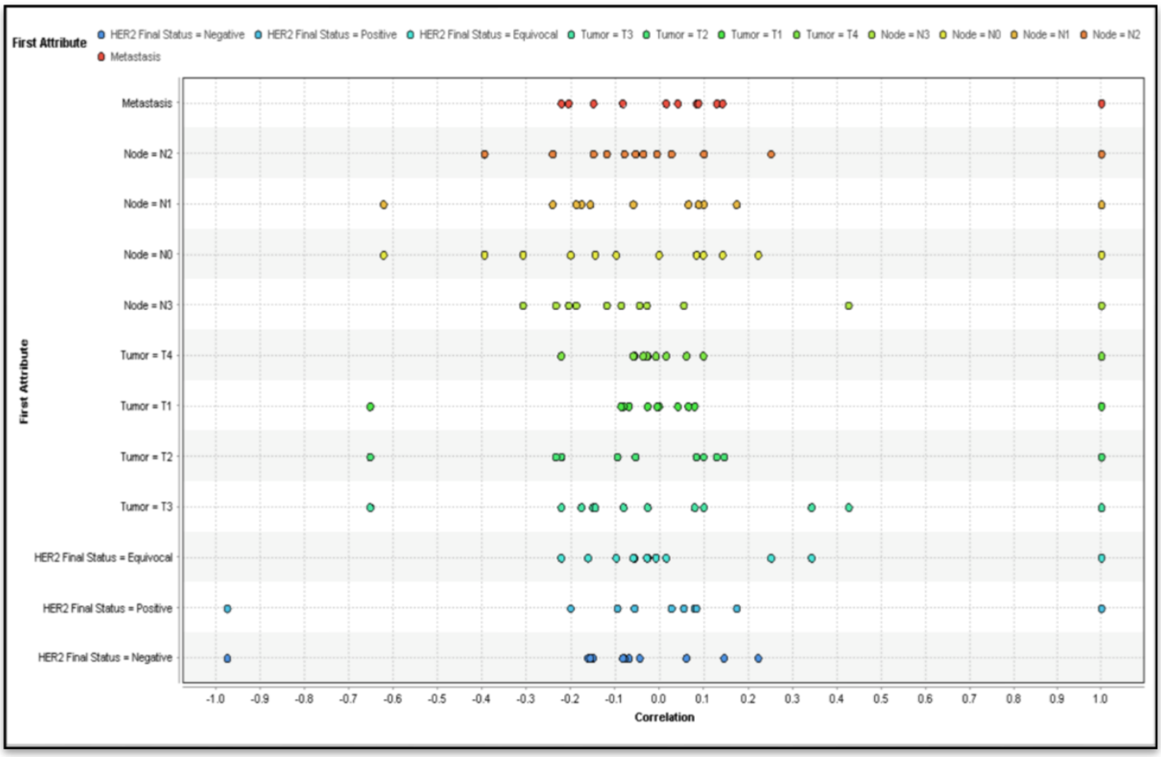


Figure 37: Scatter diagram of correlation matrix

# Conclusion

Using prediction model to classify recurrent or noncurrent cases breast cancer is a research that is statically in nature. Still this work can be linked to biomedical evidence. In this paper CANCER.csv dataset is used for finding an efficient predictor algorithm to predict the recurring or non-recurring nature of the diseases. This might help Oncologists to differentiate a good prognosis (non-recurrent) from a bad one (recurrent) and can treat the patients more effectively.

# Bibliography

[1] AMITAY, M., HONOHAN, A. M., TRAUTMAN, M., and GLEZER, A., “Use of small caps in numerical citation,” AIAA Paper 97-2004, Presented at the AIAA Shear Flow Control Conference, Snowmass, CO, 1997.

[2] BROWN, G. L. and ROSHKO, A. “The effect of names in full upper case in numerical references,” J. Fluid Mech., vol. 26, pp. 225–236, 1966.

[3] Maslen, P.E. and Gordon, M. Head., Chem. Phys. Lett. 283, 102 (1998)..

[4] Anon., “Example of a web citation”, www.thesis.gatech.edu/web_citation.html (Accessed June 30, 2005). [To remove the color and underline from the url name (required) you can either select Insert/Hyperlink and “Remove Link” or go into the Format/Style menu and modify the style named “hyperlink”.] The latter will leave the actual link intact.