Faculty of Engineering & Technology

Electrical & Computer Engineering Department

ENCS3340

Project 2 Report

Machine Learning for Classification

Prepared by :

Rama Abdlrahman - 1191344

Maymona Obaid - 1190703

Section : 2

Date : 12/6/2022

Contents

Table of figures

Table of tables

- **Abstract :**

The aim of this project is to learn how to use basic classification algorithms and compare the results by varying different parameters. The data was taken from Speaker Accent Recognition Dataset. The algorithms used are J48, Random forest, and Navi bays. The tool used is Weka.

- # Introduction

➢ About Weka

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from our own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well suited for developing new machine learning schemes. Weka supports several standard data mining tasks, more specifically, data pre-processing, clustering, classification, regression, visualization, and feature selection. All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported)[3].

➢ Machine learning

Machine learning is branch of artificial intelligent that concerns how the computer can learn and adapt new circumstances, there are many learning techniques based on the desired outcome from the technique and the input available at the training process[2].

The Most well Known Learning Techniques are:

1-Supervised learning:

When an agent tries to find the function that matches one of the example from the data set, for each input in the data set there exist an specified output class ,this type of algorithms are used for classification problems e.g. Decision Tree, Artificial neural network[2].

2-Unsupervised learning:

When an agent tries to learn from patterns without a specified output e.g. clustering[2]

- **What Is Decision Tree**

Decision Tree is the classification technique that consists of three components root node, branch (edge or link), and leaf node. Root represents the test condition for different attributes, the branch represents all possible outcomes that can be there in the test, and leaf nodes contain the label of the class to which it belongs. The root node is at the starting of the tree which is also called the top of the tree[2][3].

- **J48 Classifier**

It is an algorithm to generate a decision tree that is generated by C4.5 (an extension of ID3). It is also known as a statistical classifier. For decision tree classification, we need a database[3].

- **Random forest Classifier**

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset[3].

- **Naïve Bayes Classifier**

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object[3].

- # Discussion

➢ The tool implementation:

We opened the excel sheet, Add some details and save the file as an .arff format .

```
@relation accent

@attribute language {ES,FR,GE,IT,UK,US}
@attribute X1 real
@attribute X2 real
@attribute X3 real
@attribute X4 real
@attribute X5 real
@attribute X6 real
@attribute X7 real
@attribute X8 real
@attribute X9 real
@attribute X10 real
@attribute X11 real
@attribute X12 real

@data
ES,7.071475633,-6.512899664,7.650799805,11.1507832,-7.657312341,12.484020
ES,10.98296717,-5.15744505,3.952060221,11.52938113,-7.638046692,12.136097
ES,7.827108364,-5.477471938,7.816257284,9.187591834,-7.172510589,11.71529
ES,6.744082664,-5.688920291,6.546788659,9.000183182,-6.924963401,11.71076
ES,5.83684251,-5.32655675,7.472264712,8.847439795,-6.773244436,12.6772177
```

Figure 1 file

➢ Understanding Data

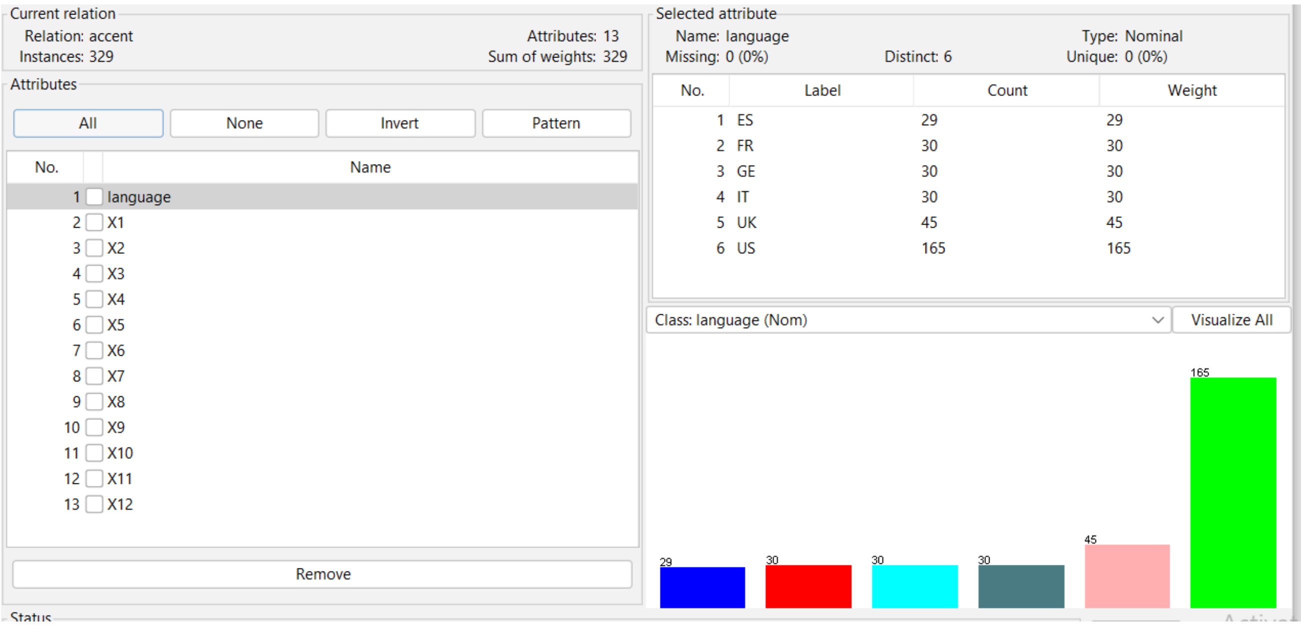


Figure 2 Understanding Data

When we opened the program, we observed that the tool provided with all of the information about the data entered by the file, there are no missing attribute values. The current relation displays the name of the data(accent), the number of attributes, and the number of examples for each attribute, in addition to the number of instances.

On the left side on figure2, notice the Attributes sub window that displays the various fields in the dataset. The Speaker Accent Recognition Dataset contains 13 fields. When you select an attribute from this list by clicking on it, further details on the attribute itself are displayed on the right hand side. When we selected the language attribute first. we would see the screen above. In the Attribute subwindow, we can see The name and the type of the attribute ,The type for the language attribute is Nominal, The number of Missing values is zero. There are six distinct values with no unique value. At the bottom of the window, you see the visual representation of the class values.

- **Visualizing all attributes before filtering :**

If you click on the Visualize All button, you will be able to see all features in one single window as shown here:

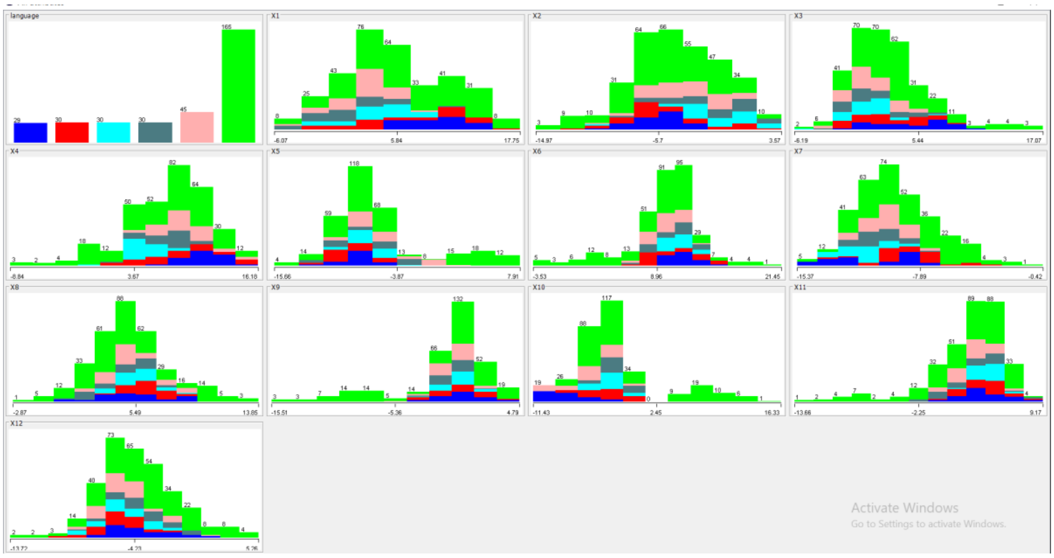


Figure 3 Visualizing all attributes before filtering

➢ Applying Filters(pre-processing)

I used Discretization filter with 6 bins to convert attribute(X3) to nominal instead of real:

weka→filters→unsupervised→attribute→Discretize

Discretizing your real valued attributes is most useful when working with decision tree type algorithms. It is perhaps more useful when you believe that there are natural groupings within the values of given attributes.

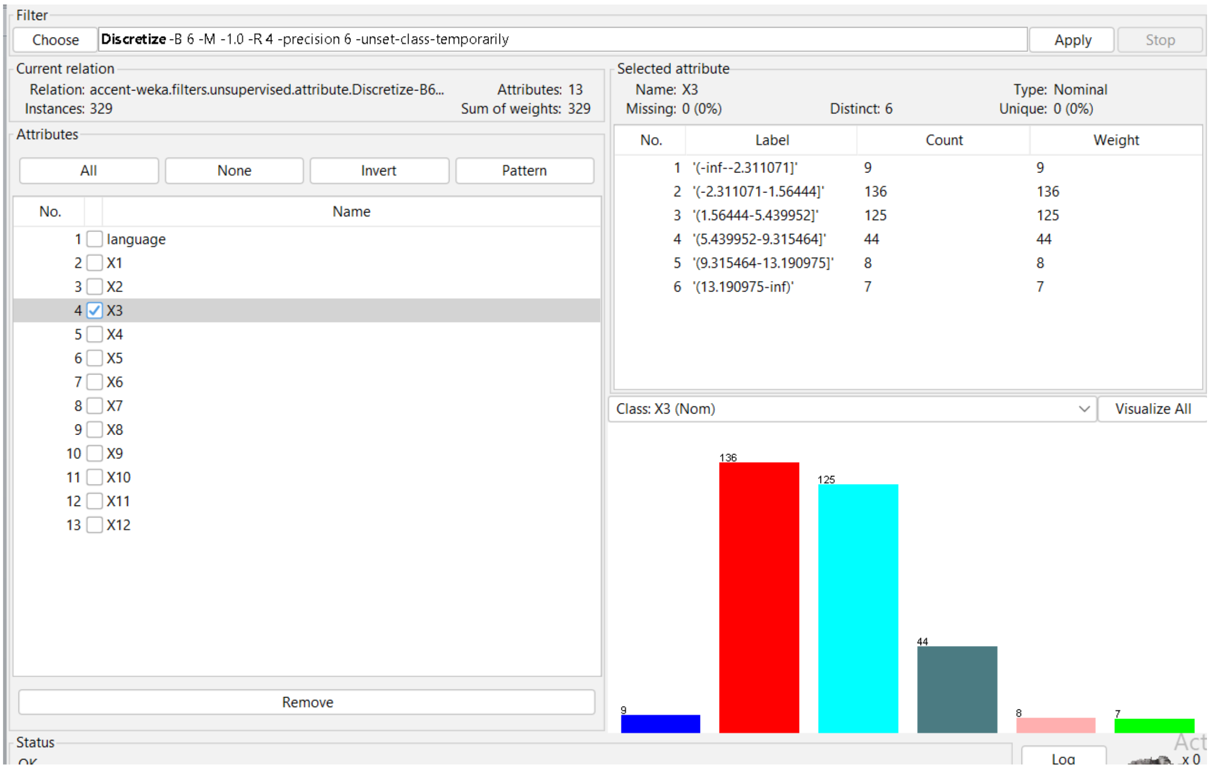


Figure 4 Applying Filters(pre-processing)

➢ Setting Test Data

We used the pre-processed dataset(pre-processed attribute X3) by choosing the Classify tab and Test the three models using 5-fold cross validation.

✓ Decision Tree (J48)

It's the most popular tool for classification. Its like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label. decision tree is known as J48 algorithm . testing was performed using 5 fold cross validation .

weka→classifiers>trees>J48

Attribute chosen : X3

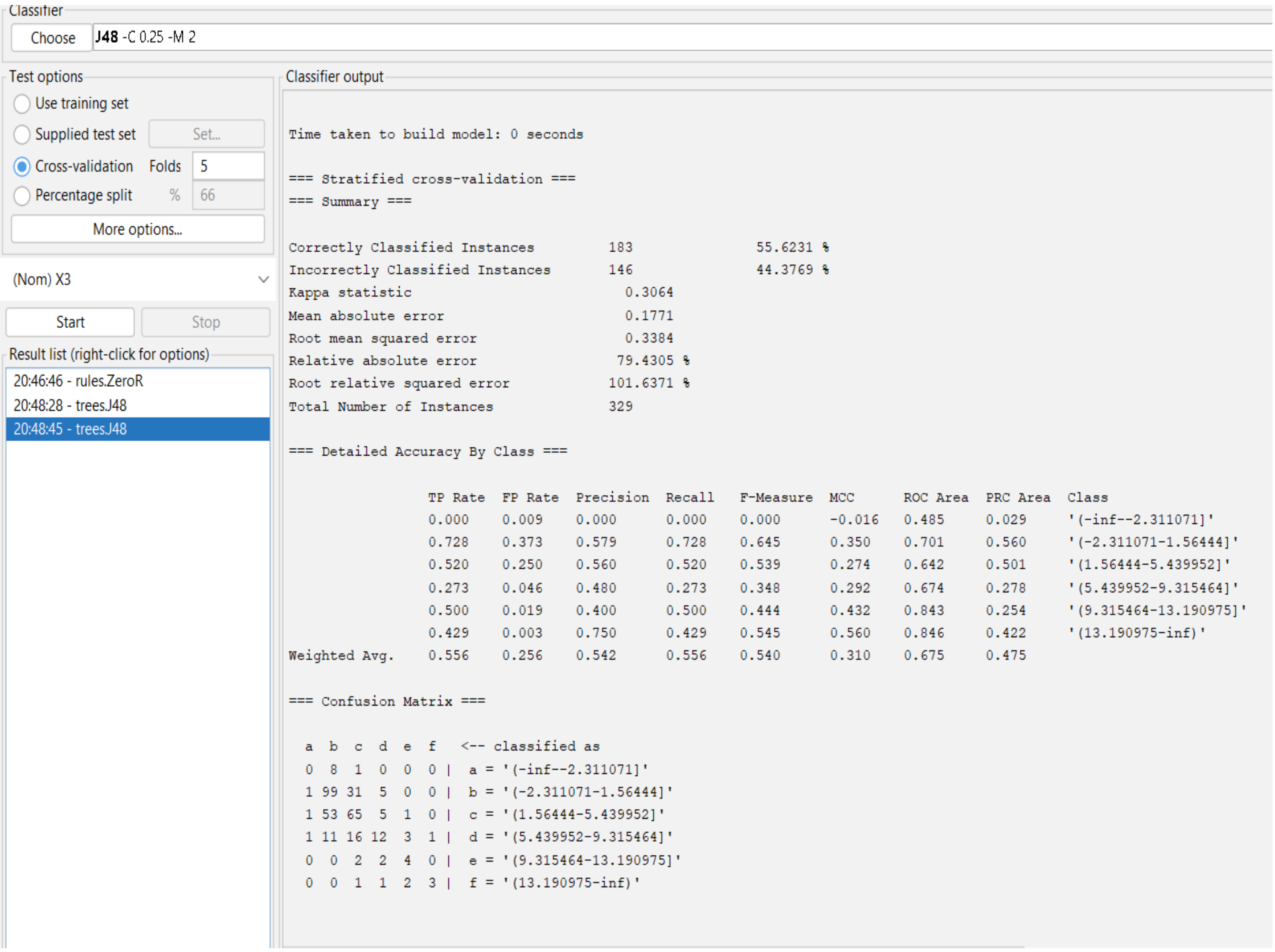


Figure 5 J48 DECISION TREE matrix and detailed tables

This screen says the size of the tree is 137. It says that the correctly classified instances 183(55.6231 %) and the incorrectly classified instances 146(44.3769 %), It also says that the Relative absolute error is 79.4305 %. It also shows the Confusion Matrix and detailes.

- ***Analysis*:**

Many measures appear in the screen. For Example, Recall, Precision and F-Score.

Precision (P) = TP / (TP + FP)

Recall (R) = TP / (TP + FN)

F-Score = 2PR / (P + R)

More explanation:

- True Positives (TP): These are cases in which we predicted yes, and the word is present.
- True Negatives (TN): We predicted no, and the word is not present.
- False Positives (FP): We predicted yes, but the word don't actually present. (Also known as a "Type I error.")
- False Negatives (FN): We predicted no, but the word actually do present. (Also known as a "Type II error.")
- Accuracy: Overall, how often classifier is correct (TP+TN)/total
- True Positive Rate: When it is actually yes, how often does it predict yes TP/actual yes also known as "Sensitivity" or "Recall"
- False Positive Rate: When it is actually no, how often does it predict yes FP/actual no
- Precision: When it predicts yes, how often is it correct TP/predicted yes
- F-measure: 2. $(Precision.\ recall)/(Precision + recall)$

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.009    0.000      0.000    0.000      -0.016   0.485     0.029     '(-inf--2.311071]'
                 0.728    0.373    0.579      0.728    0.645      0.350    0.701     0.560     '(-2.311071-1.56444]'
                 0.520    0.250    0.560      0.520    0.539      0.274    0.642     0.501     '(1.56444-5.439952]'
                 0.273    0.046    0.480      0.273    0.348      0.292    0.674     0.278     '(5.439952-9.315464]'
                 0.500    0.019    0.400      0.500    0.444      0.432    0.843     0.254     '(9.315464-13.190975]'
                 0.429    0.003    0.750      0.429    0.545      0.560    0.846     0.422     '(13.190975-inf)'
Weighted Avg.    0.556    0.256    0.542      0.556    0.540      0.310    0.675     0.475

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
  0  8  1  0  0  0 |  a = '(-inf--2.311071]'
  1 99 31  5  0  0 |  b = '(-2.311071-1.56444]'
  1 53 65  5  1  0 |  c = '(1.56444-5.439952]'
  1 11 16 12  3  1 |  d = '(5.439952-9.315464]'
  0  0  2  2  4  0 |  e = '(9.315464-13.190975]'
  0  0  1  1  2  3 |  f = '(13.190975-inf)'
```

Figure 6 the measures for X3 attributes

The above screen shows the measures for X3 attribute, we can calculate them from matrix:

For example:

TP rate for class a = 0/(1+0+8) =0 which is shown on the output screen above.

TP rate for class b = 99/(1+31+5)=0.728

TP rate for class c = 0.520

TP rate for class d = 0.273

TP rate for class e =0.500

TP rate for class f = 0.429

And all other calculations in figure.6

	TP	**FP**	Prec.	Recall	F-Measure	MCC	**ROC**	**PRC**
Weighted Avg.	**0.556**	**0.256**	**0.542**	**0.556**	**0.540**	**0.310**	**0.675**	**0.475**

Table 1 The average values of measures of J48

The average values will be used later when comparing j48 with other models.

- changing hyper parameter (Binary split --- > yes) – for X3 class attribute -

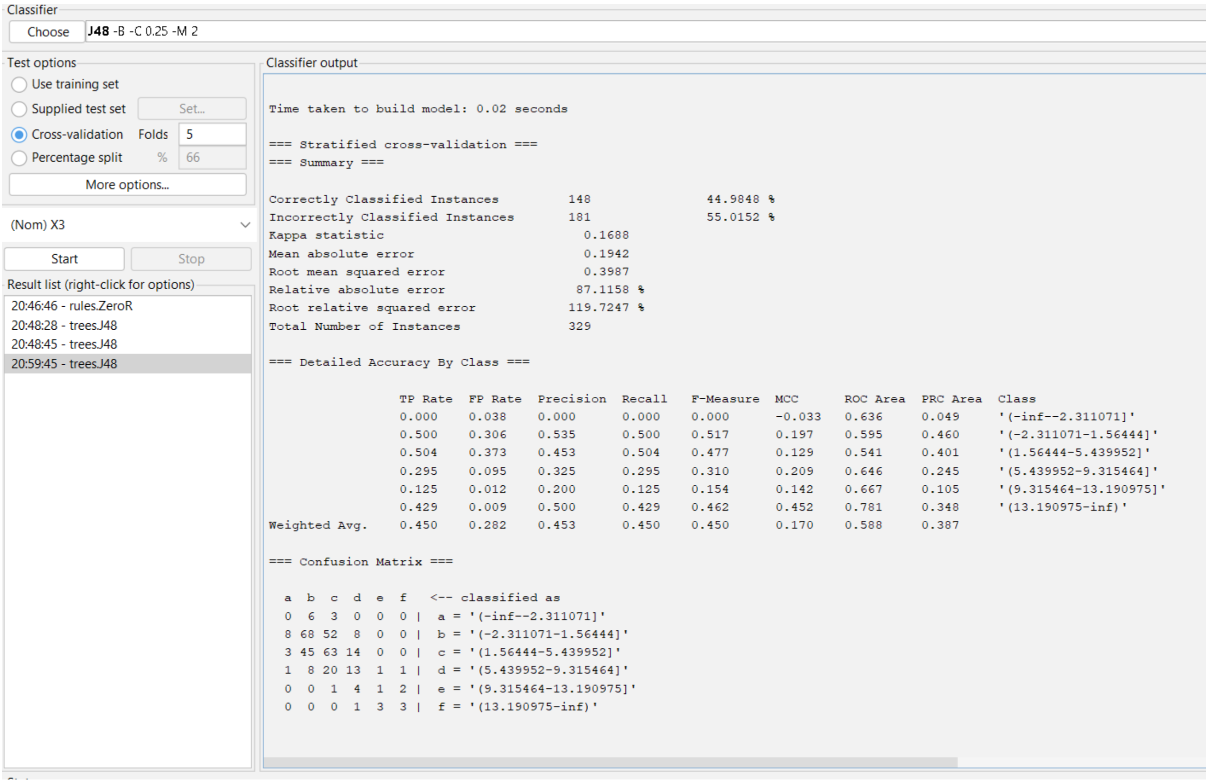


Figure 7 changing hybre parameter Binary split

By Comparing accuracy before changing Binary split parameter and after changing, we notice that before changing we had 183 Correct Classified Instances (55.6231%) & 146 Incorrectly Classified Instances (44.3769 %) and after changing it becomes 148 Correctly Classified Instances (44.9848 %) & 181 Incorrectly Classified Instances (55.0152 %). That's mean making Binary split yes caused decreasing accuracy from 55.6231% to 44.9848 %.

J48	Correct instances	Incorrect instances
before	(55.6231%)	(44.3769 %)
after	(44.9848 %)	(55.0152 %)

Table 2 accuracy before and after (J48)

Time taken to creat model using J48 is 0.07.

Also the changing affect the avg. values for TP,FP and other measures, we can notice the effect from this table :

TP	FP	Precision	Recall	F-Measure	MCC	ROC	PRC	
0.556	**0.256**	**0.542**	**0.556**	**0.540**	**0.310**	**0.675**	**0.475**	**before change**
0.450	**0.282**	**0.453**	**0.450**	**0.450**	**0.170**	**0.588**	**0.387**	**after change**

Table 3 the avg. values for TP,FP and other measures before and after

The table shows that changing binary split to yes decrease TP value, Precision , Recall , F-Measure , MCC , ROC and PRC, and increase FP.

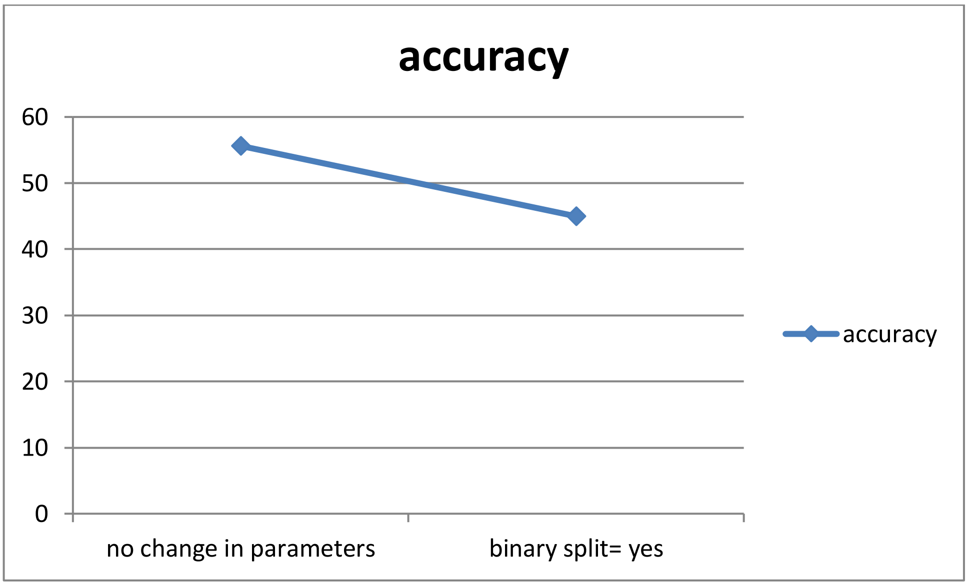


Figure 8 accuracy after changing parameter

- **Also by changing other parameter (Confidence factor):**

We got the following result (by increasing confidence factor, the accuracy decreases).

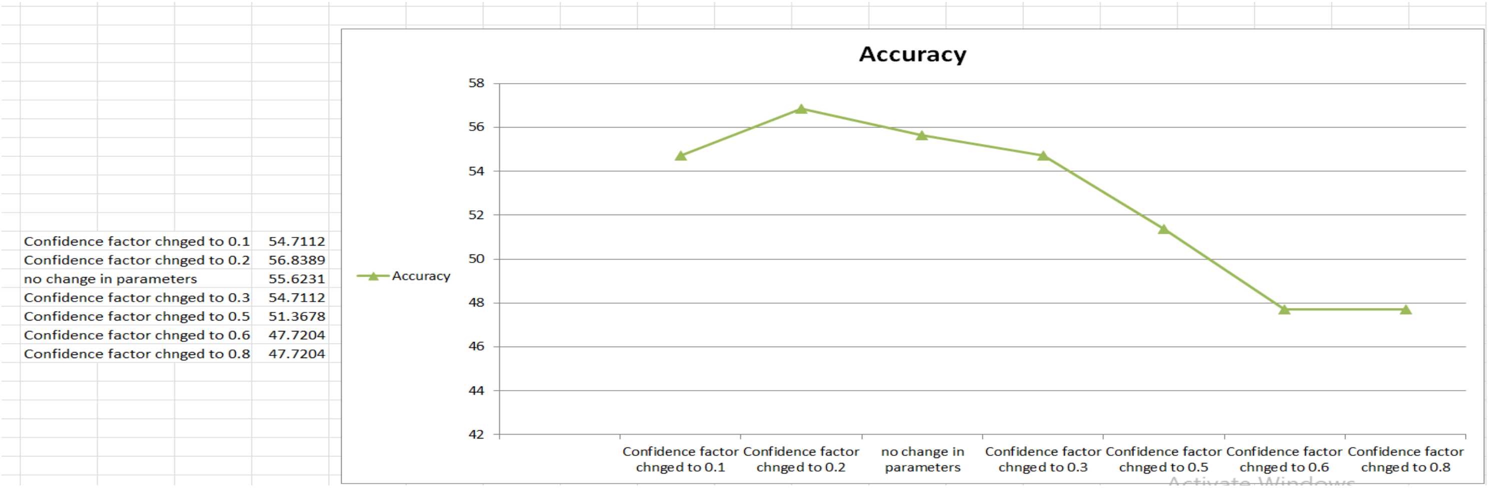


Figure 9 changing other parameter (Confidence factor)

✓ Second model (Naïve Bayes)

The result after applying the NaiveBayes algorithm like the percentage of the correctly classified instance and incorrectly classified instance shown in the screen bellow.

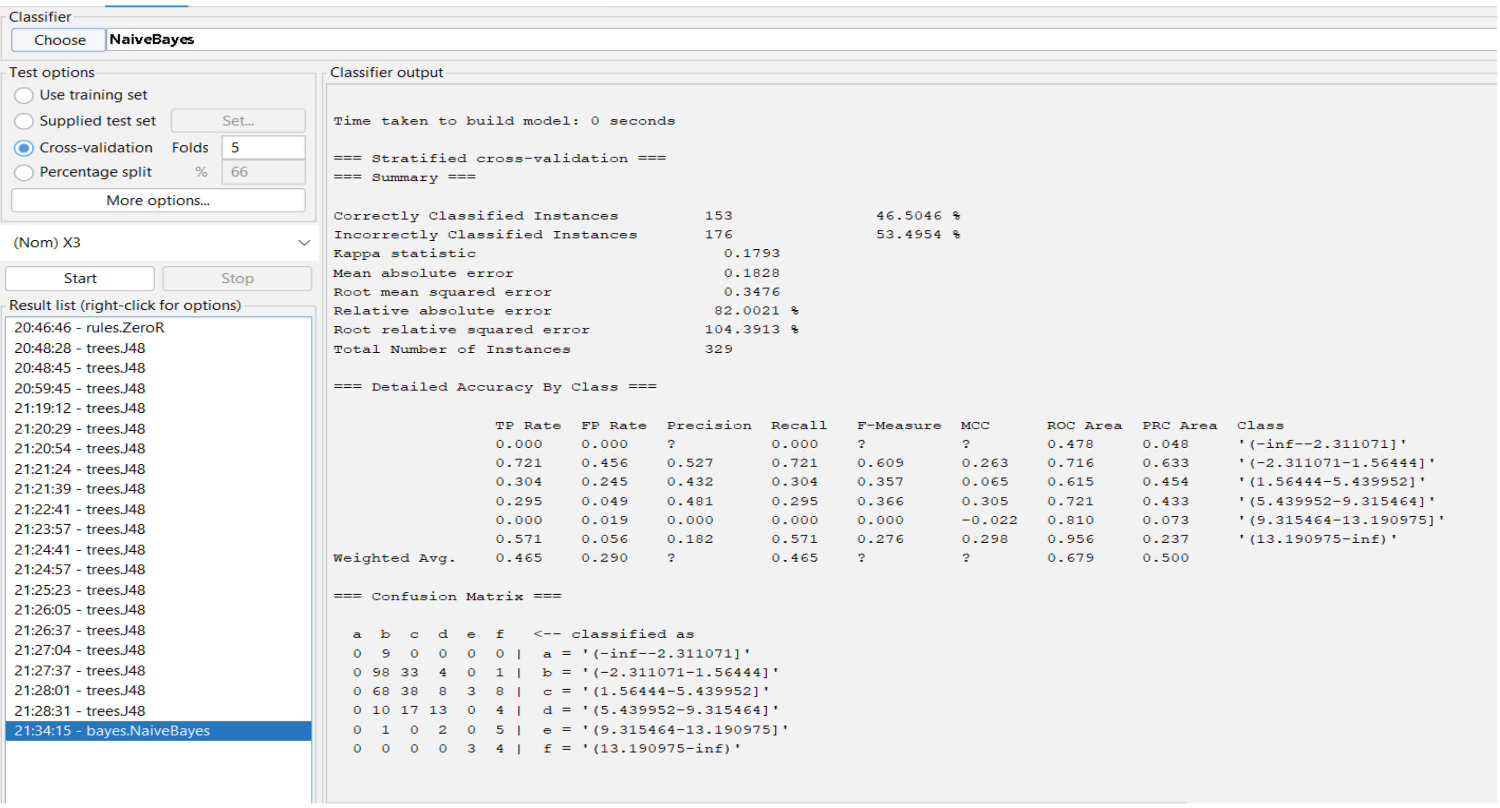


Figure 10 naïve bayes matrix and detailed tables

This screen says that the correctly classified instances 153(46.5046 %) and the incorrectly classified instances 176(53.4954 %), It also says that the Relative absolute error is 82.0021 %. It also shows the Confusion Matrix and details.

- **By changing Don’t check capabilities parameter to true**

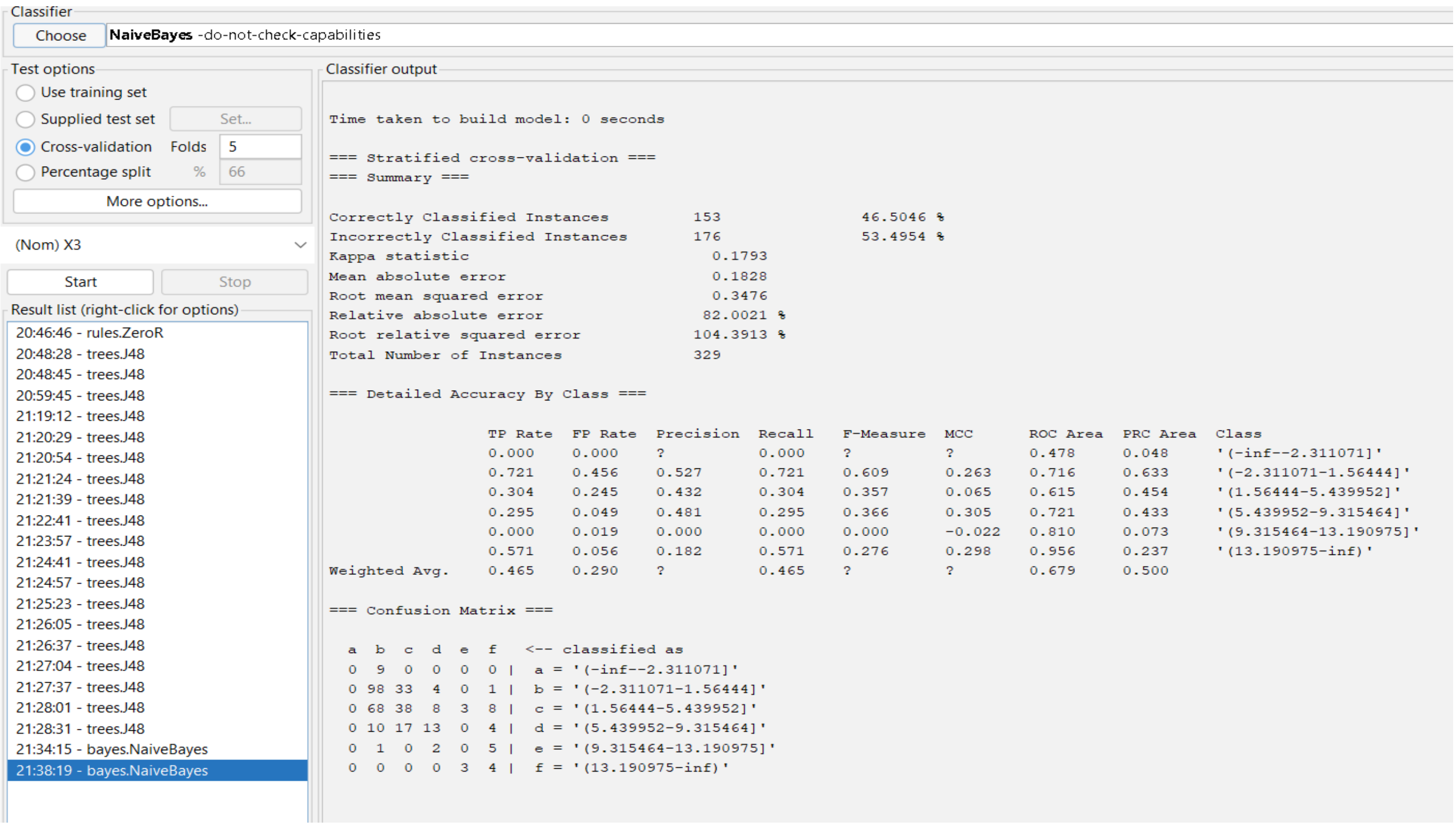


Figure 11 changing Don’t check capabilities parameter

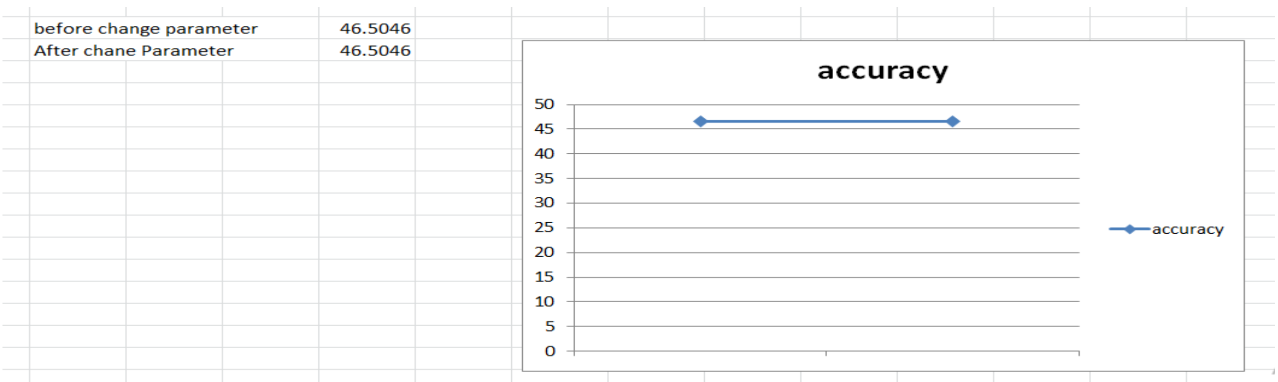


Figure 12 Accuracy after changing

Naïve Bayes	**Correct instances**	**Incorrect instances**
before	**(**46.5046**%)**	**(**53.4954**%)**
after	**(**46.5046**%)**	**(** 53.4954 **%)**

Table 4 accuracy of Naïve Bayes before and after changing parameter

Time taken to creat model using Naïve Bayes is 0.01.

And by trying changing other parameters, It concluded also that correctly properly was not changed after altering any hyper parameter of the naïve bayes, also precision, recall, and F1-score not affected .

We can conclude that decision tree j48 produced higher accuracy than naïve bayes , but naïve bayes create the model in less time than j48, which is a good advantage.

- ✓ **Final model (Random forest)**

The result after applying the Random Forest algorithm like the percentage of the correctly classified instance and incorrectly classified instance shown in screen bellow.

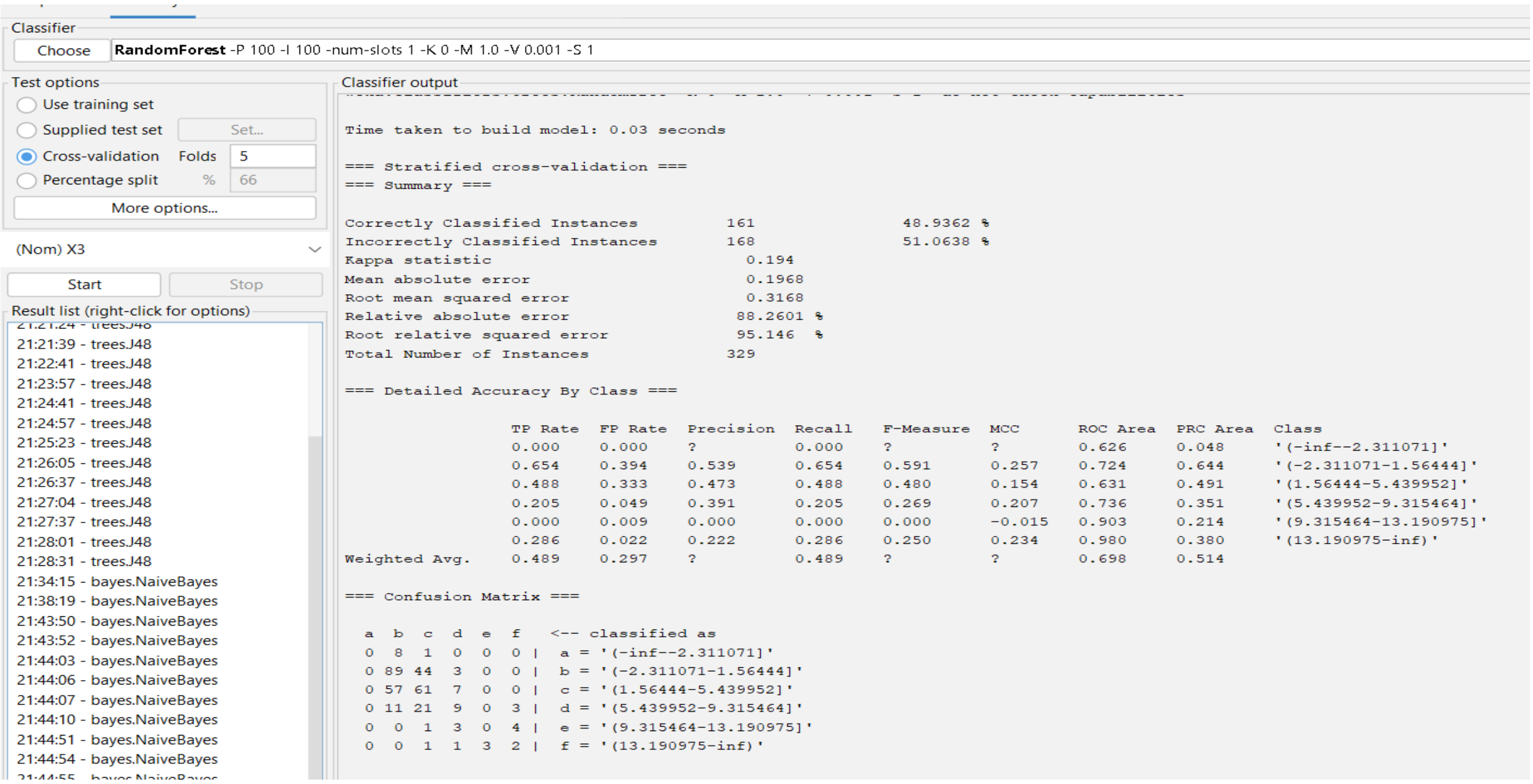


Figure 13 Random Forest matrix and detailed tables

This screen says that the correctly classified instances 161(48.9362 %) and the incorrectly classified instances 160(51.0630 %), It also says that the Relative absolute error is 88.2601 %. It also shows the Confusion Matrix and details.

- **By changing Don't check capabilities parameter to true**

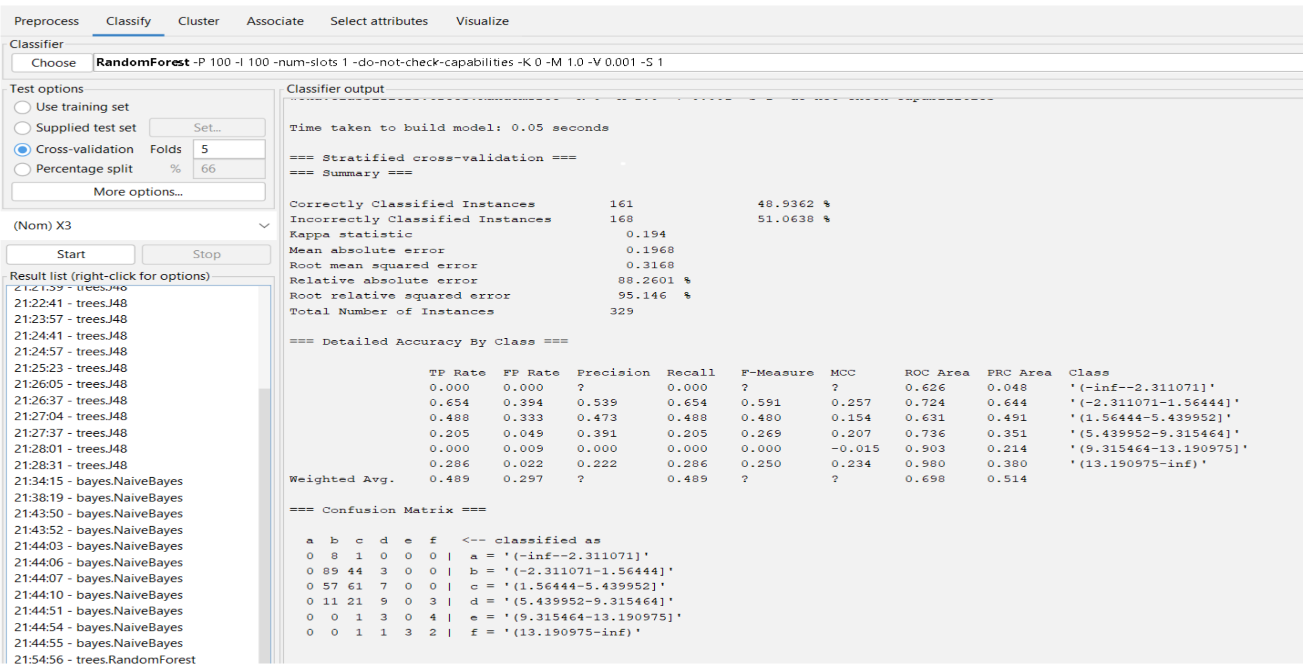


Figure 14 By changing Don't check capabilities parameter to true

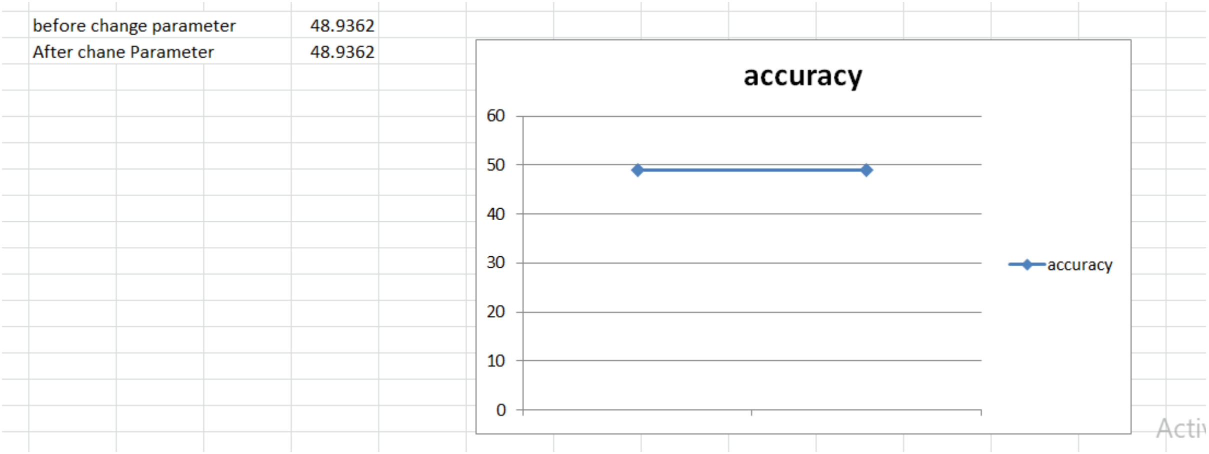


Figure 15 accuracy before and after changing

Randomforest	**Correct instances**	**Incorrect instances**
before	**(**48.9362 **%)**	**(**51.0630 **%)**
after	**(**48.9362 **%)**	**(** 51.0630 **%)**

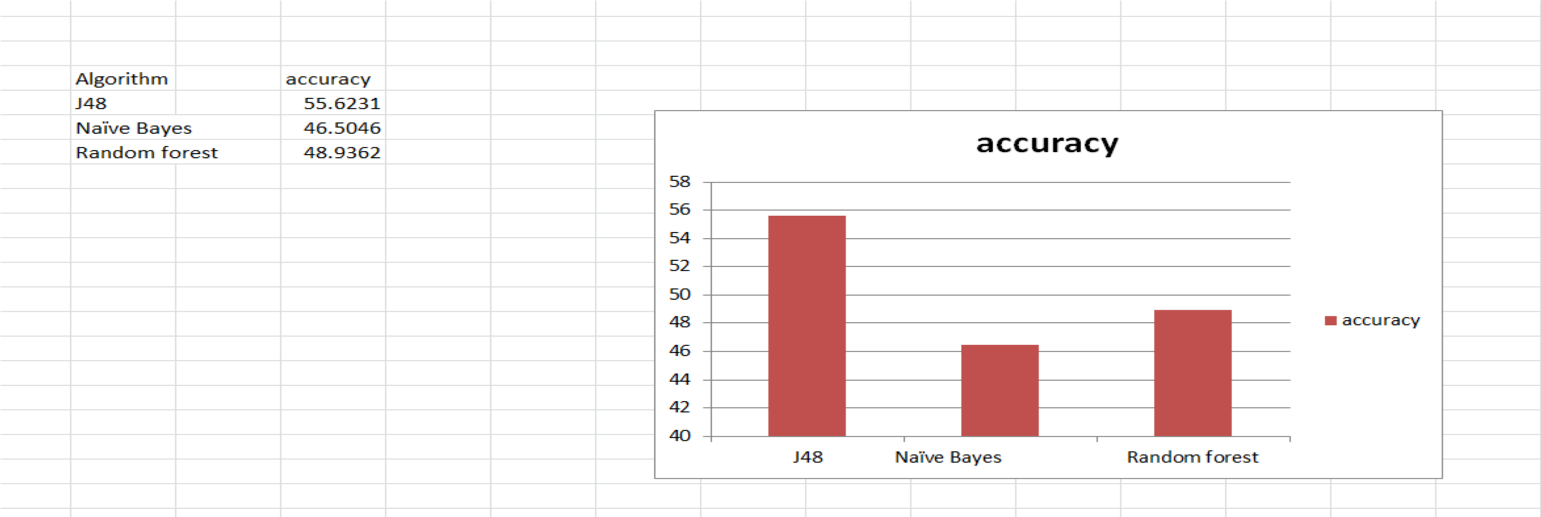
Table 5 accuracy before and after changing

Time taken to creat model using Random forest is 0.11.

And by trying changing other parameters, It concluded also that correctly properly was not changed after altering any hyper parameter of the Random forest, also precision, recall, and F1-score not affected .

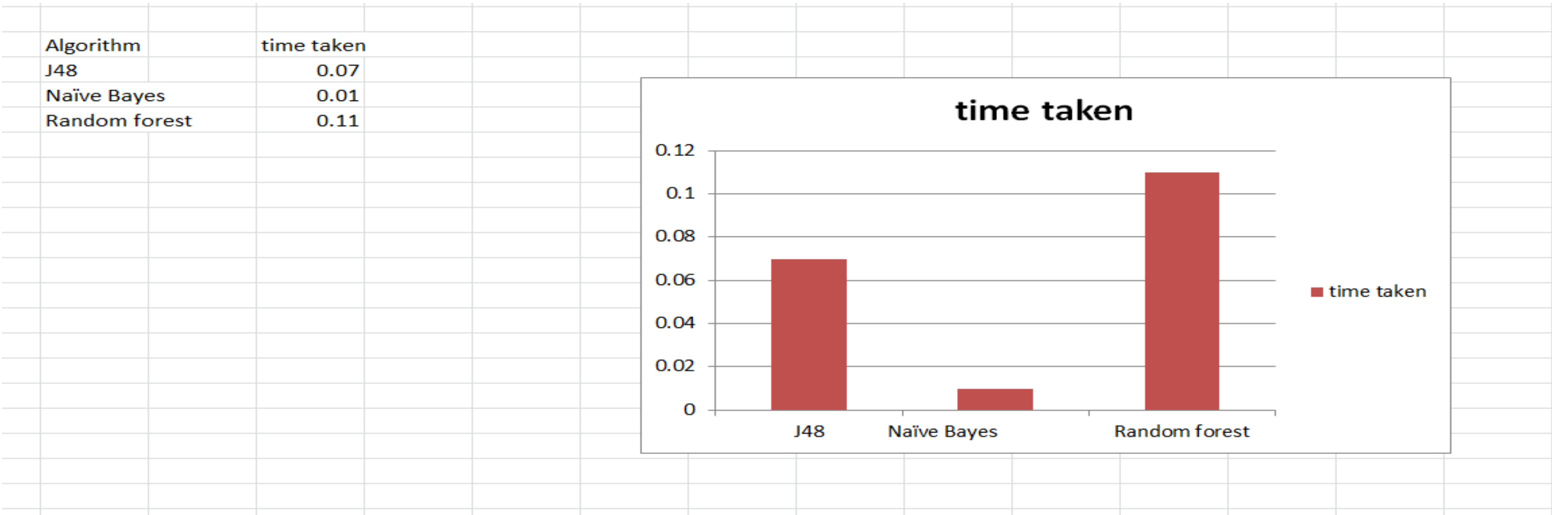
We can see that it taking time to build the Random forest model, the time is almost the same with the decision tree algorithm(j48) time .

✓ Discusion for all 3 models :



Algorithm	accuracy
J48	55.6231
Naïve Bayes	46.5046
Random forest	48.9362

Figure 16 The accuracy of the classifier models on 5-fold cross validation



Algorithm	time taken
J48	0.07
Naïve Bayes	0.01
Random forest	0.11

Figure 17 The time taken from the 3 classifier models

From the results we we got , we conclude that decision tree(J48) is the best algorithm since it has the highest correctly instance percentage(highest accuracy) ,Precision , Recall and F-measur which are the standard measures used to compare between one algorithm and another . all remaining algorithms almost have close results . Also , we concluded that Naïve Bayes is the fastest algorithm regarding to the other algorithms. it would be most efficient when the data size is small, but in the same way when we took larger data the accuracy decreased, and some other algorithms got better accuracy results like j48 and random forest.

- ## Conclusion :

In this project, Three algorithms were tested on the dataset. Which are Decision tree (J48), Naïve Bayes and random forest .And from the result of testing these algorithms, we concluded that decision tree(J48) is the best algorithm since it has the highest correctly instance percentage(highest accuracy) ,Precision , Recall and F-measure which are the standard measures used to compare between one algorithm and another . all remaining algorithms almost have close results . Also , we concluded that Naïve Bayes is the fastest algorithm regarding to the other algorithms. it would be most efficient when the data size is small, but in the same way when we took larger data the accuracy decreased, and some other algorithms got better accuracy results like j48 and random forest.

- **References**

[1] performance-metrics-confusion-matrix-precision-recall-and-f1-score , Accessed on 12 June 2022

[2] transform-machine-learning-data-weka , Accessed on 12 June 2022

[3] weka_classifiers , Accessed on 12 June 2022