INTRODUCTION

Artificial Intelligence is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristics of humans such as the ability to reason, discover meaning, generalize or learn from the past. From the Oxford dictionary, Intelligence is defined as the ability to acquire and apply knowledge and skills. Psychologists generally do not characterize human intelligence by just one trait but by the combination of many diverse ability. Research in AI has focused chiefly on the following components of intelligence: learning, reasoning, problem solving, perception and using language. These are the definitive components that have helped shape the face of modern artificial intelligence. The aim of this project is to challenge ourselves to some problems in the fields of machine learning, natural language processing, knowledge representation, optimization and game theory.

PART 1 – MACHINE LEARNING USING WEKA

OBJECTIVES

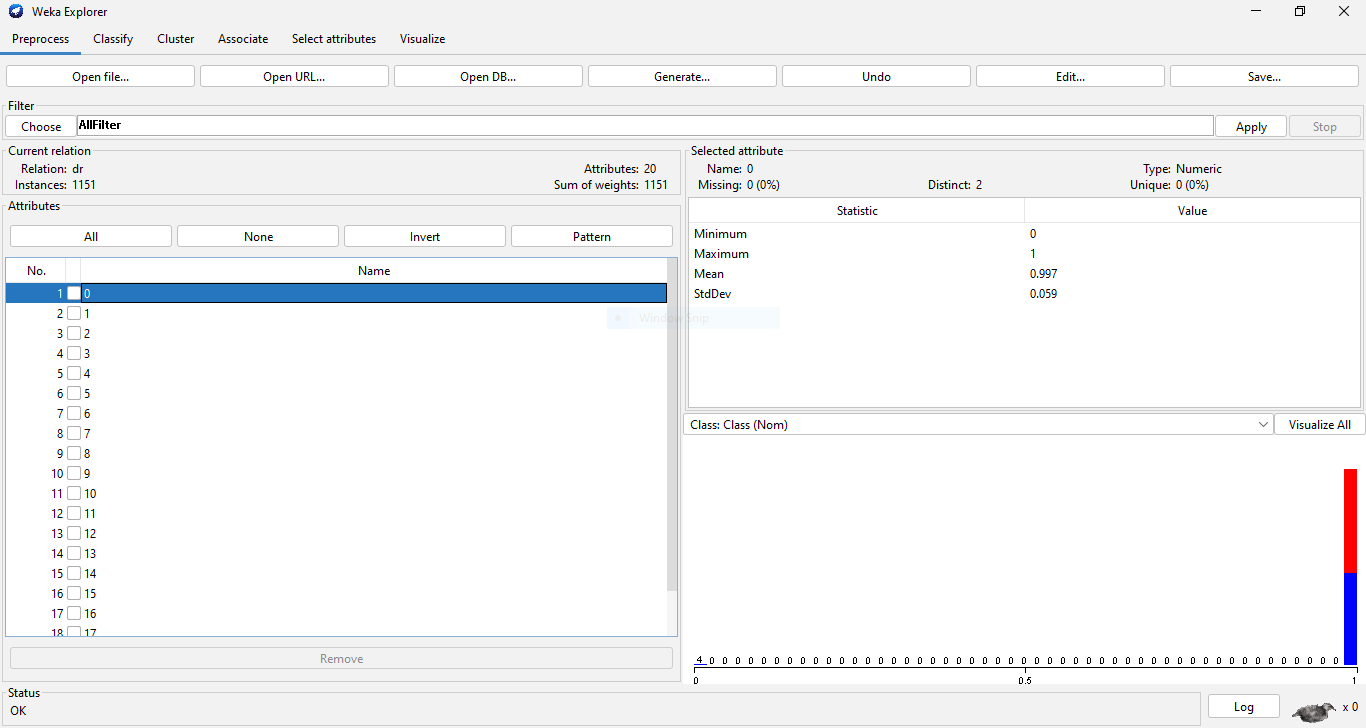
The aim of this is to have some hands-on experience with machine learning algorithms using WEKA for solving real world data-mining problems.

PROBLEM FORMULATION

To evaluate and analyze the performance of at least three classification schemes on a Diabetic Retinopathy Debrecen data set. The data set was obtained from the open-source UCI Machine Learning Repository. The data set contains features extracted from the Messidor image set to predict whether an image contains signs of diabetic retinopathy or not.

SOLUTION APPROACH AND ALGORITHMS

The Diabetic Retinopathy Debrecen data set has 1151 instances and 20 attributes as shown in figure 1 below.



For a data set with over 1000 instances, 2/3 of the data set was used for training and 1/3 was used for testing. The classification schemes applied to the data set were J48 decision tree classification algorithm, Naive Bayes classification algorithm and KNN algorithm. A brief definition of the chosen algorithms and reasons they were chosen is given below.

J48 TREE CLASSIFICATION ALGORITHM

The J48 classification algorithm is one of the best machine learning algorithms to examine the data categorically and continuously. When it is used for instance purposes, it occupies more memory space and depletes the performance and accuracy in classifying medical data.

NAÏVE BAYES CLASSIFICATION ALGORITHM

Naïve Bayes classification calculates explicit probabilities for superposition among the most practical approaches to definite types of learning problems. Even when Bayesian techniques are computationally difficult, they can produce standard and optimal decision making against all other methods of classification.

KNN CLASSIFICATION ALGORITHM

K-nearest neighbor algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. To classify a new instance it searches the training set for one that is “most like” it. KNN algorithm decides a number k which is the nearest neighbor to the data point that is to be classified. If the value of k is 5 it will look for 5 nearest neighbors to that data point. It is easy to implement and understand but has a major drawback of becoming significantly slow as the size of data grows.

RESULTS AND DISCUSSION

Although having a high percentage of correctly classified instances is good, that alone cannot accurately determine if the model used is good. You can have high classification accuracy and a bad model. For instance, if you have 90% of instances all in one class you can say all of them belong to that class. And you're going to be right 90% of the time but the model is not good.

This section discusses and analyzes the statistics of the output of the models of the different algorithms that were applied and the changes in the results as the parameters were varied. In order to do that there are certain concepts we need to understand.

Kappa statistic takes into account the fact that you could randomly guess and correctly classify an instance. It is a statistic of how well you would perform if you take into consideration the fact that you can randomly guess the class of an instance.

Kappa = (observed accuracy - expected accuracy)/(1 - expected accuracy)

Weighted average gives a general average of how well the algorithm is doing.

True positive is the percentage of classified instances that actually belong to a particular class.

False positive is the percentage of classified instances that do not actually belong to a particular class.

Precision is the ratio of true positives to total data classified as a particular class.

Recall is ratio of true positives to total data that actually belongs to a particular class, say class 0

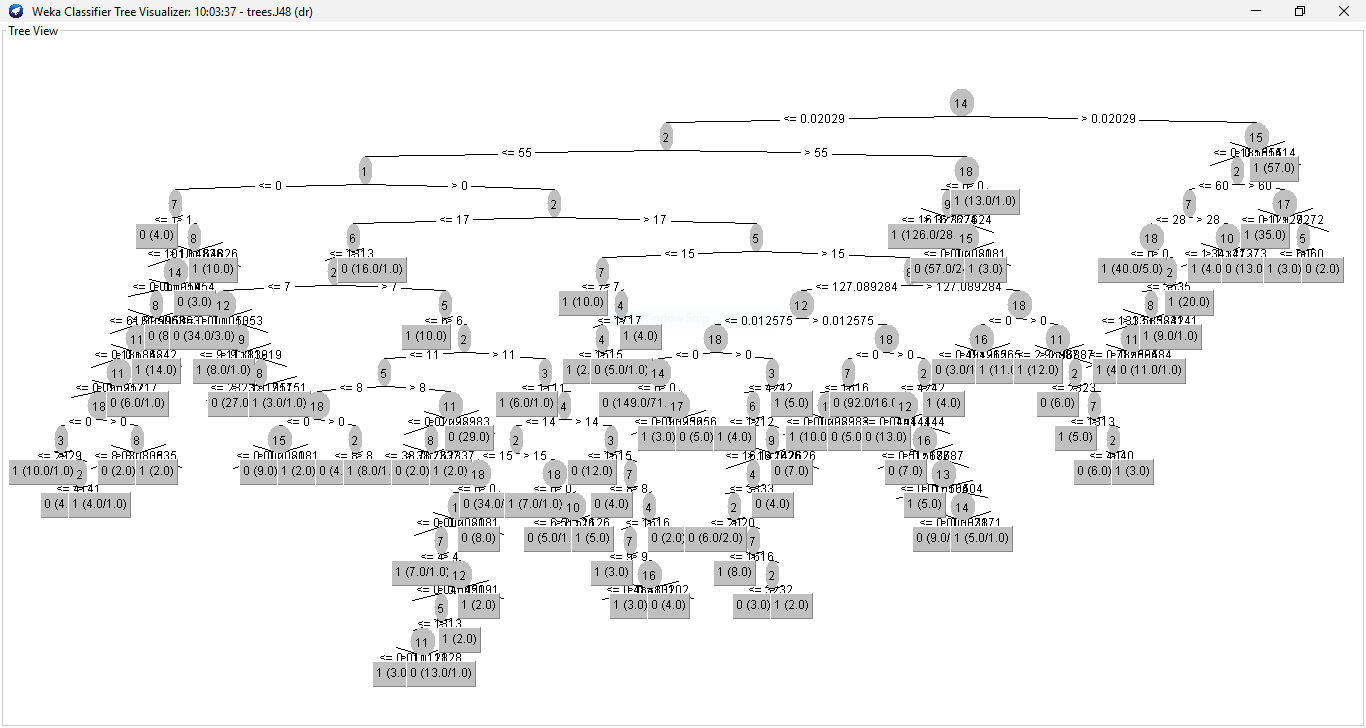
ROC area is the receiver operator characteristic area under the curve. It shows what percentage of the time you are going to correctly classify items if given one of each class.

PRC area is the precision and recall area and it does not account for true positives in the statistics.

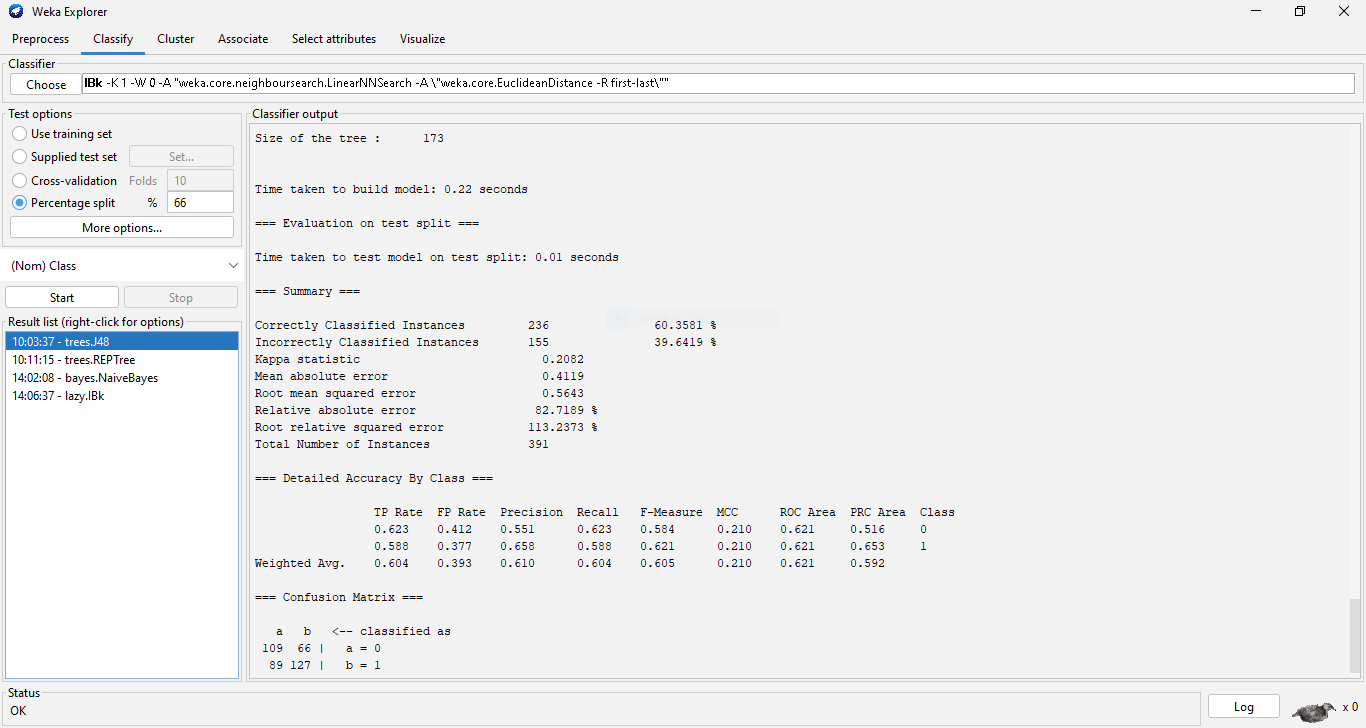
Confusion matrix shows values that are correctly classified for each class and those that are incorrectly classified.

J48 TREE CLASSIFICATION ALGORITHM

The visualized tree for the J48 algorithm on the dataset is given below:



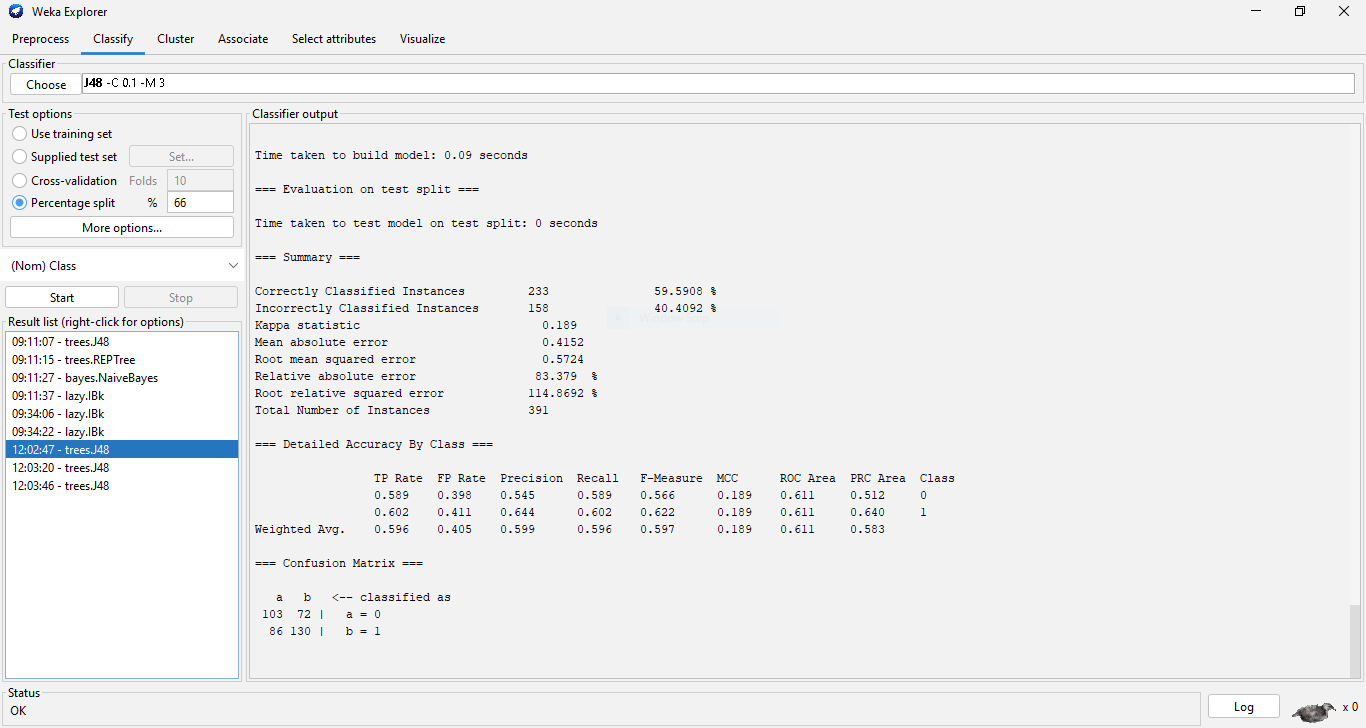
The output for the classifier model is given below:



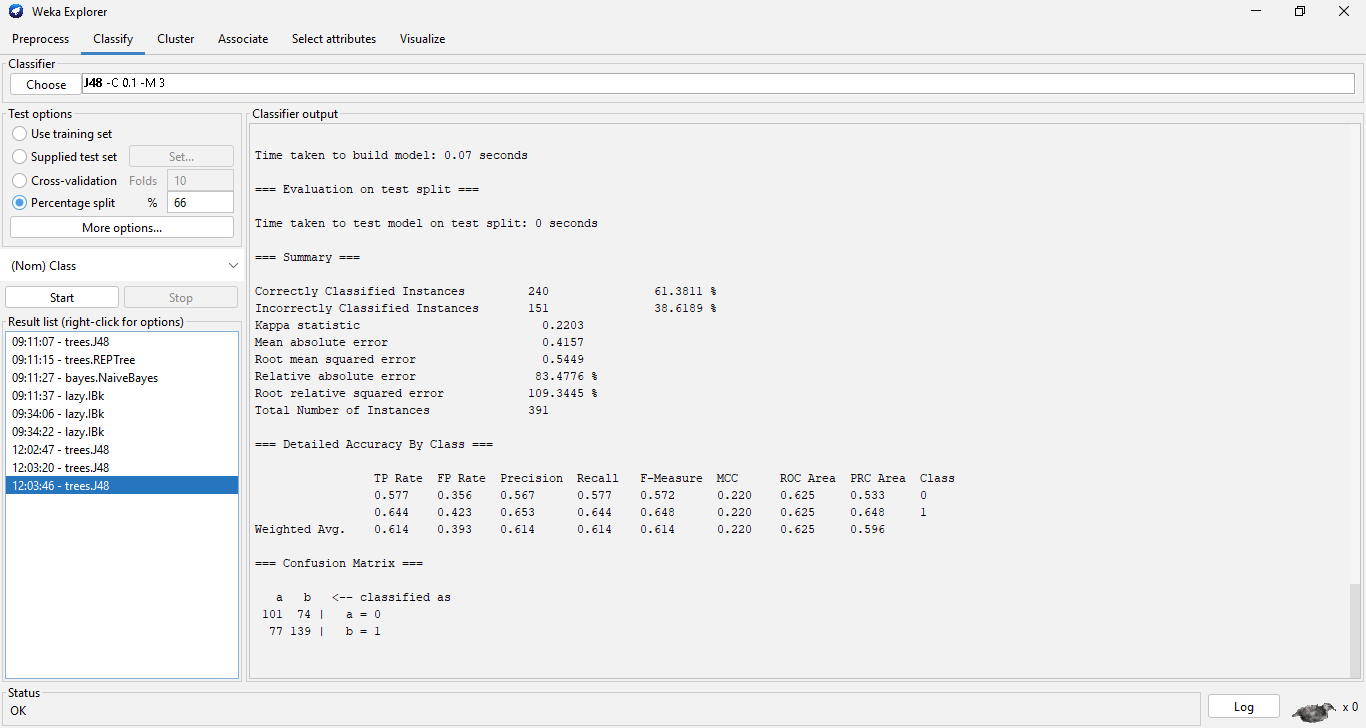
The time taken to build the model was 0.27s. The time taken to test the model on test split was 0.8s.

The tree size is 173 and the number of leaves is 87. The correctly classified instances are 236 at a 60.3581% accuracy and the incorrectly classified instances are 155 at a 39.6419%. It has a kappa statistic of 0.2082 which is poor. It has a weighted true positive rate of 0.604 and a false positive rate of 0.393. It has a weighted precision rate of 0.610 and a recall rate of 0.604. It has an F-measure of 0.605 and an ROC area of 0.621. For the confusion matrix, 109 instances of class a were correctly classified and 66 instances of class a were incorrectly classified as class b = 1. 89 instances of class b were incorrectly classified as class a and 127 instances of class b were correctly classified.

The model performs poorly. The confidenceFactor and the minNumObj parameters were varied to test how they affect the results of the model. For the first test the confidenceFactor was changed from 0.25 to 0.3 and the minNumObj was maintained at 2. The results were given as below:



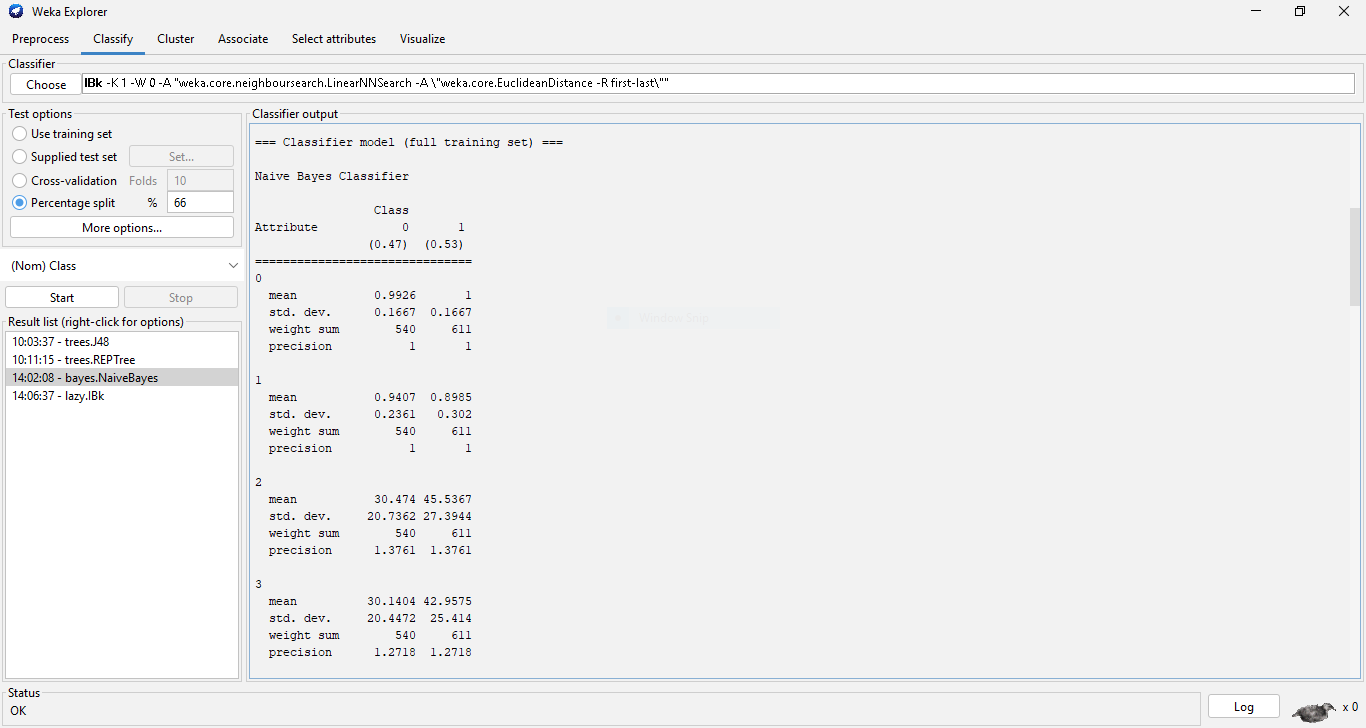
It can be observed that the overall accuracy of the model worsens. For the second test, the confidence factor was changed to 0.1 and the minNumObj was changed to 3 and the results are shown below:

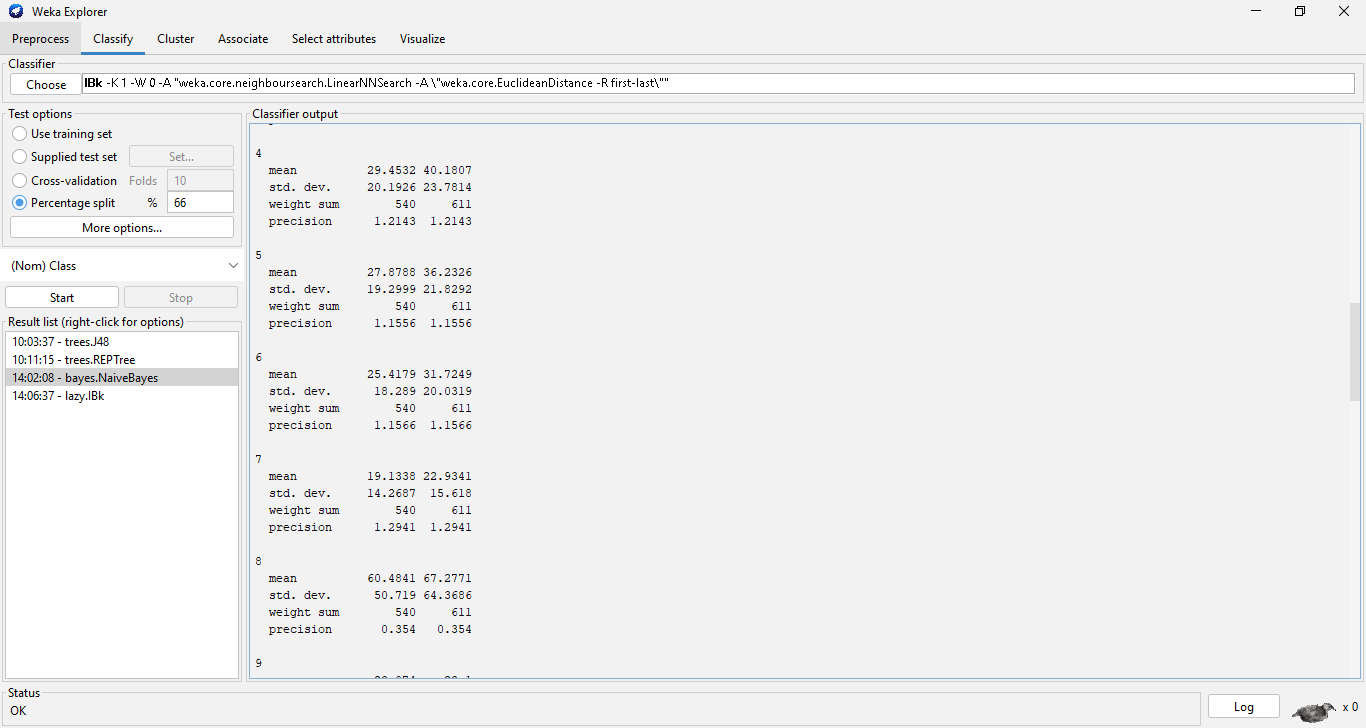


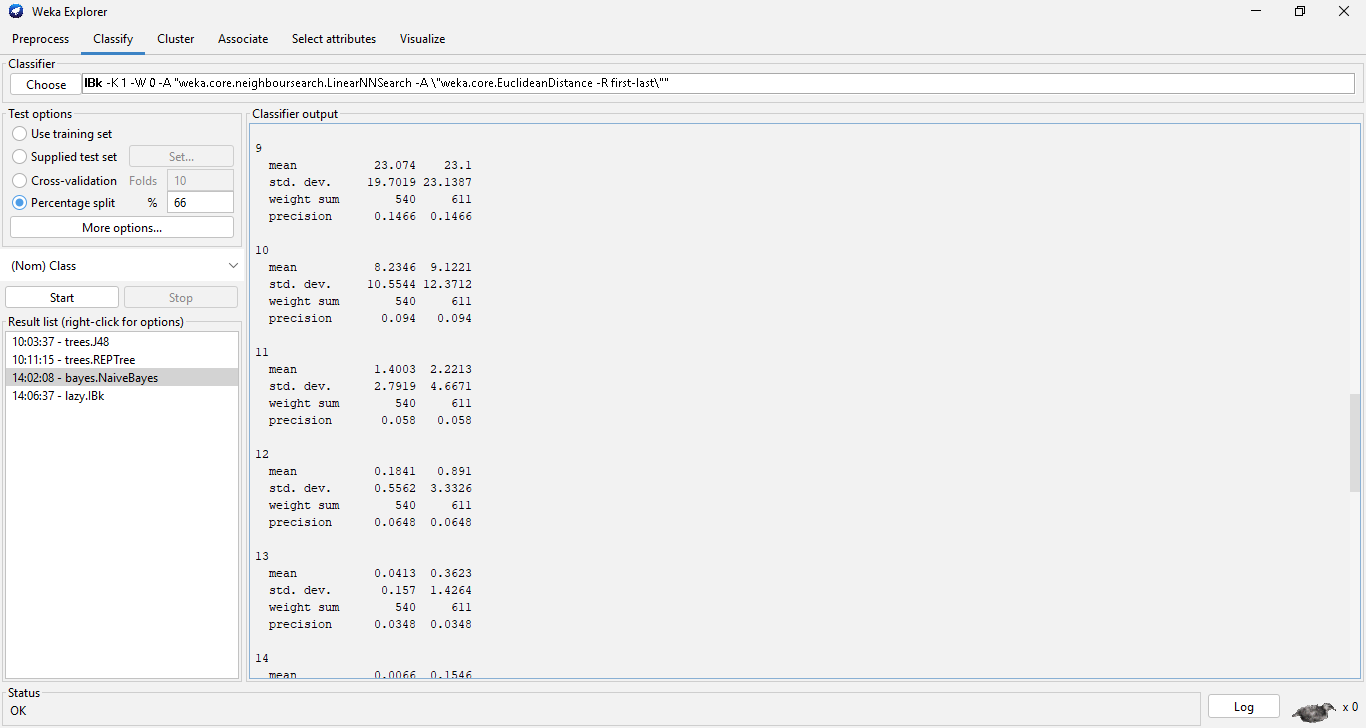
It can be observed that the model’s performance improves but still performs poorly.

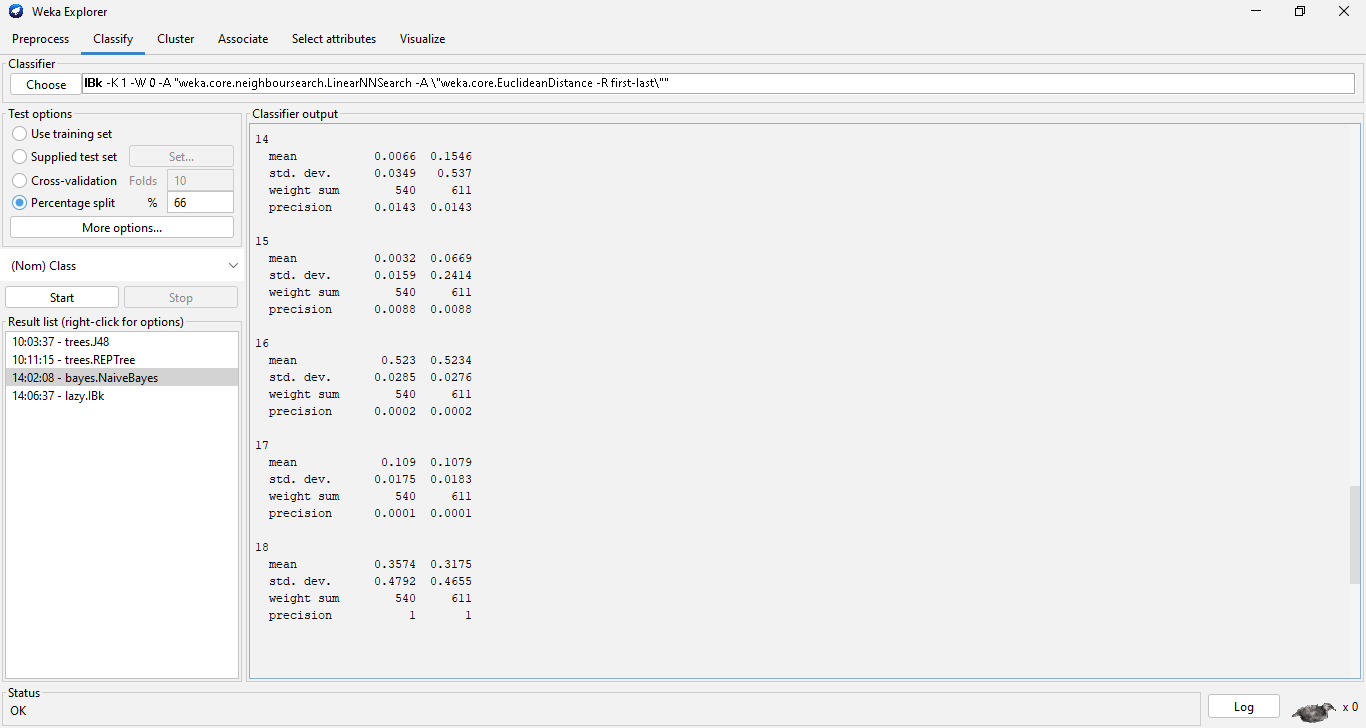
NAÏVE BAYES CLASSIFICATION ALGORITHM

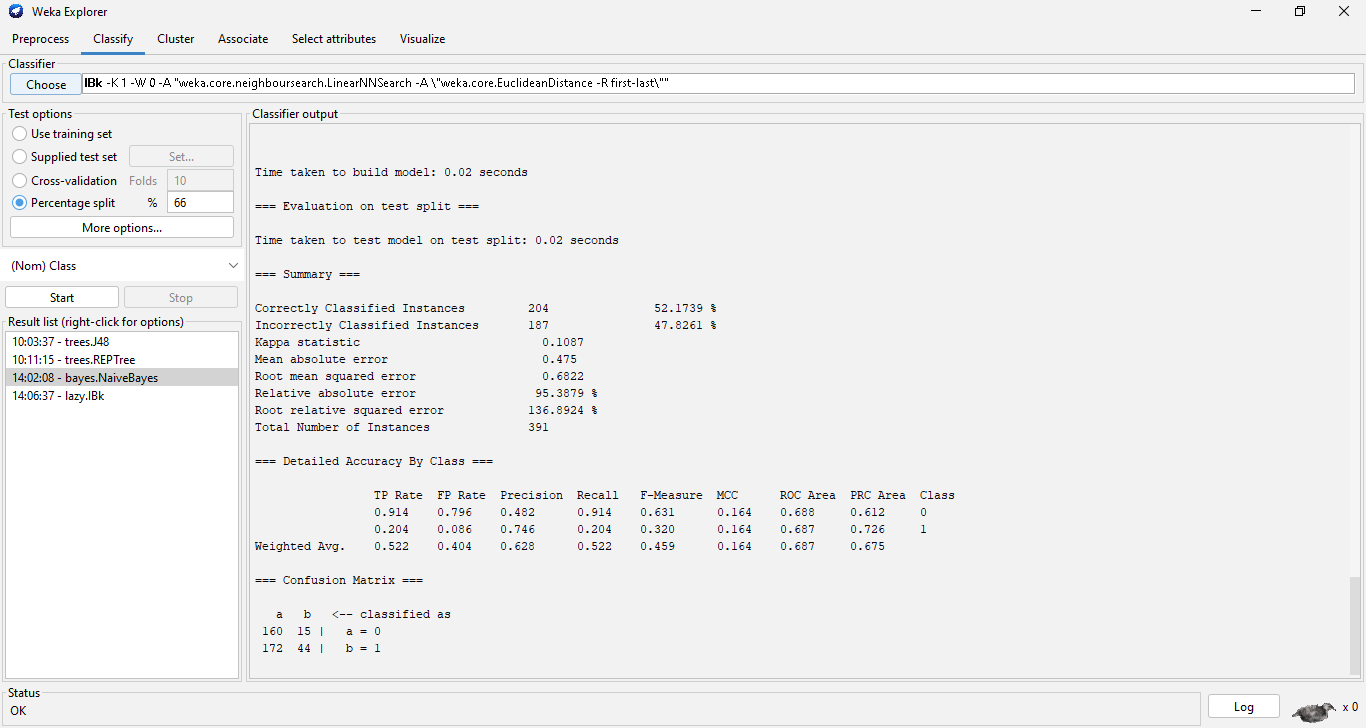
The classifier output for naïve bayes algorithm is given below:







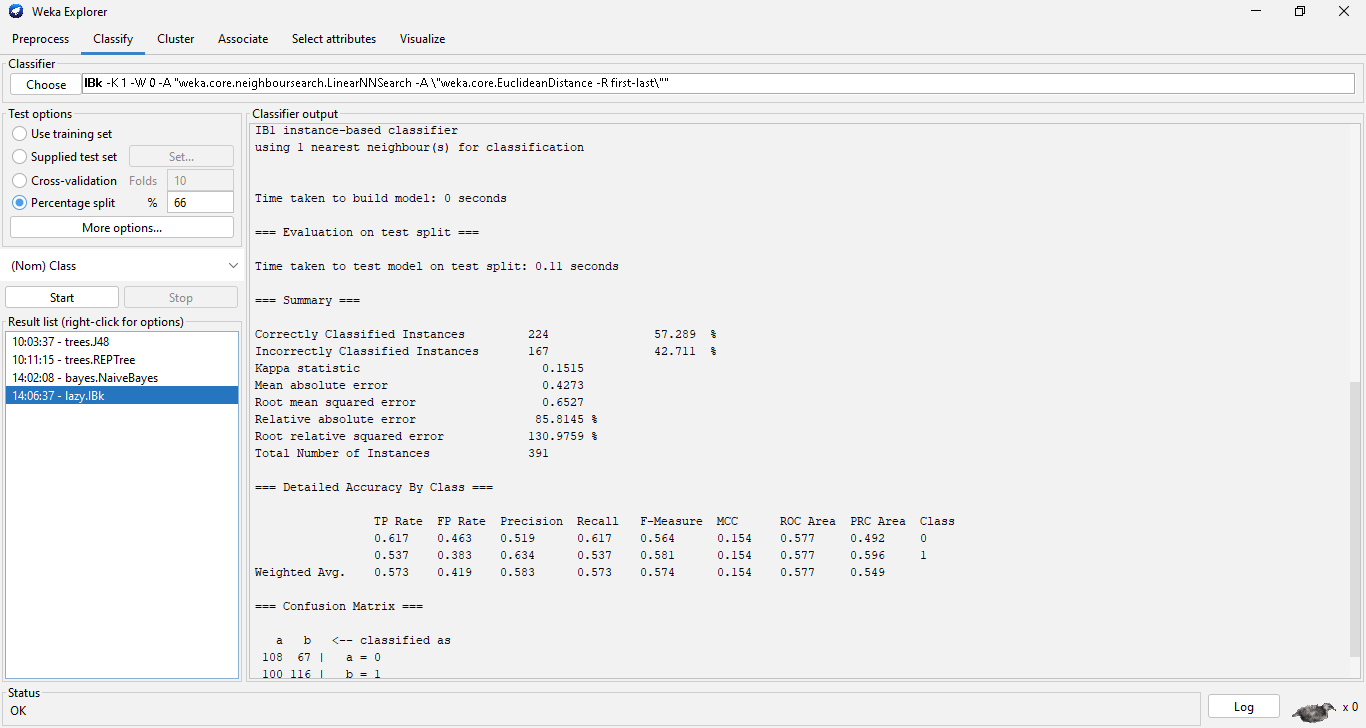




The time taken to build the model was 0.02s. The time taken to test the model on test split was 0.03s. The model generally performs poorly with a classification accuracy of 52.1739%. It has a kappa statistic of 0.1087 which is poor. It has a weighted ROC Area 0.687. It has a weighted precision rate of 0.628 and a recall rate of 0.522. For the confusion matrix, 160 instances of class a were correctly classified and 15 instances of class a were incorrectly classified as class b = 1. 172 instances of class b were incorrectly classified as class a and 44 instances of class b were correctly classified. Overall the model performs poorly.

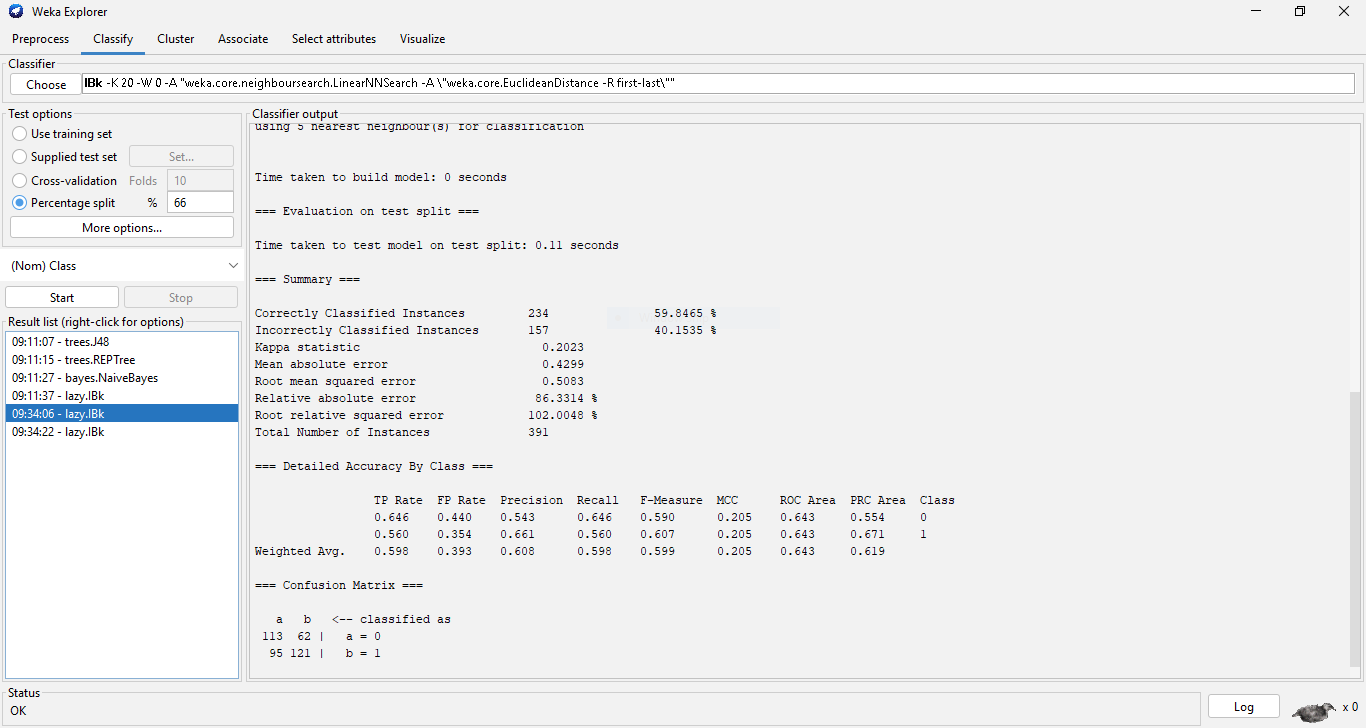
KNN CLASSIFICATION ALGORITHM

The KNN algorithm has no model. When the algorithm is applied to the dataset with a KNN value of 1, the results are shown in figure below.

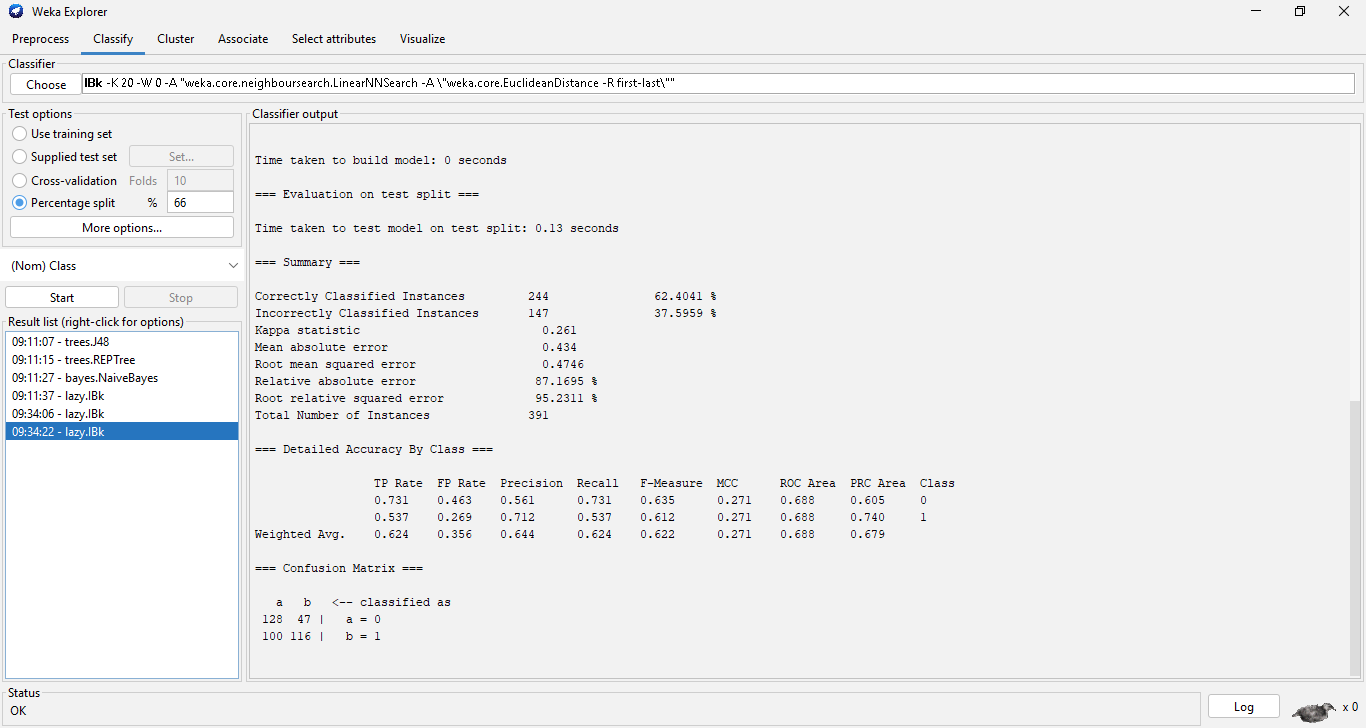


The time taken to build the model was 0s. The time taken to test the model on test split was 0.14s. The number of correctly classified Instances were 224 at a 57.289% accuracy and the number of incorrectly classified instances were 167 at a 42.711% accuracy. It has a kappa statistic of 0.1515. 0.1515 is a poor kappa statistic which is close to random guessing. The algorithm has a weighted true positive rate of 0.573 and a weighted false positive rate of 0.419. It has a weighted precision rate of 0.583 and a recall rate of 0.573. It has an F-measure of 0.573 and an ROC area of 0.577.

For the confusion matrix, 108 instances of class a were correctly classified and 67 instances of class a were incorrectly classified as class b = 1. 100 instances of class b were incorrectly classified as class a and 116 instances of class b were correctly classified. From the above statistics is can be determined that the model is not good and is close to random guessing. The KNN value was then changed to 5 which gave the following results:



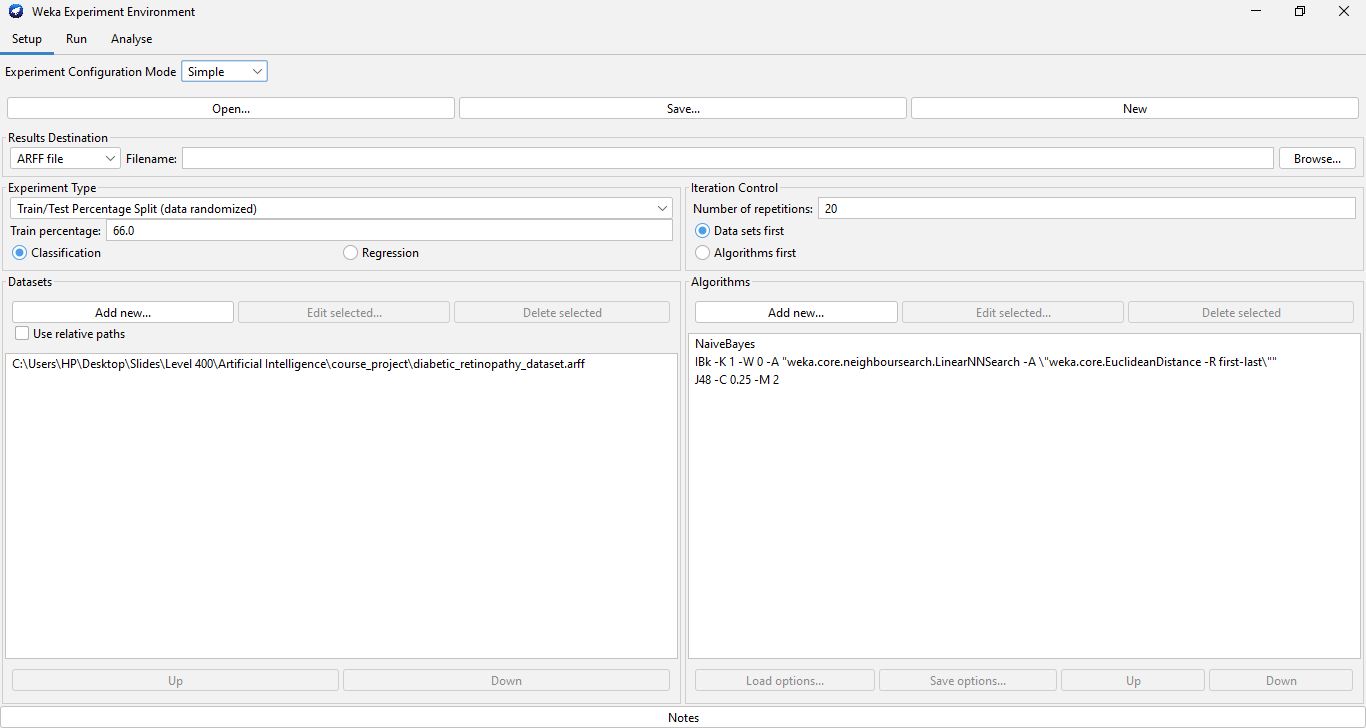
The percentage of correctly classified instances improves to 59.8465% and the precision, recall, f-measure and ROC area rates improve as well but the model still is not good enough. The KNN value was varied to 20 and the results are shown below:



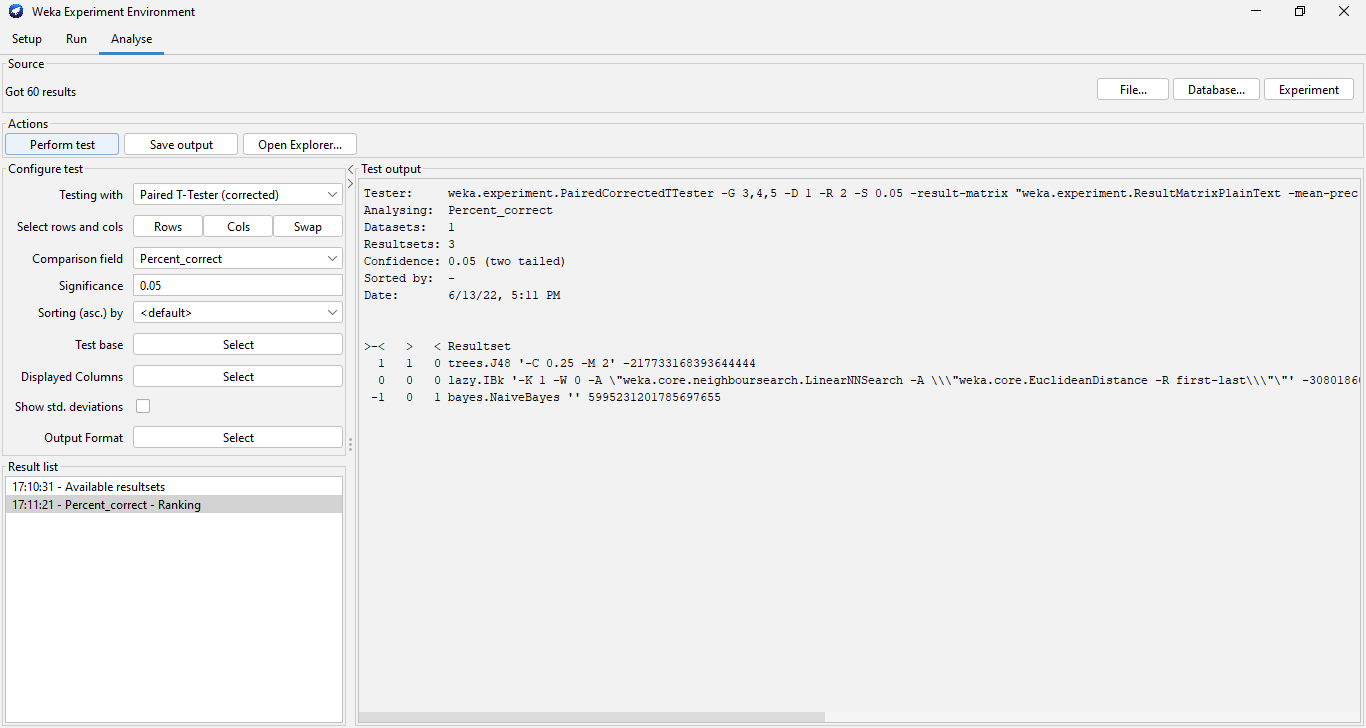
The percentage of correctly classified instances improves to 62.4041% and the precision, recall, f-measure and ROC area rates improve significantly. It can then be observed that for such a noisy dataset the accuracy of the model improves as the KNN value increases.

COMPARING THE ALGORITHMS AND TESTING FOR ACCURACY

The three algorithms were added to the experiment explorer in weka as shown below:

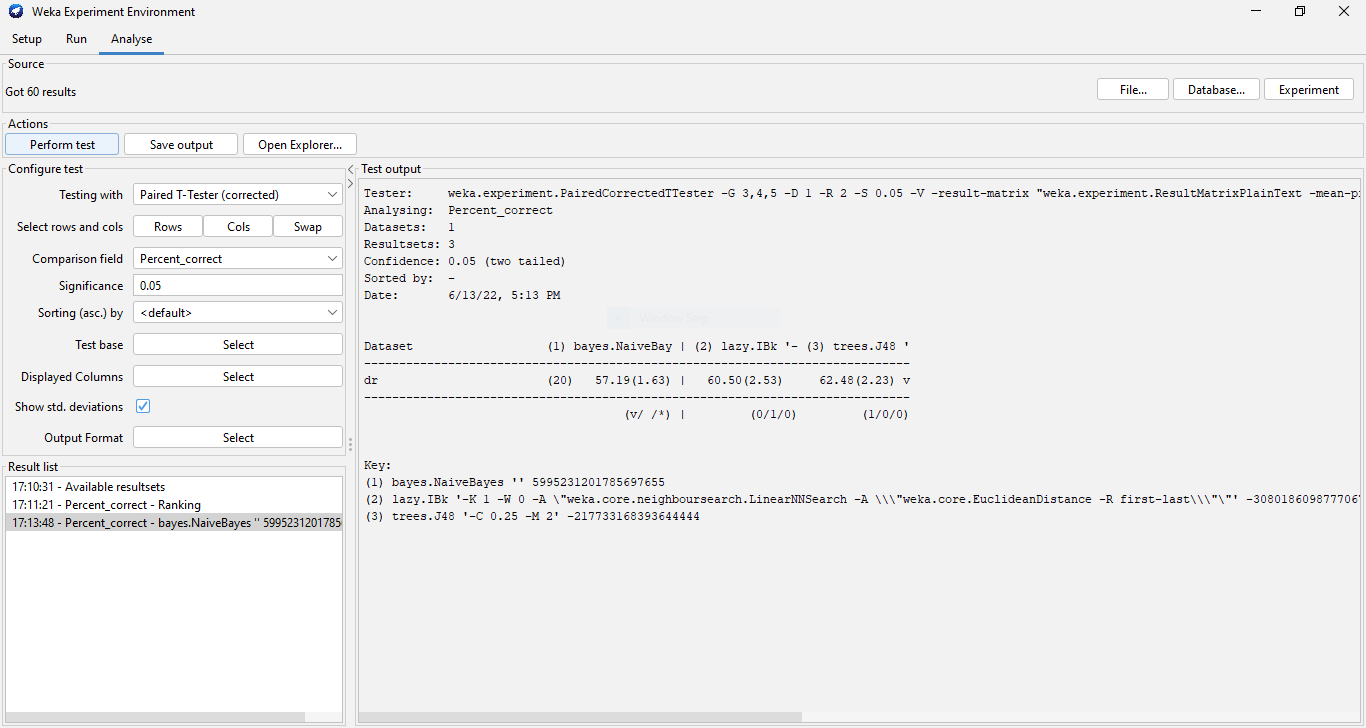


The algorithms were ranked as shown below:

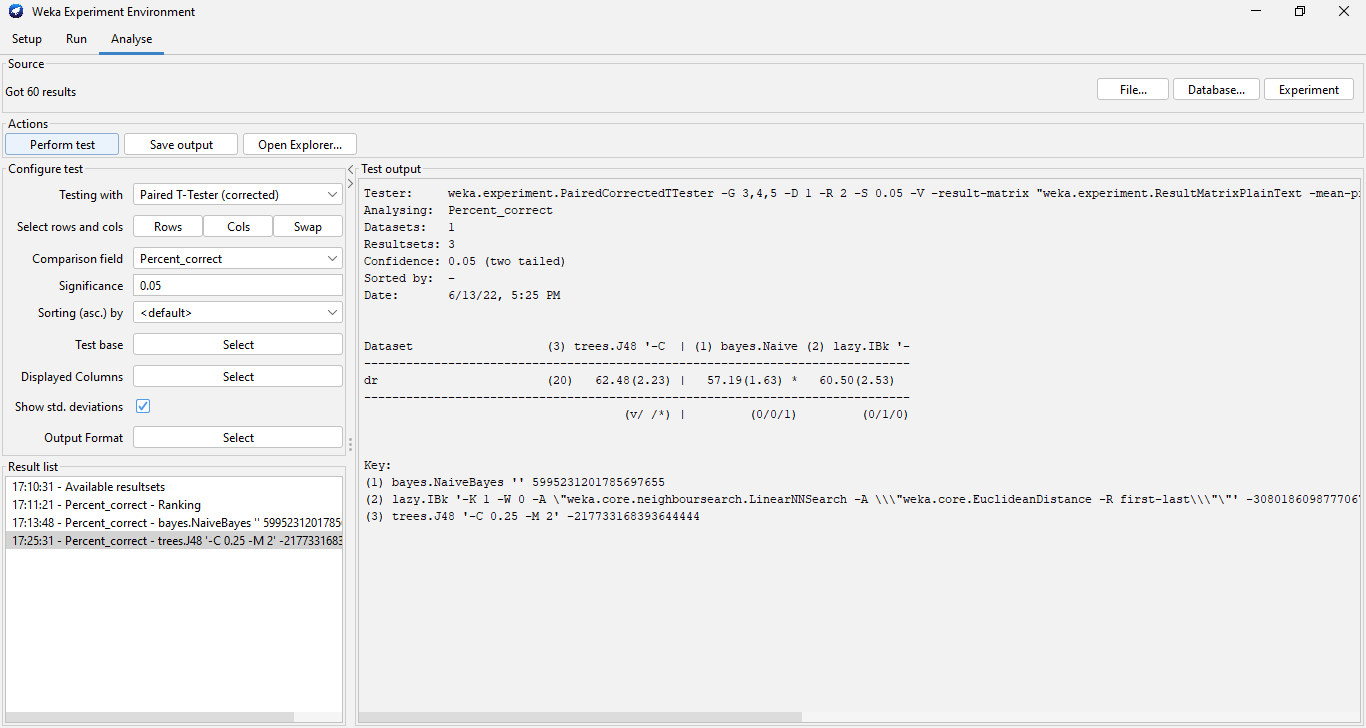


It can be observed the J48 algorithm performs best followed by the KNN algorithm and then the Naïve bayes algorithm.

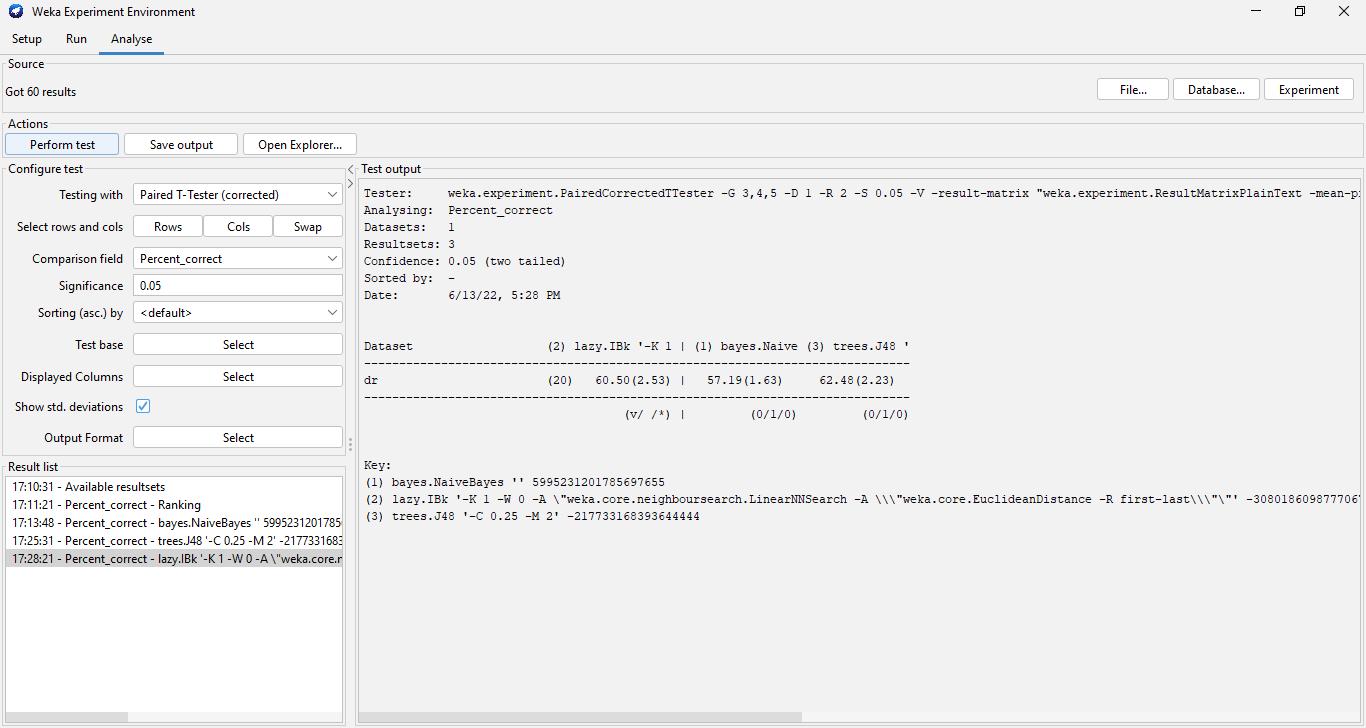
The algorithms accuracy against each other was then tested and the standard deviation shown below:



It can be observed the J48 algorithm has a higher accuracy and better performance as compared to Naïve Bayes algorithm.



It can be observed the Naïve Bayes algorithm has a worse accuracy and performance as compared to the J48 algorithm.



It can be observed the accuracy and performance of the Naïve Bayes and the J48 algorithm as compared to the KNN algorithm is statically insignificant.

REFERENCES

1. N.SaravanaN, Dr.V.Gayathri "Performance and Classification Evaluation of J48 Algorithm and Kendall’s Based J48 Algorithm (KNJ48)". International Journal of Computer Trends and Technology (IJCTT) V59(2):73-80, May 2018. ISSN:2231-2803. www.ijcttjournal.org. Published by Seventh Sense Research Group.
2. <https://www.analyticsvidhya.com/blog/2020/03/decision-tree-weka-no-coding/>
3. <https://www.tutorialspoint.com/weka/weka_classifiers.htm>
4. Pattern Recognition and Machine Learning, Christopher Bishop, 2006
5. https://www.britannica.com/technology/artificial-intelligence/Evolutionary-computing