DMAT – Assignment 1

Name: NITHIN UMAMAHESH Student Number: ­2996085

|  |  |
| --- | --- |
| **Course** | MSCBD-DMAT |
| **Stage / Year** | 1 |
| **Module** | Data Mining Algorithms & Techniques |
| **Semester** | 2 |
| **Assignment** | Assignment 1 |
| **Date of Title Issue** | Mon 14thOct2019 |
| **Assignment Deadline** | Mon 18th Nov2019 |
| **Assignment Submission** | Upload to Moodle |
| **Assignment Weighting** | 10% of module |

# Objective of the Assignment

To successfully apply a set of data mining skills imparted through lectures and lab session to a previously unseen dataset using Weka to achieve knowledge discovery and producing a written technical paper format report.

# Deliverables

A single zip called FirstName\_LastName\_StudentNumber.\_ass1.zip to be uploaded to Moodle containing the following files

* This file edited to contain the results of your investigation. Each of the **NUMBERED** headings should be expanded to satisfy the requirements of the section.
* A set of supporting files including but not limited to the following, which should be clearly referenced from your documentation.
  + dataset.arff
  + trainigSet.arff
  + testingSet.arff
  + j48tree.arff
  + associationrules.arff
  + kmeans.arff
  + dbscan.arff

# Choosing Your Dataset

1. Your dataset should concern a real-world problem that lends itself to easy understanding by your classmates.
2. It should ideally have >1000 tuples/rows/instances.
3. It should ideally have >=6 attributes
4. It should have attributes which can serve as labels so that the accuracy of your data analysis can be determined.
5. If you cannot find one dataset which is suitable for use with all techniques, then you may choose 2. Please clearly indicate which dataset was used in which case and introduce this dataset

Your search for appropriate datasets can begin with <http://www.kdnuggets.com/datasets/index.html>please post to the student discussion forum “Assignment 1 - Dataset Selection” clearly indicating which set you are using so that other students do not select the same dataset.

# Part 1 – Classification

## 1. Description of your dataset and findings – 10%

* **Title**: Brief title to capture the data and objective of your assignment

**Predicting task to determine if the person makes over 50k a year.**

* **Objective**: What you want to uncover by examining the data in this assignment. You can update this as you progress through your project revising it and making it more specific.

**We use “Adult census income” dataset from UCI machine learning repository to determine if the person makes over 50k a year. We look forward in achieving this by examining various attributes of a person such as education level, sex, marital status, etc.**

* **Data description:** A description of the data in detail under the following subheadings:
  + The problem domain

**The problem under discussion will fall into marketing domain whereby the companies can market their products to a person as per his income.**

* + The source of the data

**The dataset for this prediction purpose was taken from UCI machine learning repository. This dataset was extracted by Barry Becker from the 1994 Census database.**

* + The agencies working with the data

**Marketing agencies, government, etc.**

* + The intended use of the data

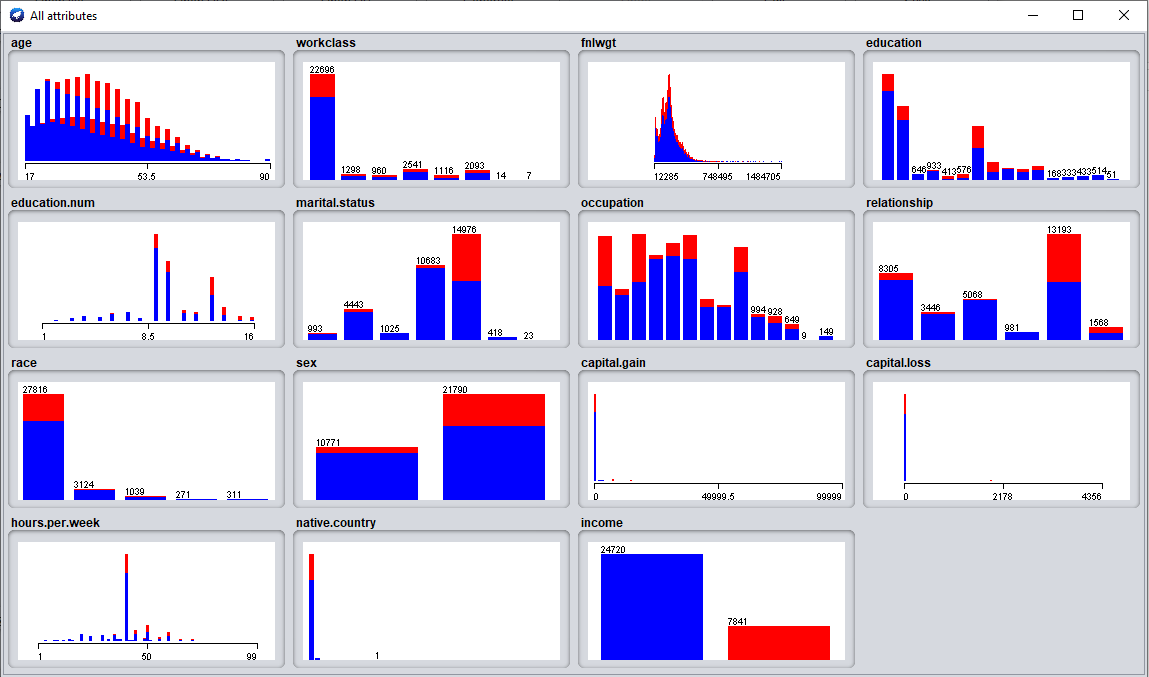
**The data is used to predict if the person income exceeds 50k or not. This in turn could be used by several agencies for customer acquisition.**

* + The attribute types of the data

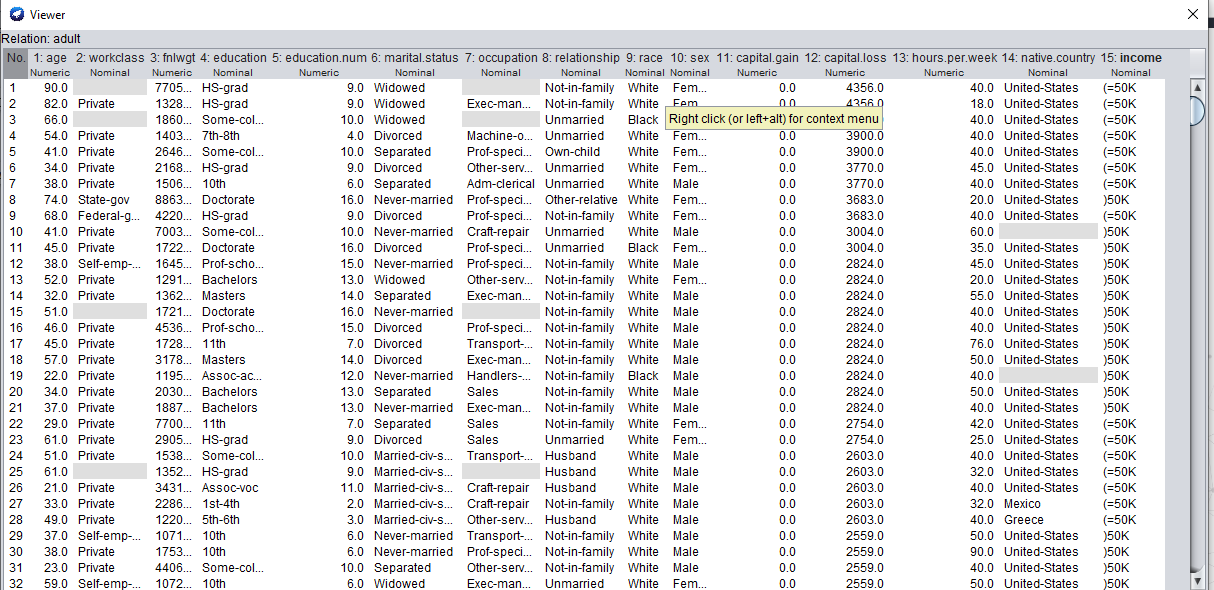
**This dataset contains 14 attributes and the description of attributes are shown below**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Description |
| age | Continuous | Age of the person |
| workclass | Categorical | Working class of the person |
| fnlwgt | Continuous | Final weight- weights on the Current Population Survey (CPS) |
| education | Categorical | Education qualification of the person |
| education-num | Continuous | Education level of the person |
| marital-status | Categorical | Marital status of the person |
| occupation | Categorical | Occuptation of the person |
| relationship | Categorical | Relationship status of the person |
| race | Categorical | Race of the person |
| sex | Categorical | Sex of the person, male or female |
| capital-gain | Continuous | Capital gain incurred |
| capital-loss | Continuous | Capital loss incurred |
| hours-per-week | Continuous | Number of hours per week worked |
| native-country | Categorical | Native country of the person |

Please include screen shots (with one or two sentences of summary) of the dataset and also of the data summaries and graphs that are available through Weka.



**Figure a – Visualization of all the attributes**



**Figure b – Displaying the data after loading into Weka.**

* **Summary of Findings**: This should feature here at the top of the document, but be written following the application of your data mining techniques. Should contain numerical values and discussion.

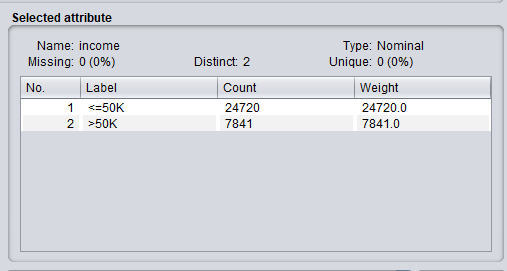
**Before performing experiments using decision tree we perform cleaning which includes replacing missing values, removing columns that does not add much importance to our model. We perform our experiment 1 by taking the default values i.e., confidence factor=0.25 for pruning. We achieved a performance of approximately 87.5% upon testing our model with the test dataset. For our experiment 2 we choose 0.4 as our confidence factor for pruning. With this parameter setting we achieve a performance of 88% upon testing our model. It is evident that our model has performed better with less pruning. For our last experiment, instead of using C45 pruning, we choose reducedErrorPruning method for our model. This model performs comparatively better with 86.5%. This method removes node that does not affect the accuracy of the model. Also, for all our experiments we see that ROC curve is close to 1, meaning our models accuracy is good for predicting the income of the person.**

## 2. Preprocessing –7%

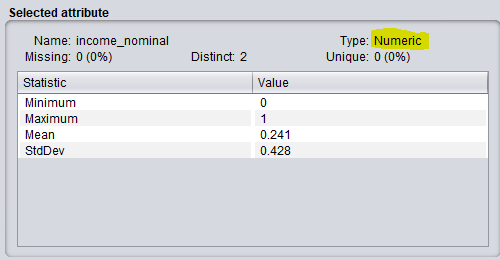
In this section you should

1. Identify the set of preprocessing techniques that can be applied to your data and clearly indicate which techniques are appropriate and which ones are not.
2. Provide evidence through screenshot of the effects of preprocessing the data along with a short explanation.
3. Generate a file called dataset.arff which is the outcome of the preprocessing.

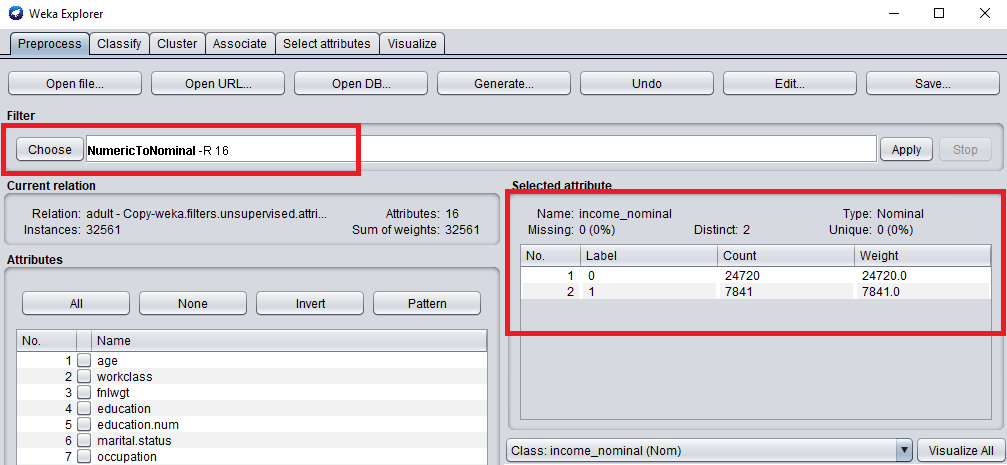
**Firstly, we convert the label income to binary 0’s and 1’s where 1 represent instances where income is greater than 50k and 0 represents income less than or equal to 50k. We are doing this process using Microsoft excel.**



**Figure 1(a) – Before transforming into 0’s and 1’s**



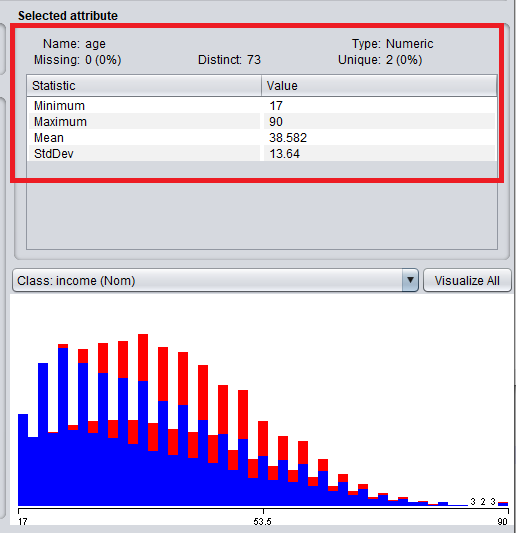
**Figure 1(b)- After transformation into 0’s and 1’s using excel**



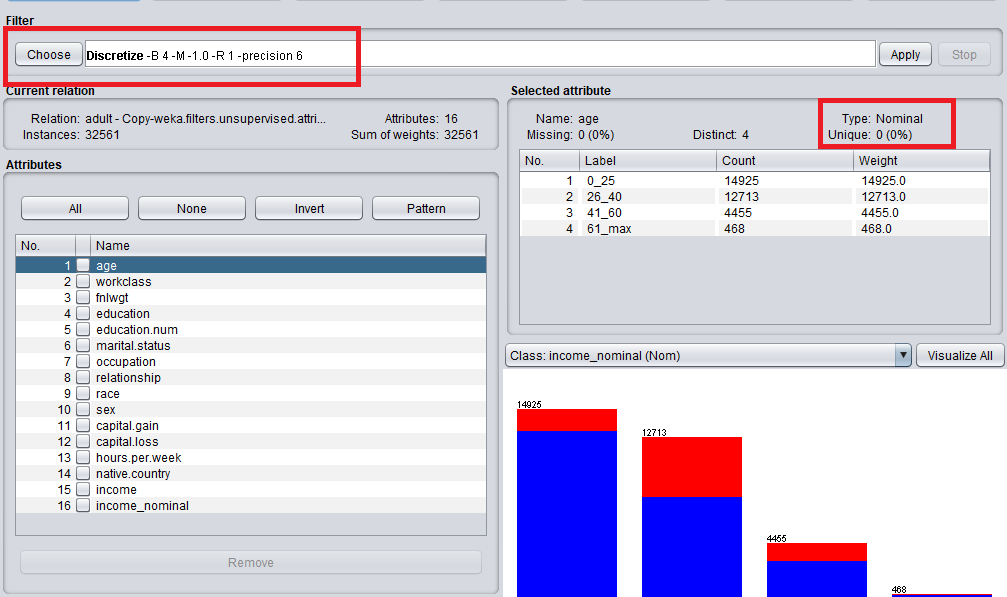
**Figure 1(c)- After converting numeric to nominal**

**As a second preprocessing step, we would discretize the “age” attribute. As we can see from the summary, minimum age in the dataset is 17 and maximum age is 90. So we will discetize this into four groups namely min-25(young), 26-40(adult), 41-60(middle aged) and 60-max(old). This achieved by using the “Discretize” filter of Weka.**

**p.s. Since “Age” cannot be in decimal values, we replace them with the integer intervals in arff file.**



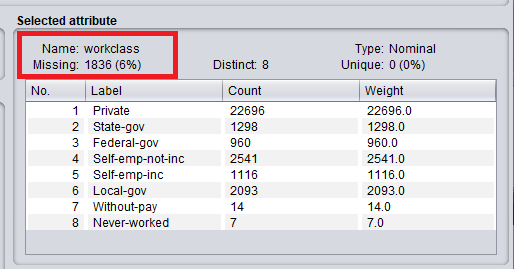
**Figure 2(a)- Age attribute before Descretization**



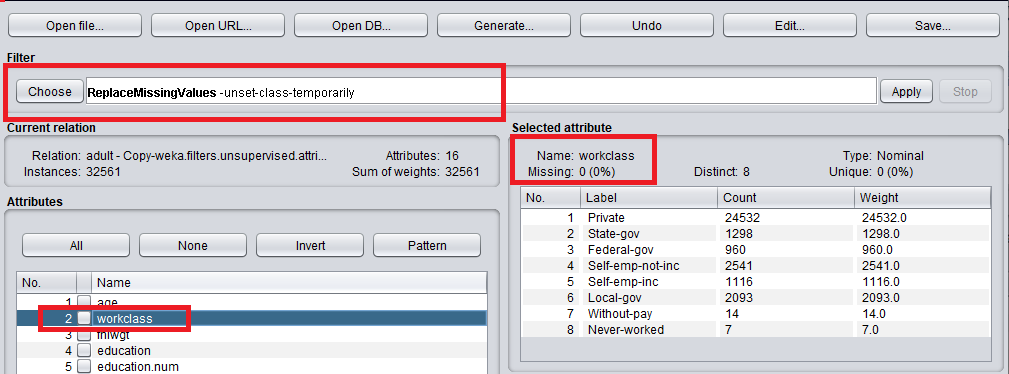
**Figure 2(b)- After descretizing the age attribute**

**After the above preprocessing, we would now replace all the missing values of the attributes using “ReplaceMissingValues” filter of Weka.**

**Below figures, 3(a) and 3(b), shows before and after applying filter for the “workclass” attribute. Similarly, filter has been applied for all the attributes.**



**Figure 3(a) –Before replacing the missing values**

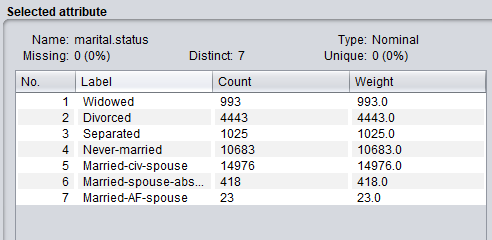


**Figure 3(b)- After replacing the missing values**

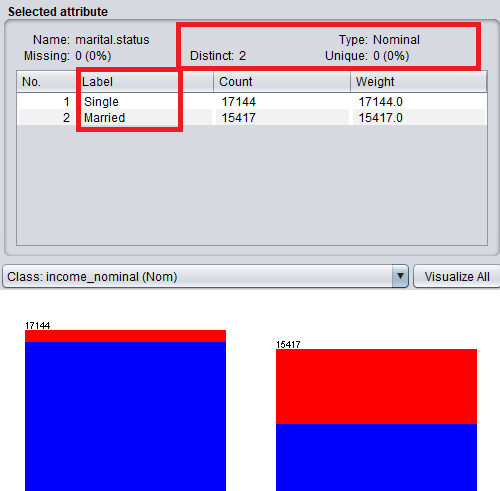
**Upon observation, we can infer that marital.status can be grouped into two categories, Single and Married. As a fourth step of preprocessing, cardinality of “marital.status” attribute has been reduced to Single('Never-married', 'Divorced', 'Separated', 'Widowed') and Married('Married-civ-spouse', 'Married-spouse-absent', 'Married-AF-spouse'). This is done using python as shown below.**

*data["marital.status"]=data["marital.status"].replace(['Nevermarried','Divorced','Separated','Widowed'], 'Single')*

*data["marital.status"] = data["marital.status"].replace(['Married-civspouse', 'Married-spouse-absent', 'Married-AF-spouse'], 'Married')*

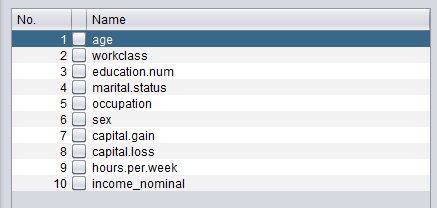


**Figure 4(a)- Before reducing the cardinality of marital.status**



**Figure 4(b)- After reducing the cardinality of marital.status**

**As a final step of preprocessing, we remove the attributes that are not of much importance in our prediction. After careful analysis, we remove income (as we have income\_nominal), fnlwght (as it does not any important information), relationship (as we have marital.status), education (as we have education.num that corresponds to the education level), race and native coutry( as the income is not determined by the race and native country)**



**Figure 5- Attributes after feature engineering**

## 3. Divide your dataset into training and test set – 3%

Follow the instructions presented in the link below divide the test into a training and testing set in the ration of (9:1).

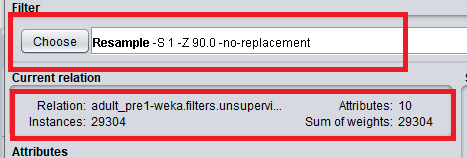
<https://www.youtube.com/watch?v=uiDFa7iY9yo> The files generated as part of this process should be saved and submitted as the following

* trainingSet.arff and
* testingSet.arff

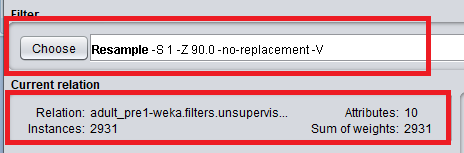
Screen shorts of these files should be included.

**We divide the dataset into training and testing set into 9:1 ratio using the “Resample” filter under Unsupervised -> instance -> resample**

**We select the “no replacement” option true to make sure the instances are not repeated.**



**Figure a – training set**



**Figure b – test set**



**Figure c – Training and test arff files**

# Experiments

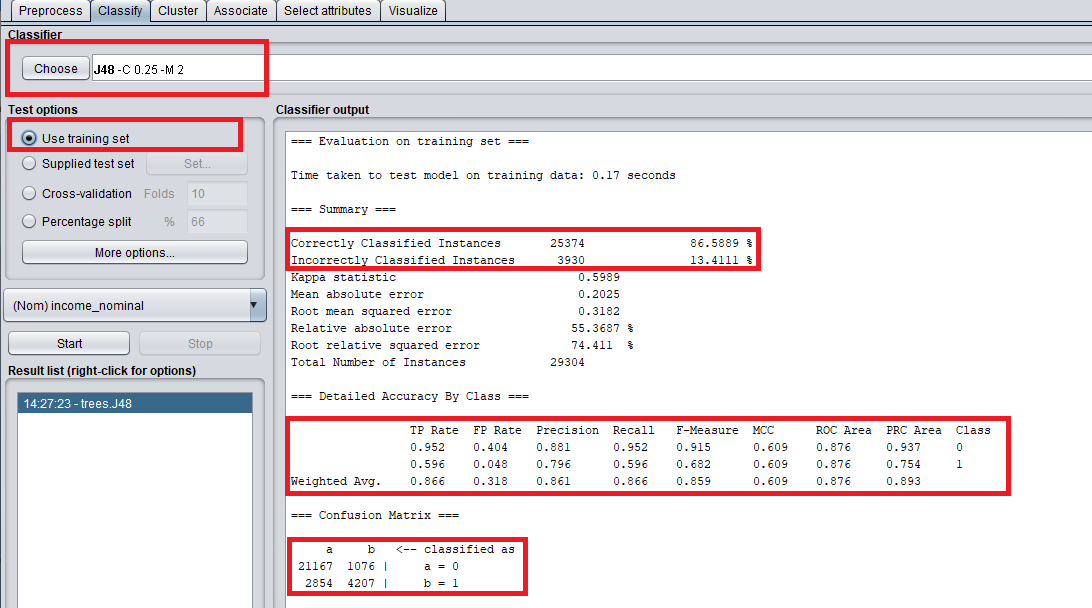
For each of the following classification techniques

1. Train your model using trainingSet.arff
2. Test your model using testingSet.arff
3. Write a few paragraphs analyzing the results. Be sure to vary parameters at least 3 times in each case. Support this analysis with screenshots of the following
   1. The model or a visualization of the model
   2. The results of the model
   3. Any additional output of the model including but not limited to
      1. Rules
      2. Confidence Values
      3. Confusion Matrixes
      4. Etc.
   4. Simple references to the notes or URL links to online resources complete with a sentence or two of explanation.

## 4. Classification: J48 Tree – 15%

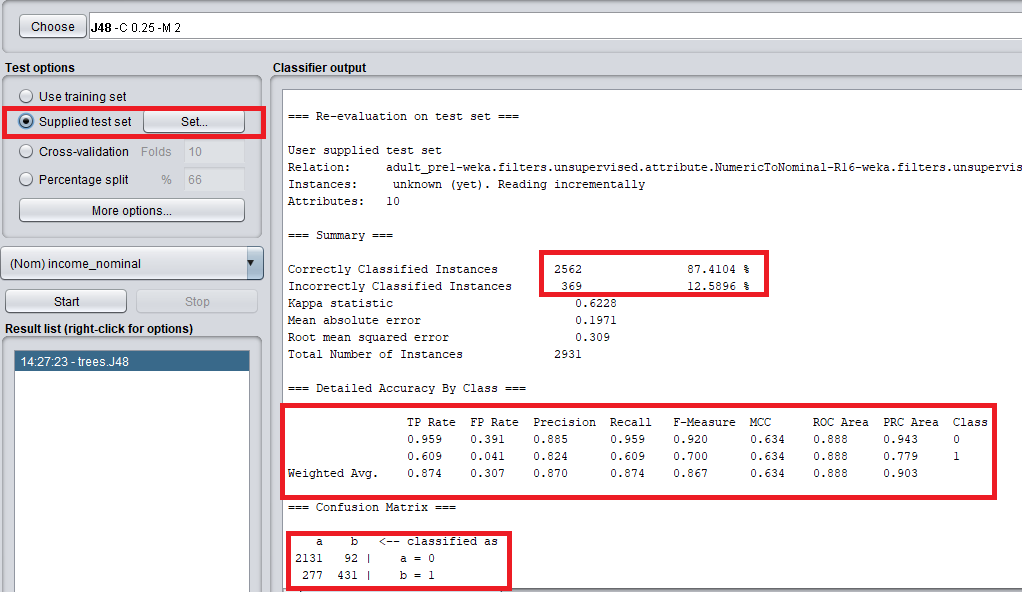
**Experiment 1: With default setting of J48**

**We train our model using trainingSet.arff and “use training set” test option. As seen in the below figure, evaluating on the training set (figure a) provide us an accuracy of approximately 86%. Also, the weighted average of the accuracy measures of the model are pretty reasonable. Since training and testing on the same dataset is not a good approach in machine learning, we next reevaluate the model providing the testingSet.arff to check the accuracy of the model.**

****

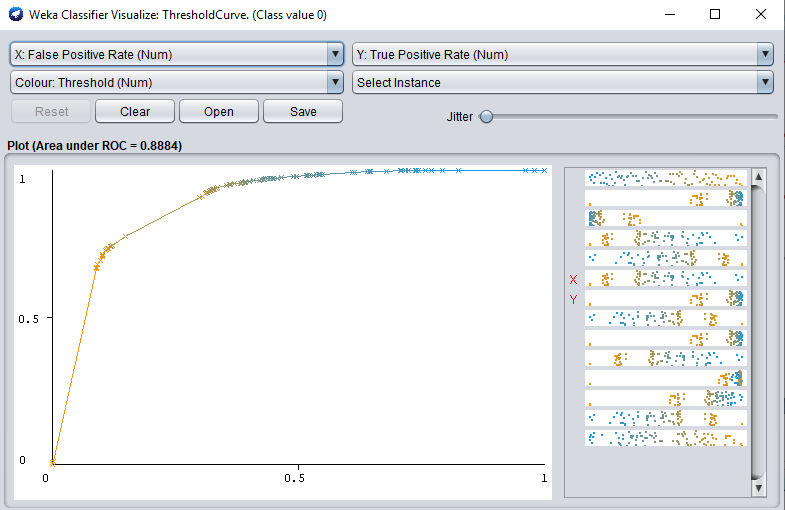
**Figure a – Evaluating on the training set**

**From the below figure b, we can observe that our model is performing better on test set by correctly classifying 88% of the instances in the test set. Also from the accuracy measures we can see that the weighted average precision model is 0.87 which is reasonably good. True positive rate for classifying if the income is less than 50k (class 0) is 0.959 which means our model has been successful in predicting class 0 in most of the cases. However, true positive rate for prediction of class 1(income >50k) is 0.609 which is not that great, but acceptable as our dataset contains less number of records for income greater than 50k and this might have affected our prediction for income above 50k. Also, F-Measure for both the classes are close to 1 which further supports our model.**

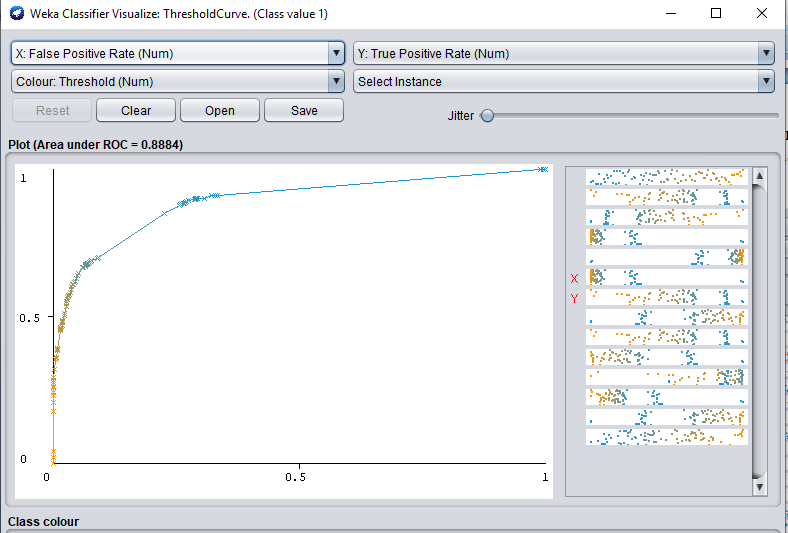
****

**Figure b- Evaluating on test set**

**From the below(figure c and d) ROC curves we can further conclude that our model performs better because the curves for both the class 0 and class 1 are close to 1.**

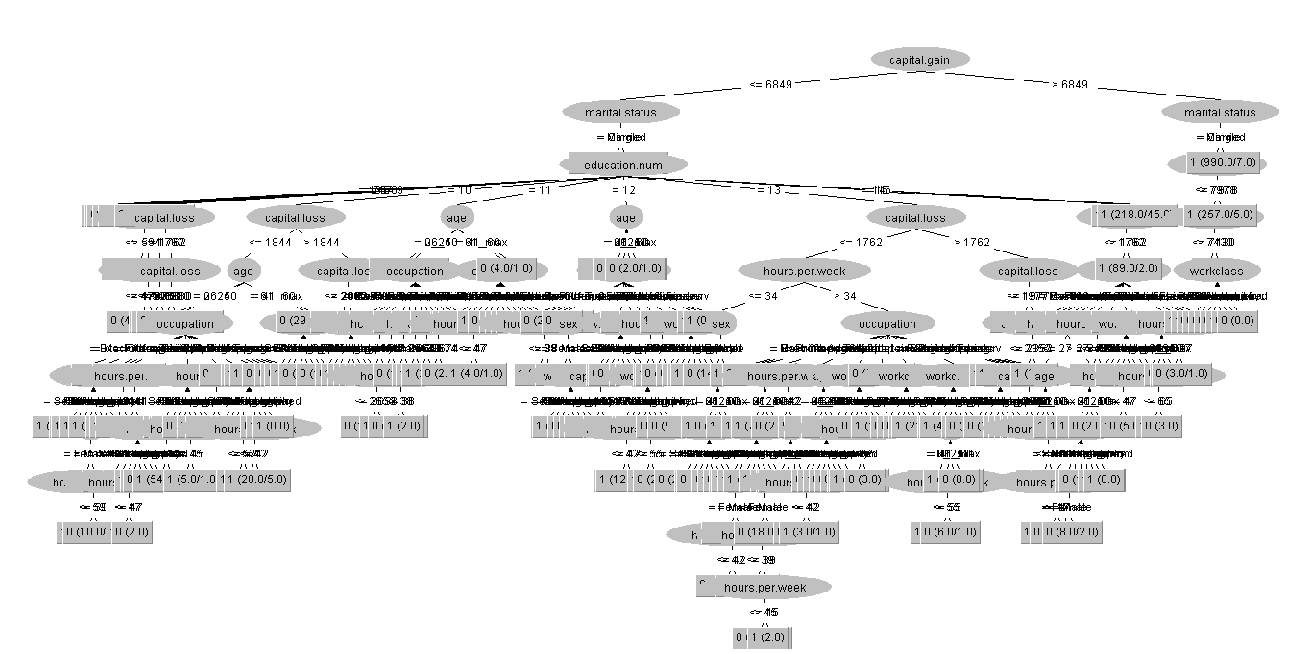
****

**Figure c – ROC for Class 0 (income <=50k)**

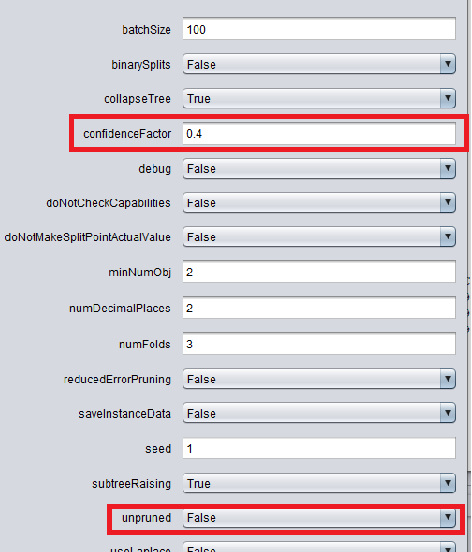
****

**Figure d – ROC curve for Class 1 (income >50k)**

**Visualisation of the tree**

****

**Experiment 2: Unpruned – False, Confidence factor-0.40**

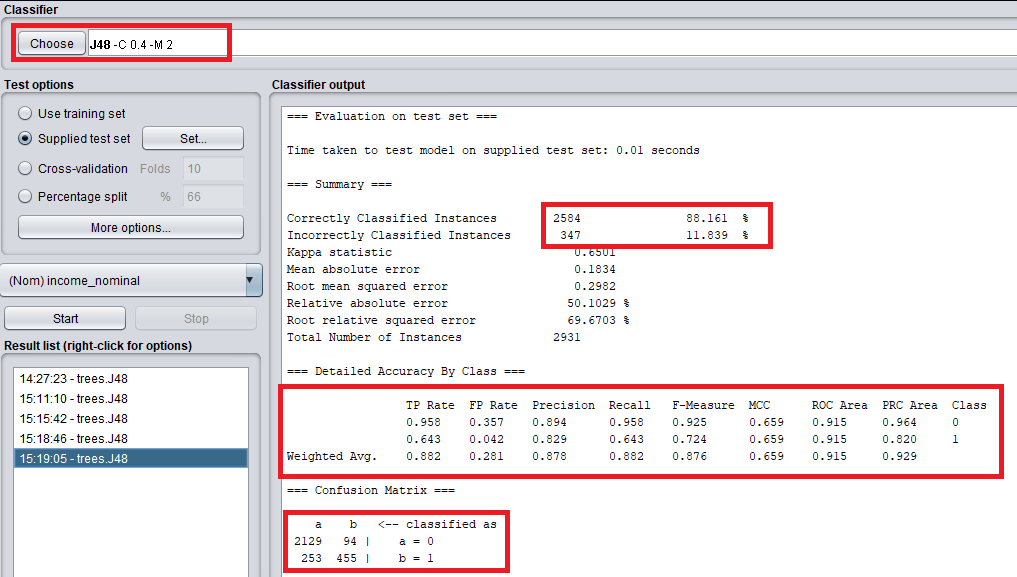
****

**Figure a – setting the confidence factor to 0.40 for pruning.**

**We perform our second experiment by increasing our confidence factor for pruning to 0.40. Higher value of confidence factor incur lesser pruning.**

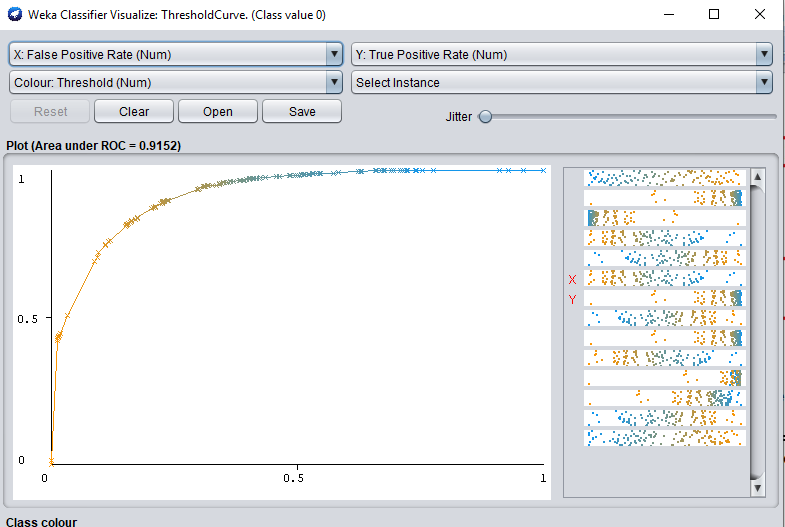
**Now, we perform our model evaluation with the above parameter settings and analyze the results obtained.**

**From the below figure b we can infer that our model performance is better when compared to our previous model. Also, this model has a higher accuracy in predicting class 1(income >50k) than the previous model. Overall performance of the model has also increased to 88%. Weighted average of the accuracy factors such as TP rate, F-measure, Precision, etc has also increased to provide further evidence for acceptance. Looking at the ROC area values which are close to 1, we can accept that this model performs better than the other model.**

****

**Figure b – Evaluation of model using testing set**

**The below figures c and d show the ROC curves for class 0 (<=50k) and class 1(>50k). We can see that ROC curve for class 1 in this model is closer to 1 than our previous model.**

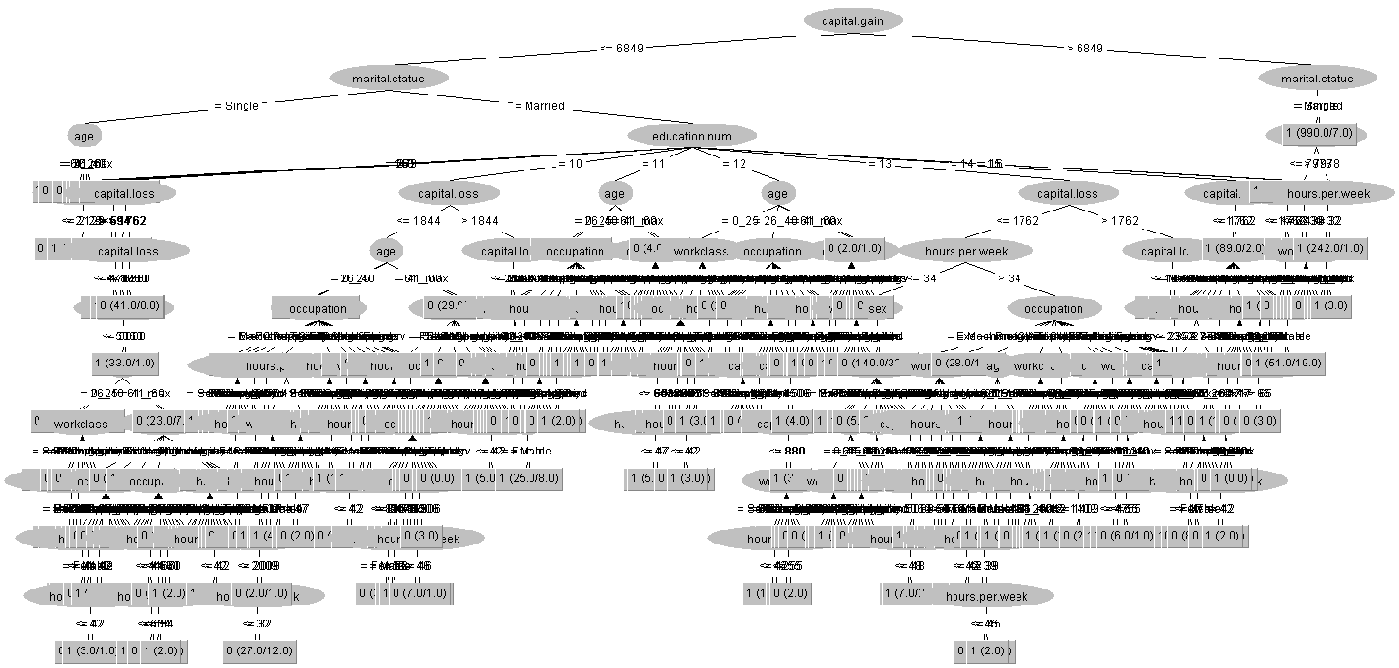
****

**Figure c – ROC curve for class 0 (income <=50k)**

## 

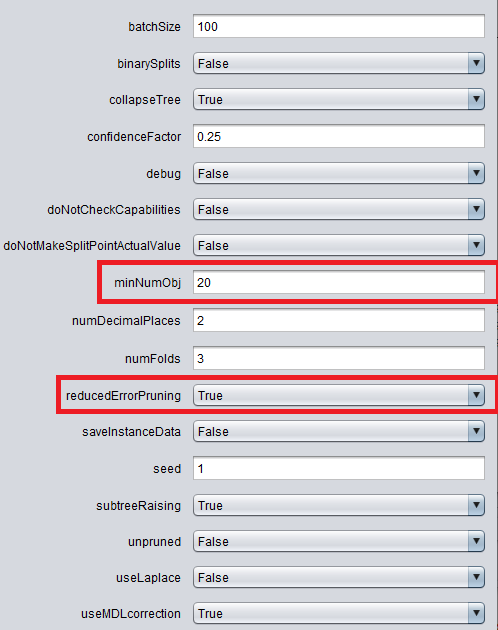
**Figure d – ROC for class 1 (income >50k)**

**Visualisation of the tree**

****

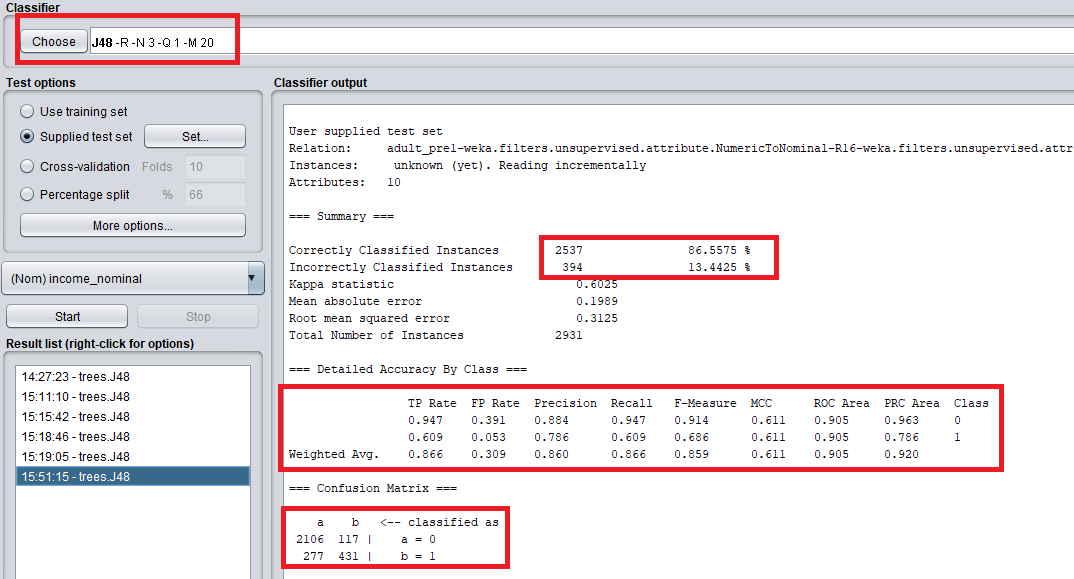
**Experiment 3:**

**For our third experiment, we set the minimum number of instances at each node to 20 and we use reduced error pruning instead of C 4.5 pruning. In this pruning process, each node is considered for pruning by removing the subtree and making it the leaf node and assigning most common class at that node. A node is removed if the resulting tree does not perform worse the accuracy of the decision tree. Pruning is continued until further pruning is harmful. Below figure a shows the changes made.**



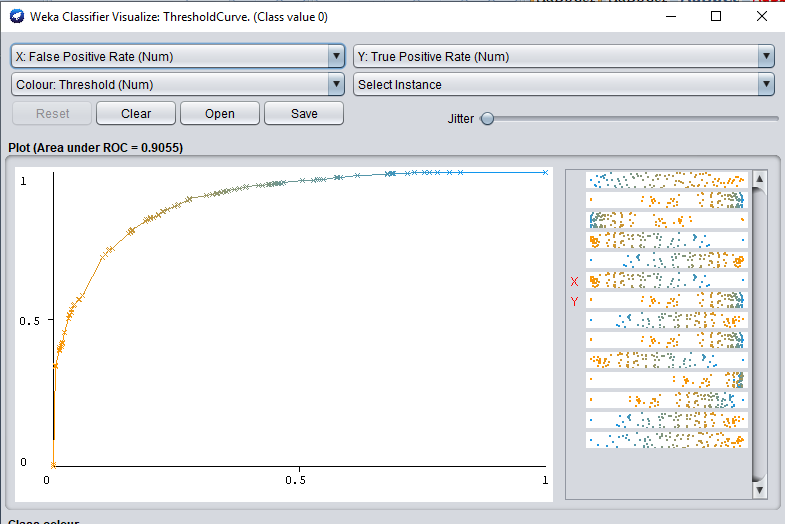
**Figure a – Reduced error pruning**

**Reduced error pruning decreases the accuracy of our model with 86% overall accuracy. This model performs worst than the other two models. However, the difference in their performance does not vary to a higher extent. This decrease I performance could be due to increasing the minimum number of instances required at each node. This model perform reasonably better in predicting class 1(>50k) but not as good as our previous model. All of the accuracy measuring factors are reasonable as shown in the below figure b.**

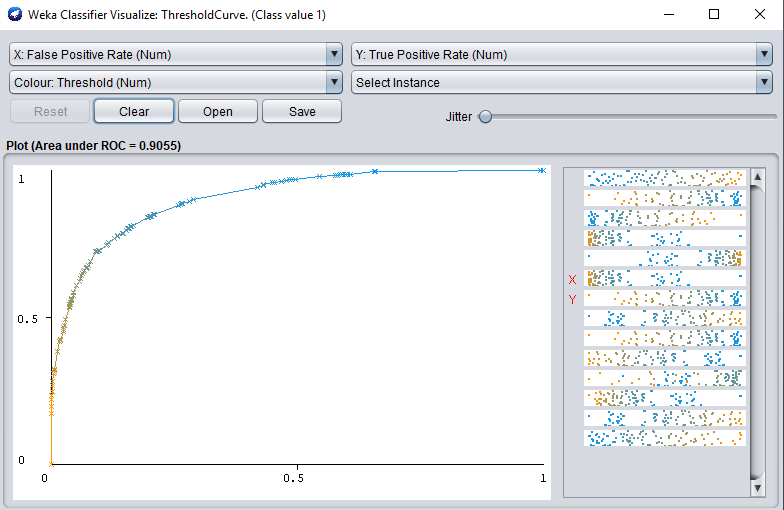


**Figure b – Evaluating the model using test set.**

**Below figures c and d, show the ROC curve for Class 0 and class 1 prediction.**

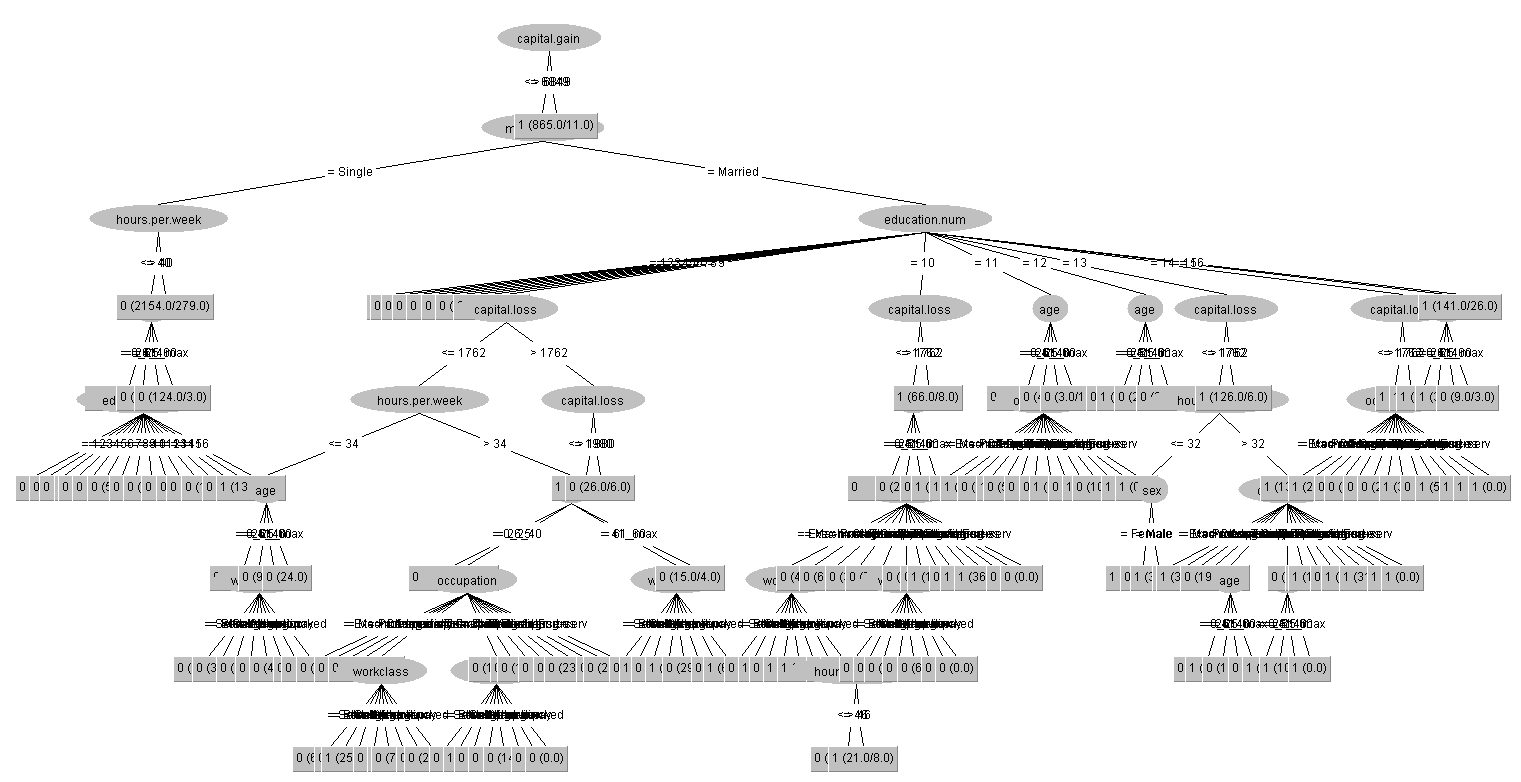


**Figure c – ROC curve for class 0 (income <=50k)**



**Figure d – ROC curve for class 1(income >50k)**

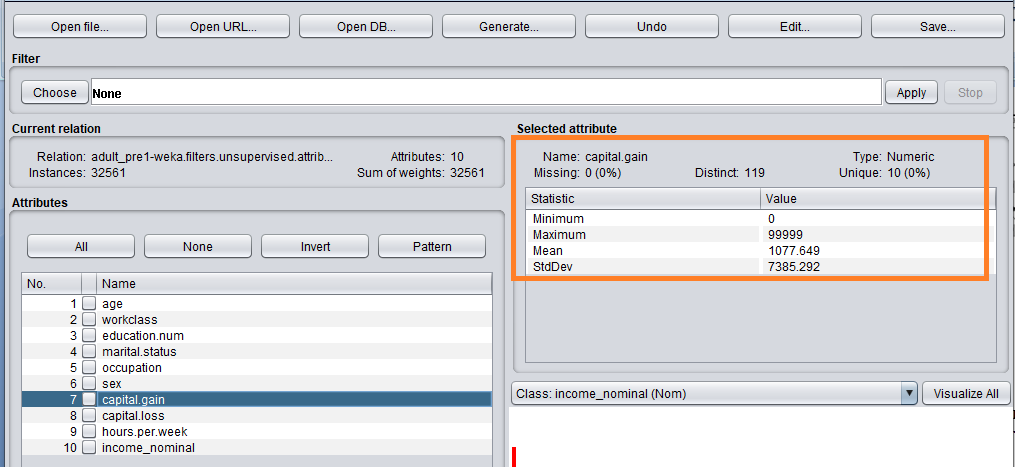
**Visualisation of the tree**



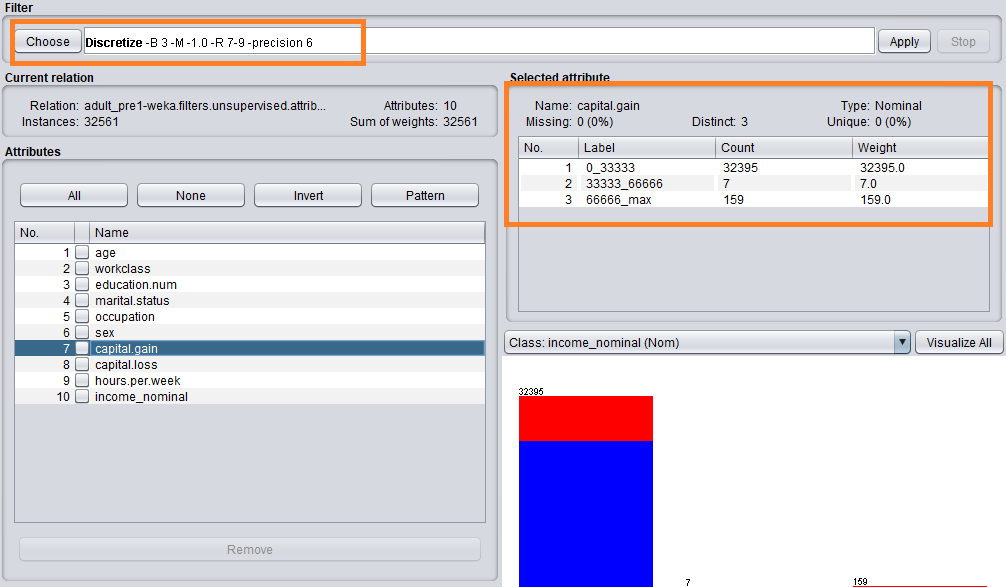
## 5. Classification: Association Rules – 15%

**Preprocessing**

**Since descretizing certain attributes such as capital loss, capital gain and hours per week would have led to loss of information while using decision trees, we had ignored them. However, in order to perform association rules it is necessary that the numerical attributes are descretized. Hence we perform descretization of the attributes as a preprocessing step as shown in the below figures (a) and (b).**



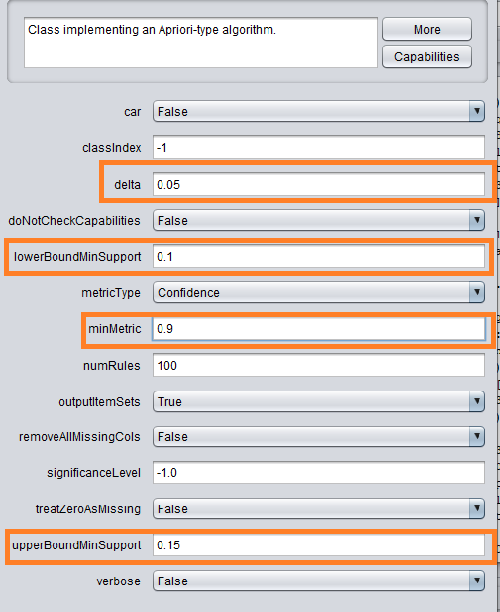
**Figure (a) – before descretization**



**Figure(b)-After descretization**

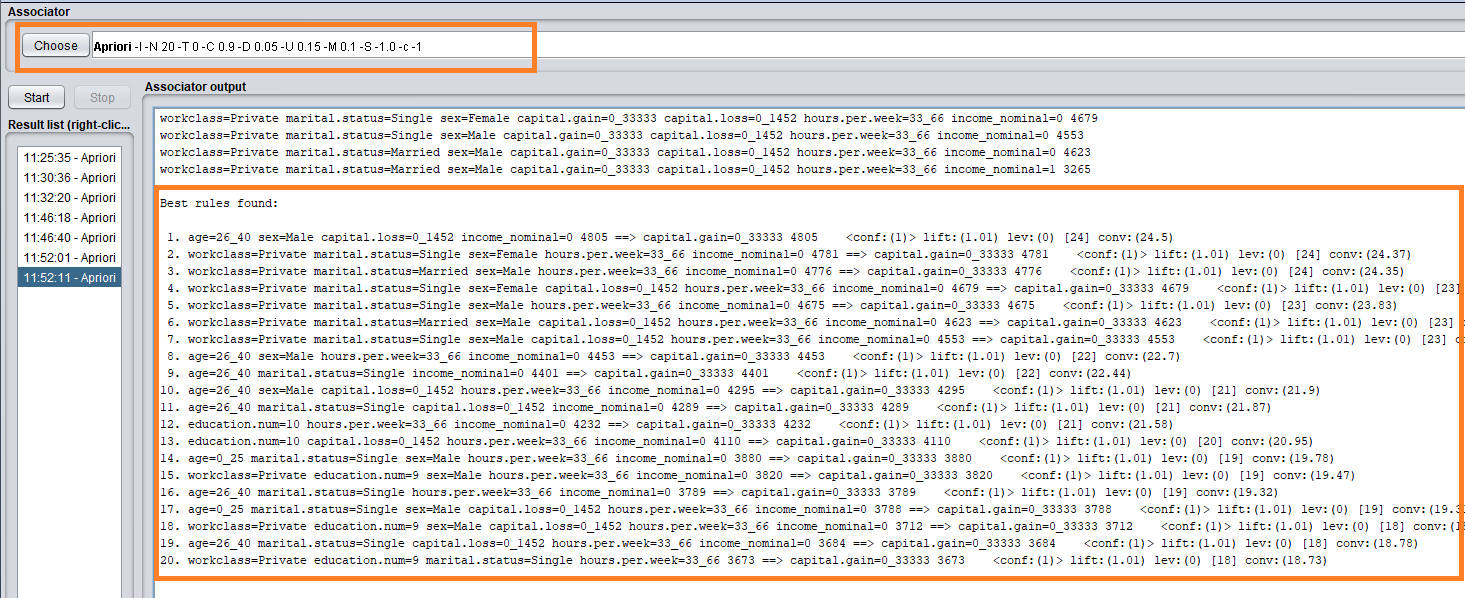
**Experiment 1:**

**Using Apriori with default settings i.e., minimum support of 0.1 and confidence interval of 0.9 and 20 best results. We set our upper bound limit to 0.15 and delta t0 0.05 to achieve the minimum support of 0.1. If the upper bound support is set to high then our minimum support automatically increases and gets close to upper limit. The settings are shown in the below figure 1(a).**



**Figure 1(a) – Apriori settings**

**Below figure provides the best rules obtained from the above parameter settings**



**Figure 1(b) – Top 20 rules using apriori**

**Interesting Rules:**

**Rule2:**



From the above rule we can depict that Females who are single and working under private workclass for 33\_66 hours and income less than 50k are highly likely to have a capital gain between 0\_33333.

**Rule20:**

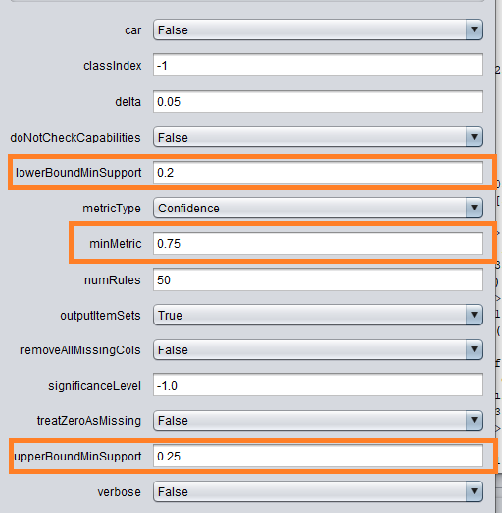
****

**From the above rule we can see that people who have worked under private class and have a educational qualification of level 9, single and works 33\_66 hours a week are highly associated with capital gain of 0\_33333.**

**The above parameter setting does not provide much information. Hence, we change the parameters further and check for better rules.**

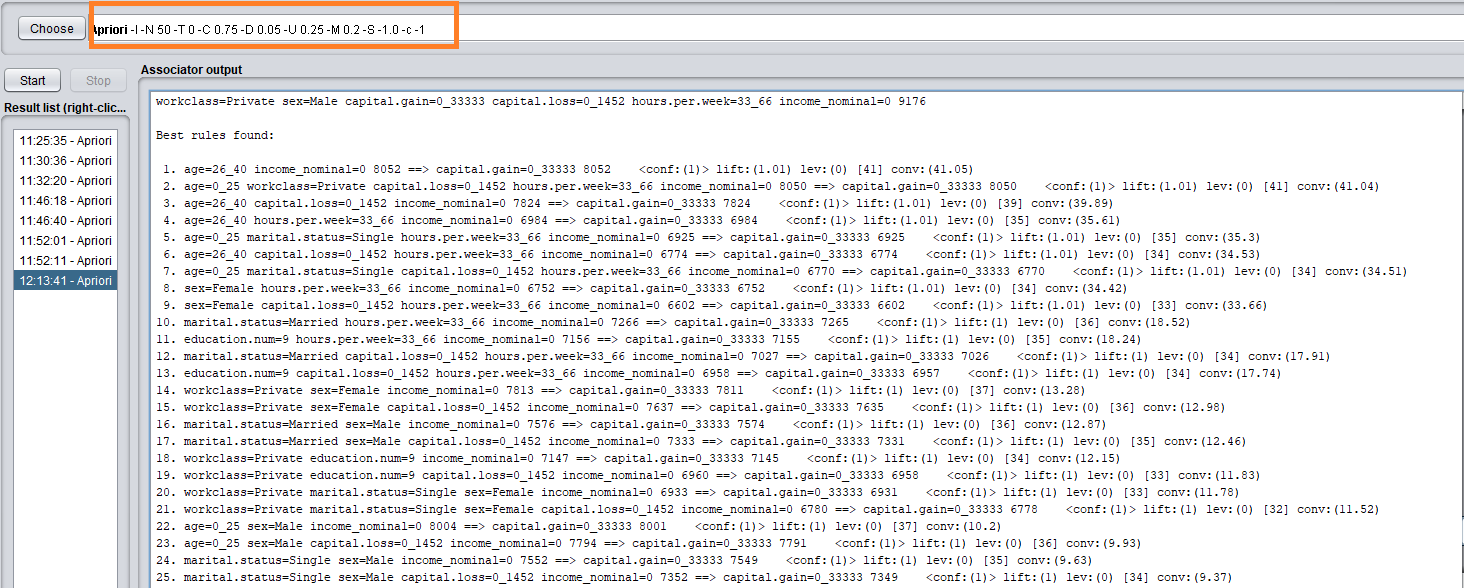
**Experiment 2:**

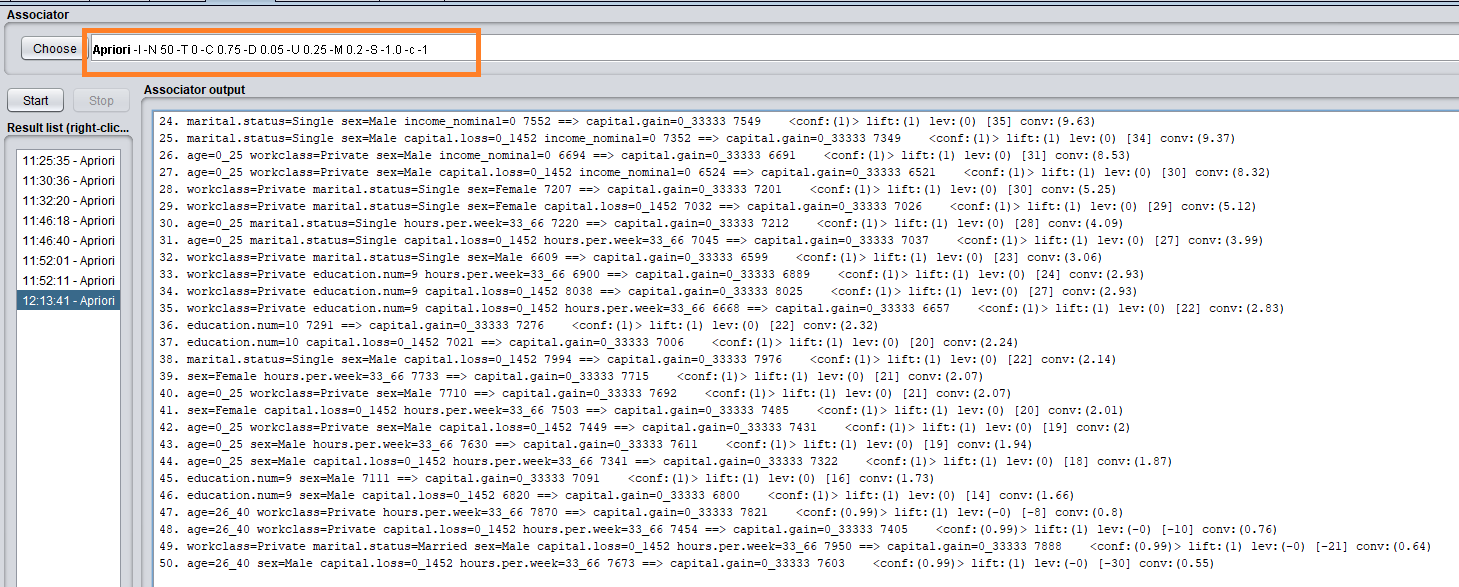
**Using Apriori with minimum support of 0.2 and confidence interval of 0.75 and 50 best results. We set our upper bound limit to 0.25 and delta t0 0.05 to achieve the minimum support of 0.2. If the upper bound support is set to high then our minimum support automatically increases and gets close to upper limit. The settings are shown in the below figure 2(a).**



**Figure 2(a) – Apriori settings**

**Below figure 2(b) shows the best 50 rules achieved from the apriori algorithm.**

****

****

**Figure 2(b)- 50 best rules**

**Interesting rules**

****

**From the above we can infer that single male who are working in private sector are highly associated with capital gain between 0\_33333.**



**From the above rule we can see that people of age 26\_40 who are working in private sector and works for 33\_66 hours a week with capital loss 1452 are strongly associated with capital gain of 0\_33333.**

****

**From the above rule we can see that capital gain of 0\_3333 and education level 10 are strongly associated. Higher the education level higher the capital gain.**

**Experiment 3:**

**For this experiment we will be using FP growth algorithm with 1984 US vote dataset.**

**Title: 1984 United States Congressional Voting Records Database**

**Source:**

**Congressional Quarterly Almanac, 98th Congress, 2nd session 1984,**

**Volume XL: Congressional Quarterly Inc. Washington, D.C., 1985.**

**About Dataset:**

**This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known(these three simplified to an unknown disposition).**

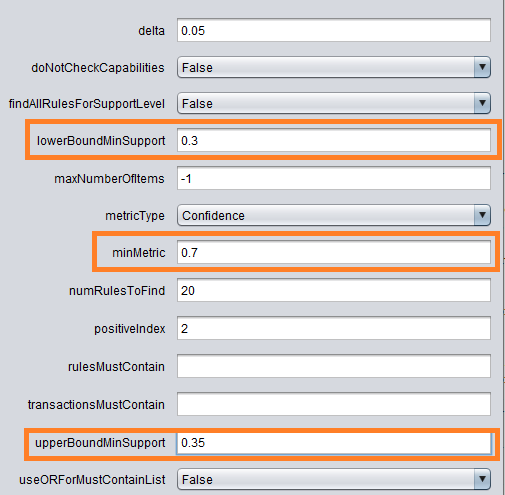
**Attributes:**

**handicapped-infants,water-project-cost-sharing,adoption-of-the-budget-resolution,physician-fee-freeze,el-salvador-aid,religious-groups-in-schools,anti-satellite-test-ban,aid-to-nicaraguan-contras,mx-missile,immigration,synfuels-corporation-cutback,education-spending,superfund-right-to-sue,crime,duty-free-exports,export-administration-act-south-africa, and Class**

**Number of Instances: 435 (267 democrats, 168 republicans)**

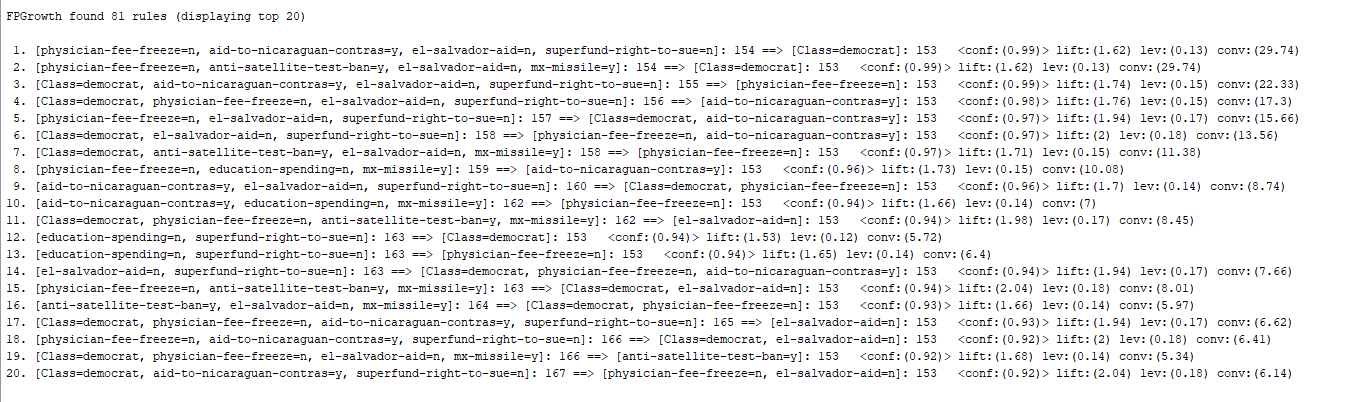
**Missing Attribute Values: Denoted by "?" (This simply means that value is not “yea” or “nay”)**

**Using FP growth with minimm support 0.3 and confidence 0.7 and 20 best rules. Below figure 3(a) shows the parameter setting.**

****

**Figure 3(a)- FP Growth parameter setting**

**The below figure 3(b) shows the top 20 best rules obtained from the above parameter setting.**

****

**Figure 3(b) – Top 20 best rules**

**Interesting Rules**

****

**From the above rule we can say that physical fess not being frozen is strongly associated with no to education spending and super fund right to sue.**

****

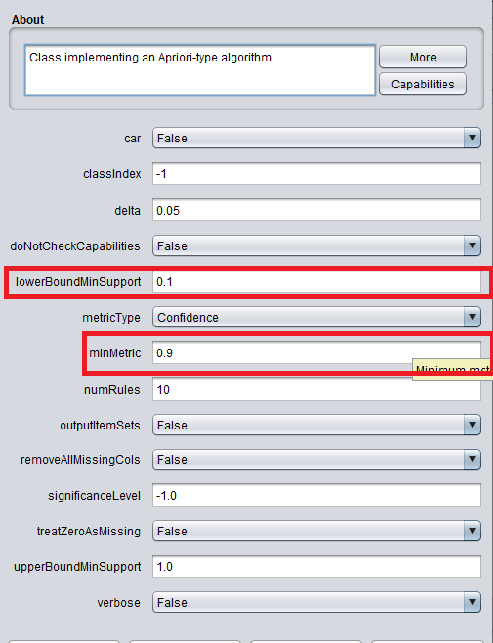
**From the above rule we can depict that aid to Nicaraguans and no aid to Salvadorans and no super fund to sue is strongly associated with democrat and no to freezing of physician fee.**

** The above rule shows the strong association between physician fess not being frozen with the democrats and anti satellite test ban=y and providing aid to Salvadorans and yes to missile.**

**Experiment 4: Using Apriori for United States Congressional Voting Records Database**

**Apriori with default settings i.e., minimum support of 0.1 and confidence of 0.90.**

**Please find the below screenshot for the setting in figure 1(a)**

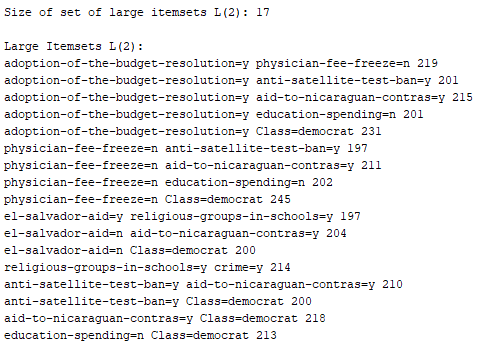
****

**Figure 1(a)-Apriori default settings**

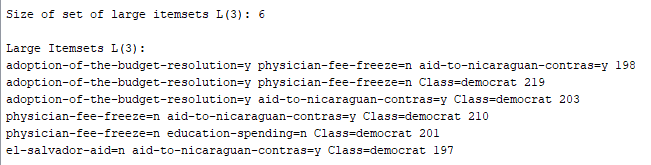
**Below figures show the itemsets generated by Apriori algorithm.**

****

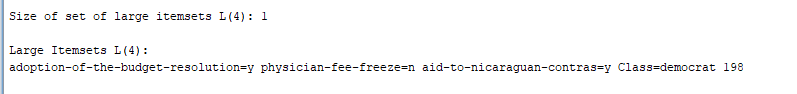
**Figure 1(b)- Itemset 1**

****

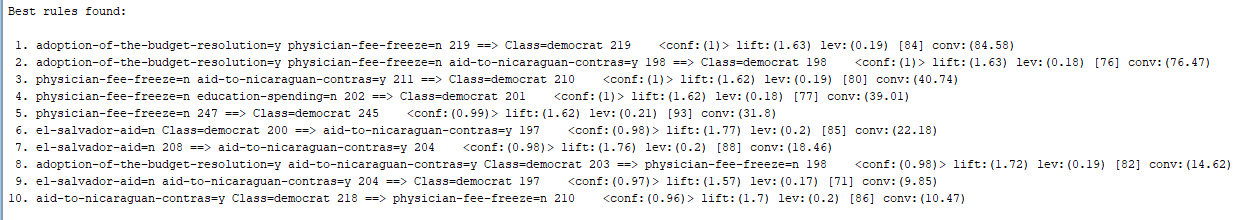
**Figure 1C - Itemset 2**



**Figure 1(d) – Itemset 3**



**Figure 1(e)- Itemset 4**



**Figure 1(f) – Best rules found**

# Interesting rules

**Rule 7 : We can see that when the government does not provide aid to Salvadorans there are chances for Nicaraguans to receive aid from USAID.**

**Rule 2 : When adoption of budget resolution is yes, physician fee frozen is no and the aid to Nicaraguan is yes it is highly associated with the Democrats.**

**Rule 5: physician fee not frozen is highly associated with the Democrats.**

# Part 2 - Clustering

## 6. Description of your dataset and findings – 10%

* **Title**: Predicting a pulsar star
* **Data description:** A description of the data in detail under the following subheadings:
  + The problem domain

**Physics and astronomy**

* + The source of the data

**This data is taken from Kaggle. However, the actual data source is**

<https://archive.ics.uci.edu/ml/datasets/HTRU2>

* + The agencies working with the data

**Physicist and astrophysicist**

* + The intended use of the data

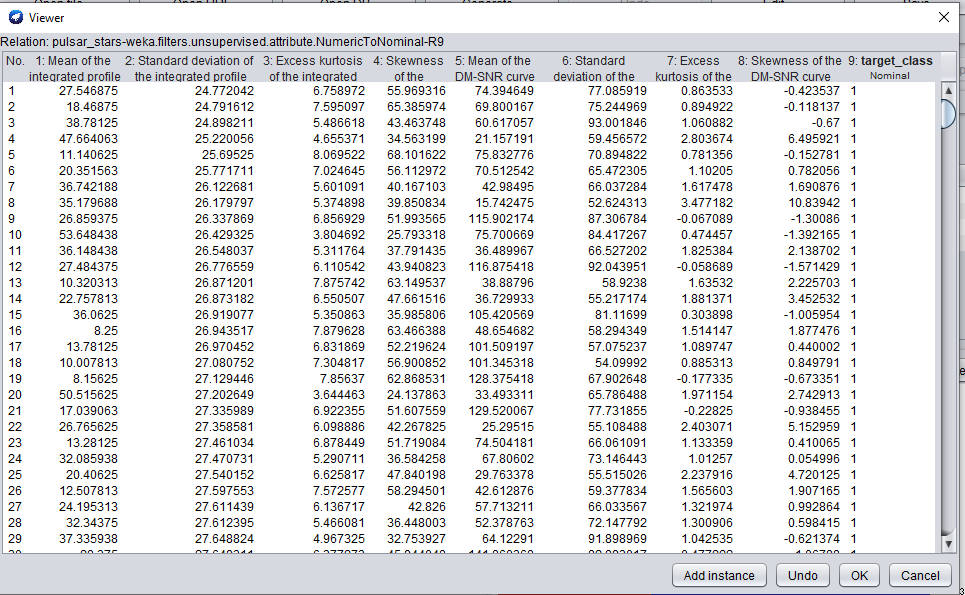
**This data is used to find whether it is a pulsar star or not.**

* + The attribute types of the data

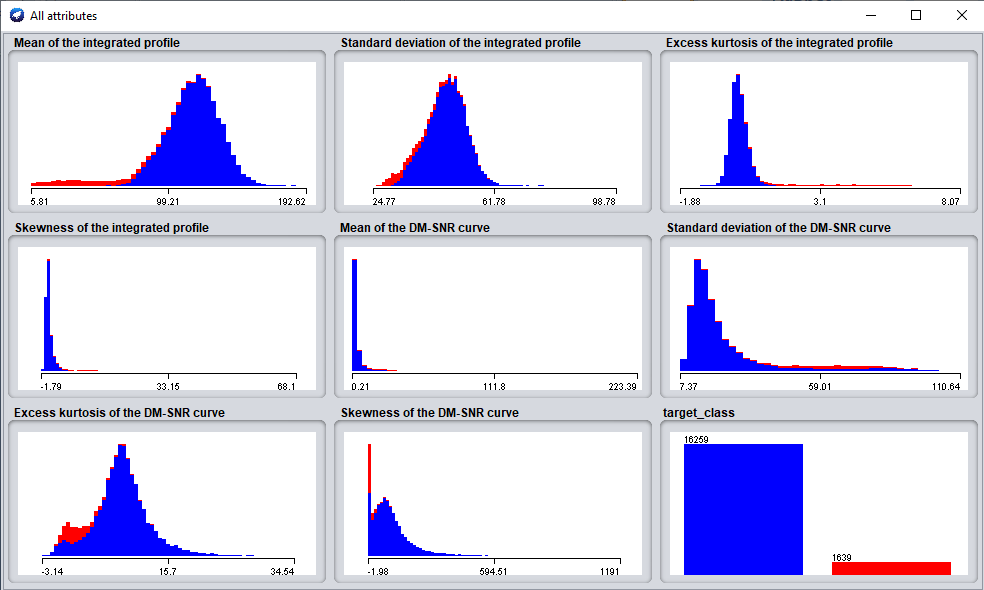
**This dataset contains 17898 instances and 9 attributes including label.**

|  |  |
| --- | --- |
| Attributes | Type |
| Mean of the integrated profile | Numeric |
| Standard deviation of the integrated profile | Numeric |
| Excess kurtosis of the integrated profile | Numeric |
| Skewness of the integrated profile | Numeric |
| Mean of the DM-SNR curve | Numeric |
| Standard deviation of the DM-SNR curve | Numeric |
| Excess kurtosis of the DM-SNR curve | Numeric |
| Skewness of the DM-SNR curve | Numeric |
| Class | Nominal |

Please include screen shots (with one or two sentences of summary) of the dataset and also of the data summaries that are available through Weka.



**From the above we can see that we have 8 attributes that are numeric and one class attribute or target.**



**The above figure provides the visualization of all the attributes present in the dataset.**

* **Objective**: What you want to uncover by examining the data in this assignment. You can update this as you progress through your project revising it and making it more specific.

**Each pulsar produces a slightly different emission pattern, which varies slightly with each rotation . Thus a potential signal detection known as a 'candidate', is averaged over many rotations of the pulsar, as determined by the length of an observation. Using this data we group them into two clusters that are pulsars and not pulsar stars.**

* **Summary of Findings**: This should be written following the application of your data mining techniques.

**After performing several clustering techniques, we can see that very less information on the clusters is obtained as there are no distinct clusters. Also, we see that there are many noise points while performing k-means and dbscan algorithms. This could be because of very less distance between points which does not help in forming distinct clusters. Also, we can see that with k-means and Euclidian distance, there are 1457 instances that are not correctly classified. However, k-means with Manhattan distance discovers 1547 instances that are classified incorrectly upon evaluating our model using “classes to classify” option. DBSCAN performs better because the clusters are formed based on the density of the point. However, with our model there are many noise points with DBSCAN. We can see that with large number of min points (1000) our model performs worst as the points required to form a cluster is huge and all points are noise. By choosing correct minimum number of points, we can discover patterns in our dataset.**

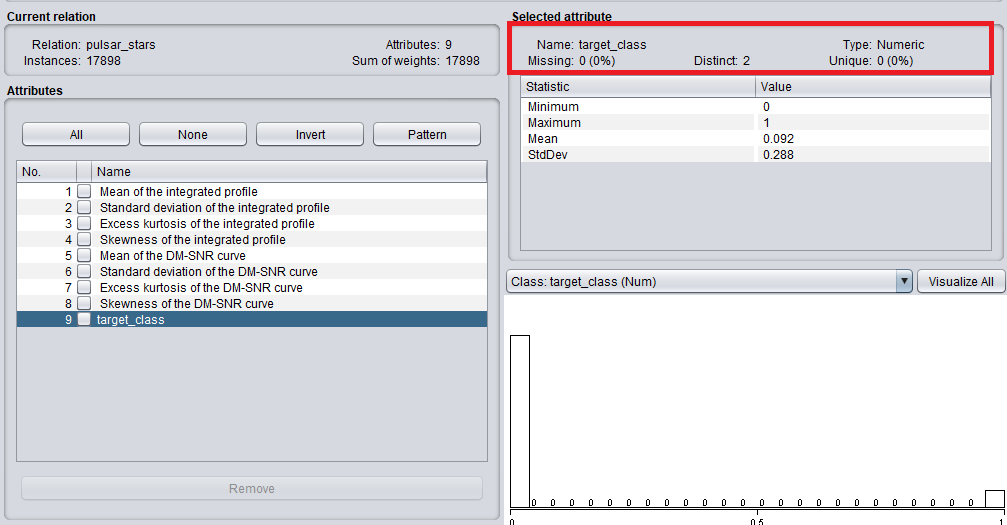
## 7. Preprocessing – 10%

In this section you should

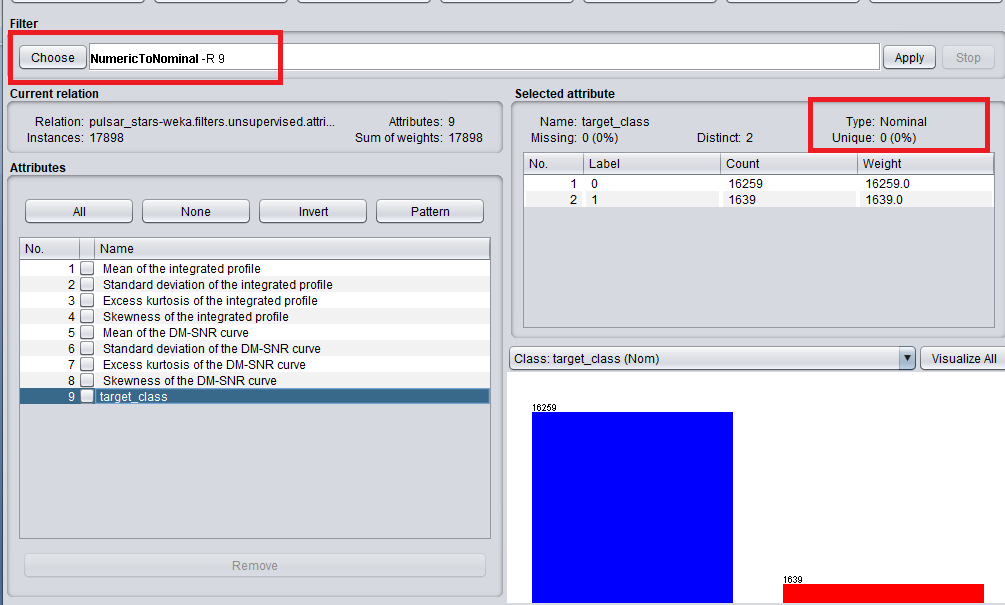
1. Identify the set of preprocessing techniques that can be applied to your data and clearly indicate which techniques are appropriate and which ones are not.
2. Provide evidence through screenshot of the effects of preprocessing the data along with a short explanation.
3. Generate a file called dataset.arff which is the outcome of the preprocessing.

**Preprocessing**

**In order to classify the cluster by classes, the target class should be a nominal attribute. In our dataset chosen, the target attribute is numeric. Hence we convert in to nominal using “Numerictonominal” filter in weka as shown in the below figures (a) and (b).**



**Figure (a)– Before converting target variable to nominal type.**



**Figure (b) – After converting target variable to nominal.**

# Experiments

For each of the following 2 clustering techniques

1. Use dataset.aff as input. If adaptions are necessary clearly indicate them.
2. Write one or two paragraph analyzing the results of the clustering. Be sure to vary parameters at least 3 times in each case. Support this analysis with screenshots of the following
   1. The clusters and/or a visualization of the clusters
   2. The results of the clusters
   3. Any additional output of the clustering process
   4. Simple references to the notes or URL links to online resources complete with a sentence or two of explanation.
   5. Evaluate the clusters using the “classes to clusters evaluation”. A worked example may be found here <http://www.cs.ccsu.edu/~markov/ccsu_courses/datamining-ex3.html>

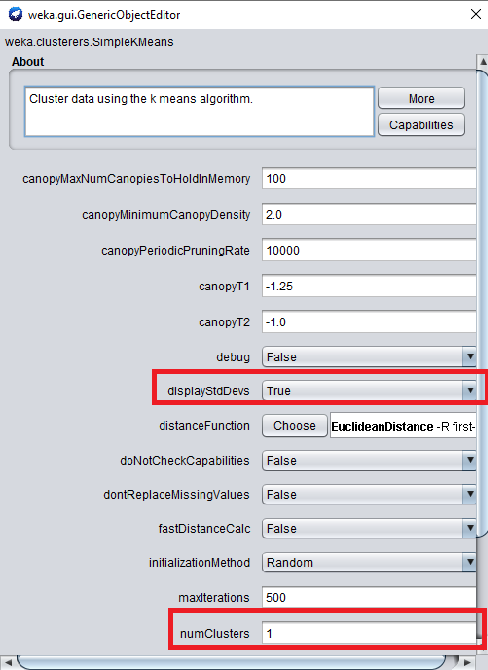
## 8. Clustering: K-Means – 15%

**Experiment 1: Clusters -1**

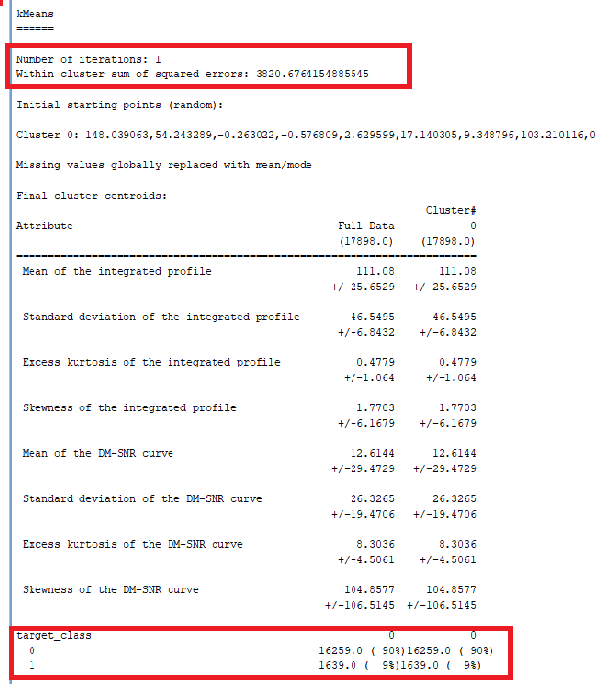
**For this experiment, we start with cluster 1 and by enabling displayStdDevs as shown in the below figure 1(a). We use Euclidean distance for clustering in this experiment**

**The result of this trial is as shown in the figure 1(b). As shown in the figure, kmeans performed one iteration for this clustering. This is because of the number of clusters chosen i.e., 1. Also, the sum squared errors within the cluster is too high for clustering with the selected parameters.**

**We can see that the target class 1 also has been added in cluster 0.**

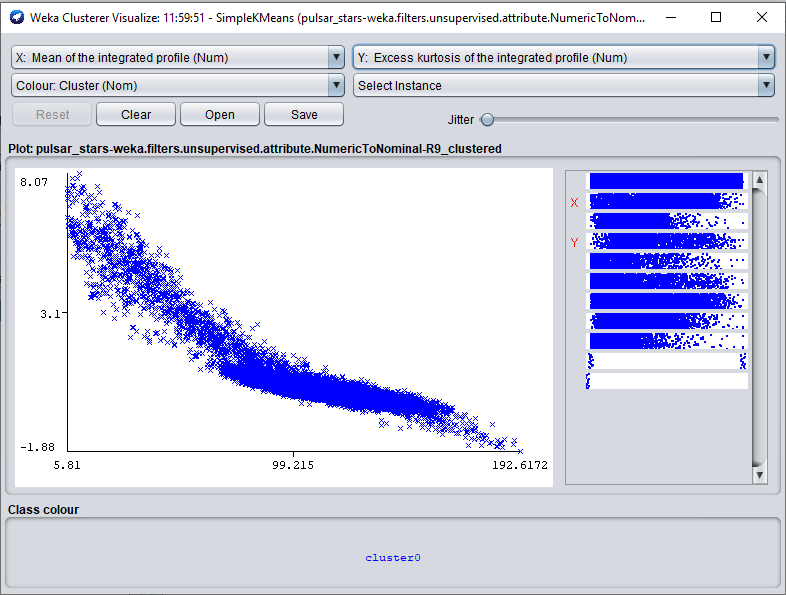


**Figure 1(a) – parameter setting**



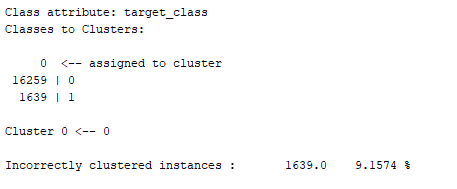
**Figure 1(b) – result of the experiment.**

**The below figure 1(C) gives the visualization of cluster. Here, we check the cluster by plotting mean of the integrated profile vs Excess kurtosis of the integrated profile as shown below.**

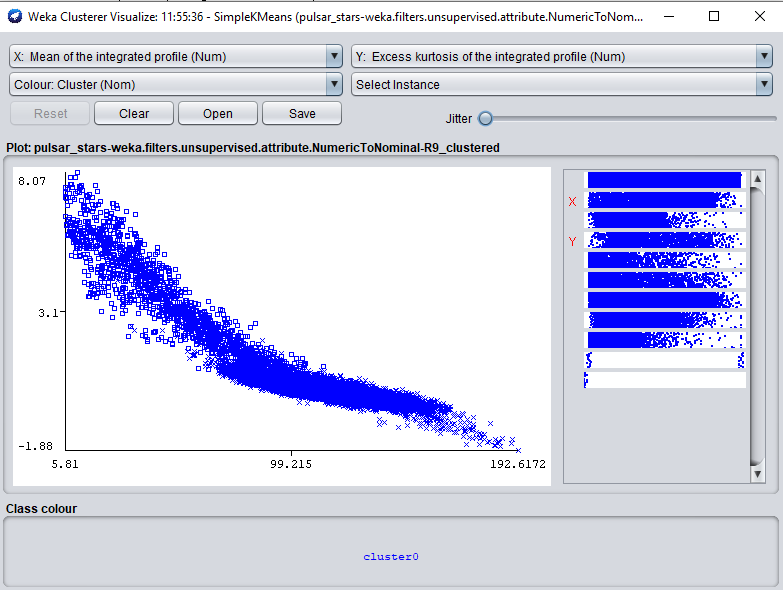


**Figure 1(c) – Visualization of the cluster.**

**Evaluating cluster using “classes to clusters evaluation” as shown in the below figure 1(d). We can from the figure that approximately 9% of the classes have been classified incorrectly into cluster 0. All the incorrectly clustered instances are for class 1 i.e., pulsar star.**



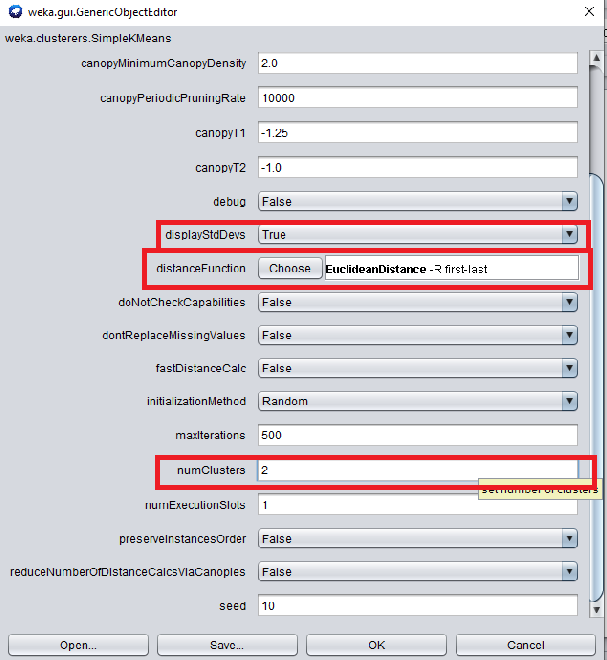
**Figure 1(d) – Evaluation using class attribute.**



**Figure 1(e) - Visualization after using classes to cluster evaluation.**

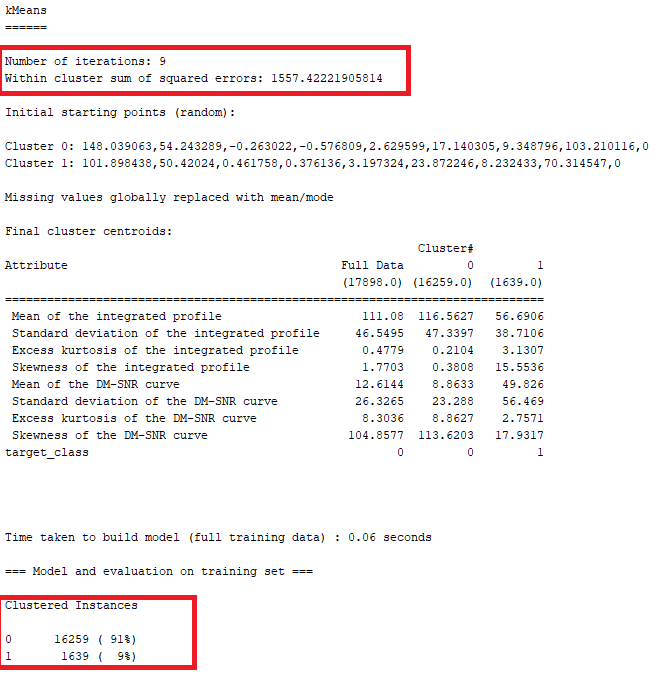
**Experiment 2: number of clusters = 2**

**For this experiment, we start with cluster 2 and by enabling displayStdDevs as shown in the below figure 2(a). We use Euclidean distance for clustering in this experiment**



**Figure 2(a) – parameter setting**

**The below figure 2(b) shows the result of the clustering algorithm for the above parameter setting.**



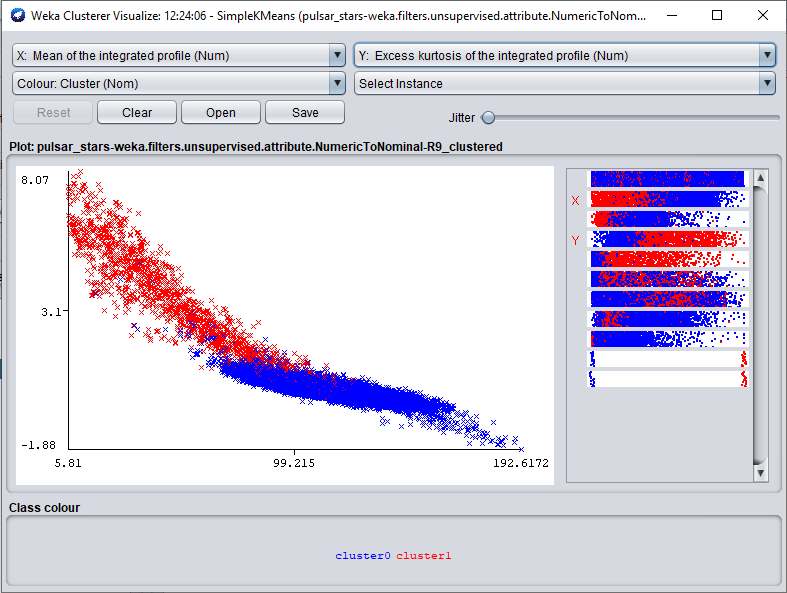
**Figure 2(b) – Result of K-means clustering for 2 clusters**

**From the above figure we can infer that k-means has performed 9 iterations to achieve these two clusters. Also, we can see that the sum squared errors for the above is less compared to the previous clustering model.**

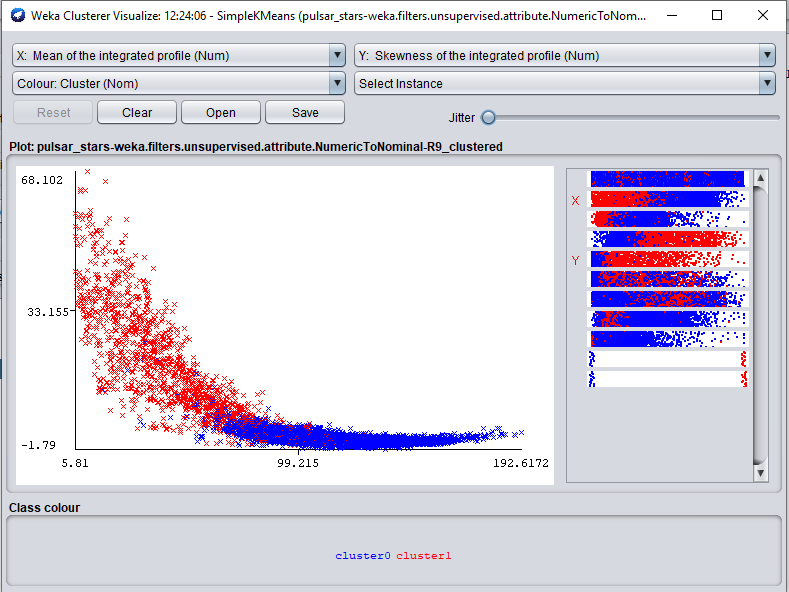
**Cluster 0 contains 91% of the instances and cluster 1 contains 9% of the total instances.**

**Visualization**

**The below figure 2(C) gives the visualization of cluster. Here, we check the cluster by plotting *mean of the integrated profile vs Excess kurtosis of the integrated profile* as shown below. We can see that there is certain degree of distinction among the clusters**

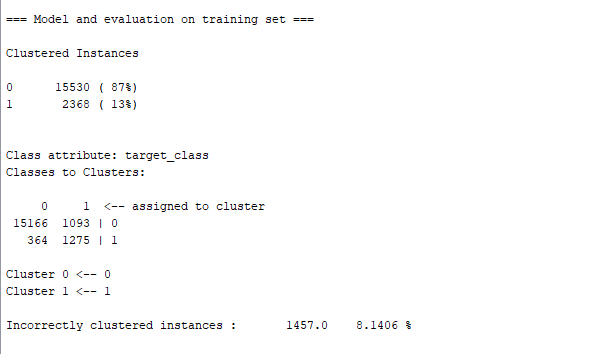


**Figure 2(c) – *mean of the integrated profile vs Excess kurtosis of the int profile***

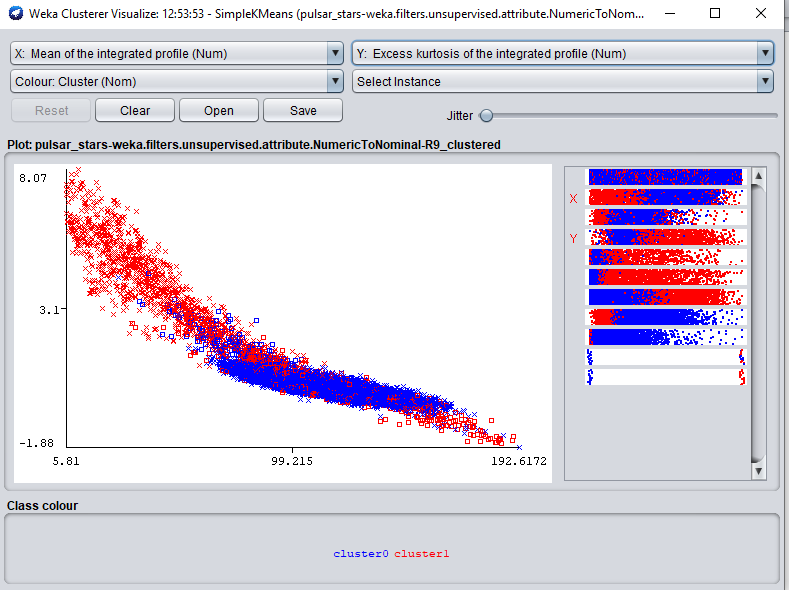


**Figure 2(d) - *mean of the integrated profile vs Skewness of the int profile***

**Evaluating cluster using “classes to clusters evaluation” as shown in the below figure 2(e). We can see that this model still has some incorrect clustered instance. But, it performs better than our previous model. We can see that 1093 instances have been incorrectly assigned to cluster 0 and 364 instances incorrectly assigned to cluster 1.**



**Figure 2(e) – Evaluating model using classes to cluster evaluation**

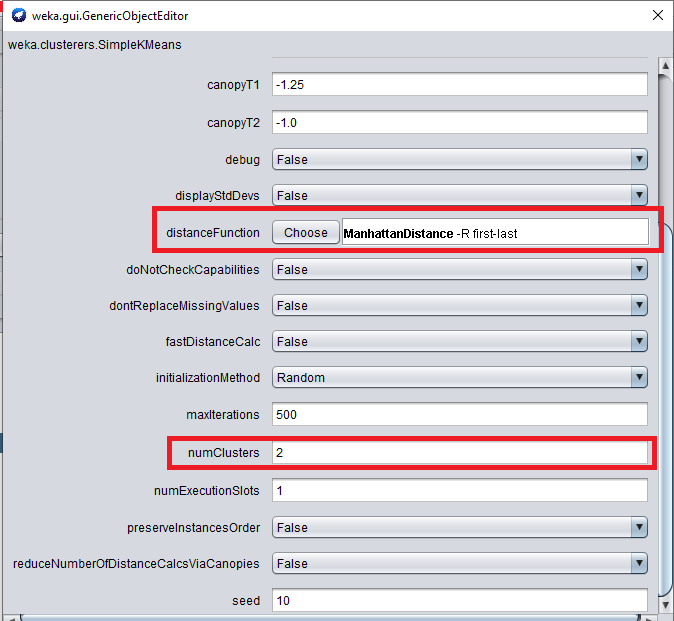


**Figure 2(f) – Visualization after using classes to cluster evaluation.**

**From the above we can see that some of the instances have wrongly clustered into cluster 0 and cluster 1.**

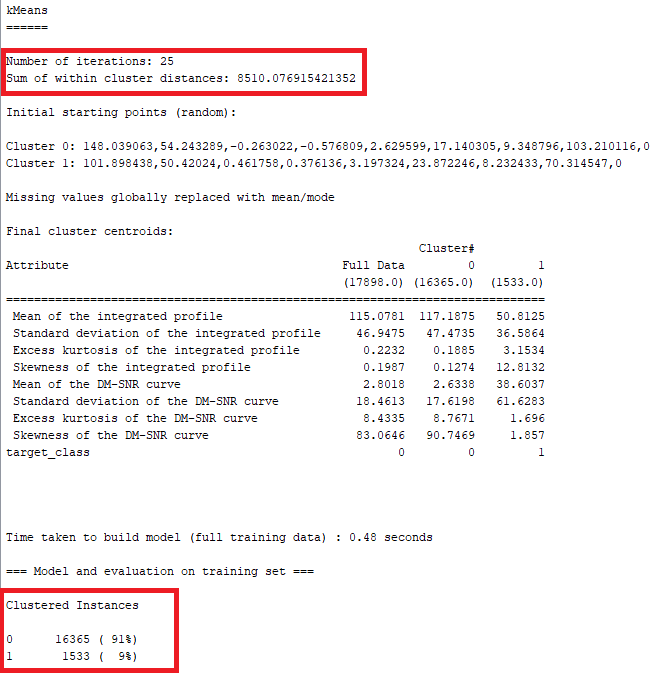
**Experiment 3: number of clusters =2 and Manhattan distance**

**For this experiment, we create two clusters as shown in the below figure 2(a). We use Manhattan distance for clustering in this experiment**



**Figure 3(a) – Manhattan distance parameter setting.**

**The below figure 3(b) shows the result of the clustering algorithm for the above parameter setting.**



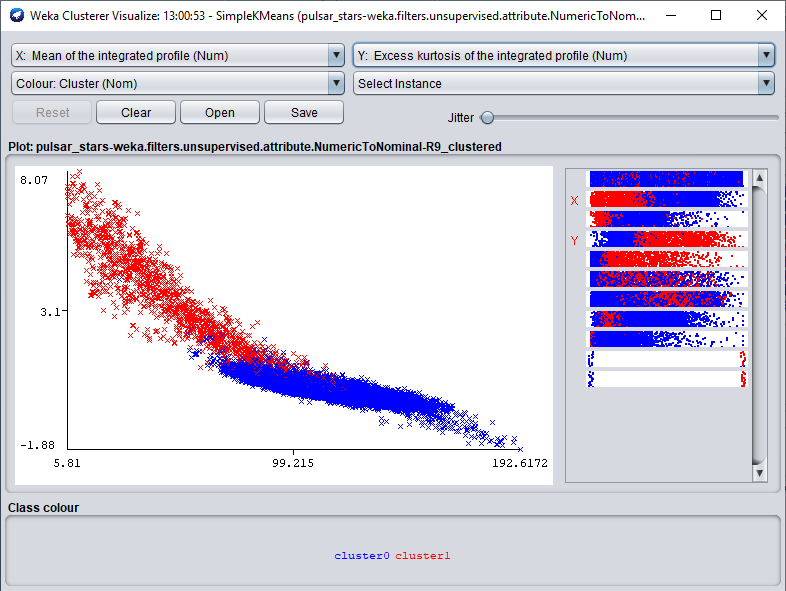
**Figure 3(b) – Result of K-means clustering for 2 clusters using Manhattan distance**

**From the above figure we can infer that k-means has performed 25 iterations to achieve these two clusters. Also, we can see that the sum squared errors for the above is very high compared to the previous clustering models.**

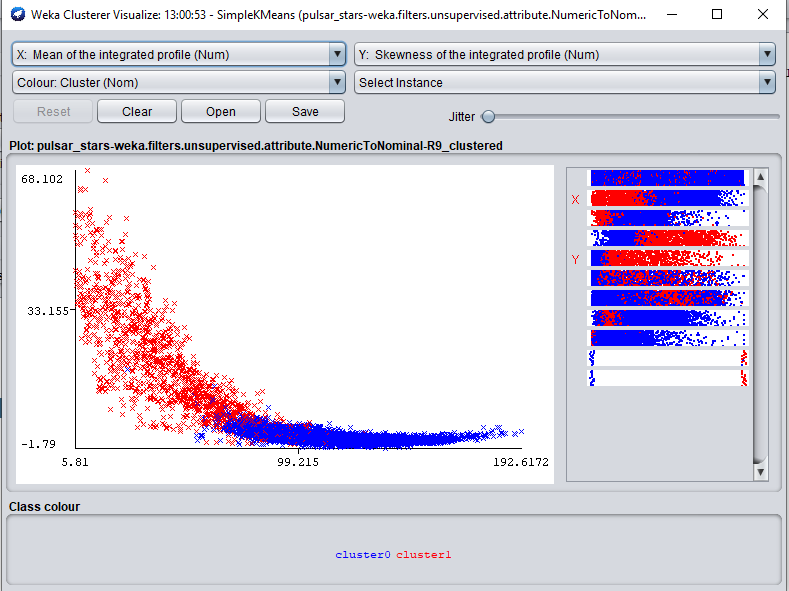
**Cluster 0 contains 91% of the instances and cluster 1 contains 9% of the total instances. However, we see that more number of instances have been assigned to cluster 0 compared to Euclidean distance.**

**Visualization**

**The below figure 3(C) gives the visualization of cluster. Here, we check the cluster by plotting mean of the integrated profile vs Excess kurtosis of the integrated profile as shown below. We can see that there is certain degree of distinction among the clusters**

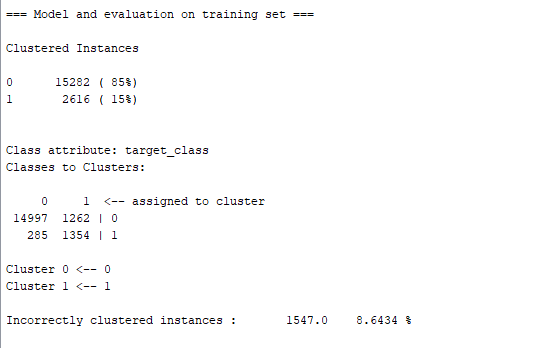


**Figure 3(c) – *mean of the integrated profile vs Excess kurtosis of the int profile***

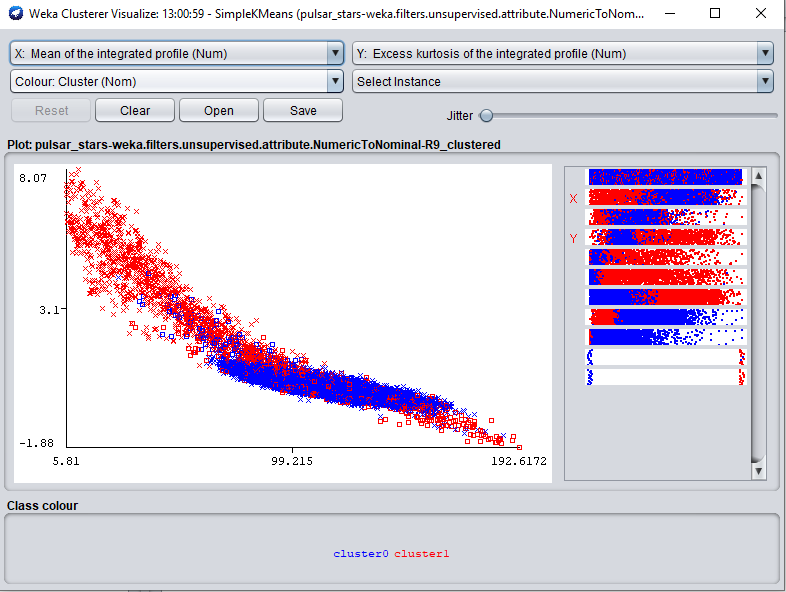


**Figure 3(d) - *mean of the integrated profile vs Skewness of the int profile***

**Evaluating cluster using “classes to clusters evaluation” as shown in the below figure 3(e). We can see that this model still has some incorrect clustered instance i.e., 1547 incorrectly clustered. We can see that 1262 instances have been incorrectly assigned to cluster 0 and 285 instances incorrectly assigned to cluster 1. This model performs reasonable better in clustering class 1 instances.**



**Figure 3(e) – Evaluating model using classes to cluster evaluation**

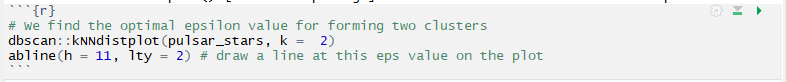


**Figure 3(f) – Visualization after using classes to cluster evaluation.**

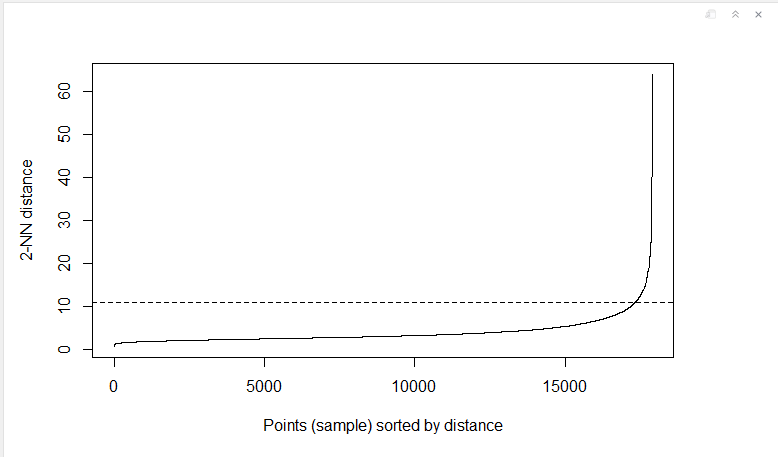
**From the above we can see that some of the instances have wrongly clustered into cluster 0 and cluster 1.**

## 9. Clustering: DBSCAN – 10%

**First, we determine the optimum epsilon value for the pulsar star dataset as shown below in figure (a).**



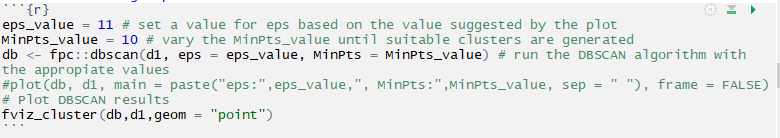
**Figure (a) – Code to find the optimal epsilon value**



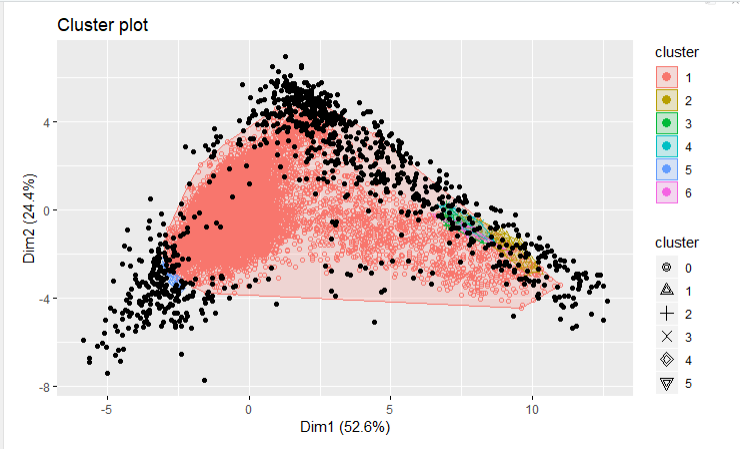
**Figure (b) – Representation of optimal epsilon value**

**From the above figure we can see that nearly 16k instances have their neighbors within a maximum distance of 11. Hence we take 11 as our epsilon value and perform the experiment by varying the minimum points.**

**Experiment 1: At low minimum points i.e., epsilon=11 and min pts =10.**



**Figure 1(a) – Code for eps=11 and min pts=10**

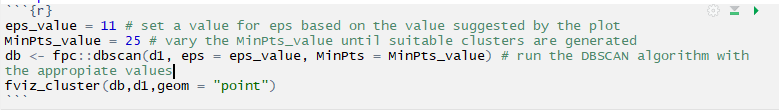


**Figure 1(b) – Plot for eps=11 and min pts=10**

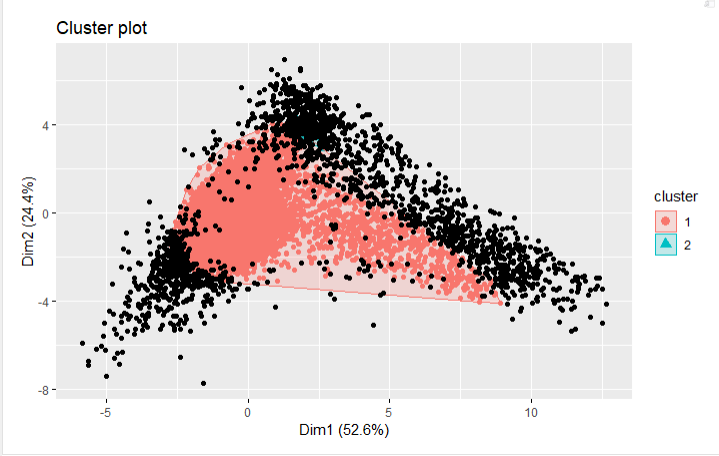
**From the above figure 1(b), we can see that at low minimum points there are 6 clusters formed due to less number of points required to form the cluster.**

**Hence in our next experiment we increase the minimum number of points in a way that forms less number of clusters.**

**Experiment 2: epsilon=11 and minimum points = 25**



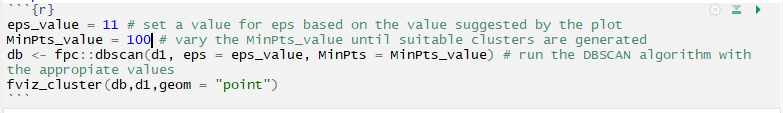
**Figure 2(a) – Code for eps=11 and min pts=25**



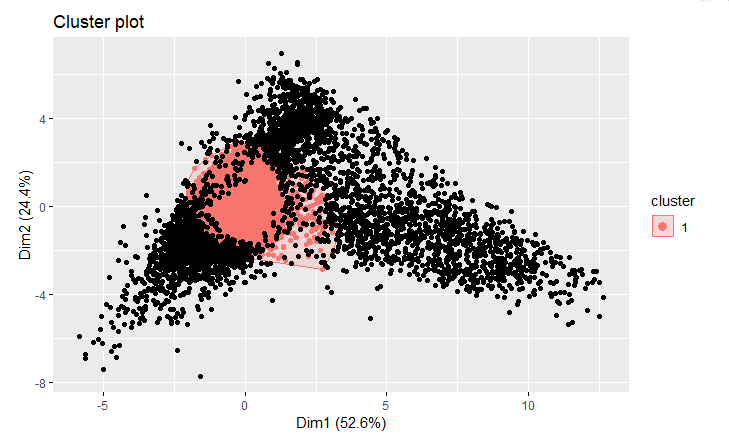
**Figure 2(b) – Plot for eps=11 and min pts=25**

**From the above figure 2(b), we can see that at min pts =25, there are two clusters that are formed. However, the second cluster is not dense enough to be distinctly visible. Also, we can see that there is large number of noise points. We increase our minpts further and check what happens.**

**Experiment 3: epsilon = 11 and min pts = 100**



**Figure 3(a) – Code for eps=11 and min pts =100**

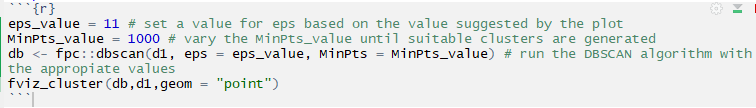


**Figure 3(b) – Plot for eps=11 and min pts=100**

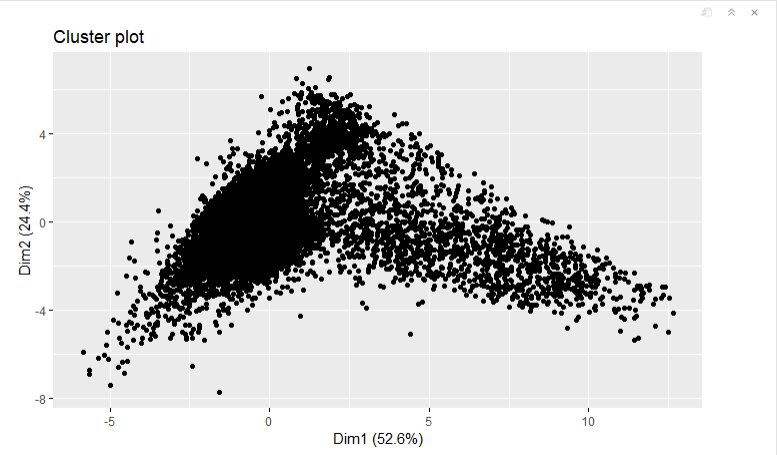
**In the above graph only one cluster has been formed. This is because of the large minimum points required to form the core point.**

**In our next experiment, we will see how the graph varies at extremely high minimum points.**

**Experiment 4: = 11 and min pts = 1000**



**Figure 4(a) – Code for eps=11 and min pts =1000**



**Figure 4(b)- Plot for eps=11 and min pts=1000**

**From the above figure we can see that no cluster has been formed due to the very large number of points required to form a core point. Hence all the points are considered as noise in the above graph.**

## 10. References – 5%

* Ron Kohavi, ["Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid"](http://robotics.stanford.edu/~ronnyk/nbtree.pdf), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996. (PDF)
* <https://www.kaggle.com/ipbyrne/income-prediction-84-369-accuracy>
* <https://archive.ics.uci.edu/ml/datasets/congressional+voting+records>
* Schlimmer, J. C. (1987). Concept acquisition through representational adjustment. Doctoral dissertation, Department of Information and Computer Science, University of California, Irvine, CA.
* <https://www.kaggle.com/pavanraj159/predicting-pulsar-star-in-the-universe>

## 11. Appendices –0%