EX.NO: 01
Date:

CENTRAL TENDENCY AND DATA DISPERSION MEASURES USING R-TOOL

PROBLEM STATEMENT:

Download the dataset from the UCI repository (or) any other appropriate website and perform (or) implement the central tendency measures.(mean, median, mode and midrange) and Data dispersion technique including summary.

DESCRIPTION:

This data comes from the 2010 census profile of general population and housing characteristics. Zip codes and limited to those that fall at least partially within LA city boundaries. The dataset will be updated after the next census in 2020.

CENTRAL TENDENCY:

i. **Mean** :The mean is the average of the numbers: a calculated "central" value of a set of numbers.

ii. **Median :**The median is a statistical term that is one way of finding the 'average' of a set of data points.

iii. **Mode :**The mode of a set of data values is the value that appears most often.

iv. **Summary :**A summary table stores data that has been aggregated in a way that answers a meancommon (or resource-intensive) business query.

```
> View(census)
> mean(census$Total.Males)
[1] 16391.56
> median(census$Total.Males)
[1] 15283
> mode(census$Total.Males)
[1] "numeric"
> summary(census$Total.Males)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0    9764   15283   16392   22220   52794
> IQR=(census$Total.Males)
>
```

MEASURES OF DISPERSION:

i. **Inter Quartile Range :**The interquartile range (IQR) is a measure of variability, based on dividing a data set into quartiles. Quartiles divide a rank-ordered data set into four equal parts.

ii. **Quartiles :**A quartile is a statistical term describing a division of observations into four defined intervals based upon the values of the data and how they compare to the entire set of observations.

iii. **Mid Range :**The arithmetic mean of the largest and the smallest values in a sample or other group.

iv. **Outlier :**An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.

a) Lower Fence : Q1 - 1.5*IQ
b) Upper Fence : Q3 + 1.5*IQ
c) Outlier Values

```
> IQR(census$Total.Males)
[1] 12456
> quantile(census$Total.Males,0.25)
   25%
9763.5
> quantile(census$Total.Males,0.75)
    75%
22219.5
> range(census$Total.Males)
[1]     0 52794
> mean(range(census$Total.Males))
[1] 26397
> Lf<-quantile(census$Total.Males,0.25)-1.5*(IQR(census$Total.Males))
> print(Lf)
    25%
-8920.5
```

```
> Lf<-quantile(census$Total.Males,0.25)-1.5*(IQR(census$Total.Males))
> print(Lf)
    25%
-8920.5
> Uf<-quantile(census$Total.Males,0.25)+1.5*(IQR(census$Total.Males))
> print(Uf)
    25%
28447.5
> outlier_values<-boxplot.stats(census$Total.Males)$out
> print(outlier_values)
 [1] 52794 43128 50658 45113 46321 52364 45229 52358 45786 42283 42564
>
```

RESULT:

Thus the central tendency and measures of dispersion have been executed successfully. The outlier values are from more than upper fence there are no lower fence values.

EX.NO: 02
Date :

PLOTTING GRAPHS USING R-TOOL

PROBLEM STATEMENT:

Plot the boxplot, histogram and scatterplot for the dataset which was taken in the previous exercise.

DESCRIPTION:

Consider a dataset travel.times.csv, where it contains the attributes are Date, StartTime, DayOfWeek , Goingto , Distance, MaxSpeed , AvgSpeed, AvgMovingSpeed, FuelEconomy, TotalTime, MovingTime, Take407All, Comments for plotting the graphs.

IMPLEMENTATION:

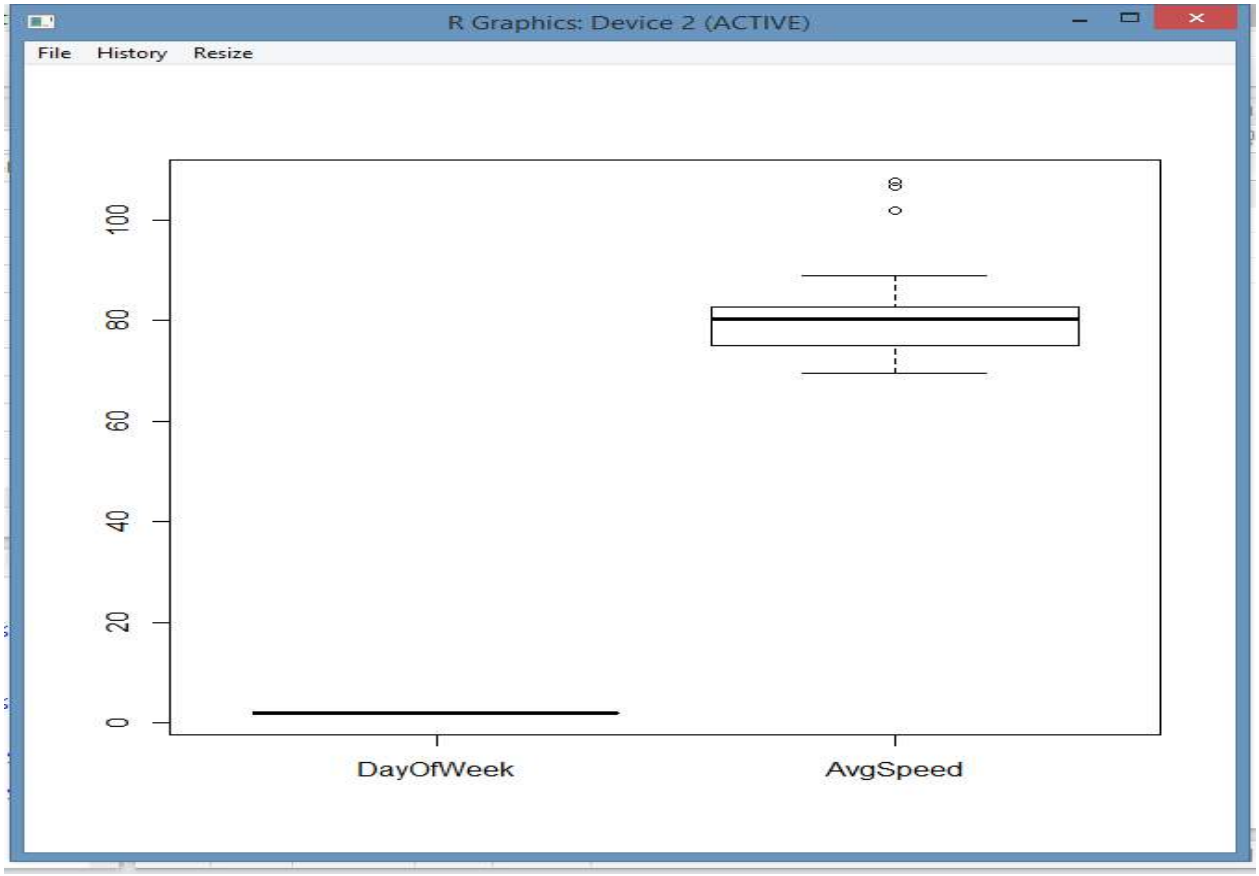
i. BoxPlot
ii. Histogram
iii. ScatterPlot

i. BOXPLOT :

A box plot is a graphical rendition of statistical data based on the minimum, first quartile, median, third quartile, and maximum. The term "box plot" comes from the fact that the graph looks like a rectangle with lines extending from the top and bottom. Because of the extending lines, this type of graph is sometimes called a box-and-whisker plot.Boxplot analysis made among the DayOfWeek and AvgSpeed.

- travel1<-travel.times[which(travel.times$DayOfWeek==”Friday”),names(travel.times)%in% c(“DayOfWeek”,”AvgSpeed”)]
- travel2<-travel.times[which(travel.times$DayOfWeek==”Monday”),names(travel.times)%in% c(“DayOfWeek”,”AvgSpeed”)]
- boxplot(travel1,travel2)

OUTPUT:

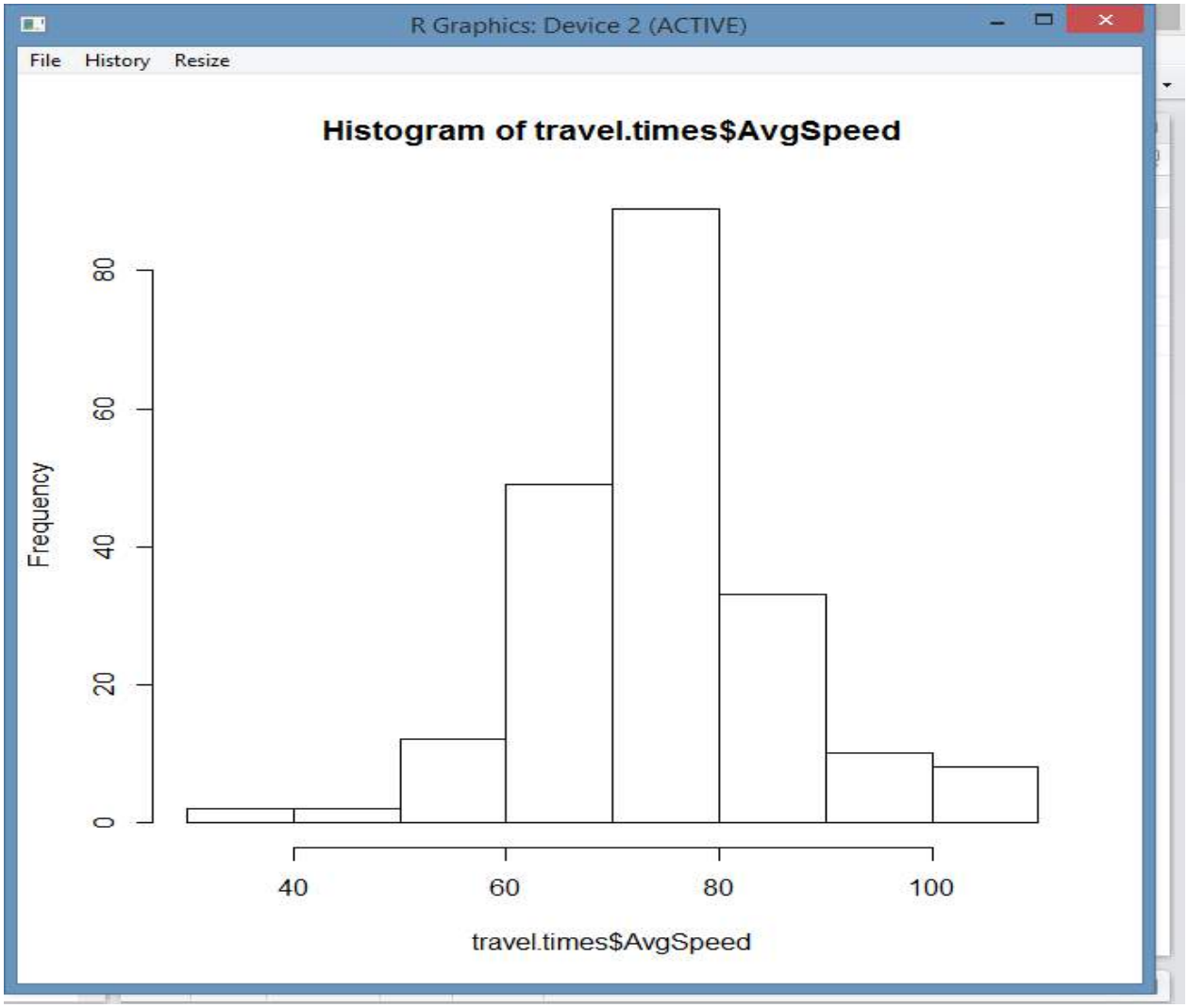


ii. HISTOGRAM:

A histogram is a display of statistical information that uses rectangles to show the frequency of data items in successive numerical intervals of equal size. In the most common form of histogram, the independent variable is plotted along the horizontal axis and the dependent variable is plotted along the vertical axis. The data appears as coloured or shaded rectangles of variable area.

- Hist(travel.times$AvgSpeed)

OUTPUT:

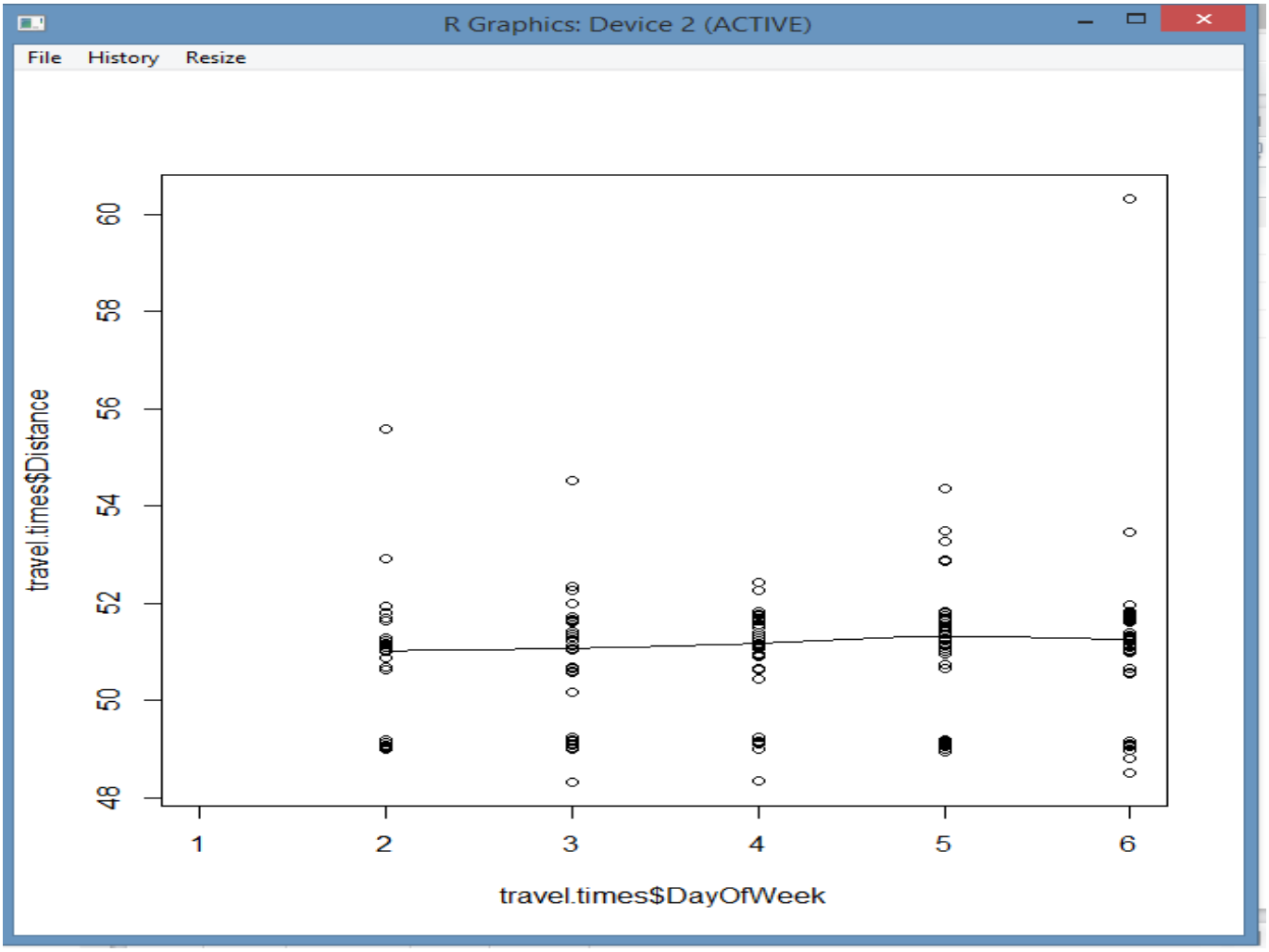


iii. SCATTERPLOT:

Scatter plots are important in statistics because they can show the extent of correlation, if any, between the values of observed quantities or phenomena (called variables). If no correlation exists between the variables, the points appear randomly scattered on the coordinate plane. Scatterplot is made between DayOfWeek and distance of the dataset travel.times.csv

- Scatter.smooth(travel.times$DayOfWeek,travel.times$Distance)

OUTPUT:



RESULT:

Thus, the plotting of graphs like boxplot, histogram and scatterplot for the given dataset has been successfully completed.

EX.NO : 03
Date :

PERFORM CORRELATION ANALYSIS AND NORMALIZATION USING R-TOOL

PROBLEM STATEMENT :

Perform the correlation analysis for the numerical attribute using pearson coefficient and for categorical attribute using chi-square and also, perform the normalization technique using min, max, z score and decimal scaling for the given data frames of particular dataset.

DESCRIPTION :

A dataset of name diabetes.csv is given for the correlation analysis, to calculate or to correlate between Age and Insulin and the same dataset for the performance of normalization technique.

- **CORRELATION ANALYSIS:**

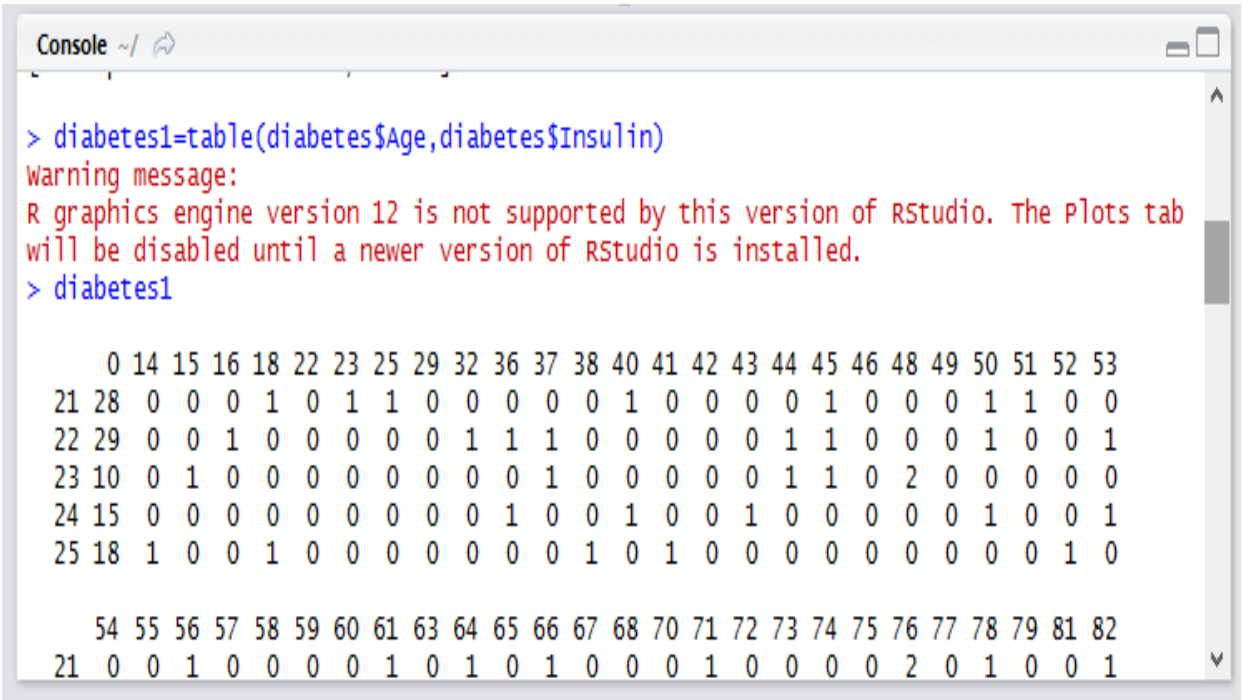
  STEPS INVOLVED:

  i. Create a new table with required dataframes.
  ii. After that apply the formula or query for the chi-square test.

  QUERIES:

  - diabetes1<-table(diabetes$Age,diabetes$Insulin)
  - diabetes1
  - chi sq.test(diabetes1)

OUTPUT:



```
Console ~/
 22   0   0   0   0   0   0   0   0   0   0
 23   1   0   0   0   0   0   0   0   1   0   0
 24   0   0   0   0   0   0   0   0   0   0   0
 25   0   0   0   1   0   0   0   0   0   0   0
 [ reached getOption("max.print") -- omitted 47 rows ]
> chisq.test(diabetes1)

        Pearson's Chi-squared test

data:  diabetes1
X-squared = 7561.7, df = 9435, p-value = 1

Warning message:
In chisq.test(diabetes1) : Chi-squared approximation may be incorrect
>
```

- **NORMALIZATION :**

  i. **MIN MAX NORMALIZATION:**

  - A<- c(diabetes$Age)
  - Mean<-mean(A)
  - Minimum<-min(diabetes$Age)
  - Maximum<-max(diabetes$Age)
  - MinMax<- (A-Minimum)/(Maximum-Minimum)
  - MinMax

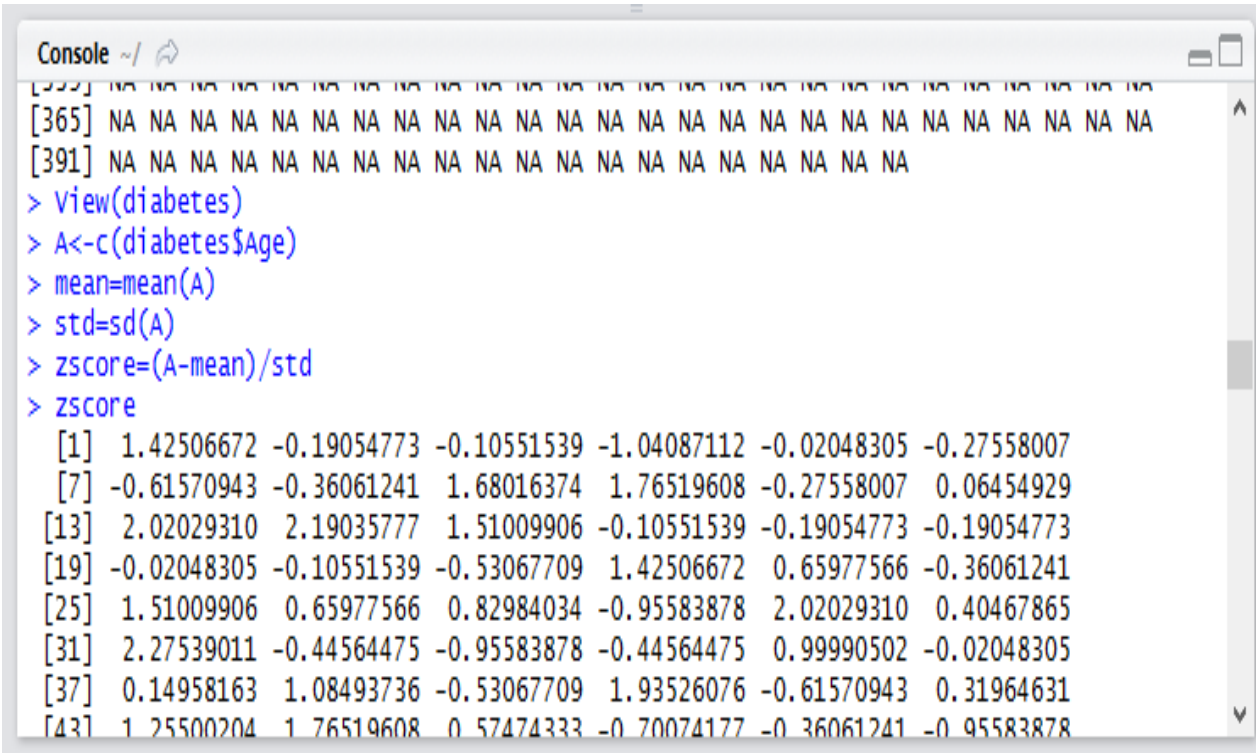
OUTPUT:

```
Console ~/
[745]  0.48971099  1.08493736 -0.53067709 -0.10551539  0.23461397  1.42506672
[751] -0.95583878 -0.44564475 -0.70074177 -0.61570943  0.99990502  0.31964631
[757]  0.48971099  1.59513140 -0.61570943  2.78558415 -0.95583878  0.82984034
[763] -0.02048305  2.53048713 -0.53067709 -0.27558007  1.16996970 -0.87080644
> minimum=min(diabetes$Age)
> maximum=max(diabetes$Age)
> minmax=(A-minimum)/(maximum-minimum)
> minmax
  [1] 0.48333333 0.16666667 0.18333333 0.00000000 0.20000000 0.15000000 0.08333333
  [8] 0.13333333 0.53333333 0.55000000 0.15000000 0.21666667 0.60000000 0.63333333
 [15] 0.50000000 0.18333333 0.16666667 0.16666667 0.20000000 0.18333333 0.10000000
 [22] 0.48333333 0.33333333 0.13333333 0.50000000 0.33333333 0.36666667 0.01666667
 [29] 0.60000000 0.28333333 0.65000000 0.11666667 0.01666667 0.11666667 0.40000000
 [36] 0.20000000 0.23333333 0.41666667 0.10000000 0.58333333 0.08333333 0.26666667
 [43] 0.45000000 0.55000000 0.31666667 0.06666667 0.13333333 0.01666667 0.16666667
 [50] 0.05000000 0.01666667 0.08333333 0.15000000 0.61666667 0.35000000 0.00000000
```

ii. Z SCORE NORMALIZATION :

- A<- c(diabetes$Age)
- Mean<- mean(A)
- Std<- sd(A)
- Zscore<- (A-Mean)/Std
- Zscore

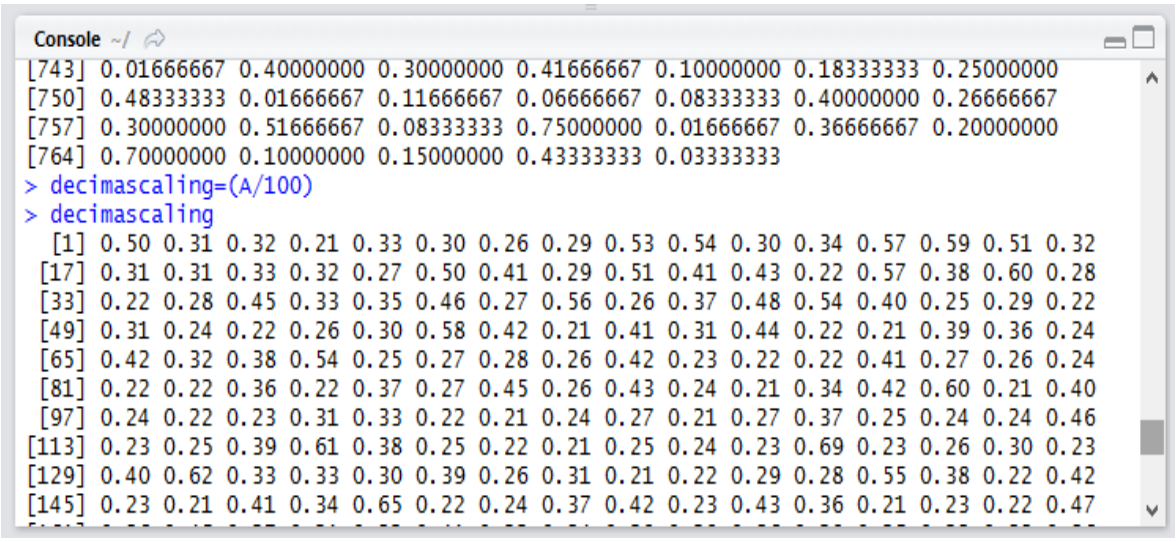
OUTPUT:



iii. DECIMAL SCALING NORMALIZATION :

- Decimalscaling =(A/100)
- Decimalscaling

OUTPUT:



RESULT:

Thus, the correlation analysis and normalization for the given dataset has been successfully executed and observed.

EX.No: 04
Date :

REGRESSION ANALYSIS USING R TOOL

PROBLEM STATEMENT :

Perform the linear regression and multiple regression for the given dataset.

DESCRIPTION :

Consider a dataset of diabetes.csv with the attributes pregnancies, Glucose, BloodPressure, SkinThickness, BMI, Diabetes, Age, Outcome for the analysis. There will be linear regression analysis between Age and BloodPressure. Where, for the multiple regression, the analysis is between Age, BloodPressure, Glucose from the dataset.

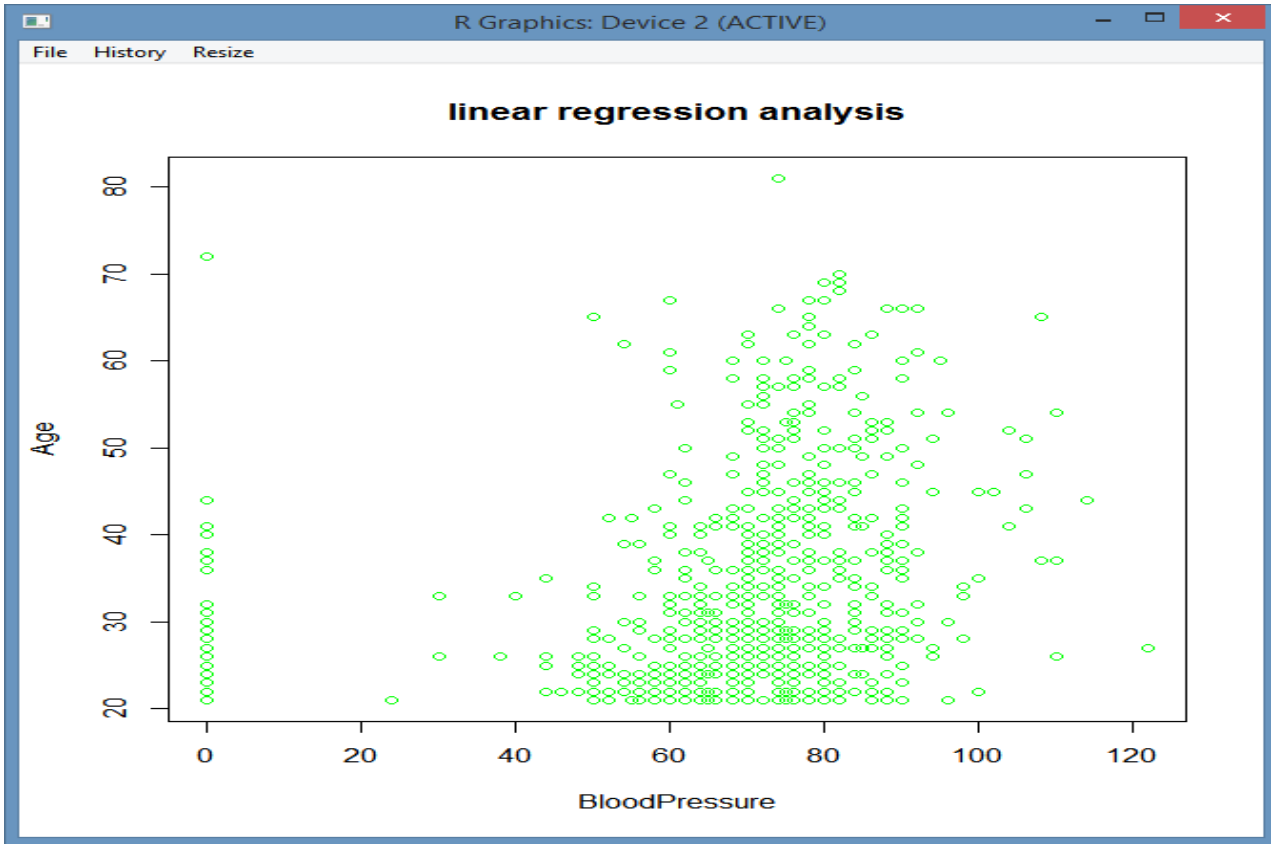
- **LINEAR REGRESSION :**

Linear regression is a kind of statistical analysis that attempts to show a relationship between two variables. Linear regression looks at various data points and plots a trend line. Linear regression can create a predictive model on apparently random data, showing trends in data, such as in cancer diagnoses or in stock prices.

QUERIES :

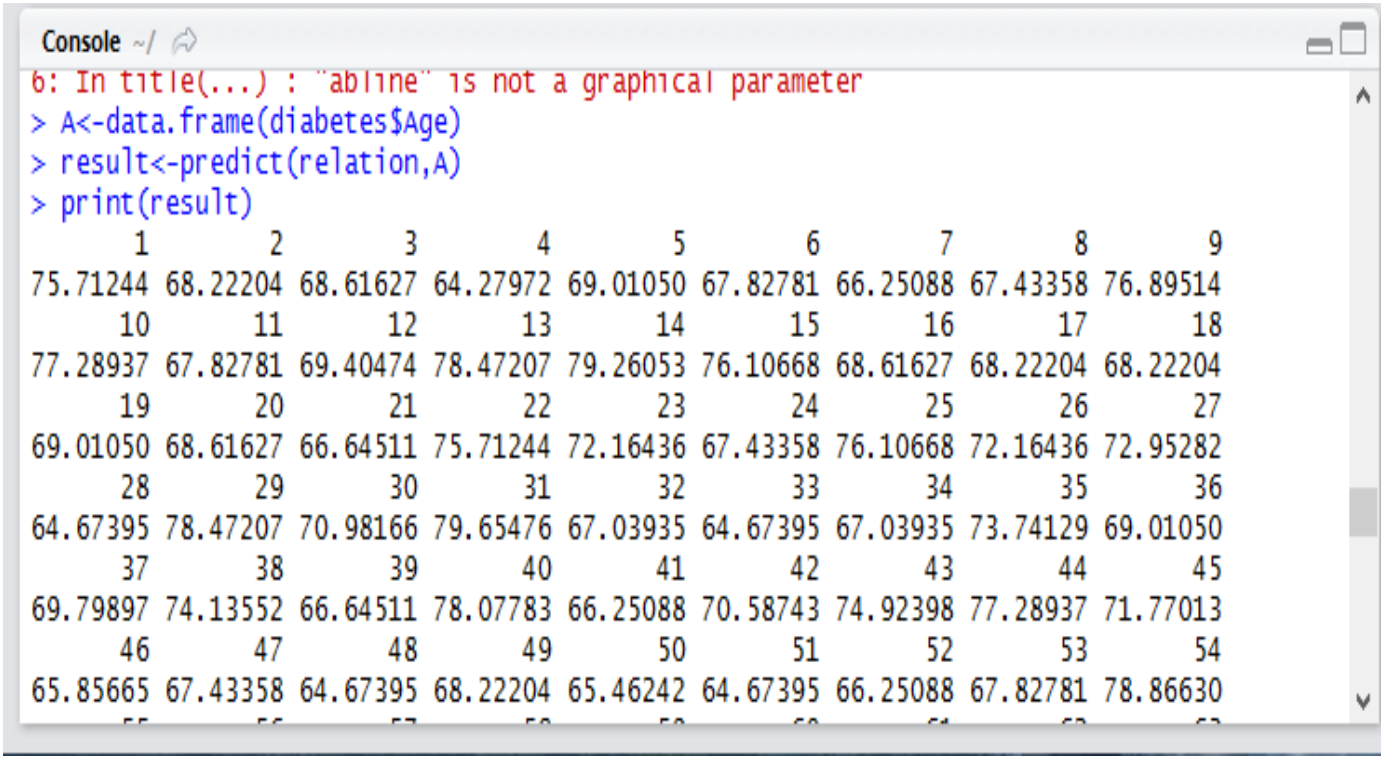
- Relation <- lm(diabetes$BloodPressure~diabetes$Age)
- Png<- (file="linear regression.png")
- Plot(diabetes$Age, diabetes$BloodPressure, col="green", main= " Linear Regression Analysis" , abline= (lm(diabetes$BloodPressure~ diabetes$Age)), xlab = "BloodPressure", ylanb= "Age")

OUTPUT:



- A<- data.frame(diabetes$Age)
- Result<- predict(relation, A)
- Print(Result)

OUTPUT :



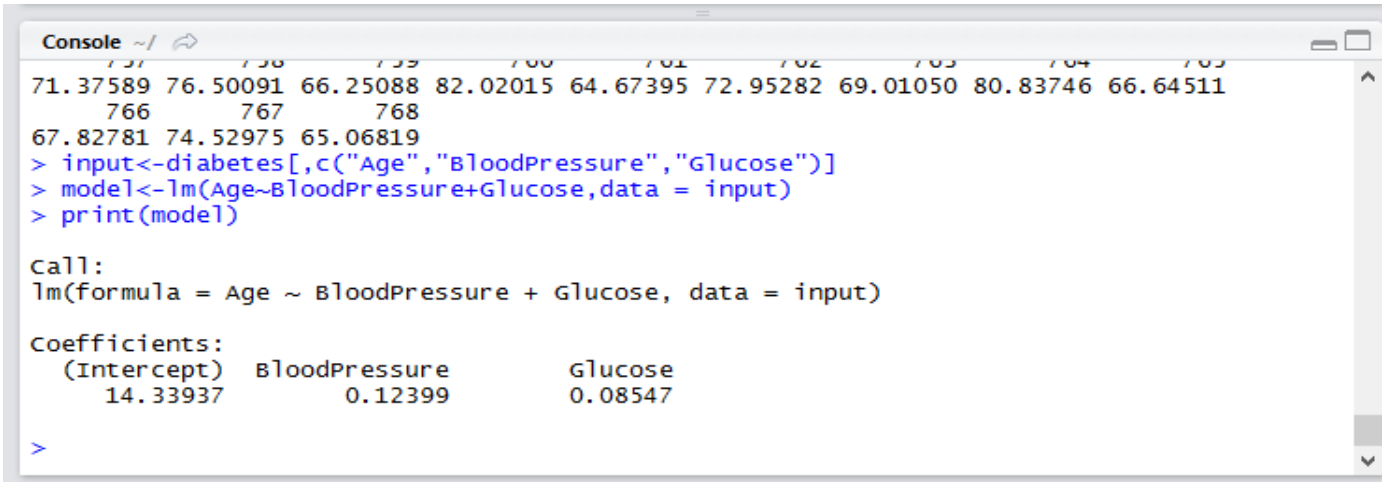
❖ **MULTIPLE REGRESSION :**

Multiple regression is a statistical tool used to derive the value of a criterion from several other independent, or predictor, variables. It is the simultaneous combination of multiple factors to assess how and to what extent they affect a certain outcome.

- **QUERIES :**

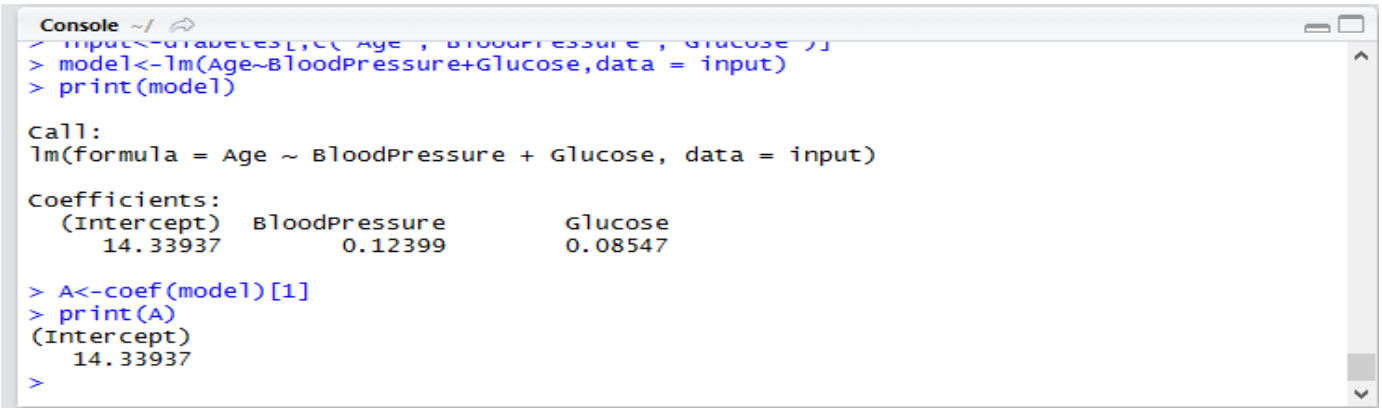
  - Input <- diabetes[,c(“Age”, “BloodPressure”, “Glucose”)]
  - Model <- lm(Age~ BloodPressure+Glucose,data=input)
  - Print(model)

OUTPUT:



- A<- coef(model)[1]
- Print(A)

OUTPUT:



- xBloodPressure<- coef(model)[2]
- yGlucose<- coef(model)[3]
- print(xBloodPressure)
- print(yGlucose)

OUTPUT:

- y = A+xBloodPressure + yGlucose
- print(y)

OUTPUT:

RESULT :

Thus, the linear regression and the multiple regression analysis for the given dataset has been successfully completed.

EX.No: 05
Date :

DATA PREPROCESSING AND ANALYSIS FOR DATASET USING WEKA

PROBLEM STATEMENT :

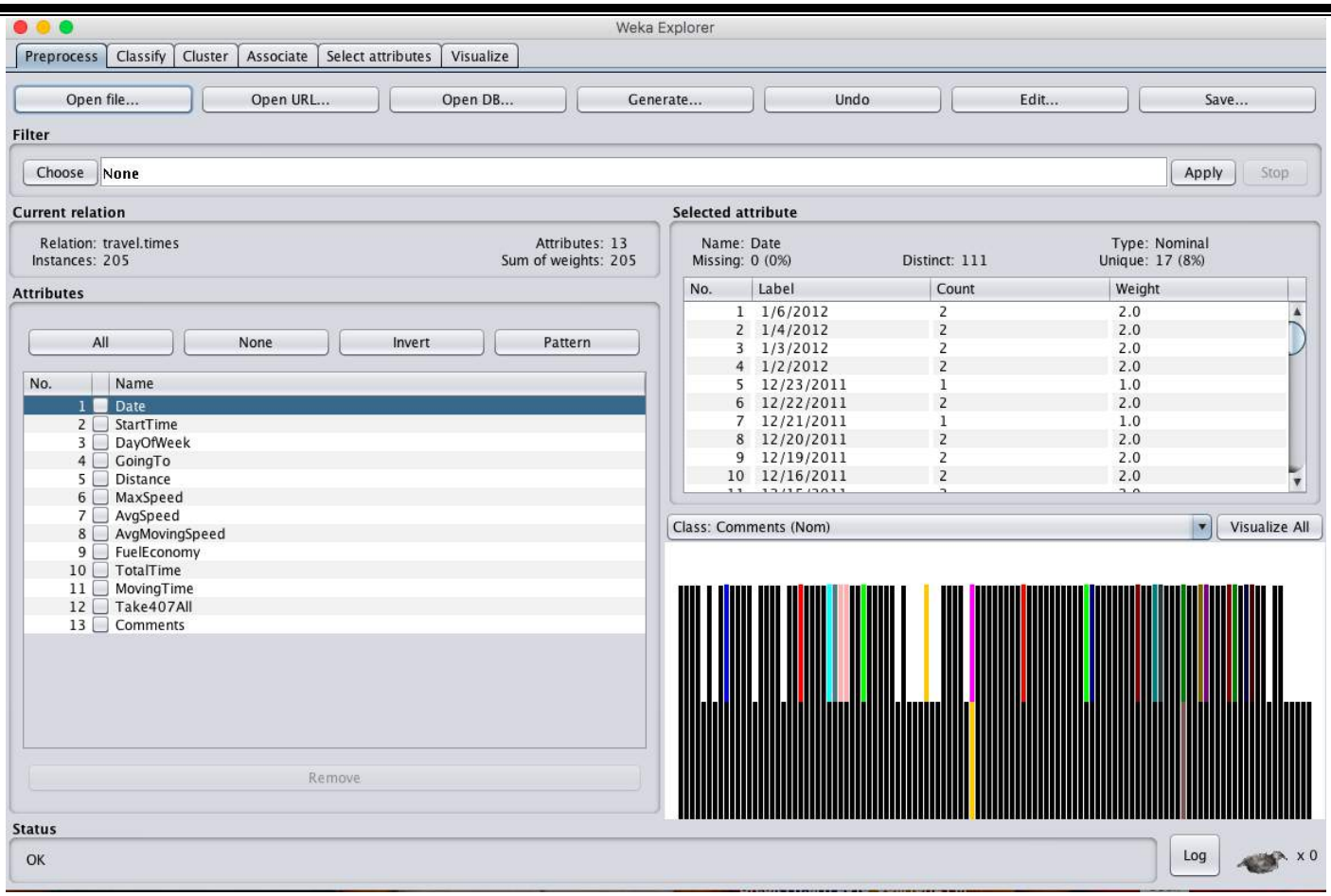
For each attribute in the dataset find the following information:

A. Attribute type
B. Percentage of missing values
C. Find the minimum, maximum, mean, Standard Deviation with numerical attributes.
D. Are there any records that have a values that no other record has ?
E. Write a note on class distribution for each of the attributes.
F. Apply attribute selection measures under filter supervised selection attribute.

DESCRIPTION :

Consider a dataset of traveltimes.csv file where it contains the columns of (or) attributes as Date, StartTime, DayOfWeek, GoingTo, Distance, MaxSpeed, AvgSpeed, AvgMovingSpeed, FuelEconomy, TotalTime, MovingTime, Take407All comments.

https://www.kaggle.com/datasets/minnieliang/travel-times-data?resource=download



OBSERVATION :

A. ATTRIBUTE TYPE :

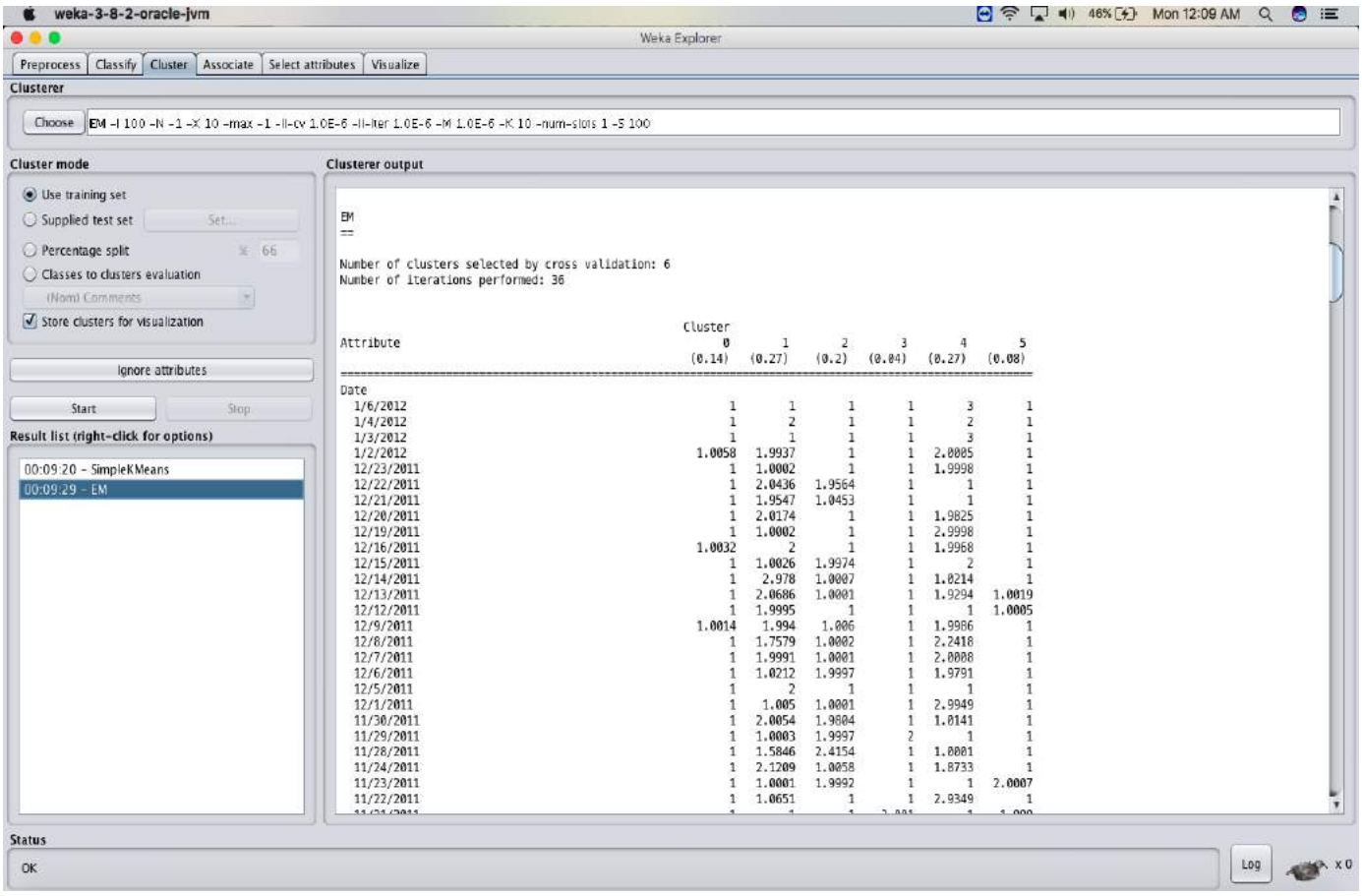
S.NO	ATTRIBUTE	TYPE
1.	Date	Nominal
2.	Start Time	Nominal
3.	Day Of Week	Nominal
4.	Going To	Nominal
5.	Distance	Numeric
6.	Max Speed	Numeric
7.	Avg Speed	Numeric

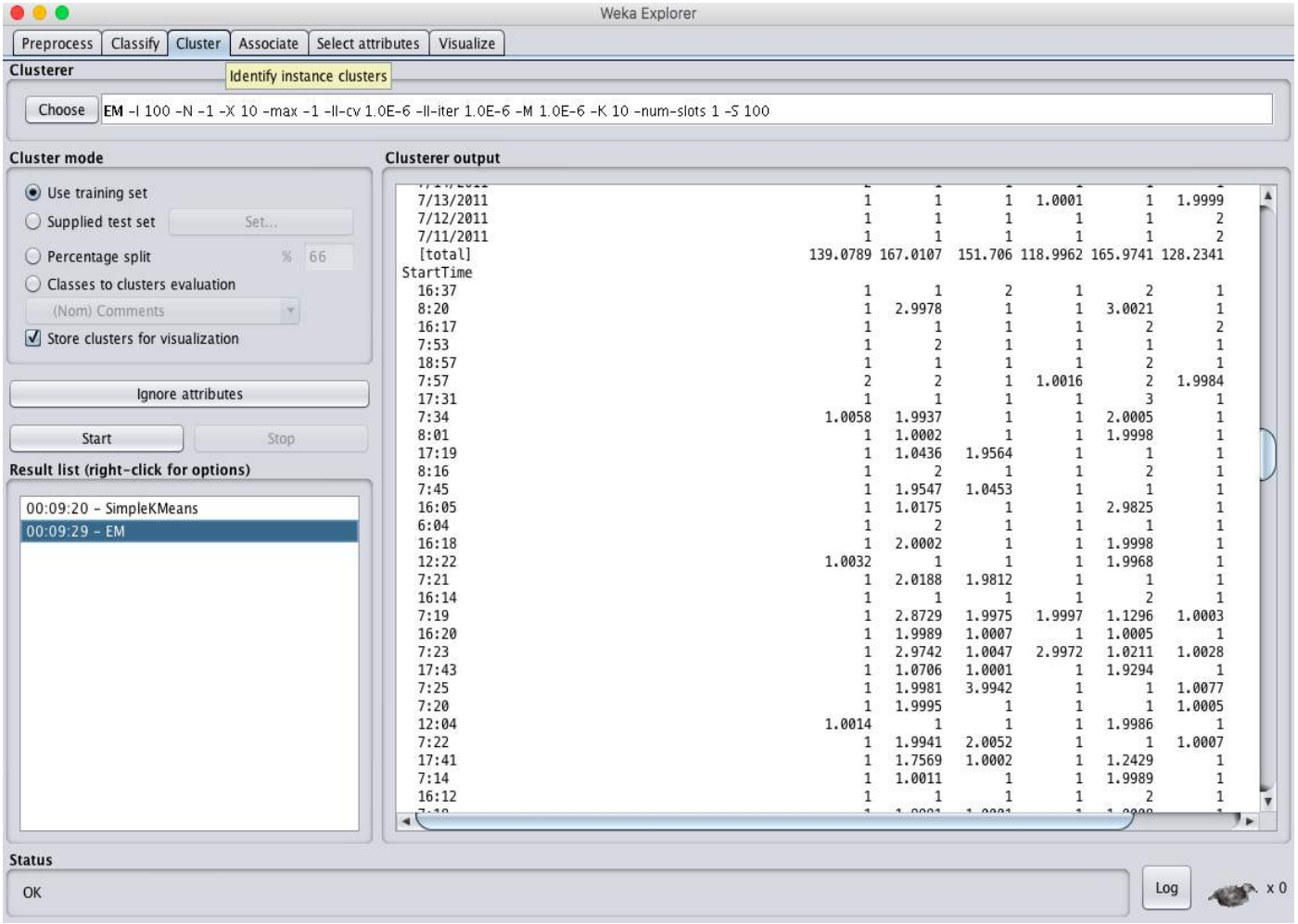
8.	Avg Moving Speed	Numeric
9.	Fuel Economy	Nominal
10.	Total Time	Numeric
11.	Moving Time	Numeric
12.	Comments	Nominal
13.	Take 407 All	Nominal

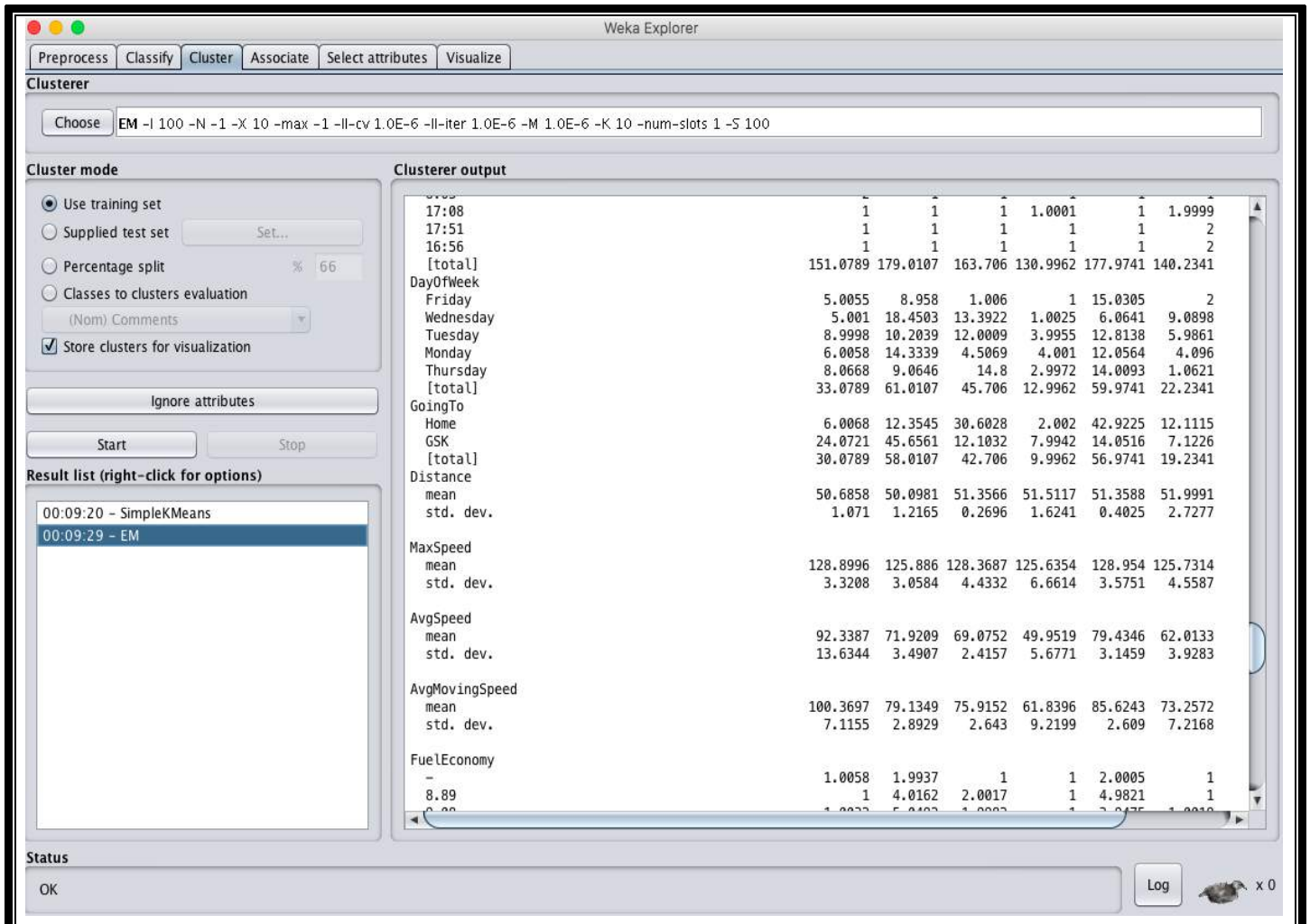
B. PERCENTAGE OF MISSING VALUES :

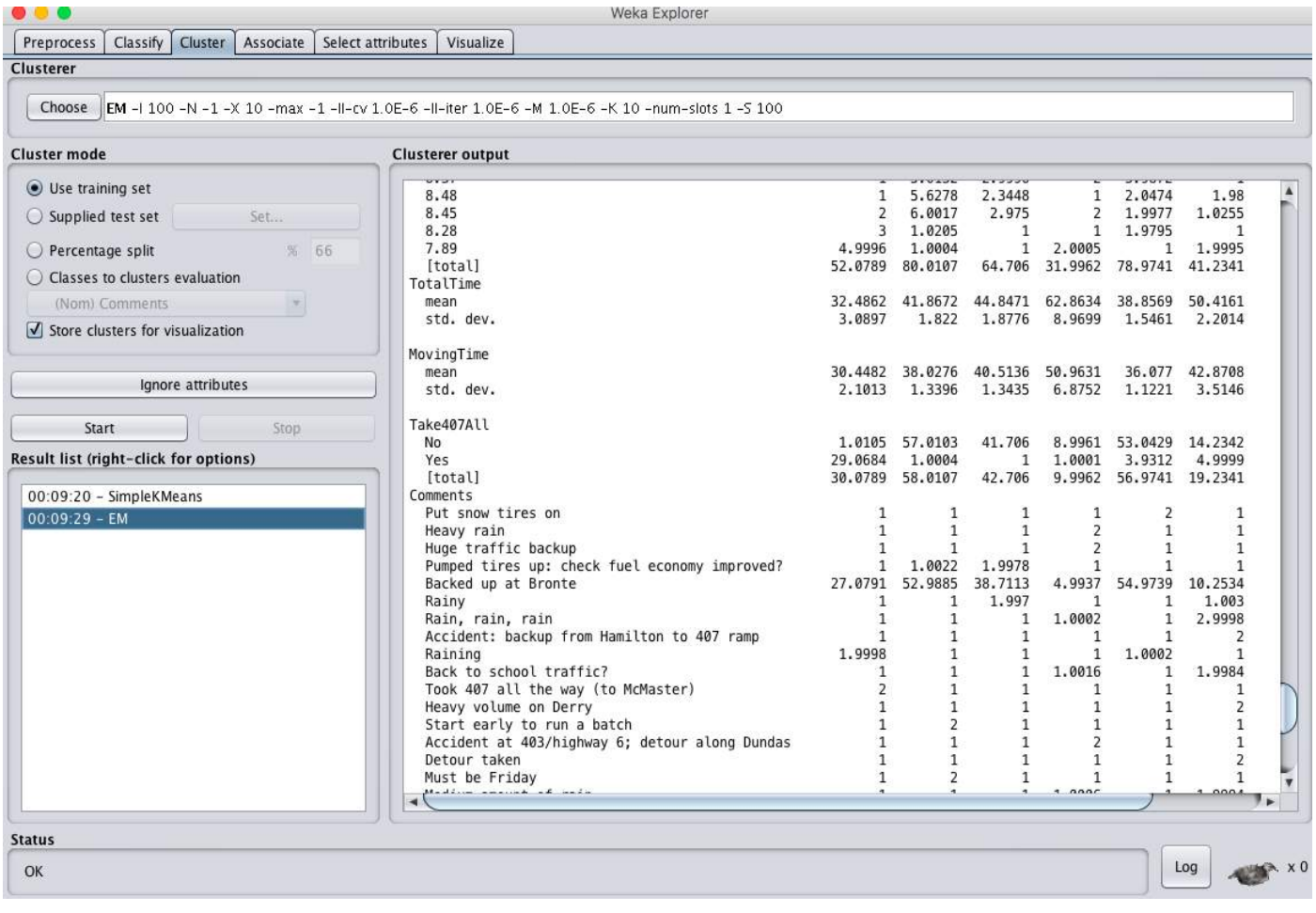
S.NO	ATTRIBUTE	Percentage Of Missing Values
1.	Date	0 %
2.	Start Time	0 %
3.	Day Of Week	0 %
4.	Going To	0 %
5.	Distance	0 %
6.	Max Speed	0 %
7.	Avg Speed	0 %
8.	Avg Moving Speed	0 %
9.	Fuel Economy	8 %
10.	Total Time	0 %
11.	Moving Time	0 %
12.	Comments	88 %
13.	Take 407 All	0 %

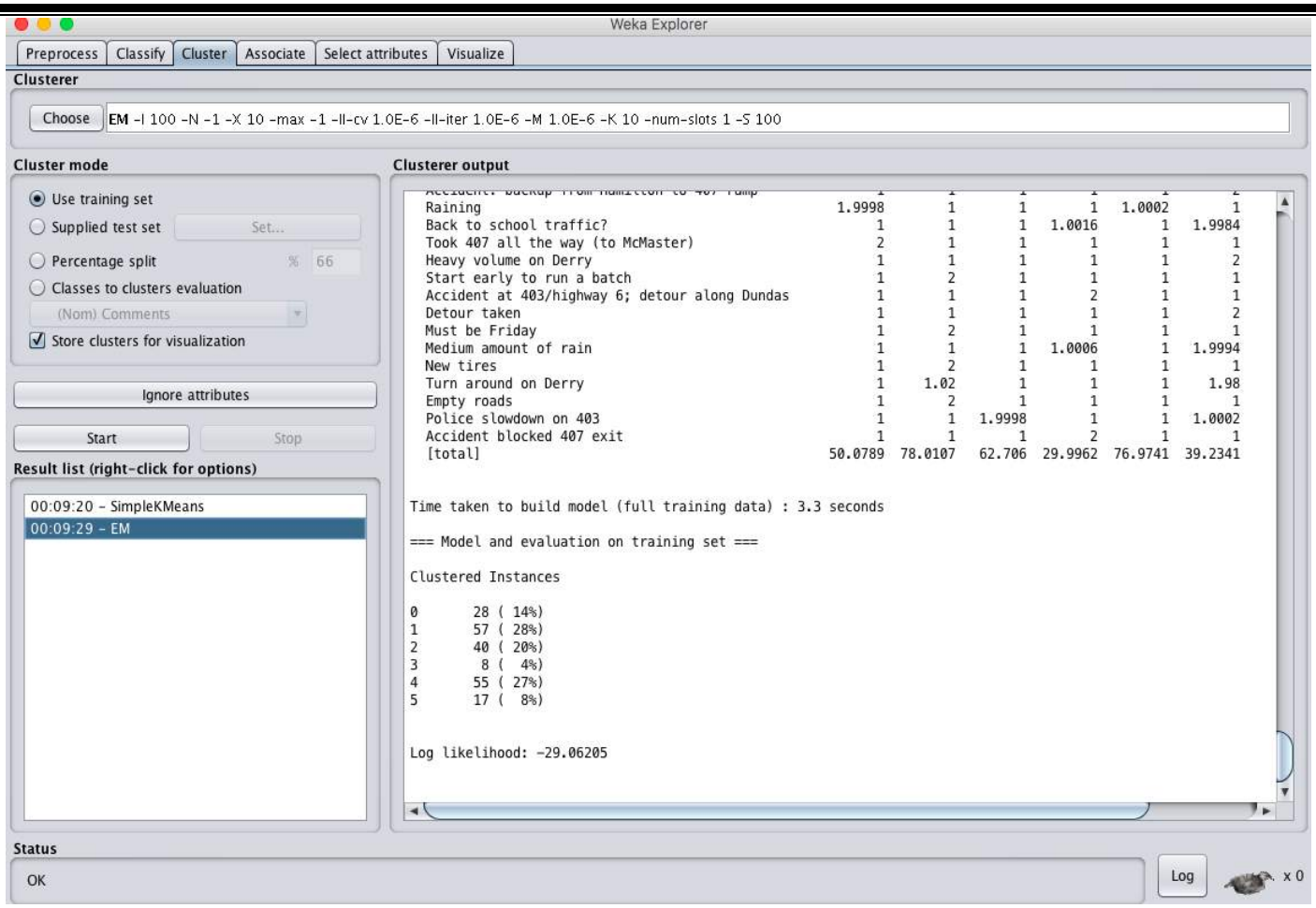
C. MIN, MAX, MEAN, STANDARD DEVIATION :






Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) Comments
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
00:09:20 - SimpleKMeans
00:09:29 - EM
Clusterer output
17:08 1 1 1 1.0001 1 1.9999
17:51 1 1 1 1 1 2
16:56 1 1 1 1 1 2
[total] 151.0789 179.0107 163.706 130.9962 177.9741 140.2341
DayOfWeek
Friday 5.0055 8.958 1.006 1 15.0305 2
Wednesday 5.001 18.4503 13.3922 1.0025 6.0641 9.0898
Tuesday 8.9998 10.2039 12.0009 3.9955 12.8138 5.9861
Monday 6.0058 14.3339 4.5069 4.001 12.0564 4.096
Thursday 8.0668 9.0646 14.8 2.9972 14.0093 1.0621
[total] 33.0789 61.0107 45.706 12.9962 59.9741 22.2341
GoingTo
Home 6.0068 12.3545 30.6028 2.002 42.9225 12.1115
GSK 24.0721 45.6561 12.1032 7.9942 14.0516 7.1226
[total] 30.0789 58.0107 42.706 9.9962 56.9741 19.2341
Distance
mean 50.6858 50.0981 51.3566 51.5117 51.3588 51.9991
std. dev. 1.071 1.2165 0.2696 1.6241 0.4025 2.7277
MaxSpeed
mean 128.8996 125.886 128.3687 125.6354 128.954 125.7314
std. dev. 3.3208 3.0584 4.4332 6.6614 3.5751 4.5587
AvgSpeed
mean 92.3387 71.9209 69.0752 49.9519 79.4346 62.0133
std. dev. 13.6344 3.4907 2.4157 5.6771 3.1459 3.9283
AvgMovingSpeed
mean 100.3697 79.1349 75.9152 61.8396 85.6243 73.2572
std. dev. 7.1155 2.8929 2.643 9.2199 2.609 7.2168
FuelEconomy
- 1.0058 1.9937 1 1 2.0005 1
8.89 1 4.0162 2.0017 1 4.9821 1
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) Comments
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
00:09:20 - SimpleKMeans
00:09:29 - EM
Clusterer output
8.48 1 5.6278 2.3448 1 2.0474 1.98
8.45 2 6.0017 2.975 2 1.9977 1.0255
8.28 3 1.0205 1 1 1.9795 1
7.89 4.9996 1.0004 1 2.0005 1 1.9995
[total] 52.0789 80.0107 64.706 31.9962 78.9741 41.2341
TotalTime
mean 32.4862 41.8672 44.8471 62.8634 38.8569 50.4161
std. dev. 3.0897 1.822 1.8776 8.9699 1.5461 2.2014
MovingTime
mean 30.4482 38.0276 40.5136 50.9631 36.077 42.8708
std. dev. 2.1013 1.3396 1.3435 6.8752 1.1221 3.5146
Take407All
No 1.0105 57.0103 41.706 8.9961 53.0429 14.2342
Yes 29.0684 1.0004 1 1.0001 3.9312 4.9999
[total] 30.0789 58.0107 42.706 9.9962 56.9741 19.2341
Comments
Put snow tires on 1 1 1 1 2 1
Heavy rain 1 1 1 2 1 1
Huge traffic backup 1 1 1 2 1 1
Pumped tires up: check fuel economy improved? 1 1.0022 1.9978 1 1 1
Backed up at Bronte 27.0791 52.9885 38.7113 4.9937 54.9739 10.2534
Rainy 1 1 1.997 1 1 1.003
Rain, rain, rain 1 1 1 1.0002 1 2.9998
Accident: backup from Hamilton to 407 ramp 1 1 1 1 1 2
Raining 1.9998 1 1 1 1.0002 1
Back to school traffic? 1 1 1 1.0016 1 1.9984
Took 407 all the way (to McMaster) 2 1 1 1 1 1
Heavy volume on Derry 1 1 1 1 1 2
Start early to run a batch 1 2 1 1 1 1
Accident at 403/highway 6; detour along Dundas 1 1 1 2 1 1
Detour taken 1 1 1 1 1 2
Must be Friday 1 2 1 1 1 1
Status
OK
Log
x 0



D. Are there any records that have a value that no other records has?

Avg Speed is unique from all of the other attributes records. Its uniqueness percentage is all about 60 % out of everything. So this would be the attribute where no other record has this records value.

E. Write a note on class distribution for each of the attributes.

Mainly, all the attributes are distributed to 2 types. They are :

1. Nominal
2. Numeric

- Nominal class attributes are : Date, Start Time, Day Of Week, Going To, Fuel Economy, Total 407 All, Comments.
- Numeric class attributes are : Distance, MaxSpeed, AvgSpeed, Avg Moving Speed, Total Time, Moving Time.

F. Apply attribute selection measures under filter supervised selection.

When we apply attribute selection under the filter of supervised attribute selection. Initially, it had 13 attributes but after the filter of the attributes count is been reduced to 11, where AvgMaxSpeed and Total 407 All are removed.

RESULT :

Thus, the dataprocessing and analysis for a dataset using weka tool has been successfully completed.

EX.No: 06
Date :

DATA SEGMENTATION BY K- MEANS CLUSTER USING WEKA AND R-TOOL

PROBLEM STATEMENT :

Apply K-means algorithm to your dataset experiment with the algorithm as follows: By setting the number of elements and seed of the random algorithm for generating initial cluster centres . Compare the results that has occurred between K-means and R-Tool .

DESCRIPTION :

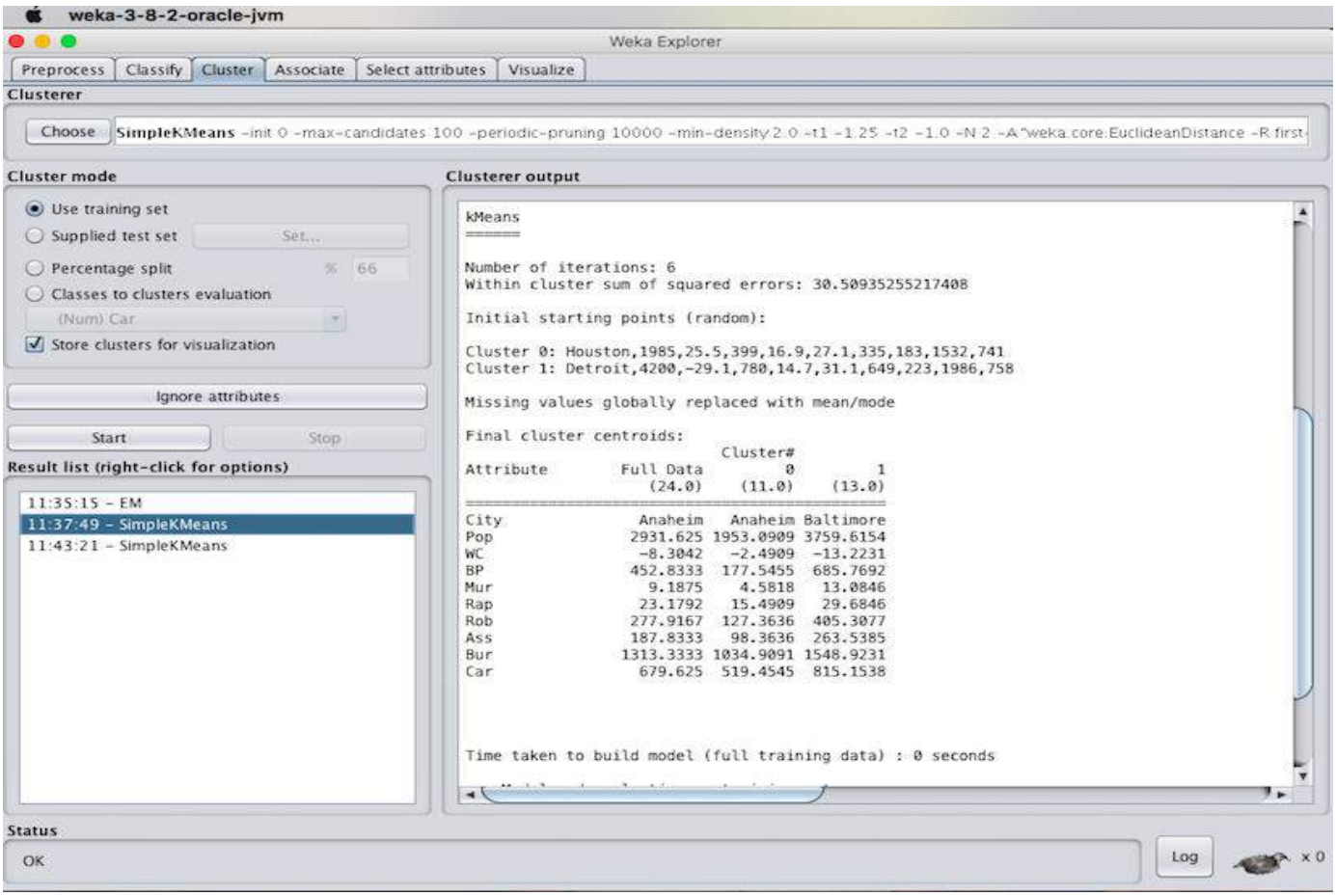
Consider a dataset of citycrimes.csv file of which it contains the attributes are City, Pop, WC, BP, Mur, Rap, Rob, Ass, Bus and car for the performance of the dataset by applying the K-means algorithm in weka and as well using R- tool.

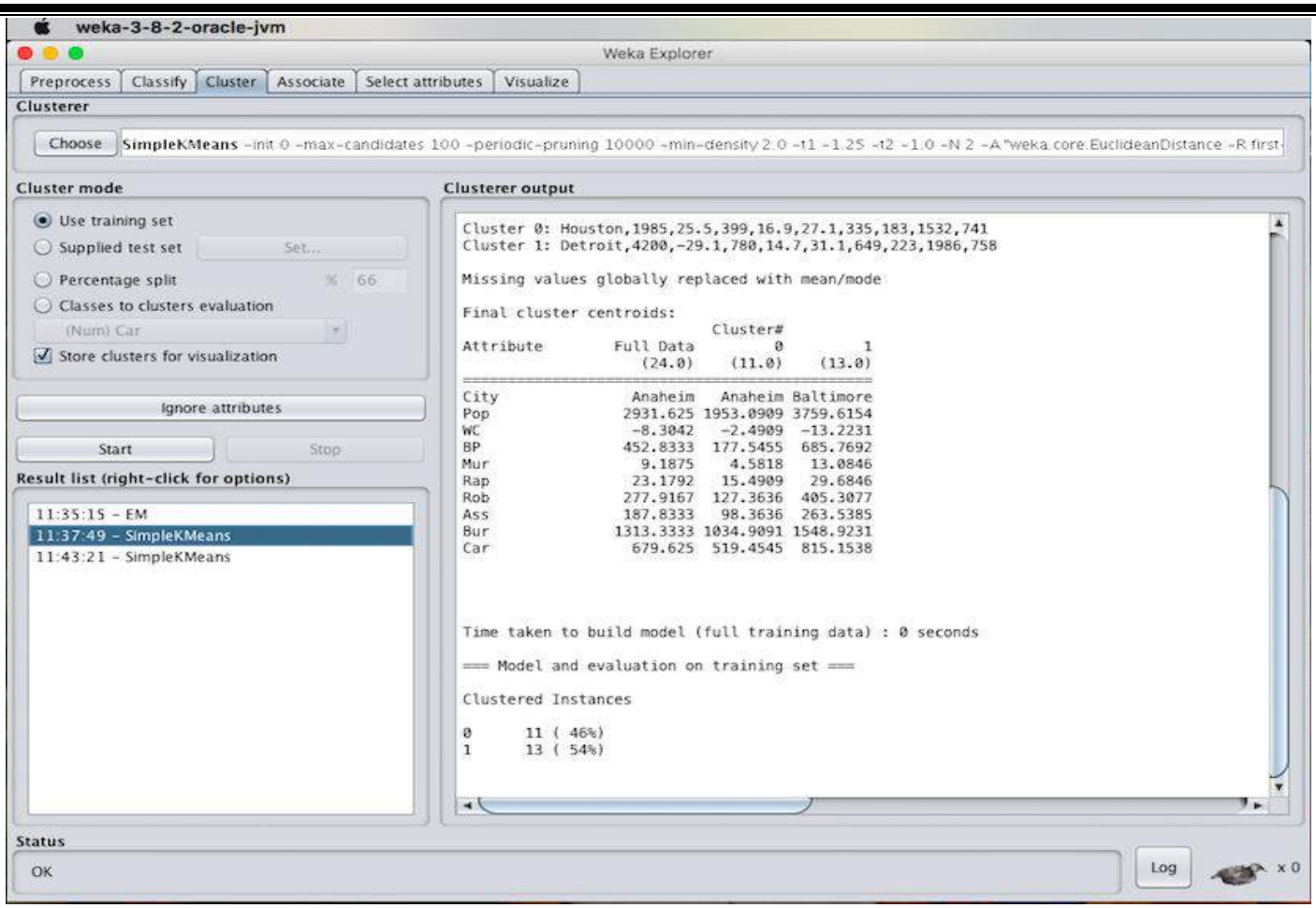
❖ USING WEKA TOOL :

A. Choose a set of attributes for clustering and give a motivation.

STEPS INVOLVED :

- Choose a set of attributes for clustering and for giving a motivation.
- Choose the dataset and import the dataset into Weka tool.
- Cluster the dataset and choose simple K-means algorithm and give the motivation.

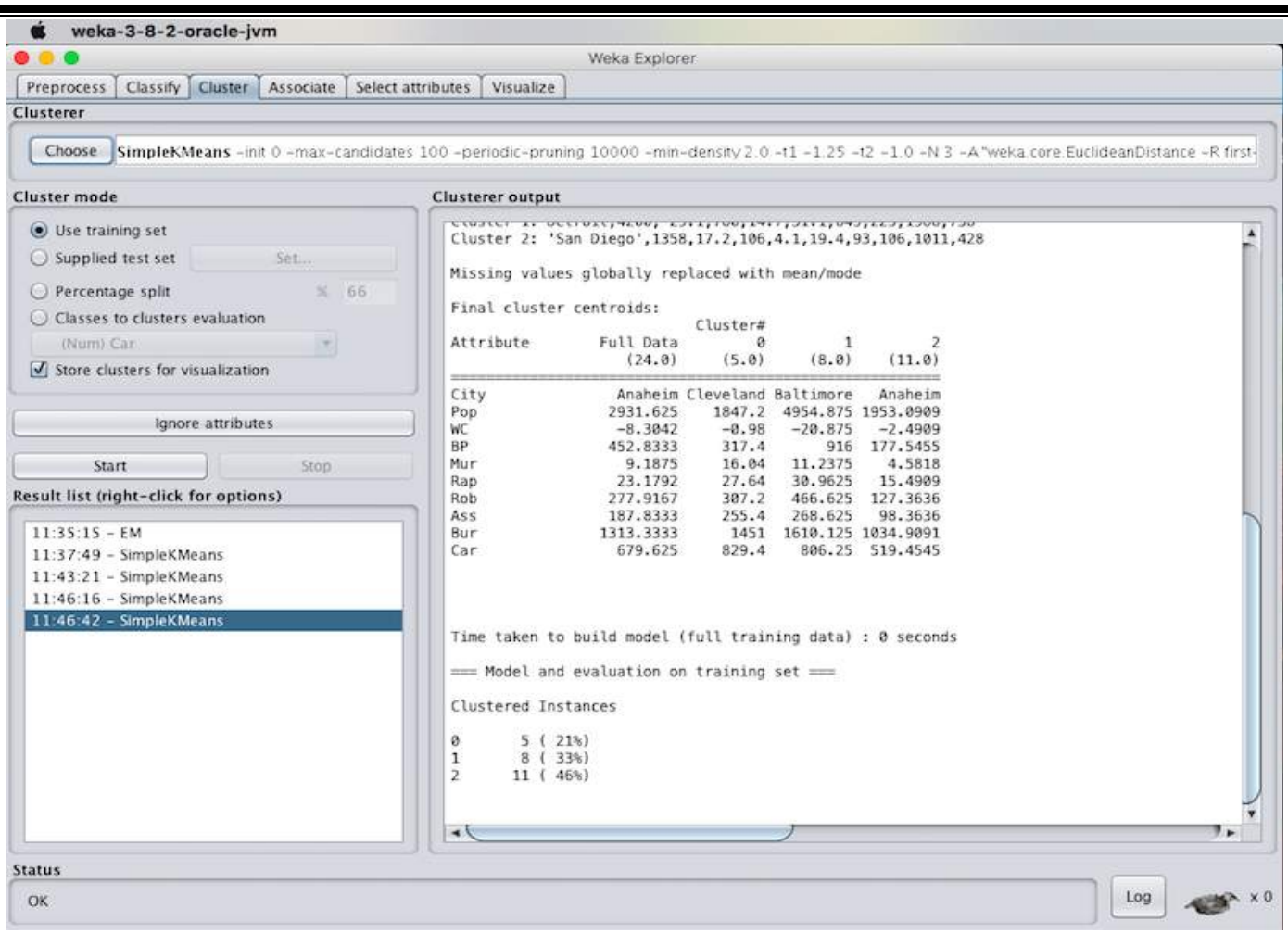


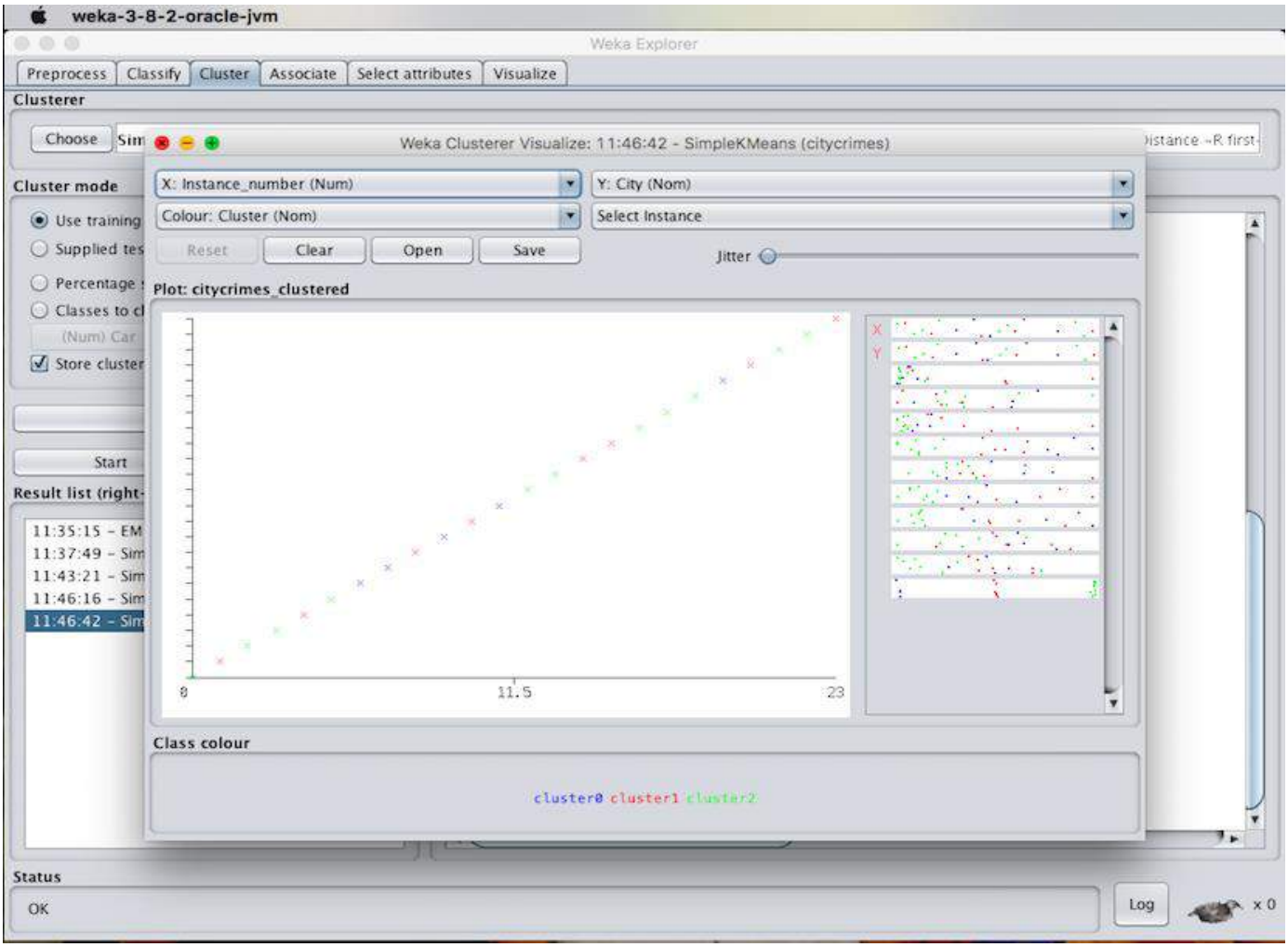


B. Experiment with atleast 2 different number of clusters but with same seed values:

STEPS INVOLVED :

- Compare the two different clusters but with the same seed values.
- Change the number of clusters value and need not to change the seed value.
- Apply the K-means algorithm and start executing the algorithm.


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
66
Classes to clusters evaluation
(Num) Car
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
11:35:15 - EM
11:37:49 - SimpleKMeans
11:43:21 - SimpleKMeans
11:46:16 - SimpleKMeans
11:46:42 - SimpleKMeans
Clusterer output
Cluster 2: 'San Diego',1358,17.2,106,4.1,19.4,93,106,1011,428
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1 2
(24.0) (5.0) (8.0) (11.0)
City Anaheim Cleveland Baltimore Anaheim
Pop 2931.625 1847.2 4954.875 1953.0909
WC -8.3042 -0.98 -20.875 -2.4909
BP 452.8333 317.4 916 177.5455
Mur 9.1875 16.04 11.2375 4.5818
Rap 23.1792 27.64 30.9625 15.4909
Rob 277.9167 307.2 466.625 127.3636
Ass 187.8333 255.4 268.625 98.3636
Bur 1313.3333 1451 1610.125 1034.9091
Car 679.625 829.4 806.25 519.4545
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 5 (21%)
1 8 (33%)
2 11 (46%)
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Weka Clusterer Visualize: 11:46:42 - SimpleKMeans (citycrimes)
X: Instance_number (Num)
Y: City (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: citycrimes_clustered
0
11.5
23
Class colour
cluster0 cluster1 cluster2
Cluster mode
Result list (right-
11:35:15 - EM
11:37:49 - Sim
11:43:21 - Sim
11:46:16 - Sim
11:46:42 - Sim
Start
Status
OK
Log
x 0

C. **Try with the different seed values . Explain what is the seed value controls**.

It was observed that when there is an increase in the seed value from the standard seed value 10 to higher ranges. The number of iterations will be reduced. In the case of seed value 10, for the given citycrime.csv dataset. It generated '6' iterations whereas for seed value 100 it has generated only '2' iterations.

Finally, there will be the change in the sequence of the tuples in the output and in clustered instances percentage will be changed. Seed value 100 controls the number of iterations.

❖ **USING R-TOOL :**

STEPS INVOLVED :

- Choose the dataset and import the dataset into the R-tool.
- View the dataset and start inserting queries for the k means clustering algorithm.

QUERIES :

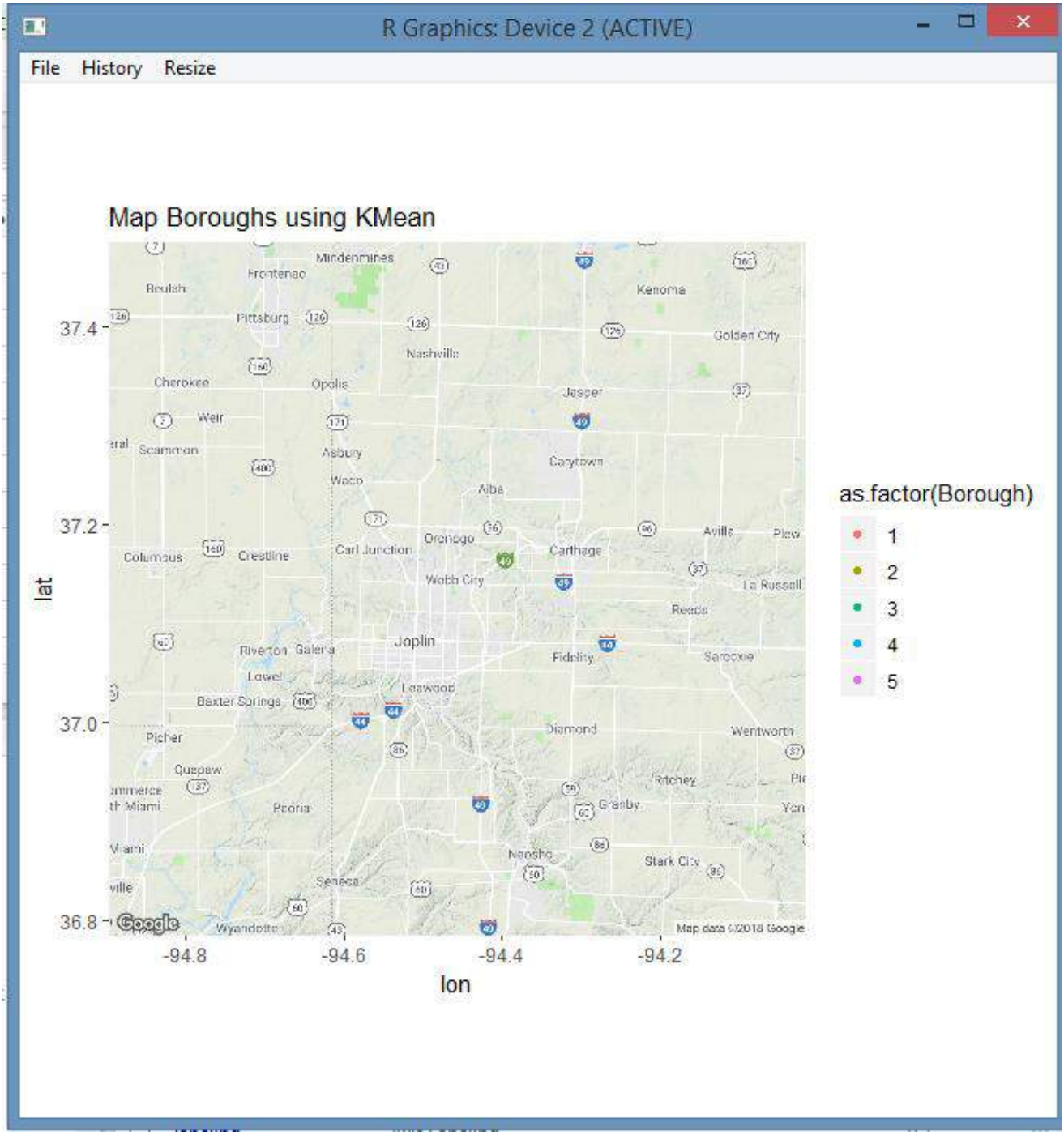
- set.seed(20)
- clusters <- kmeans(citycrimes[,2:3], 5)
- citycrimes$Borough<- as.factor(clusters$cluster)
- str(clusters)

```
>
> clusters <- kmeans(citycrimes[,2:3], 5)
>
> citycrimes$Borough <- as.factor(clusters$cluster)
> str(clusters)
```

```
Console ~/
List of 9
 $ cluster     : int [1:24] 4 1 3 4 2 4 1 4 5 1 ...
 $ centers     : num [1:5, 1:2] 2079 8513 2908 1391 4509 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5] "1" "2" "3" "4" ...
  .. ..$ : chr [1:2] "Pop" "WC"
 $ totss       : num 1.39e+08
 $ withinss    : num [1:5] 315574 13643048 67068 52387 191844
 $ tot.withinss: num 14269920
 $ betweenss   : num 1.25e+08
 $ size        : int [1:5] 7 3 3 9 2
 $ iter        : int 3
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> library(ggmap)
```

- library(ggmap)
- Map <- get_map("citycrimes:,zoom=10)
- ggmap(Map) + geom_point(aes(x = Pop[] , y = WC[] , colour = as.factor(Borough)) , data = citycrimes) +
  ggtitle("Map Boroughs using KMean")

```
Console ~/
> ggmap(Map) + geom_point(aes(x = Pop[], y = WC[], colour = as.factor(Borough)),data = c
itycrimes)
Warning message:
Removed 24 rows containing missing values (geom_point).
> ggtitle("Map Boroughs using KMean")
$title
[1] "Map Boroughs using KMean"

$subtitle
NULL

attr(,"class")
[1] "labels"
>
>
```



RESULT :

Thus, the K-means clustering analyzing using both the weka tool and R- tool has been successfully completed. In case of weka tool, the change in seed values lead to the decrease in the number of iterations. In case of R-tool, there are only 3 number of iterations.

EX.No: 07
Date :

DATA SEGMENTATION BY EXPECTATION MAXIMISATION ALGORITHM THROUGH WEKA

PROBLEM STATEMENT :

Analyze the dataset using Expectation Maximization algorithm(EM) by setting the minimum standard deviation for normal density calculation and compare the results with the simple K-means algorithm.

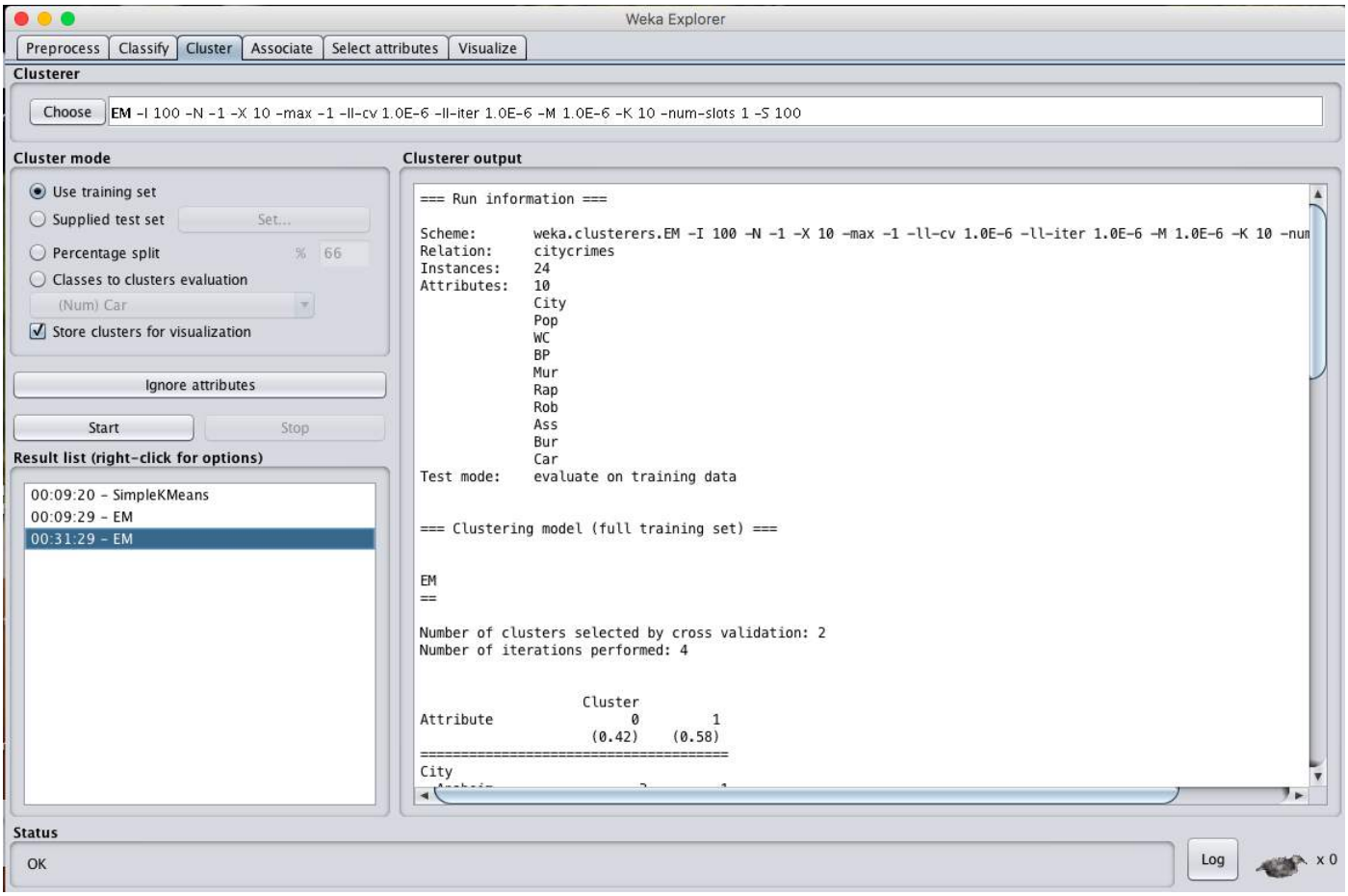
DESCRIPTION :

Consider a dataset of citycrimes.csv file of which it contains the attributes are City, Pop, WC, BP, Mur, Rap, Rob, Ass, Bus and car for the performance of the dataset by applying the K-means algorithm in weka and as well using R- tool.

When the clustering is been made through the expectation maximization algorithm by setting minimum standard deviation values then the results will be of the following :

- **Steps Involved :**

  - Initially, load the dataset into the weka tool and check for all the attributes present in the dataset.
  - Then move to cluster panel and apply the EM algorithm technique for the datasheet.
  - Finally, Observe the results that are obtained.



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer

Choose: EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode

- (•) Use training set
- ( ) Supplied test set — Set...
- ( ) Percentage split — % 66
- ( ) Classes to clusters evaluation
  (Num) Car
- [✓] Store clusters for visualization

Ignore attributes

Start | Stop

Result list (right-click for options)

00:09:20 - SimpleKMeans
00:09:29 - EM
00:31:29 - EM

Clusterer output

```
                    Cluster
Attribute               0        1
                     (0.42)   (0.58)
=====================================
City
  Anaheim               2        1
  Baltimore             1        2
  Boston                2        1
  Buffalo               2        1
  Chicago               1        2
  Cincinnatti           2        1
  Cleveland             1        2
  Dallas                1        2
  Detroit               1        2
  Houston               1        2
  Los Angeles           1        2
  Miami                 1        2
  Milwaukee             2        1
  Minneapolis           2        1
  New York              1        2
  Newark                1        2
  Paterson              2        1
  Philadelphia          1        2
  Pittsburgh            2        1
  St Louis              1        2
  San Francisco         1        2
  San Diego             2        1
  Seattle               2        1
  Washington            1        2
  [total]              34       38
Pop
  mean           1666.5993 3835.2135
  std. dev.       479.9086 2795.5345
```

Status

OK — Log — x 0

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer

Choose: EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode

- (•) Use training set
- ( ) Supplied test set — Set...
- ( ) Percentage split — % 66
- ( ) Classes to clusters evaluation
  (Num) Car
- [✓] Store clusters for visualization

Ignore attributes

Start | Stop

Result list (right-click for options)

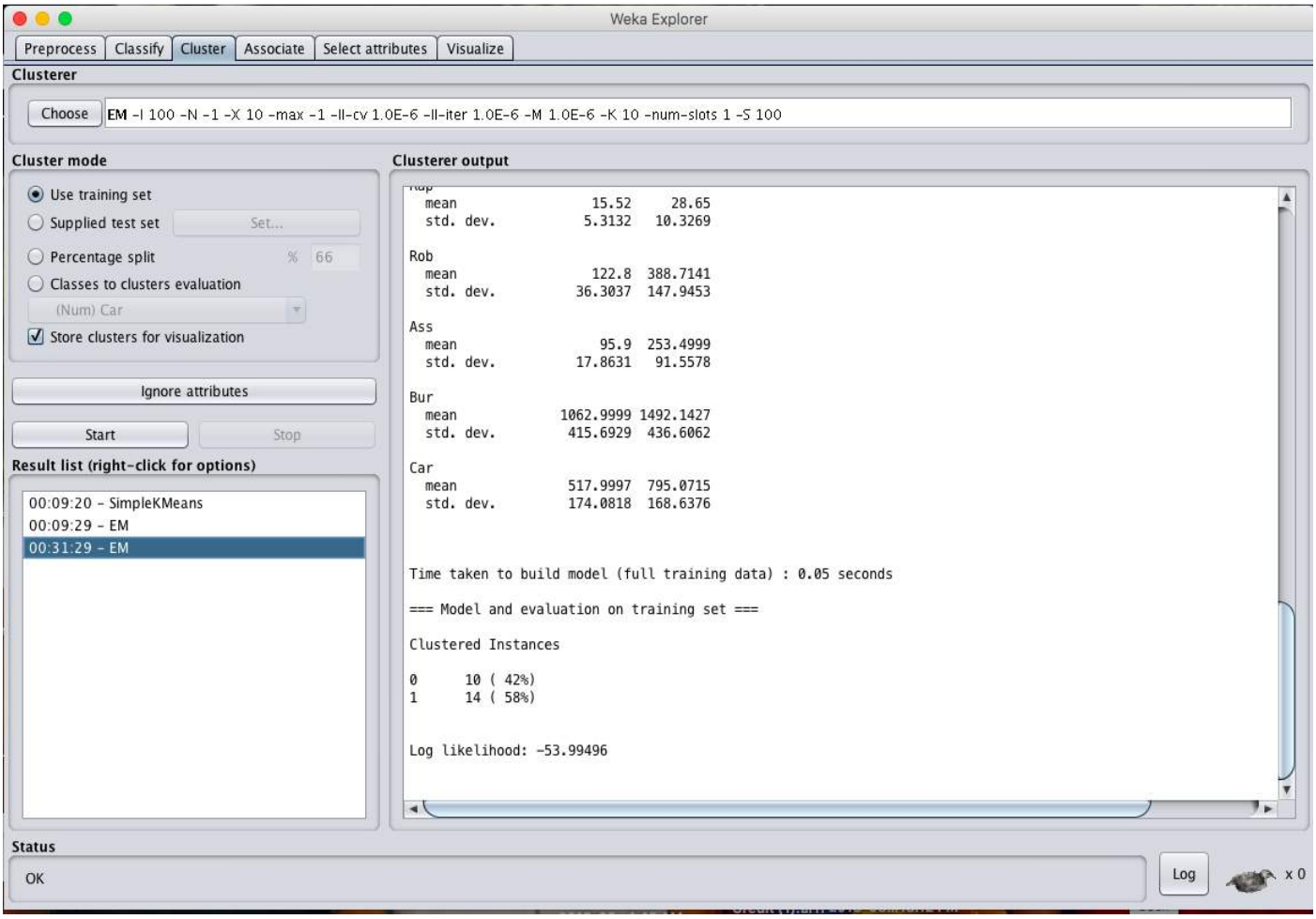
00:09:20 - SimpleKMeans
00:09:29 - EM
00:31:29 - EM

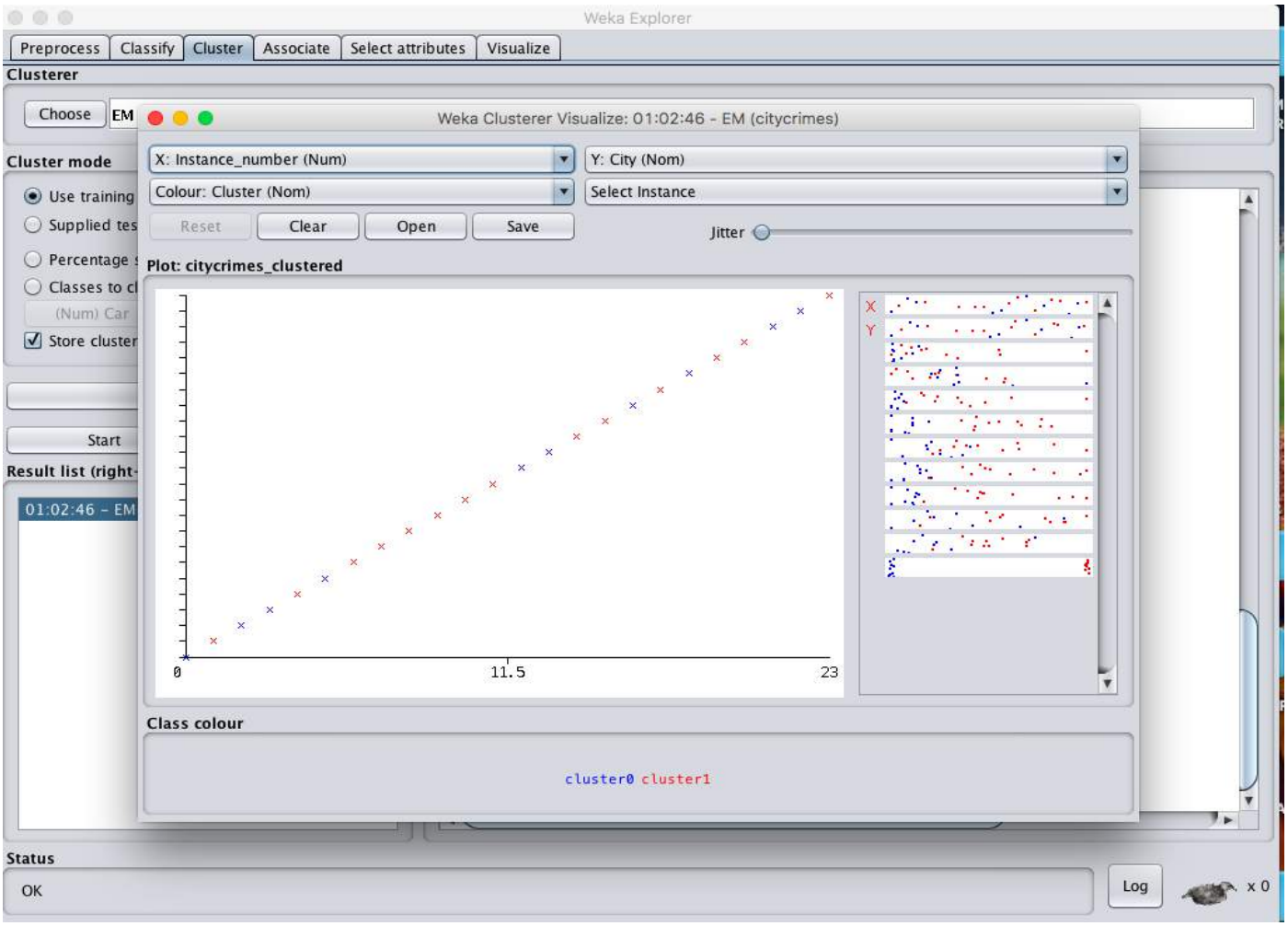
Clusterer output

```
WC
  mean              -4.03  -11.3571
  std. dev.        20.8813    20.788

BP
  mean                108  699.1425
  std. dev.         42.6896  489.7613

Mur
  mean                4.11   12.8143
  std. dev.          1.2112    3.0341

Rap
  mean               15.52     28.65
  std. dev.          5.3132   10.3269

Rob
  mean               122.8  388.7141
  std. dev.        36.3037  147.9453

Ass
  mean                95.9  253.4999
  std. dev.        17.8631   91.5578

Bur
  mean           1062.9999 1492.1427
  std. dev.       415.6929  436.6062

Car
  mean            517.9997  795.0715
  std. dev.       174.0818  168.6376
```

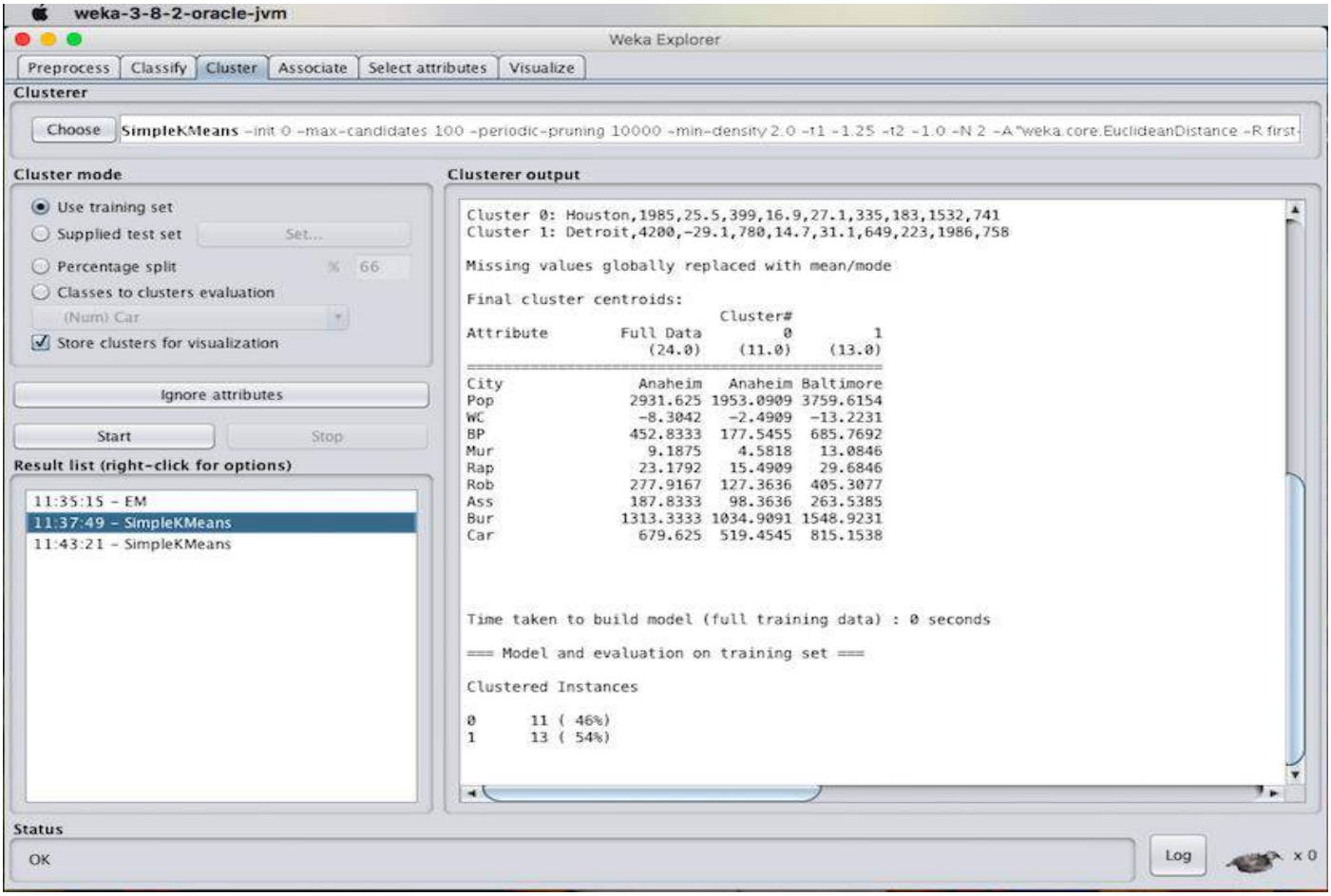
Status

OK — Log — x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) Car
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
00:09:20 - SimpleKMeans
00:09:29 - EM
00:31:29 - EM
Clusterer output
mean 15.52 28.65
std. dev. 5.3132 10.3269
Rob
mean 122.8 388.7141
std. dev. 36.3037 147.9453
Ass
mean 95.9 253.4999
std. dev. 17.8631 91.5578
Bur
mean 1062.9999 1492.1427
std. dev. 415.6929 436.6062
Car
mean 517.9997 795.0715
std. dev. 174.0818 168.6376
Time taken to build model (full training data) : 0.05 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 10 (42%)
1 14 (58%)
Log likelihood: -53.99496
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM
Weka Clusterer Visualize: 01:02:46 - EM (citycrimes)
X: Instance_number (Num)
Y: City (Nom)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: citycrimes_clustered
0
11.5
23
Class colour
cluster0 cluster1
Cluster mode
Use training
Supplied tes
Percentage
Classes to cl
(Num) Car
Store cluster
Start
Result list (right-
01:02:46 - EM
Status
OK
Log
x 0

❖ **K- MEANS ALGORITHM:**



COMPARISION :

When compared to both the algorithms for the same dataset there will be a change in time taken to build model will be little longer in EM than when compared to K-means and there will be a percentage change in the clustered instances values.

RESULT :

Thus, the data analysis by the expectation maximization algorithm using weka has been analyzed and observed properly .

EX.No: 08
Date :

DATA SEGMENTATION BY COBWEB – HIERARCHIAL CLUSTERING ALGORITHM USING WEKA TOOL

PROBLEM STATEMENT :

For the given data file find the following using weka:

A. what are the most alive groups in terms of number of people involved, cluster together.

B. what are the most active community.

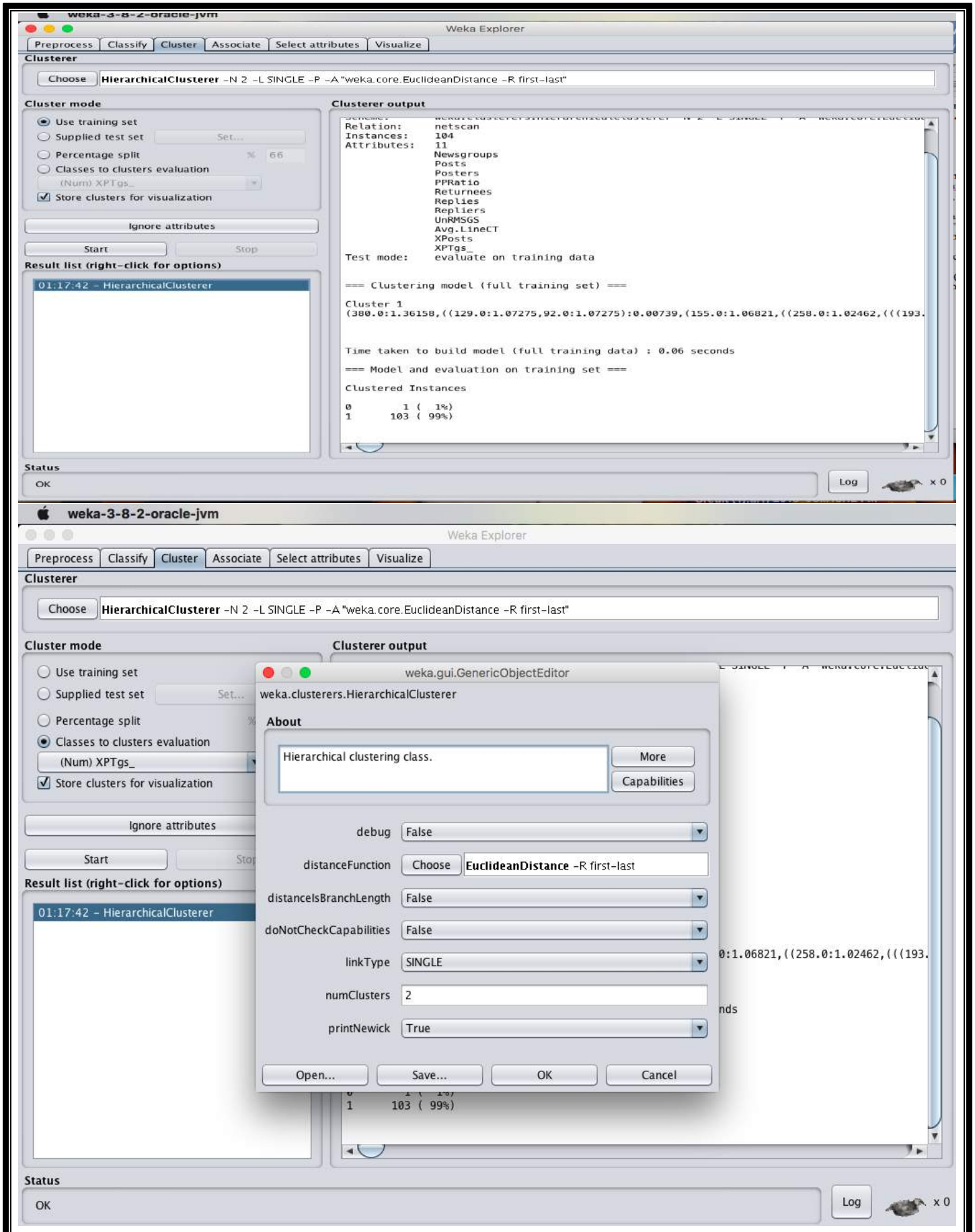
DESCRIPTION:

Consider a dataset netscan.csv where it contains the attributes of Newsgroups, posts, posters, PPRatio, Returnees, Replies, Repliers, UnRMsgs, Avg.LineCT, Xports, XPTgs. Each attribute will have different types of the meanings.

OBSERVATIONS :

A. The most active groups in terms of the number of people involved cluster together. Those groups - microsoft.public.windowsxp.perform_maintain, microsoft.public.windowsxp.network_web, microsoft.public.windowsxp.security_admin, microsoft.public.windowsxp.hardware - are all advanced user groups.
    1. They look like very active communities. (large number of posters, repliers, posts, and etc.)
    2. However, there are large number of isolated messages that might be questions with no answers yet, or might be questions ignored because they look like uninteresting to the advance users in those groups.

B. microsoft.public.es.* groups tightly cluster together except for the .windowsxp group. They share the followings.
    1. Relatively large number of XPosts (crosspostings) : reference many postings in other groups.
    2. Low PPRatio : Small number of posters post large number of postings.


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) XPTgs_
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
01:17:42 - HierarchicalClusterer
Clusterer output
Relation: netscan
Instances: 104
Attributes: 11
Newsgroups
Posts
Posters
PPRatio
Returnees
Replies
Repliers
UnRMSGS
Avg.LineCT
XPosts
XPTgs_
Test mode: evaluate on training data
=== Clustering model (full training set) ===
Cluster 1
(380.0:1.36158,((129.0:1.07275,92.0:1.07275):0.00739,(155.0:1.06821,((258.0:1.02462,(((193.
Time taken to build model (full training data) : 0.06 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 1 (1%)
1 103 (99%)
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.clusterers.HierarchicalClusterer
About
Hierarchical clustering class.
More
Capabilities
debug
False
distanceFunction
Choose
EuclideanDistance -R first-last
distanceIsBranchLength
False
doNotCheckCapabilities
False
linkType
SINGLE
numClusters
2
printNewick
True
Open...
Save...
OK
Cancel

RESULT :

Thus, the data analysis of cobweb hierarchial clustering algorithm using weka tools has been analyzed and observed successfully.

EX.No: 09
Date :

FREQUENT PATTERN MINING USING ASSOCIATION RULE THROUGH WEKA AND R TOOLS

PROBLEM STATEMENT :

Run the Apriori algorithm, and explore the association rules by changing the following parameters:

a) Upper bound min_sup
b) Lower bound min_sup
c) Metric type
d) Output itemsets

Implement the aprirori algorithm through R Tool and compare the results obtained through the weka.

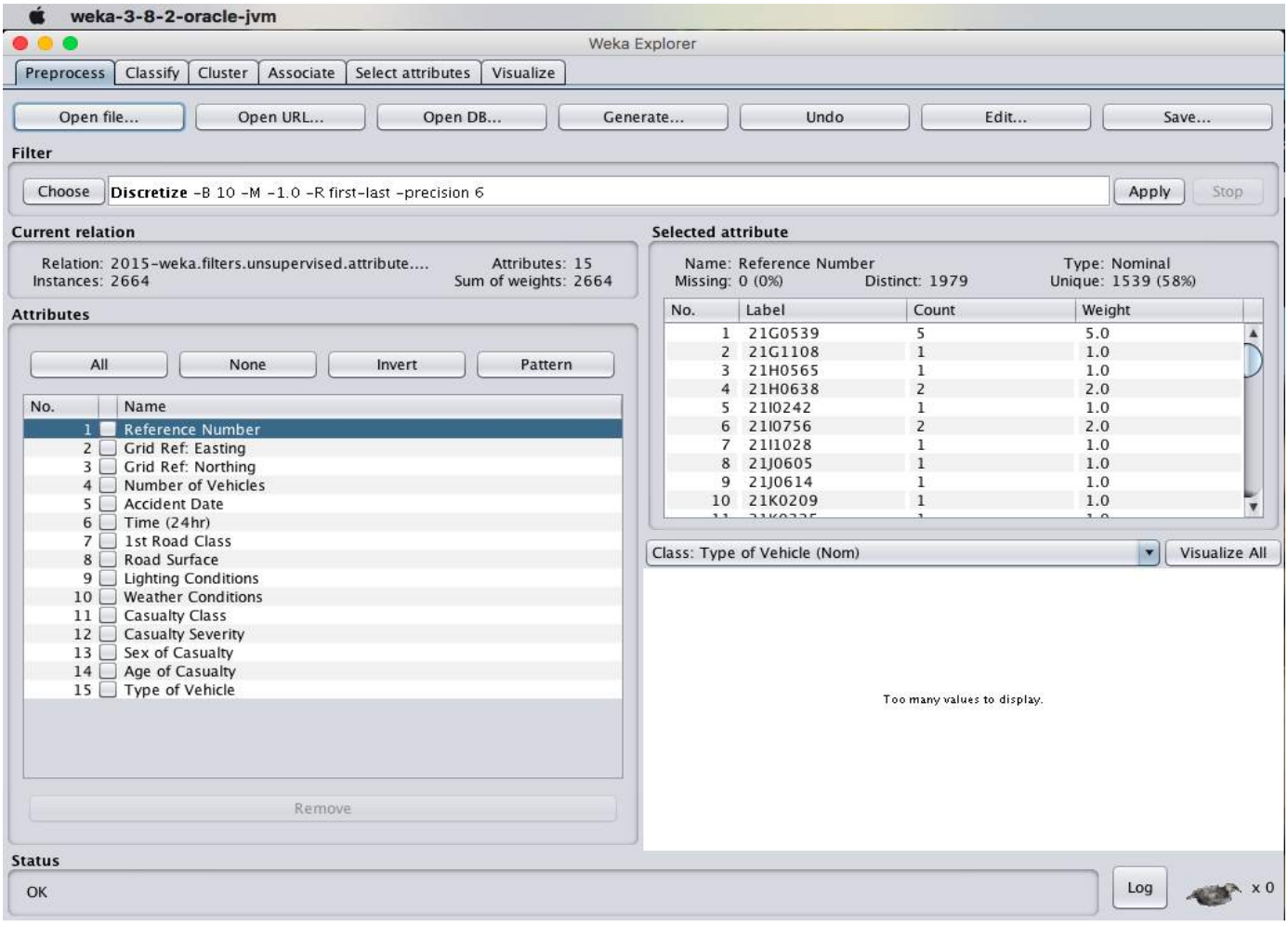
DESCRIPTION :

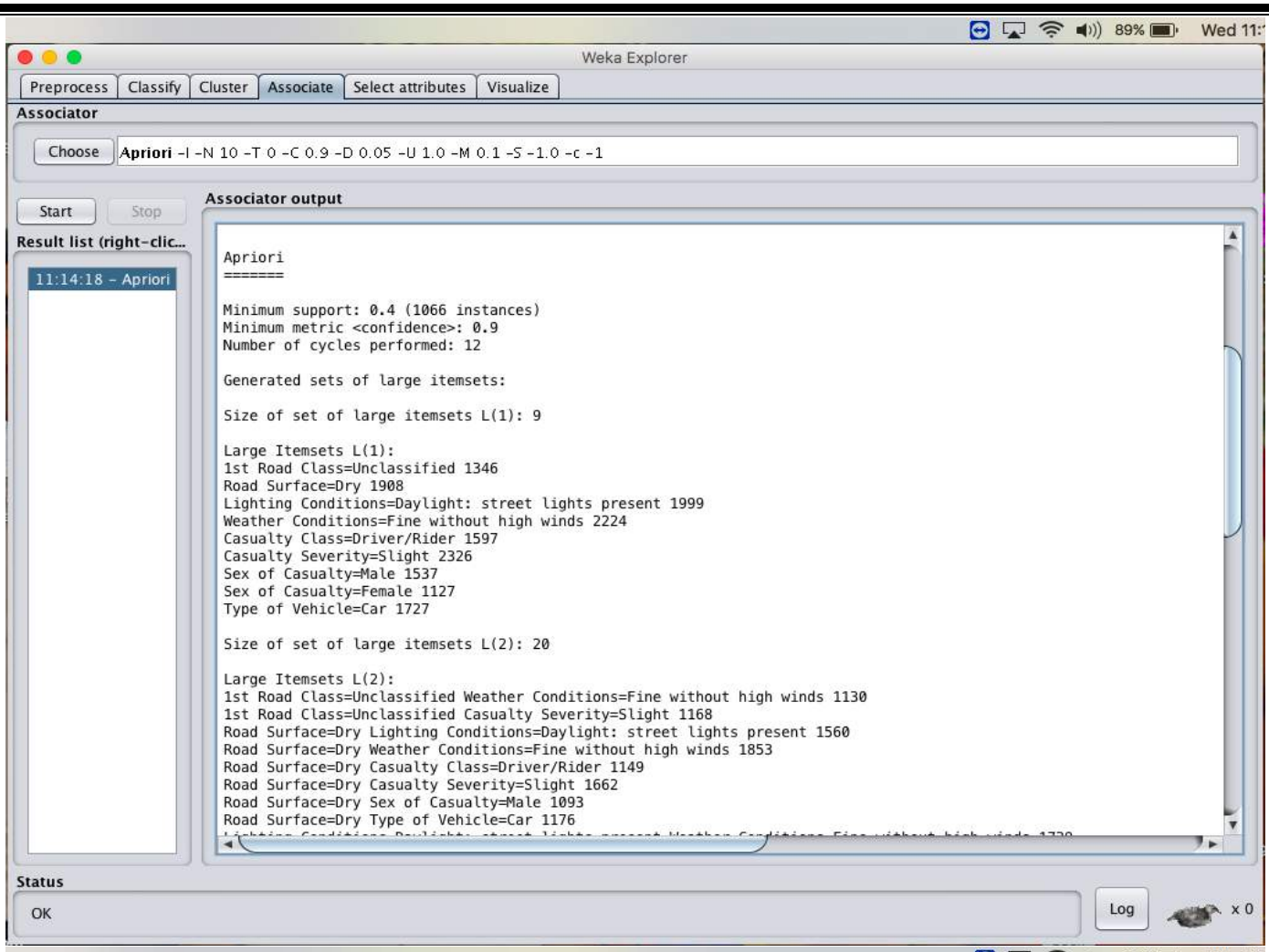
Consider a dataset of 2015.csv file of which it contains the attributes are Reference Number, Grid ref: Easting, Grid Ref: Northing, Number of vehicles, Accident date, Time(24 hr), 1st Road class, Road Surface, Lighting conditions, Weather conditions, casuality class, Sex of casuality, Age of casuality, Type of casuality for the performance of the dataset by applying the Apriori algorithm in weka and as well using R-tool.

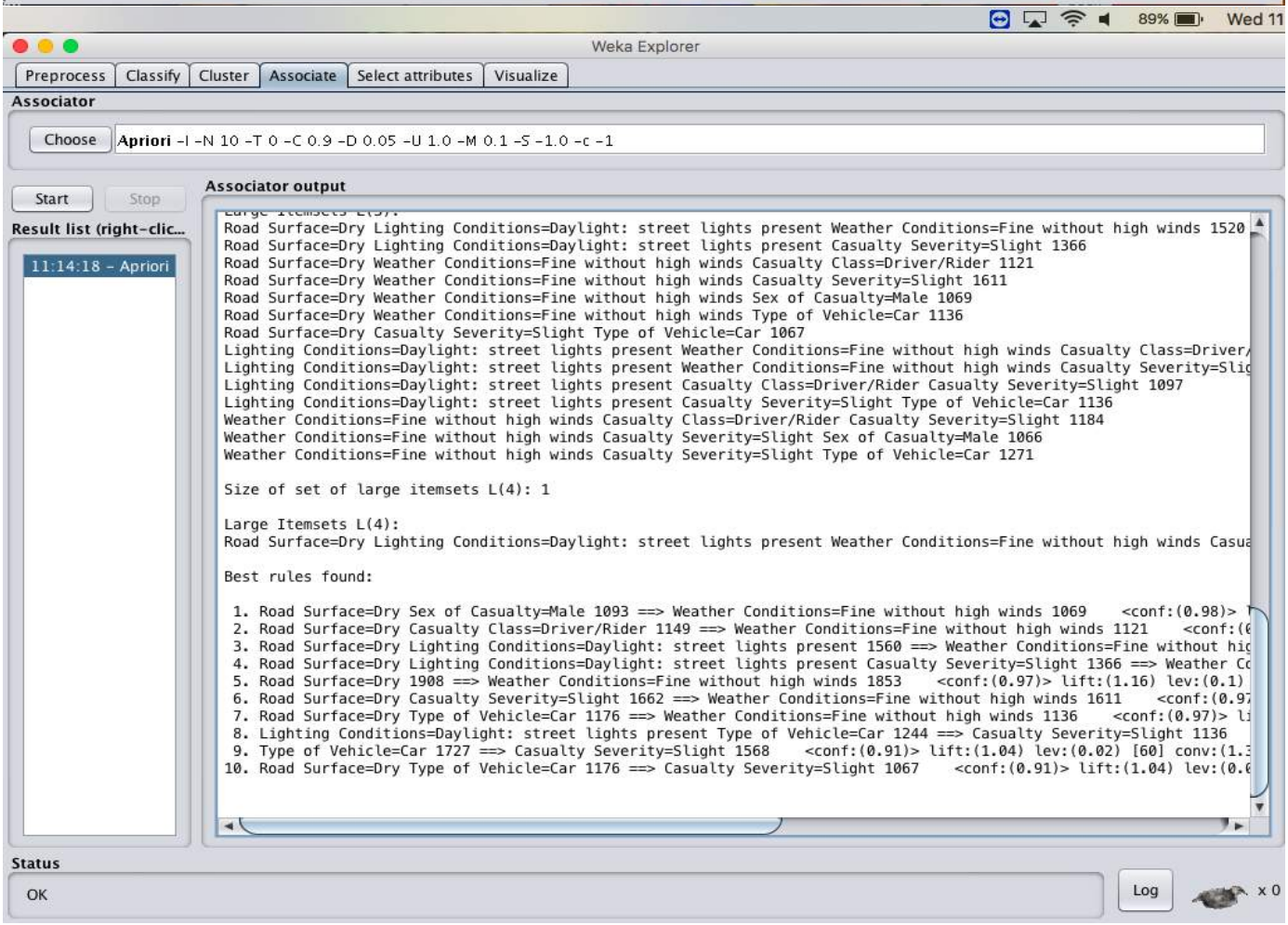
- **USING WEKA TOOL :**

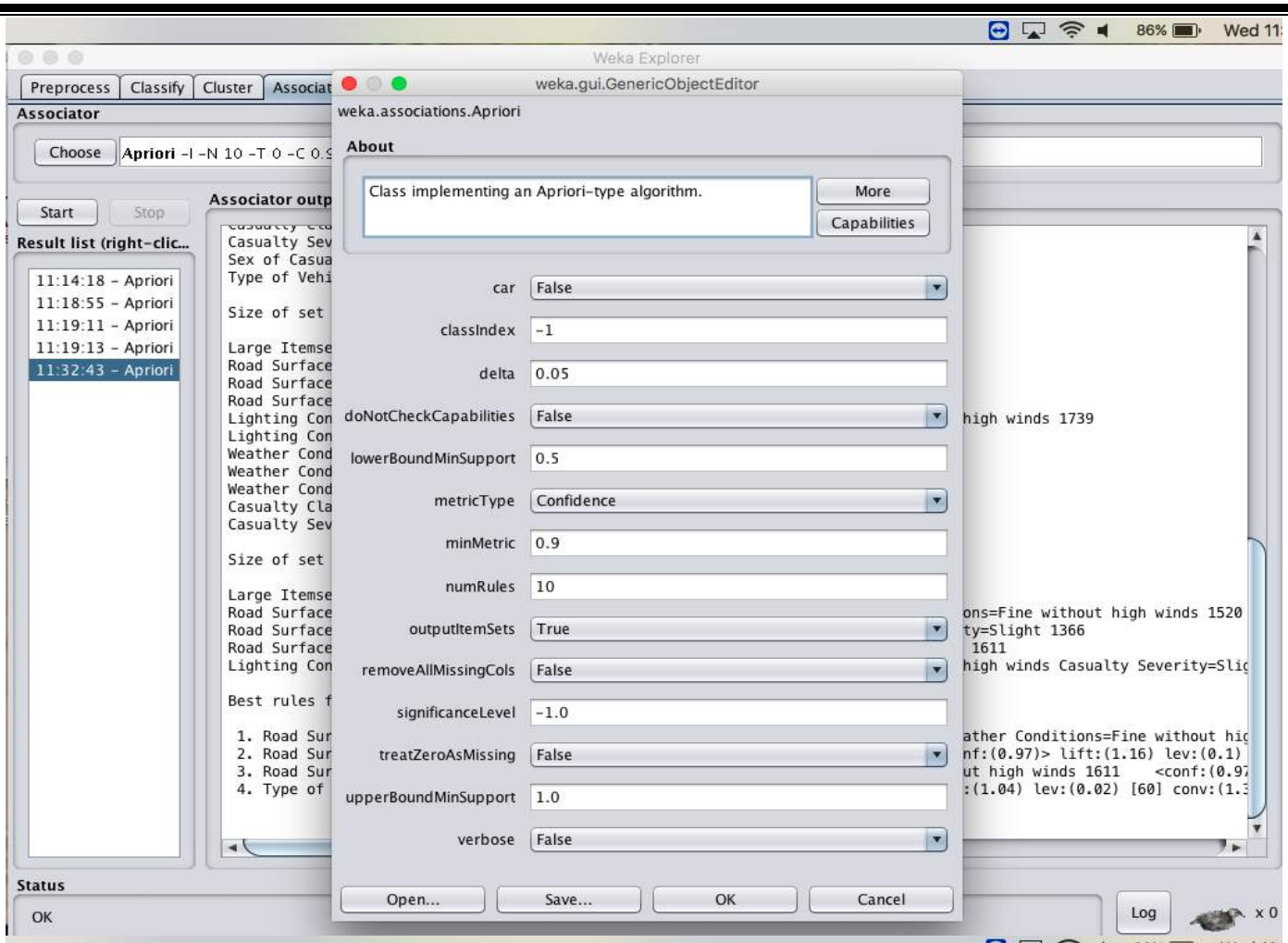
STEPS INVOLVED :

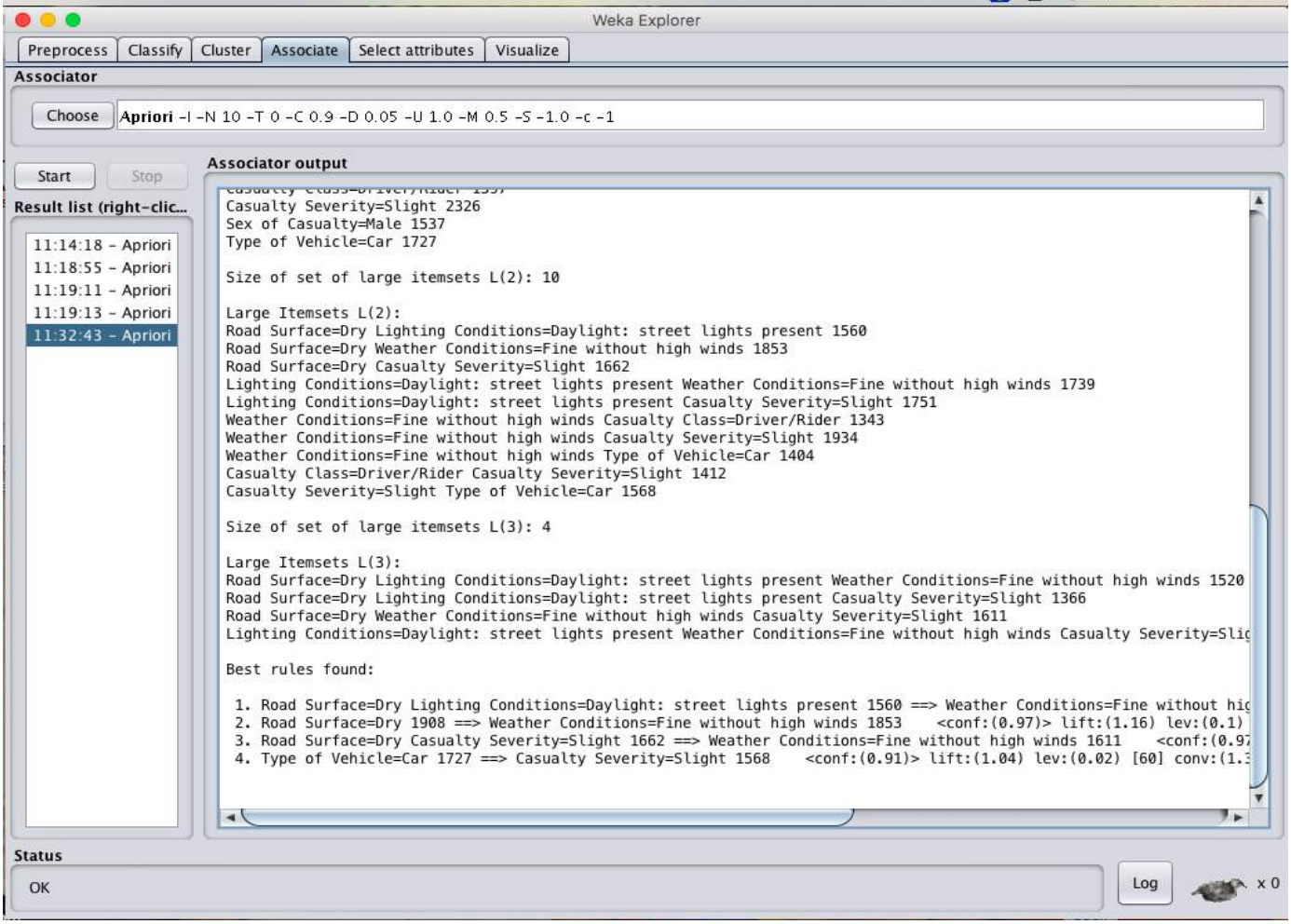
- Choose a set of attributes for clustering and for giving a motivation.
- Choose the dataset and import the dataset into Weka tool.
- Discretize the attributes from numeric to nominal to perform the algorithm.
- Cluster the dataset and choose simple Apriori algorithm.
- Set the Upper bound min_sup and lower bound min_sup values.

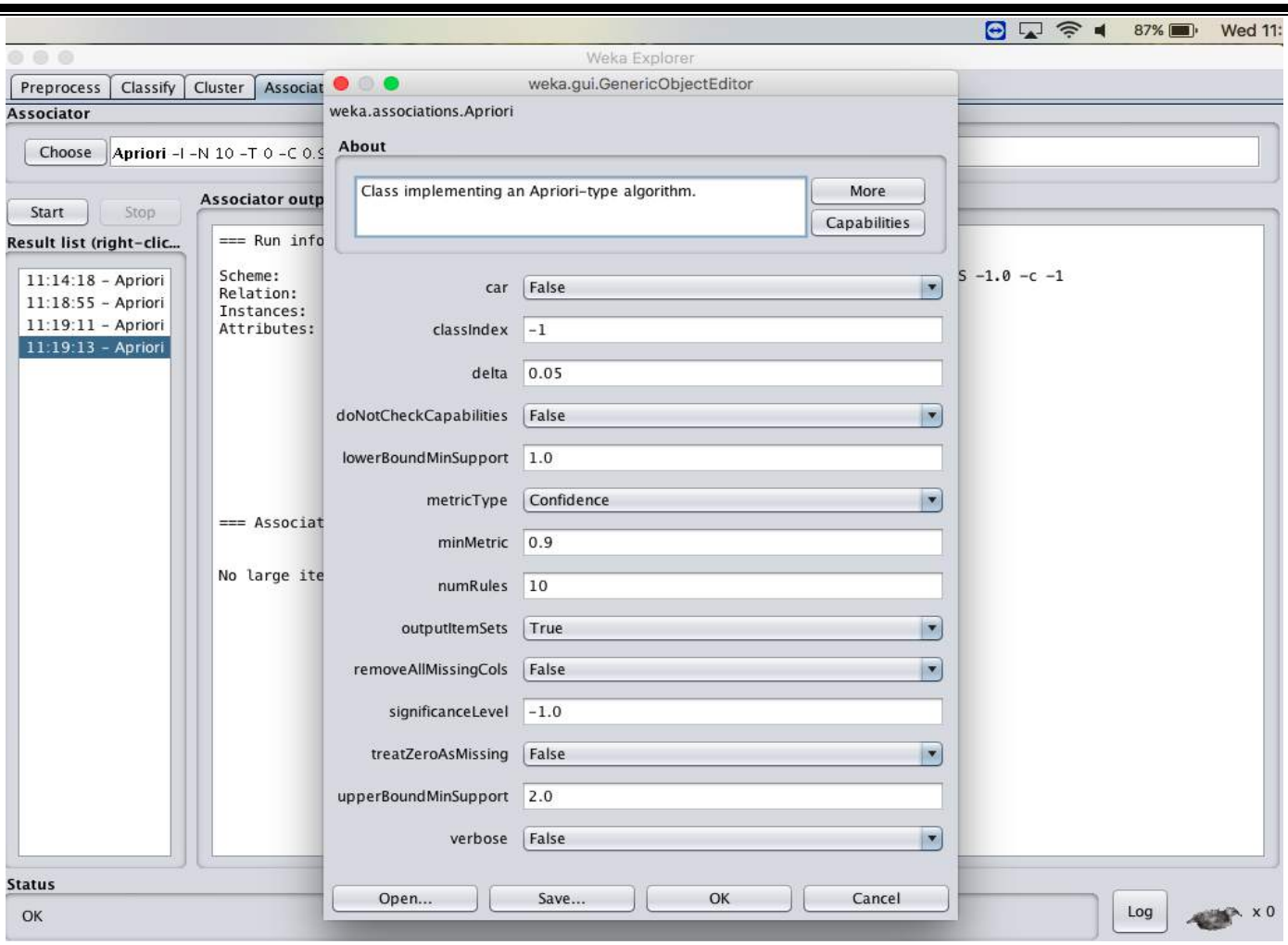


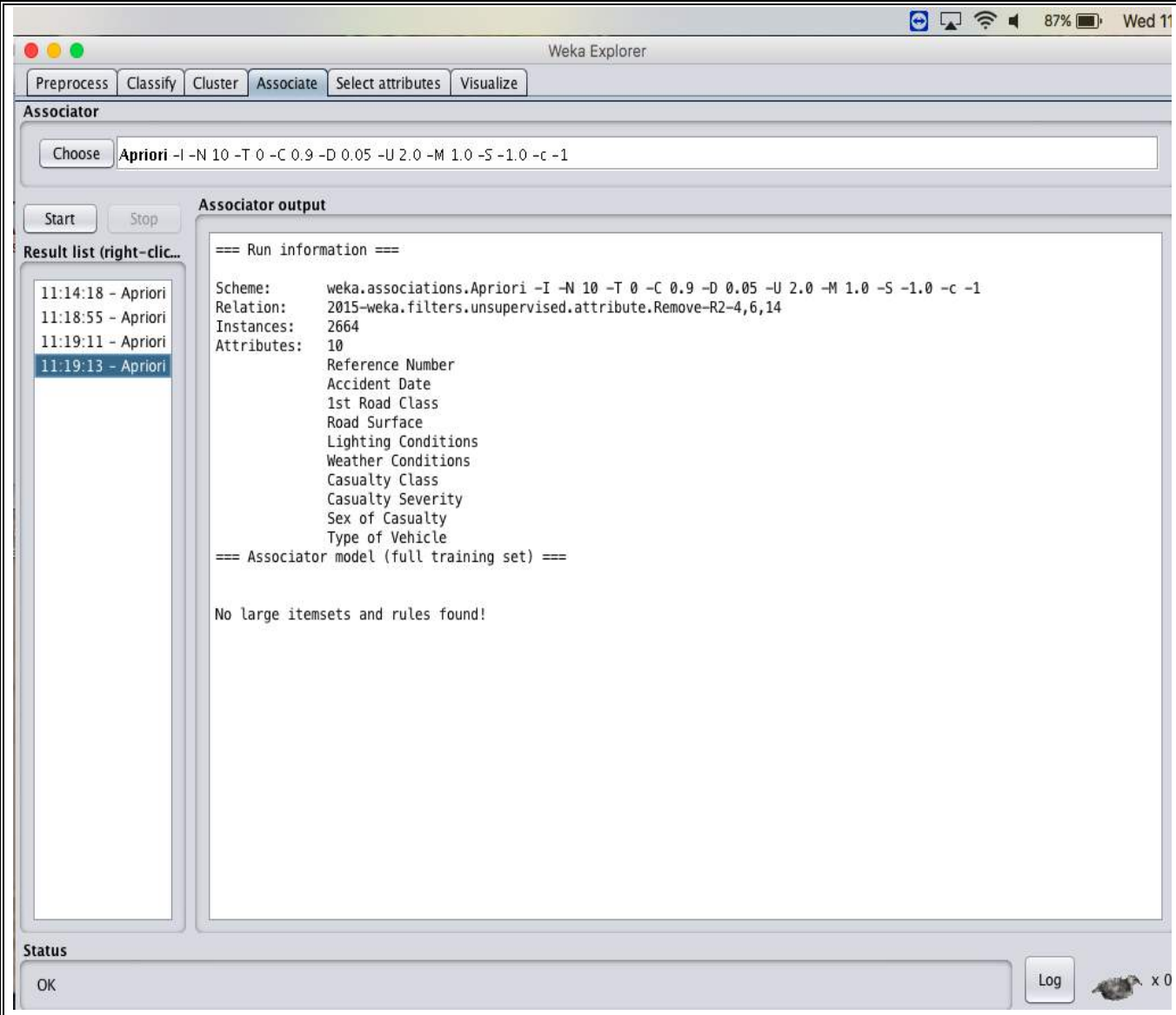

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -I -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Start
Stop
Result list (right-clic...
11:14:18 - Apriori
Associator output
Apriori
=======
Minimum support: 0.4 (1066 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 12
Generated sets of large itemsets:
Size of set of large itemsets L(1): 9
Large Itemsets L(1):
1st Road Class=Unclassified 1346
Road Surface=Dry 1908
Lighting Conditions=Daylight: street lights present 1999
Weather Conditions=Fine without high winds 2224
Casualty Class=Driver/Rider 1597
Casualty Severity=Slight 2326
Sex of Casualty=Male 1537
Sex of Casualty=Female 1127
Type of Vehicle=Car 1727
Size of set of large itemsets L(2): 20
Large Itemsets L(2):
1st Road Class=Unclassified Weather Conditions=Fine without high winds 1130
1st Road Class=Unclassified Casualty Severity=Slight 1168
Road Surface=Dry Lighting Conditions=Daylight: street lights present 1560
Road Surface=Dry Weather Conditions=Fine without high winds 1853
Road Surface=Dry Casualty Class=Driver/Rider 1149
Road Surface=Dry Casualty Severity=Slight 1662
Road Surface=Dry Sex of Casualty=Male 1093
Road Surface=Dry Type of Vehicle=Car 1176
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -I -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Start
Stop
Result list (right-clic...
11:14:18 - Apriori
Associator output
Road Surface=Dry Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds 1520
Road Surface=Dry Lighting Conditions=Daylight: street lights present Casualty Severity=Slight 1366
Road Surface=Dry Weather Conditions=Fine without high winds Casualty Class=Driver/Rider 1121
Road Surface=Dry Weather Conditions=Fine without high winds Casualty Severity=Slight 1611
Road Surface=Dry Weather Conditions=Fine without high winds Sex of Casualty=Male 1069
Road Surface=Dry Weather Conditions=Fine without high winds Type of Vehicle=Car 1136
Road Surface=Dry Casualty Severity=Slight Type of Vehicle=Car 1067
Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds Casualty Class=Driver/
Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds Casualty Severity=Slig
Lighting Conditions=Daylight: street lights present Casualty Class=Driver/Rider Casualty Severity=Slight 1097
Lighting Conditions=Daylight: street lights present Casualty Severity=Slight Type of Vehicle=Car 1136
Weather Conditions=Fine without high winds Casualty Class=Driver/Rider Casualty Severity=Slight 1184
Weather Conditions=Fine without high winds Casualty Severity=Slight Sex of Casualty=Male 1066
Weather Conditions=Fine without high winds Casualty Severity=Slight Type of Vehicle=Car 1271
Size of set of large itemsets L(4): 1
Large Itemsets L(4):
Road Surface=Dry Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds Casua
Best rules found:
1. Road Surface=Dry Sex of Casualty=Male 1093 ==> Weather Conditions=Fine without high winds 1069 <conf:(0.98)>
2. Road Surface=Dry Casualty Class=Driver/Rider 1149 ==> Weather Conditions=Fine without high winds 1121 <conf:(
3. Road Surface=Dry Lighting Conditions=Daylight: street lights present 1560 ==> Weather Conditions=Fine without hig
4. Road Surface=Dry Lighting Conditions=Daylight: street lights present Casualty Severity=Slight 1366 ==> Weather Co
5. Road Surface=Dry 1908 ==> Weather Conditions=Fine without high winds 1853 <conf:(0.97)> lift:(1.16) lev:(0.1)
6. Road Surface=Dry Casualty Severity=Slight 1662 ==> Weather Conditions=Fine without high winds 1611 <conf:(0.97
7. Road Surface=Dry Type of Vehicle=Car 1176 ==> Weather Conditions=Fine without high winds 1136 <conf:(0.97)> li
8. Lighting Conditions=Daylight: street lights present Type of Vehicle=Car 1244 ==> Casualty Severity=Slight 1136
9. Type of Vehicle=Car 1727 ==> Casualty Severity=Slight 1568 <conf:(0.91)> lift:(1.04) lev:(0.02) [60] conv:(1.3
10. Road Surface=Dry Type of Vehicle=Car 1176 ==> Casualty Severity=Slight 1067 <conf:(0.91)> lift:(1.04) lev:(0.0
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associat
Associator
Choose
Apriori -I -N 10 -T 0 -C 0.9
Start
Stop
Result list (right-clic...
11:14:18 - Apriori
11:18:55 - Apriori
11:19:11 - Apriori
11:19:13 - Apriori
11:32:43 - Apriori
Associator outp
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.associations.Apriori
About
Class implementing an Apriori-type algorithm.
More
Capabilities
car
False
classIndex
-1
delta
0.05
doNotCheckCapabilities
False
lowerBoundMinSupport
0.5
metricType
Confidence
minMetric
0.9
numRules
10
outputItemSets
True
removeAllMissingCols
False
significanceLevel
-1.0
treatZeroAsMissing
False
upperBoundMinSupport
1.0
verbose
False
Open...
Save...
OK
Cancel


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -I -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.5 -S -1.0 -c -1
Start
Stop
Result list (right-clic...
11:14:18 - Apriori
11:18:55 - Apriori
11:19:11 - Apriori
11:19:13 - Apriori
11:32:43 - Apriori
Associator output
Casualty Severity=Slight 2326
Sex of Casualty=Male 1537
Type of Vehicle=Car 1727
Size of set of large itemsets L(2): 10
Large Itemsets L(2):
Road Surface=Dry Lighting Conditions=Daylight: street lights present 1560
Road Surface=Dry Weather Conditions=Fine without high winds 1853
Road Surface=Dry Casualty Severity=Slight 1662
Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds 1739
Lighting Conditions=Daylight: street lights present Casualty Severity=Slight 1751
Weather Conditions=Fine without high winds Casualty Class=Driver/Rider 1343
Weather Conditions=Fine without high winds Casualty Severity=Slight 1934
Weather Conditions=Fine without high winds Type of Vehicle=Car 1404
Casualty Class=Driver/Rider Casualty Severity=Slight 1412
Casualty Severity=Slight Type of Vehicle=Car 1568
Size of set of large itemsets L(3): 4
Large Itemsets L(3):
Road Surface=Dry Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds 1520
Road Surface=Dry Lighting Conditions=Daylight: street lights present Casualty Severity=Slight 1366
Road Surface=Dry Weather Conditions=Fine without high winds Casualty Severity=Slight 1611
Lighting Conditions=Daylight: street lights present Weather Conditions=Fine without high winds Casualty Severity=Slig
Best rules found:
1. Road Surface=Dry Lighting Conditions=Daylight: street lights present 1560 ==> Weather Conditions=Fine without hig
2. Road Surface=Dry 1908 ==> Weather Conditions=Fine without high winds 1853 <conf:(0.97)> lift:(1.16) lev:(0.1)
3. Road Surface=Dry Casualty Severity=Slight 1662 ==> Weather Conditions=Fine without high winds 1611 <conf:(0.9
4. Type of Vehicle=Car 1727 ==> Casualty Severity=Slight 1568 <conf:(0.91)> lift:(1.04) lev:(0.02) [60] conv:(1.3
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associat
Associator
Choose
Apriori -I -N 10 -T 0 -C 0.9
Start
Stop
Result list (right-clic...
11:14:18 - Apriori
11:18:55 - Apriori
11:19:11 - Apriori
11:19:13 - Apriori
Associator outp
=== Run info
Scheme:
Relation:
Instances:
Attributes:
=== Associat
No large ite
S -1.0 -c -1
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.associations.Apriori
About
Class implementing an Apriori-type algorithm.
More
Capabilities
car
False
classIndex
-1
delta
0.05
doNotCheckCapabilities
False
lowerBoundMinSupport
1.0
metricType
Confidence
minMetric
0.9
numRules
10
outputItemSets
True
removeAllMissingCols
False
significanceLevel
-1.0
treatZeroAsMissing
False
upperBoundMinSupport
2.0
verbose
False
Open...
Save...
OK
Cancel



- **USING R-TOOL :**

  STEPS INVOLVED :

  - Choose the dataset and import the dataset into the R-tool.
  - View the dataset and start inserting queries for the Apriori algorithm.

QUERIES :

- Data=read.csv("C:/users/prasanthiemani/Desktop/2015.csv")
- View(Data)
- a = apriori(data, parameter = list(sup=0.3, conf=0.9)

```
> a=apriori(data,parameter = list(sup=0.3,conf=0.9))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
        0.9    0.1    1 none FALSE            TRUE       5     0.3      1     10
 target   ext
  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 799

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[2393 item(s), 2664 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [30 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>
```

RESULT :

Thus, the Apriori algorithm analyzing using both the weka tool and R- tool has been successfully completed. In case of weka tool, the change in upper bound and lower bound values lead to the increase and decrease of number of itemsets and rules . In case of R-tool, there is an increase in absolute minimum support count value.

EX.No: 10
Date :

FREQUENT PATTERN MINING USING FP GROWTH THROUGH WEKA TOOL

PROBLEM STATEMENT :

Run the FP growth algorithm, and explore the association rules by changing the following parameters:

a) Upper bound min_sup
b) Lower bound min_sup
c) Metric type

DESCRIPTION :

Consider a dataset of 2015.csv file of which it contains the attributes are Reference Number, Grid ref: Easting, Grid Ref: Northing, Number of vehicles, Accident date, Time(24 hr), 1st Road class, Road Surface, Lighting conditions, Weather conditions, casuality class, Sex of casuality, Age of casuality, Type of casuality for the performance of the dataset by applying the FP algorithm in weka tool.

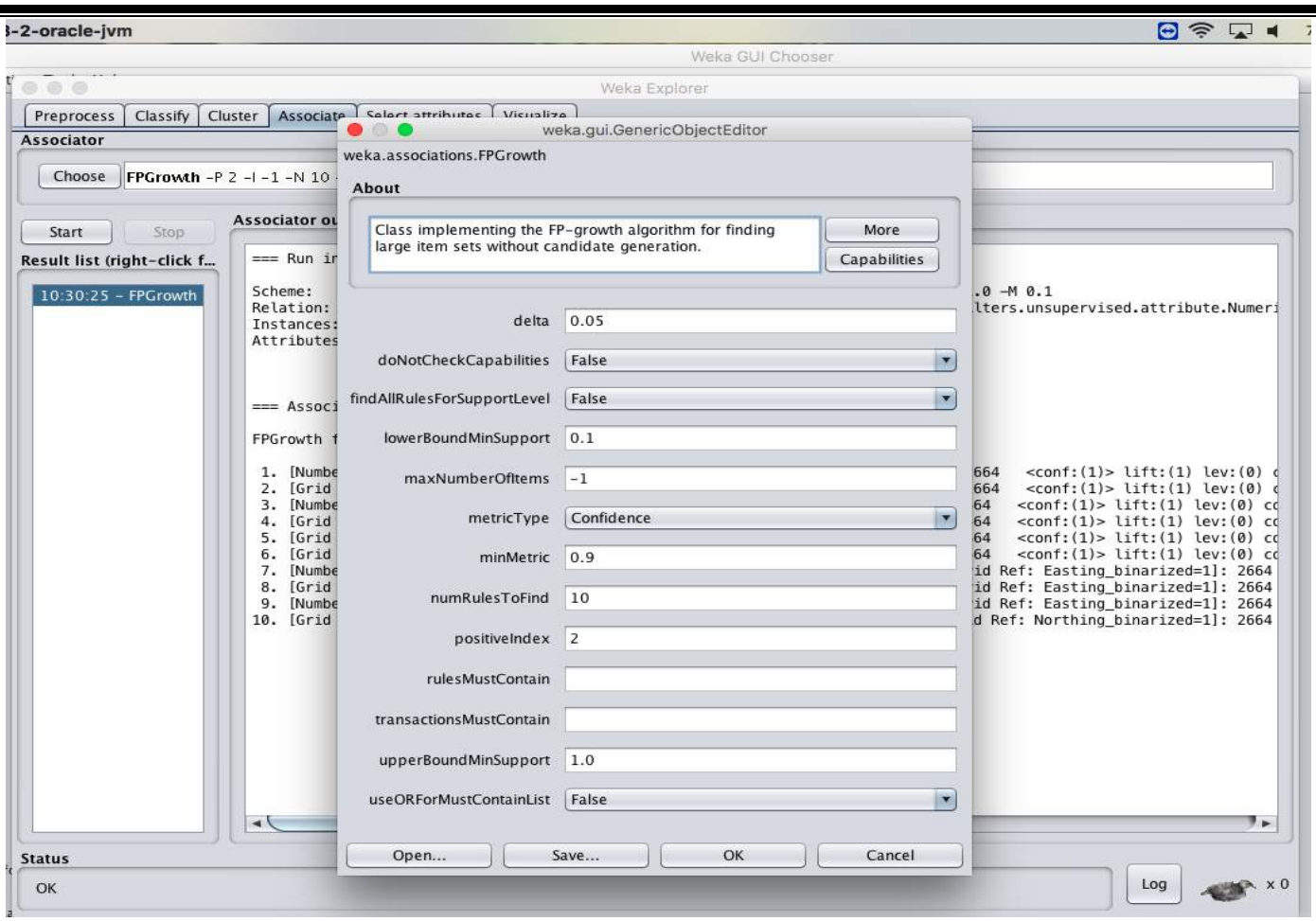
❖ **USING WEKA TOOL :**

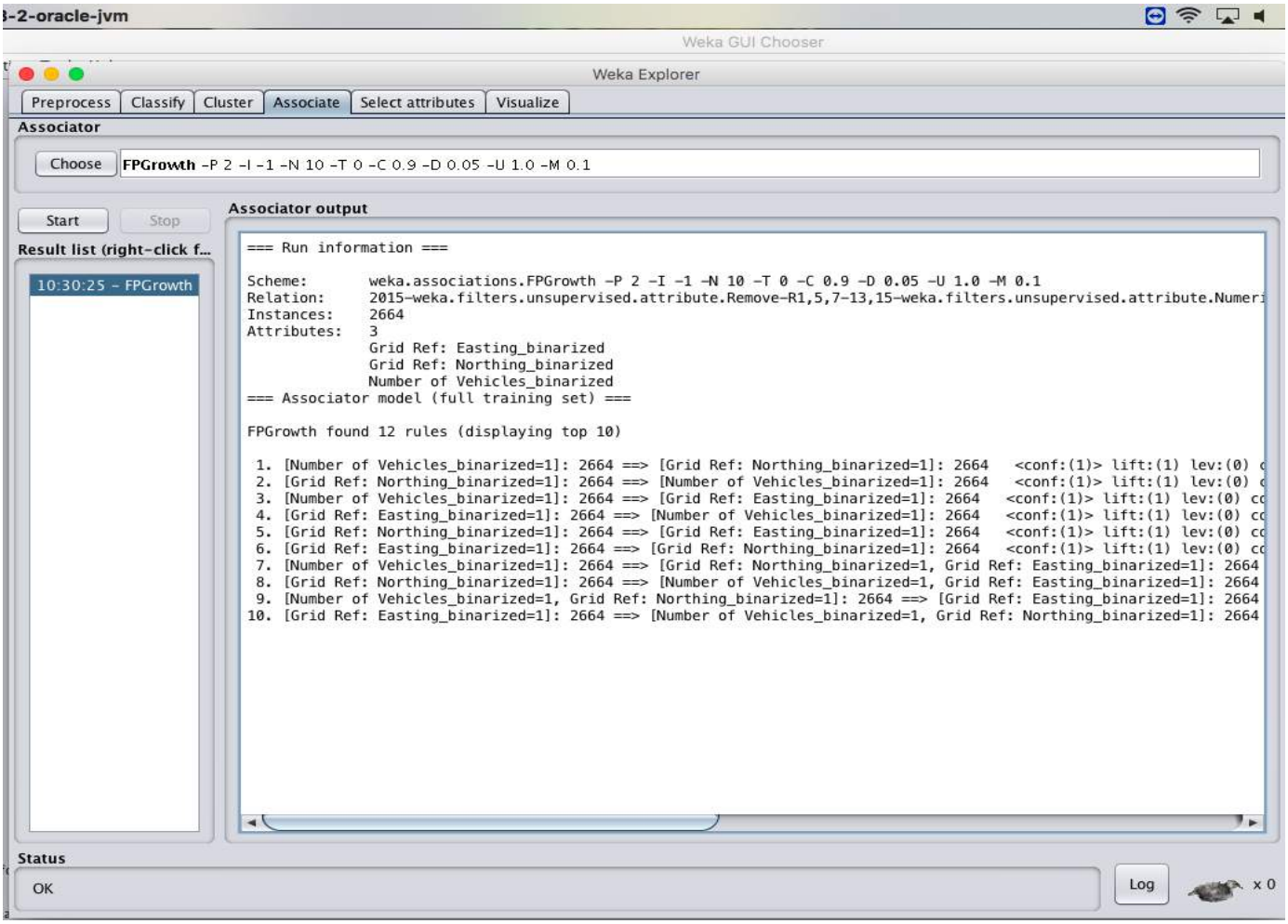
STEPS INVOLVED :

- Choose a set of attributes for clustering and for giving a motivation.
- Choose the dataset and import the dataset into Weka tool.
- Discretize the attributes from all data types to nominal to perform the algorithm.
- Associate the attributes with the FP growth algorithm.
- Set the Upper bound min_sup and lower bound min_sup values.

OBSERVATIONS :

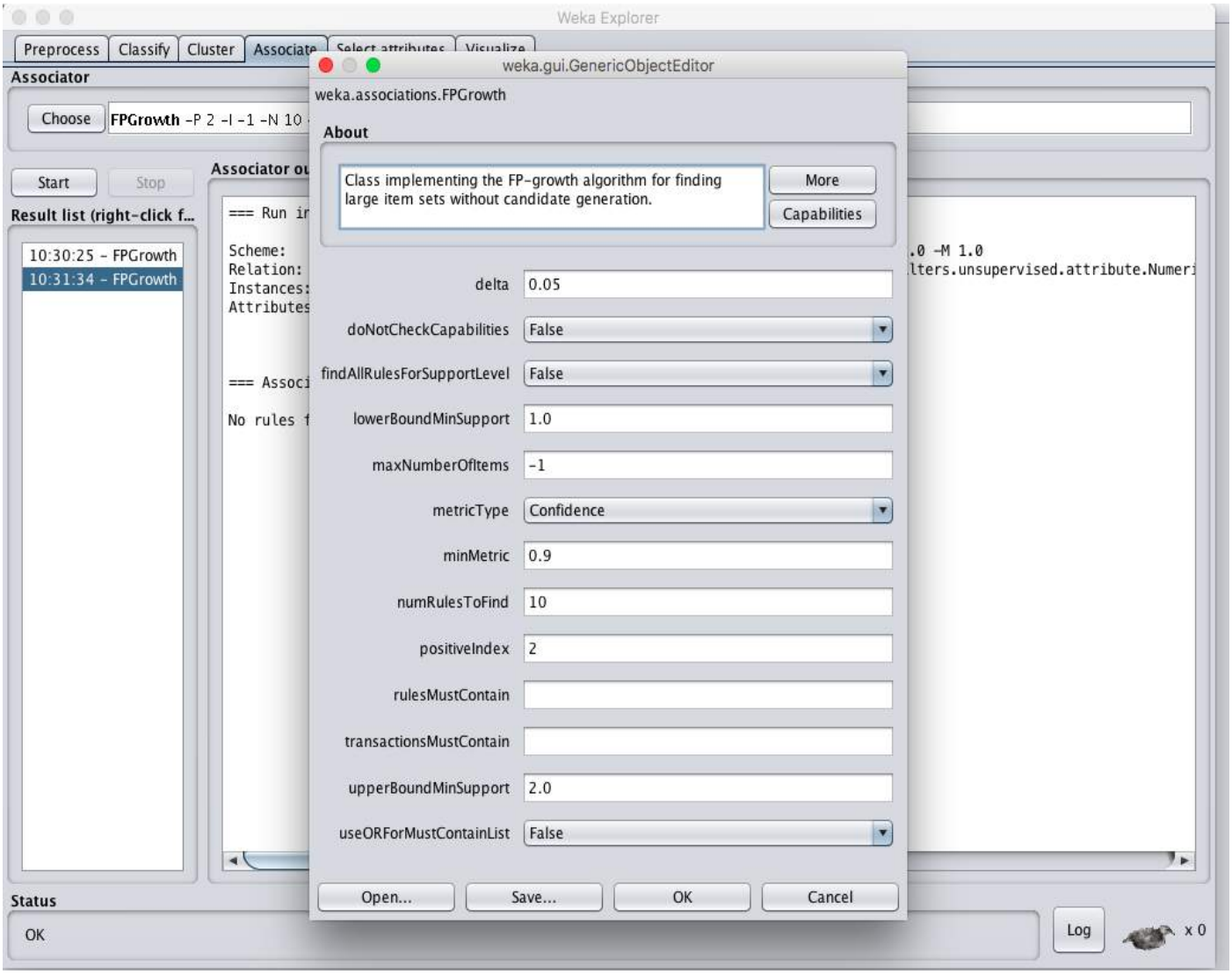
1) When the association rules are of values:
   a) Upper bound min_sup = 1.0
   b) Lower bound min_sup = 0.1
   c) Metric type = confidence.

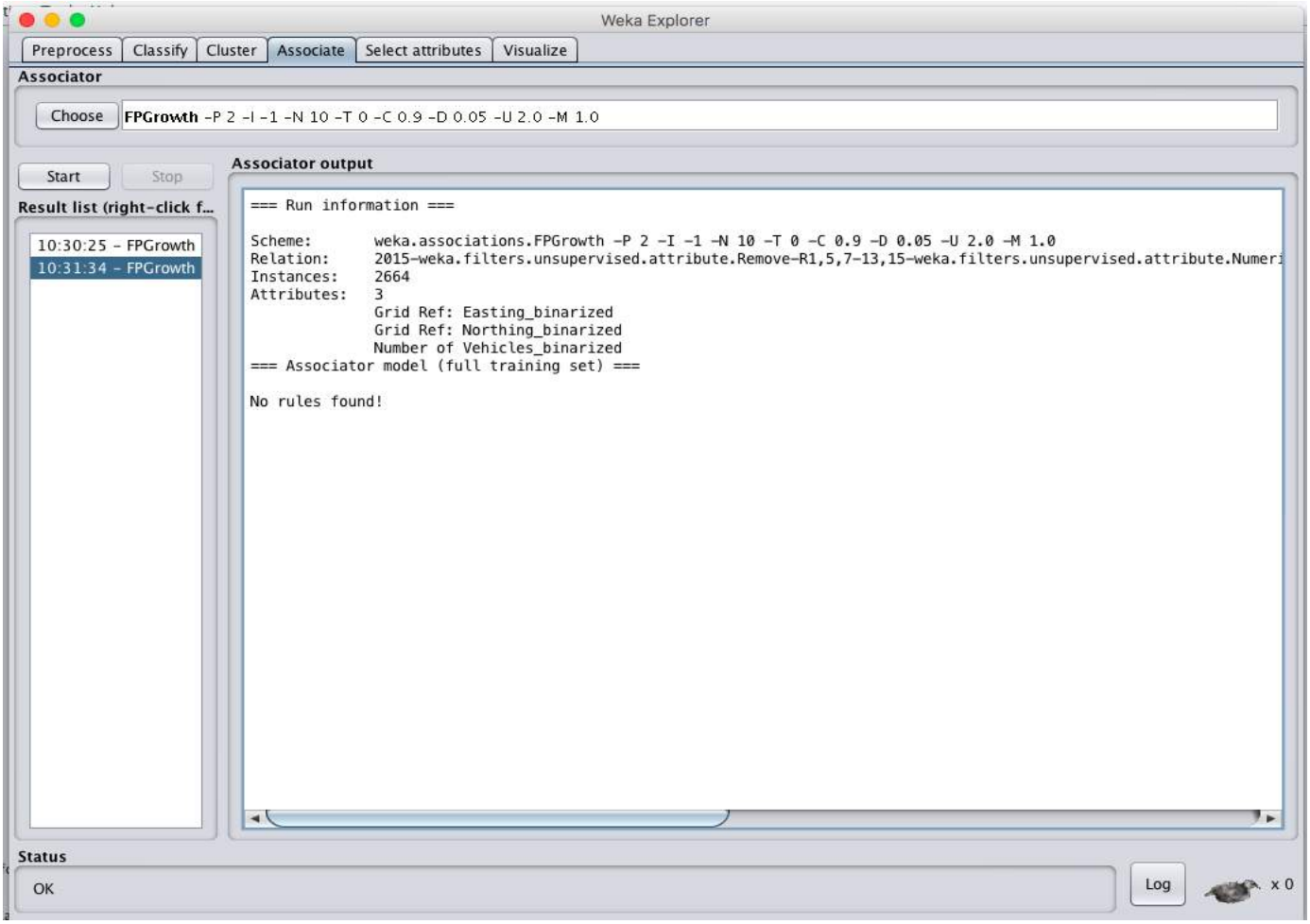

Weka GUI Chooser
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
FPGrowth -P 2 -I -1 -N 10
Start
Stop
Result list (right-click f...
10:30:25 - FPGrowth
Associator ou
=== Run in
Scheme:
Relation:
Instances:
Attributes
=== Associ
FPGrowth f
1. [Numbe
2. [Grid
3. [Numbe
4. [Grid
5. [Grid
6. [Grid
7. [Numbe
8. [Grid
9. [Numbe
10. [Grid
.0 -M 0.1
lters.unsupervised.attribute.Numeri
664 <conf:(1)> lift:(1) lev:(0) c
664 <conf:(1)> lift:(1) lev:(0) c
64 <conf:(1)> lift:(1) lev:(0) co
64 <conf:(1)> lift:(1) lev:(0) co
64 <conf:(1)> lift:(1) lev:(0) co
64 <conf:(1)> lift:(1) lev:(0) co
id Ref: Easting_binarized=1]: 2664
id Ref: Easting_binarized=1]: 2664
id Ref: Easting_binarized=1]: 2664
d Ref: Northing_binarized=1]: 2664
weka.gui.GenericObjectEditor
weka.associations.FPGrowth
About
Class implementing the FP-growth algorithm for finding large item sets without candidate generation.
More
Capabilities
delta 0.05
doNotCheckCapabilities False
findAllRulesForSupportLevel False
lowerBoundMinSupport 0.1
maxNumberOfItems -1
metricType Confidence
minMetric 0.9
numRulesToFind 10
positiveIndex 2
rulesMustContain
transactionsMustContain
upperBoundMinSupport 1.0
useORForMustContainList False
Open...
Save...
OK
Cancel
Status
OK
Log
x 0


Weka GUI Chooser
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Start
Stop
Result list (right-click f...
10:30:25 - FPGrowth
Associator output
=== Run information ===
Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation: 2015-weka.filters.unsupervised.attribute.Remove-R1,5,7-13,15-weka.filters.unsupervised.attribute.Numeri
Instances: 2664
Attributes: 3
Grid Ref: Easting_binarized
Grid Ref: Northing_binarized
Number of Vehicles_binarized
=== Associator model (full training set) ===
FPGrowth found 12 rules (displaying top 10)
1. [Number of Vehicles_binarized=1]: 2664 ==> [Grid Ref: Northing_binarized=1]: 2664 <conf:(1)> lift:(1) lev:(0) c
2. [Grid Ref: Northing_binarized=1]: 2664 ==> [Number of Vehicles_binarized=1]: 2664 <conf:(1)> lift:(1) lev:(0) c
3. [Number of Vehicles_binarized=1]: 2664 ==> [Grid Ref: Easting_binarized=1]: 2664 <conf:(1)> lift:(1) lev:(0) co
4. [Grid Ref: Easting_binarized=1]: 2664 ==> [Number of Vehicles_binarized=1]: 2664 <conf:(1)> lift:(1) lev:(0) co
5. [Grid Ref: Northing_binarized=1]: 2664 ==> [Grid Ref: Easting_binarized=1]: 2664 <conf:(1)> lift:(1) lev:(0) co
6. [Grid Ref: Easting_binarized=1]: 2664 ==> [Grid Ref: Northing_binarized=1]: 2664 <conf:(1)> lift:(1) lev:(0) co
7. [Number of Vehicles_binarized=1]: 2664 ==> [Grid Ref: Northing_binarized=1, Grid Ref: Easting_binarized=1]: 2664
8. [Grid Ref: Northing_binarized=1]: 2664 ==> [Number of Vehicles_binarized=1, Grid Ref: Easting_binarized=1]: 2664
9. [Number of Vehicles_binarized=1, Grid Ref: Northing_binarized=1]: 2664 ==> [Grid Ref: Easting_binarized=1]: 2664
10. [Grid Ref: Easting_binarized=1]: 2664 ==> [Number of Vehicles_binarized=1, Grid Ref: Northing_binarized=1]: 2664
Status
OK
Log
x 0

2) When the association rules are of values:

a) Upper bound min_sup = 2.0
b) Lower bound min_sup = 1.0
c) Metric type = confidence.





Through the comparison of both the cases 1 and 2, the rules will be changed according to the values of upper bound minimum support and lower bound minimum support. If the upperbound minimum support and lower bound minimum support are high then the number of rules will be very less and its vice-versa in the case of less upper bound and lower bound minimum support values.

RESULT :

Thus, the analysis of FP growth algorithm using weka tool has been successfully completed. Incase of changing the upper bound and lower bound values there is a change in the number of rules that are found.

EX.No: 11
Date :

PREDICTION OF CATEGORICAL DATA USING DECISION TREE ALGORTIHM THROUGH WEKA

PROBLEM STATEMENT :

Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. credit dataset (original) Excel spreadsheet version of the German credit data (Down load from web).
In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer !)

1) What attributes do you think might be crucial in making the credit assessement ? Come up with some simple rules in plain English using your selected attributes.
2) One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.
3) Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly ? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy ?
4) Is testing on the training set as you did above a good idea ? Why or Why not ?
5) One approach for solving the problem encountered in the previous question is using cross - validation ? Describe what is cross-validation briefly. Train a Decistion Tree again using cross-validation and report your results. Does your accuracy increase/decrease ?Why ?

DESCRIPTION :

1. **What attributes do you think might be crucial in making the credit assessement ? Come up with some simple rules in plain English using your selected attributes.**

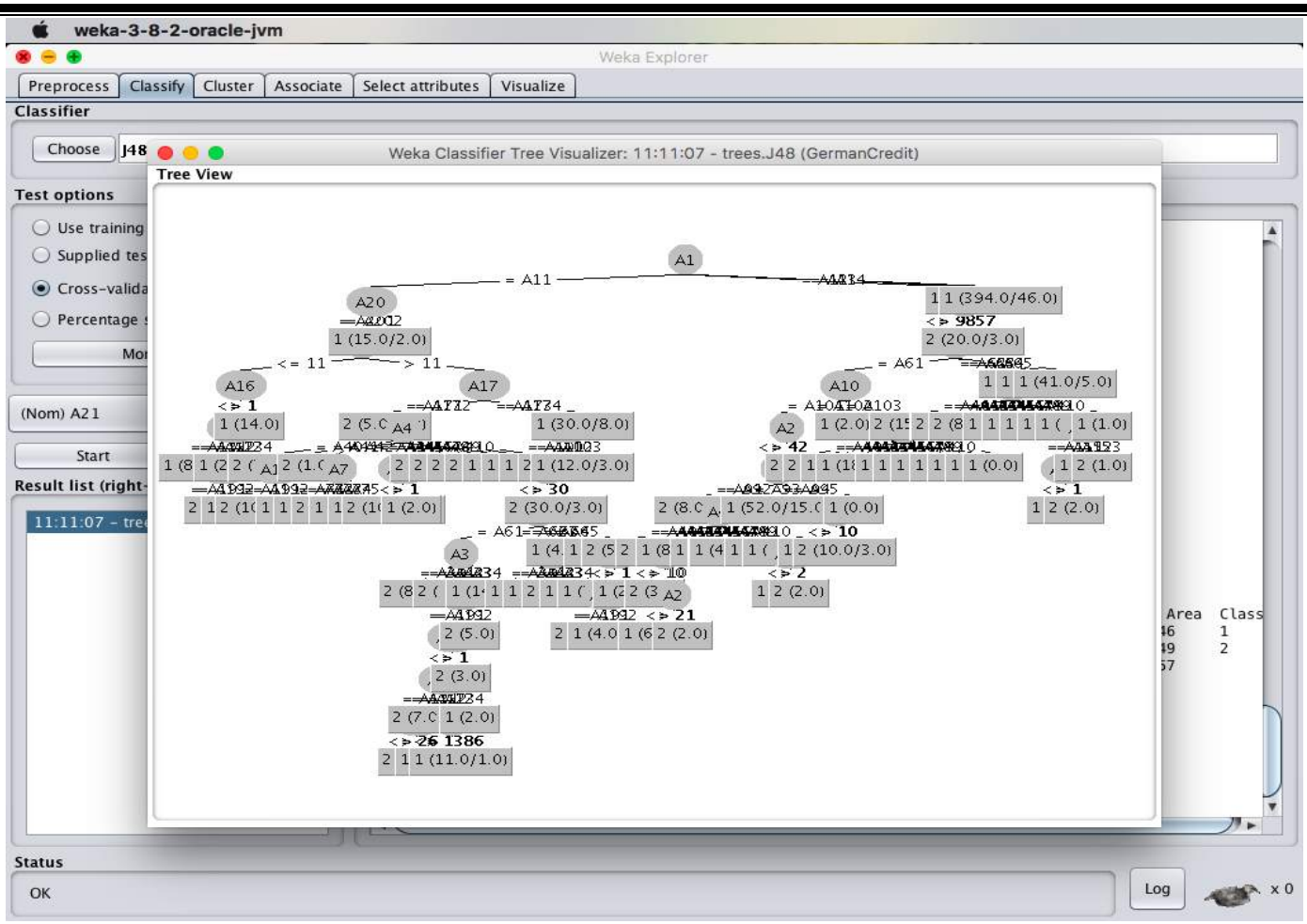
   The attributes that might be crucial in making the credit assessment are :
   - Numerical attributes
   - Nominal attributes

   ➢ Nominal and numeric attributes are the capabilities of the given dataset.

2. **One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.**

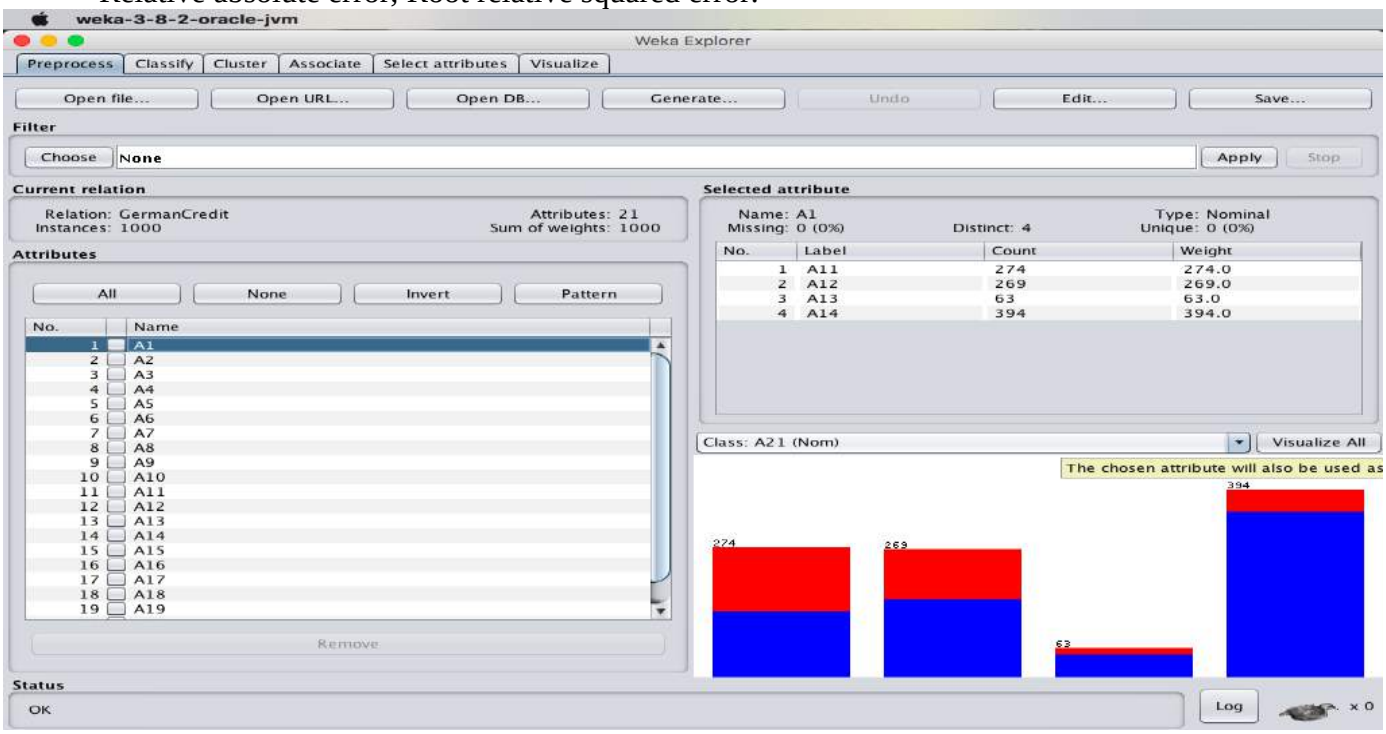
❖ **Decision Tree :**

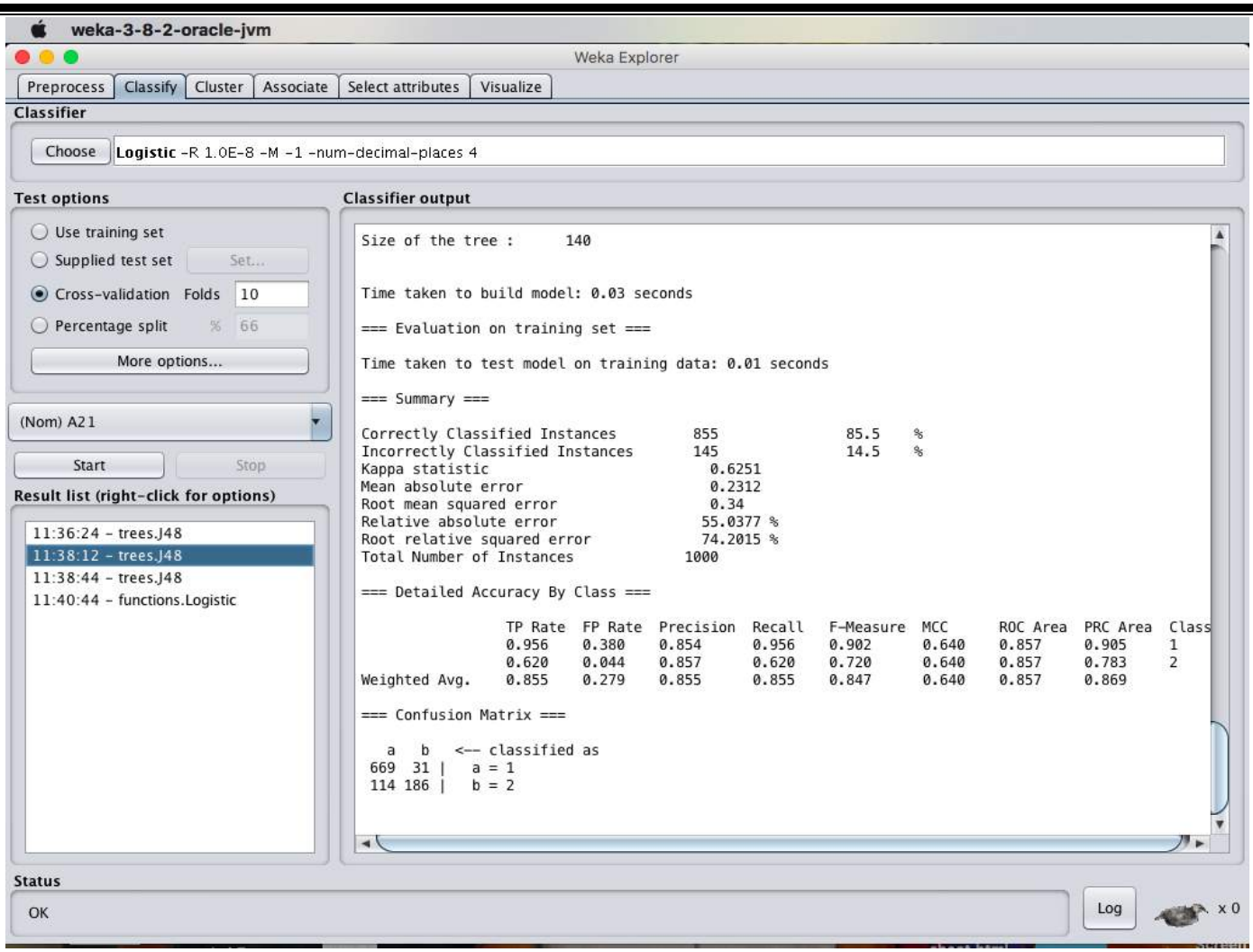
Visualize the decision tree for the given dataset.

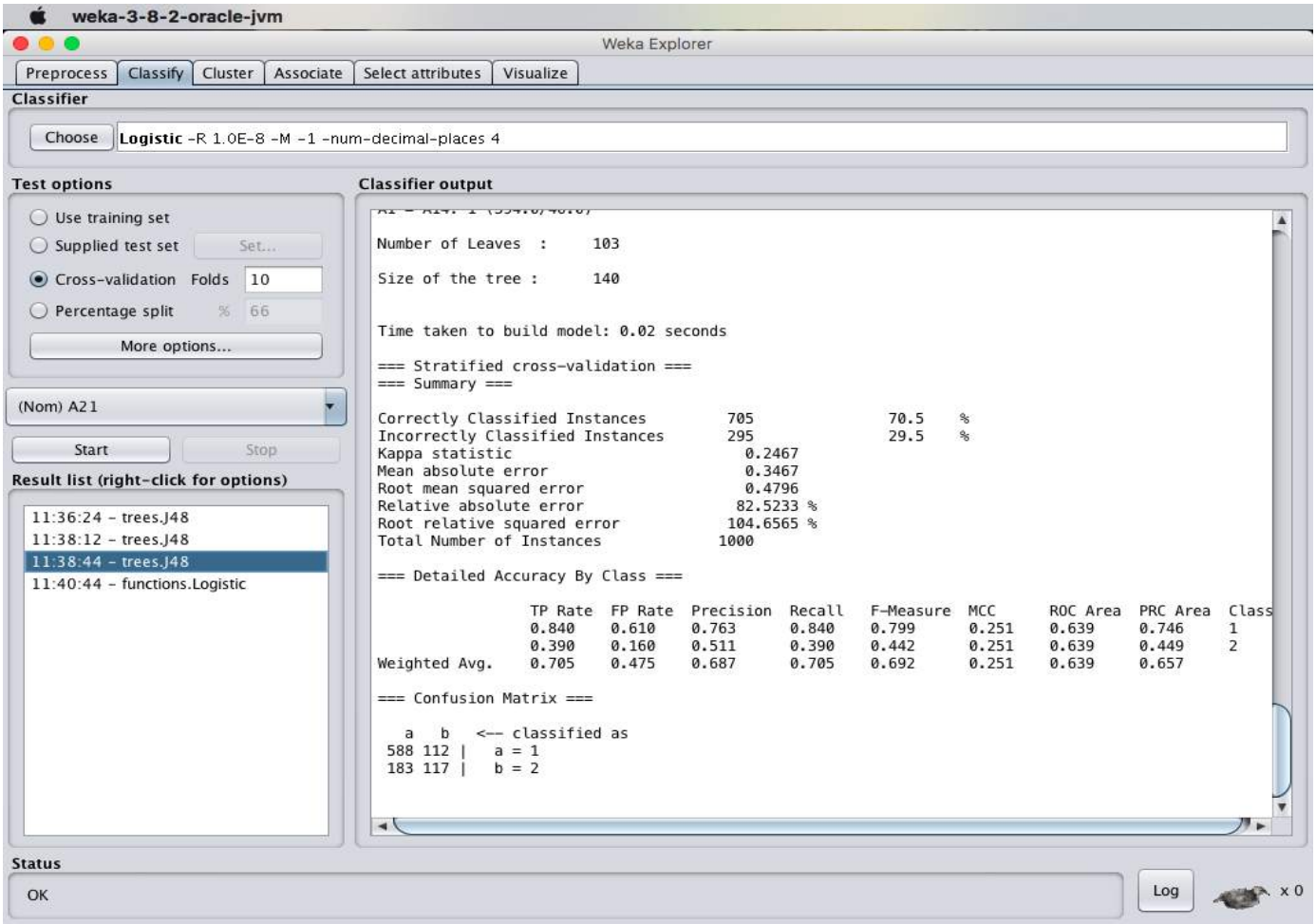


3. **Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly ? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy ?**

Usually training dataset will be evaluated by the test dataset. This can be evaluated and analyzed by the instances of the dataset and error like Mean absolute error, Root mean square error, Relative absolute error, Root relative squared error.




weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
Classifier output
Size of the tree : 140
Time taken to build model: 0.03 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.01 seconds
=== Summary ===
Correctly Classified Instances 855 85.5 %
Incorrectly Classified Instances 145 14.5 %
Kappa statistic 0.6251
Mean absolute error 0.2312
Root mean squared error 0.34
Relative absolute error 55.0377 %
Root relative squared error 74.2015 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.956 0.380 0.854 0.956 0.902 0.640 0.857 0.905 1
0.620 0.044 0.857 0.620 0.720 0.640 0.857 0.783 2
Weighted Avg. 0.855 0.279 0.855 0.855 0.847 0.640 0.857 0.869
=== Confusion Matrix ===
a b <-- classified as
669 31 | a = 1
114 186 | b = 2
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
Classifier output
Number of Leaves : 103
Size of the tree : 140
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 705 70.5 %
Incorrectly Classified Instances 295 29.5 %
Kappa statistic 0.2467
Mean absolute error 0.3467
Root mean squared error 0.4796
Relative absolute error 82.5233 %
Root relative squared error 104.6565 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.840 0.610 0.763 0.840 0.799 0.251 0.639 0.746 1
0.390 0.160 0.511 0.390 0.442 0.251 0.639 0.449 2
Weighted Avg. 0.705 0.475 0.687 0.705 0.692 0.251 0.639 0.657
=== Confusion Matrix ===
a b <-- classified as
588 112 | a = 1
183 117 | b = 2
Status
OK
Log
x 0

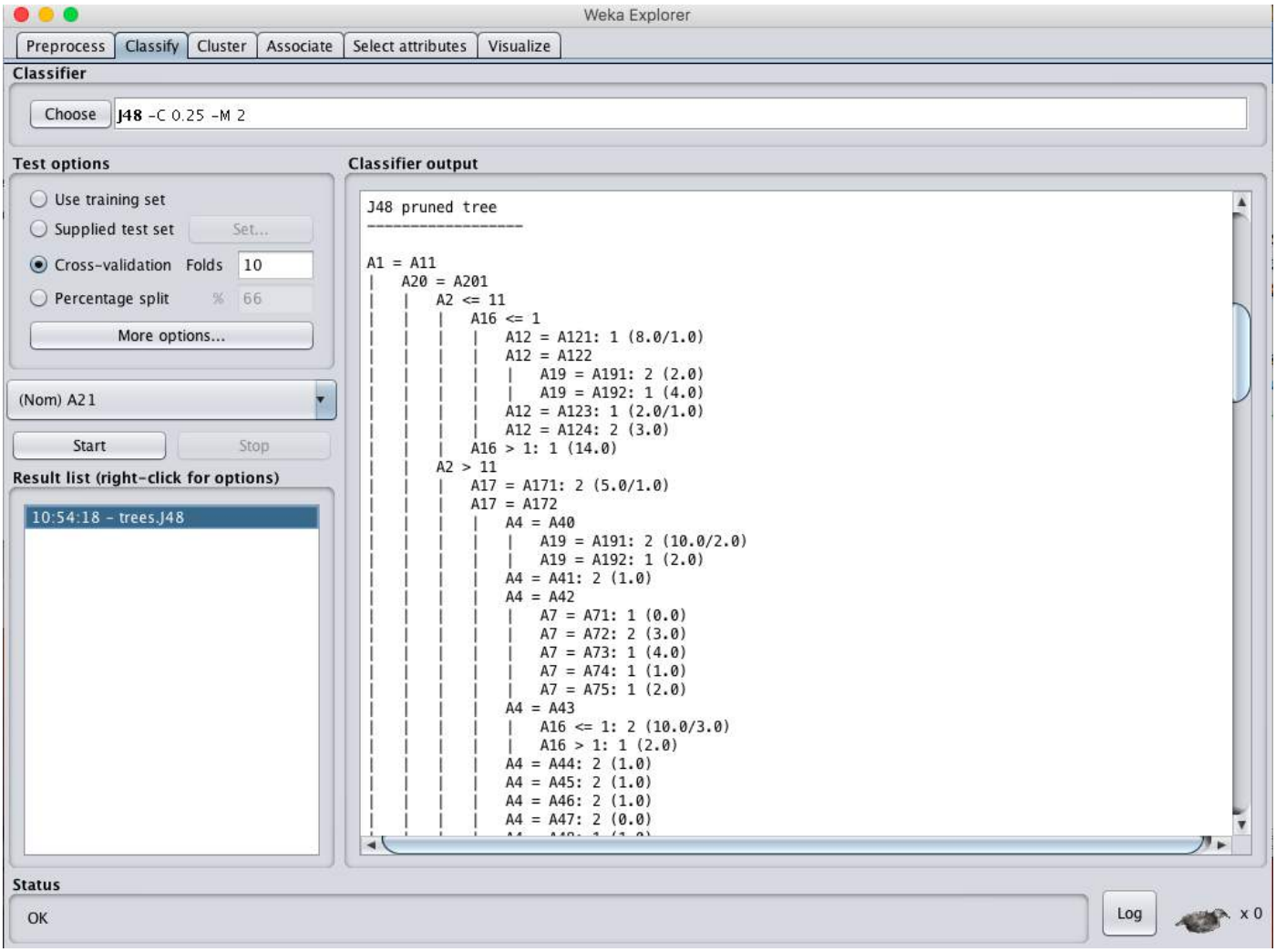
4. **Is testing on the training set as you did above a good idea ? Why or Why not ?**

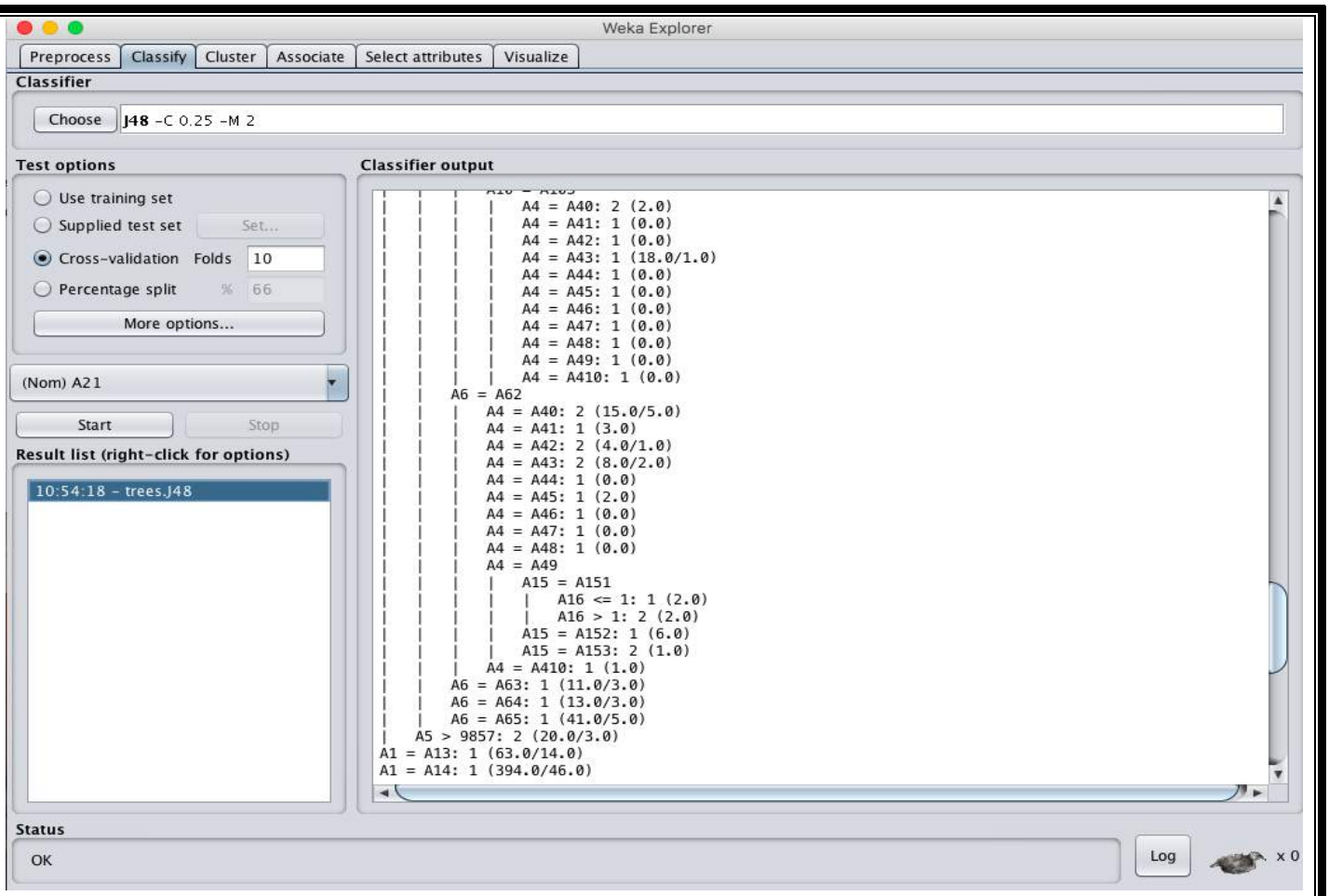
Testing on the training dataset is a good idea where it helps to increase the accuracy of your model by decreasing its complexity. In this case of the dataset, you can prune tree after training. This will decrease the amount of specification in the specific training dataset and increase generalisation on unseen data.

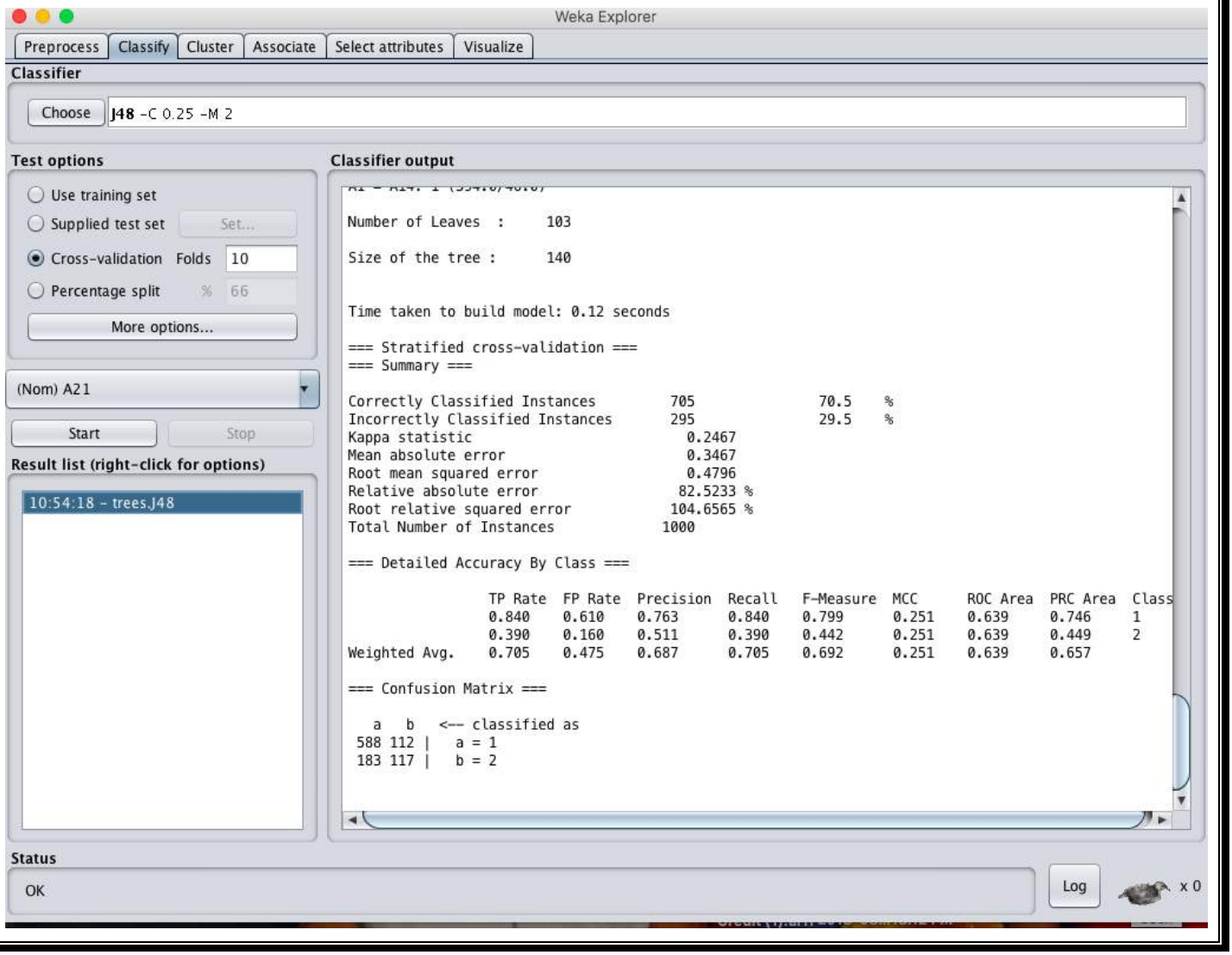
5. **One approach for solving the problem encountered in the previous question is using cross - validation ? Describe what is cross-validation briefly. Train a Decistion Tree again using cross-validation and report your results. Does your accuracy increase/decrease ?Why ?**

- **CROSS VALIDATION ANALYSIS :**

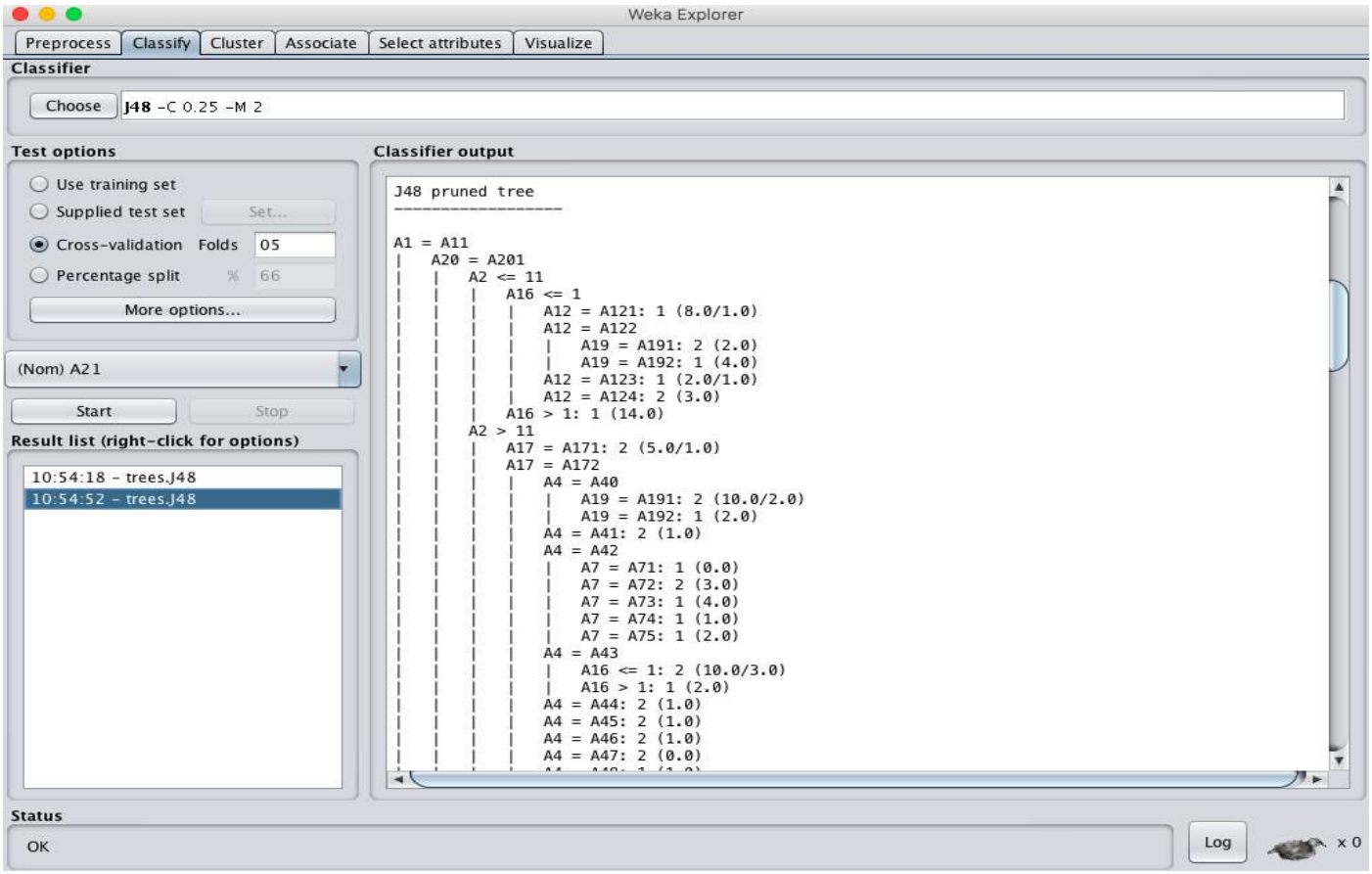
  - **When cross validation folds are 10 :**

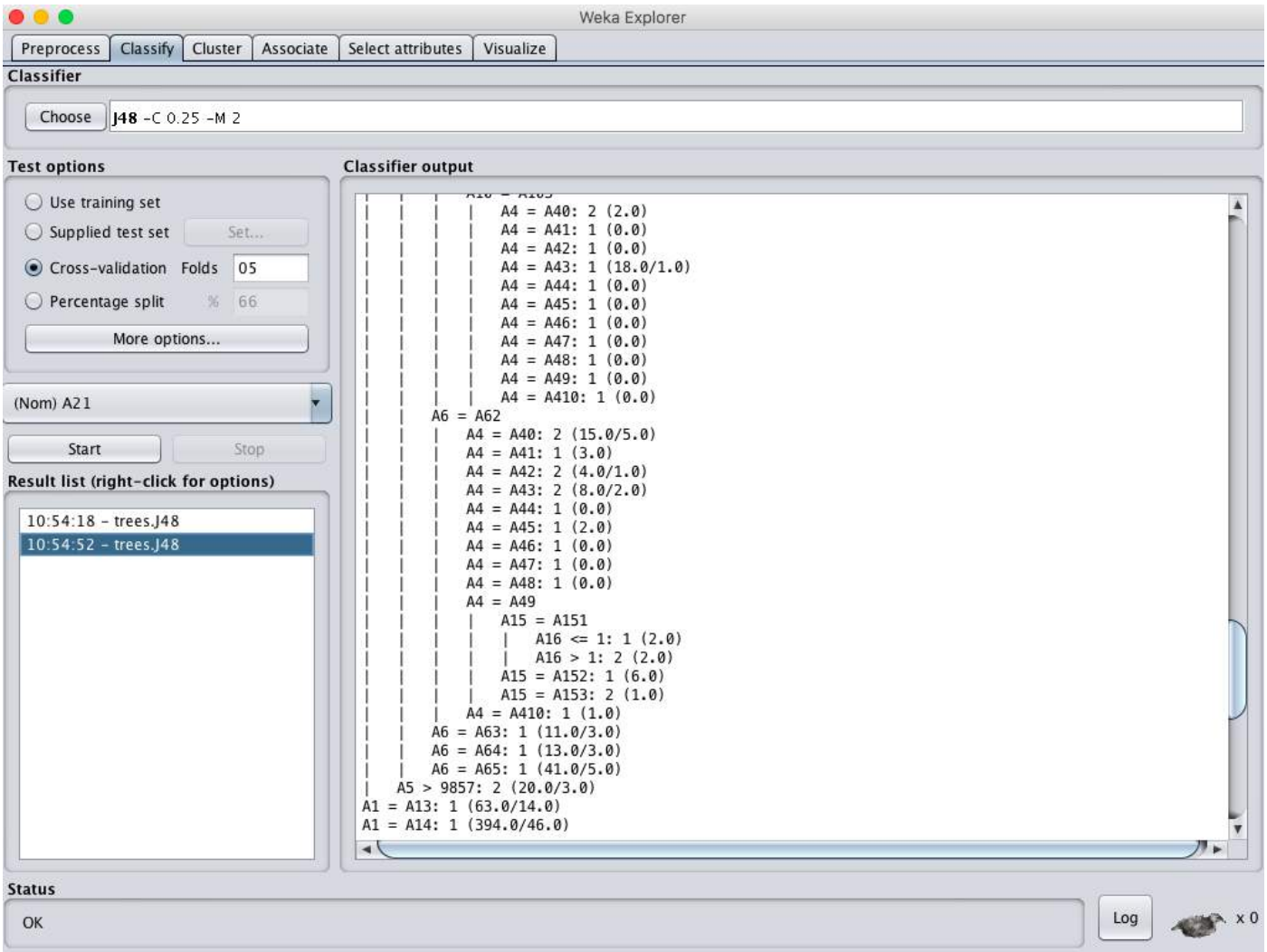


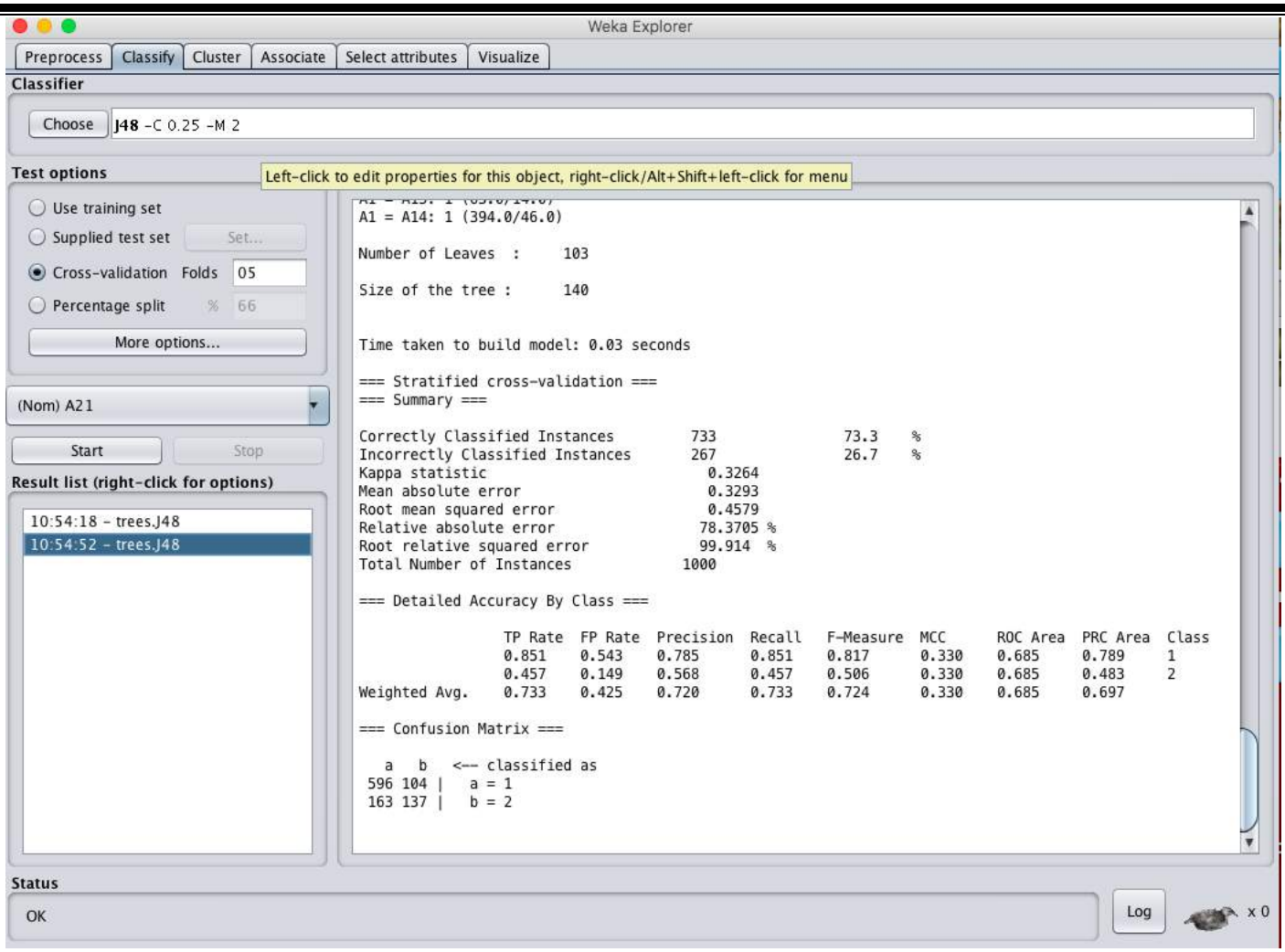

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
10:54:18 - trees.J48
Classifier output
| | | | A4 = A40: 2 (2.0)
| | | | A4 = A41: 1 (0.0)
| | | | A4 = A42: 1 (0.0)
| | | | A4 = A43: 1 (18.0/1.0)
| | | | A4 = A44: 1 (0.0)
| | | | A4 = A45: 1 (0.0)
| | | | A4 = A46: 1 (0.0)
| | | | A4 = A47: 1 (0.0)
| | | | A4 = A48: 1 (0.0)
| | | | A4 = A49: 1 (0.0)
| | | | A4 = A410: 1 (0.0)
| | A6 = A62
| | | A4 = A40: 2 (15.0/5.0)
| | | A4 = A41: 1 (3.0)
| | | A4 = A42: 2 (4.0/1.0)
| | | A4 = A43: 2 (8.0/2.0)
| | | A4 = A44: 1 (0.0)
| | | A4 = A45: 1 (2.0)
| | | A4 = A46: 1 (0.0)
| | | A4 = A47: 1 (0.0)
| | | A4 = A48: 1 (0.0)
| | | A4 = A49
| | | | A15 = A151
| | | | | A16 <= 1: 1 (2.0)
| | | | | A16 > 1: 2 (2.0)
| | | | A15 = A152: 1 (6.0)
| | | | A15 = A153: 2 (1.0)
| | | A4 = A410: 1 (1.0)
| | A6 = A63: 1 (11.0/3.0)
| | A6 = A64: 1 (13.0/3.0)
| | A6 = A65: 1 (41.0/5.0)
| A5 > 9857: 2 (20.0/3.0)
A1 = A13: 1 (63.0/14.0)
A1 = A14: 1 (394.0/46.0)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
10:54:18 - trees.J48
Classifier output
Number of Leaves : 103
Size of the tree : 140
Time taken to build model: 0.12 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 705 70.5 %
Incorrectly Classified Instances 295 29.5 %
Kappa statistic 0.2467
Mean absolute error 0.3467
Root mean squared error 0.4796
Relative absolute error 82.5233 %
Root relative squared error 104.6565 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.840 0.610 0.763 0.840 0.799 0.251 0.639 0.746 1
0.390 0.160 0.511 0.390 0.442 0.251 0.639 0.449 2
Weighted Avg. 0.705 0.475 0.687 0.705 0.692 0.251 0.639 0.657
=== Confusion Matrix ===
a b <-- classified as
588 112 | a = 1
183 117 | b = 2
Status
OK
Log
x 0

- **When cross validation folds are : 05 :-**







The change in the cross validation folds leads to the change in the stratified cross validation summary which contains the correctly classified instances and incorrectly classified instances.

RESULT :

Thus, the observations and evaluations done on the german_credit dataset are analyzed. The decision tree has been successfully visualized. Various evaluations and comparisons done through the cross validation folds change. Which lead to the change of values in confusion matrix.

EX.No: 12
Date :

PREDICTION OF CATEGORICAL DATA USING SMO ALGORTIHM THROUGH WEKA

PROBLEM STATEMENT :

Use the german credit dataset download from UCI repository and analyze the given task.

1. Do you really need to input so many attributes to get good results?
2. Compare the results obtained by decision tree and SMO ?
3. Set the cost sensitive evaluation and compare the obtained results.
4. What is the significance of the following parameters :
    a) Mean Absolute Error
    b) Root Mean Square Error
    c) Relative Absolute Error
    d) Total Number of Instances

DESCRIPTION :

Consider the german credit dataset which can be downloaded from the UCI repository.

ANALYSIS :

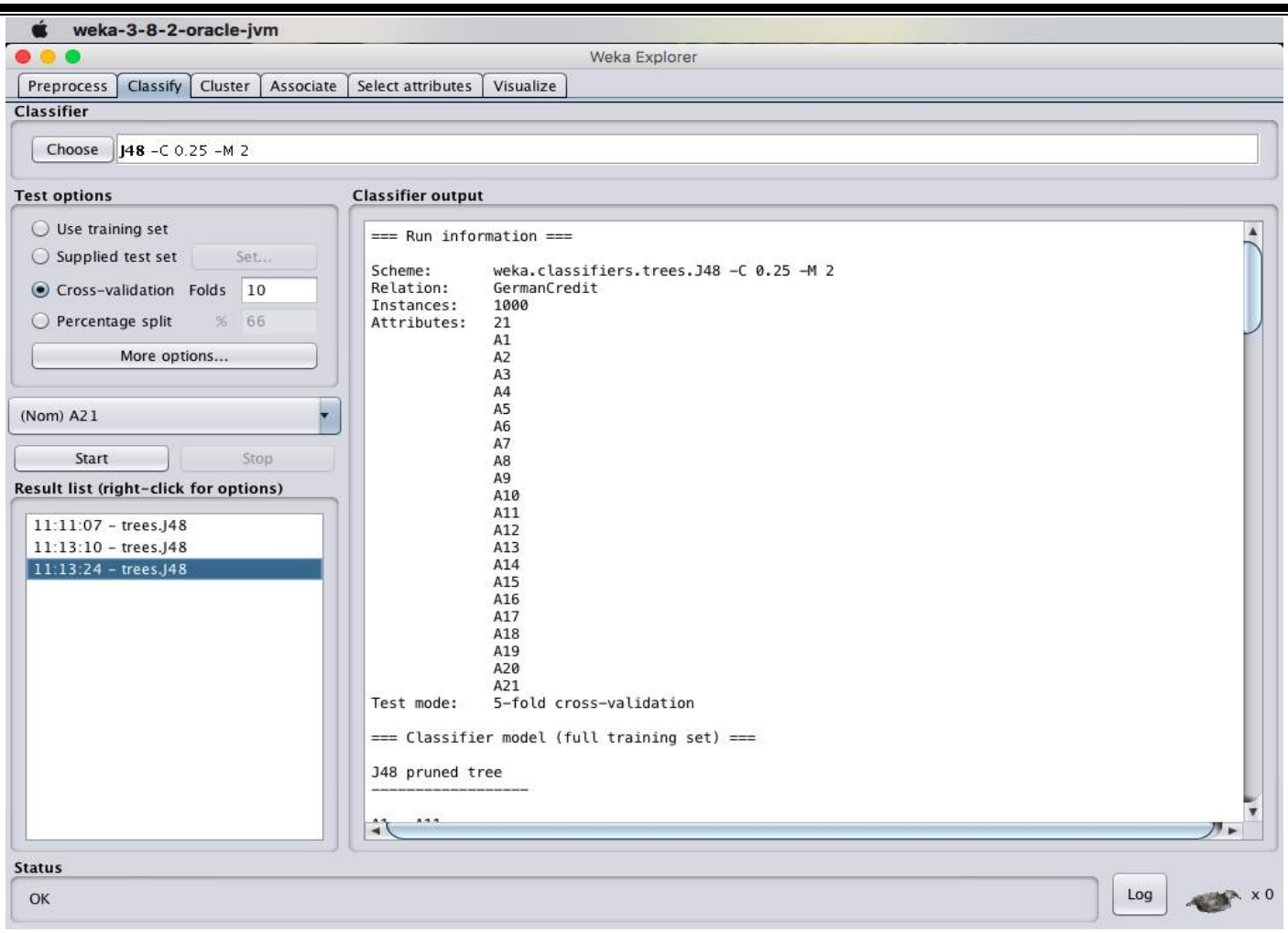
1. **Do you really need to input so many attributes to get good results?**

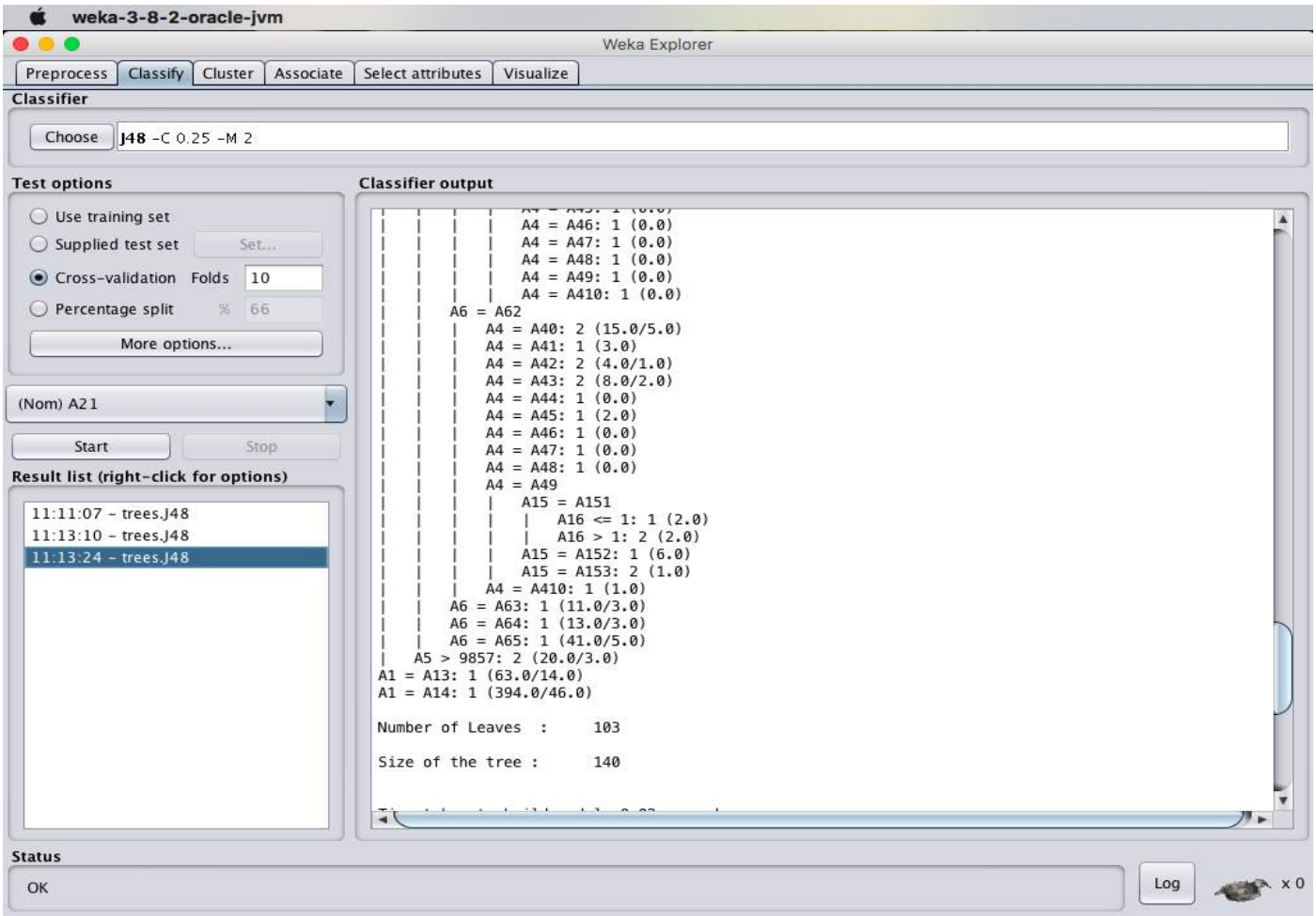
Yes, Having many attributes as the input leads to the good results regarding the dataset. Having more attributes leads to have more and different types of the evaluations.

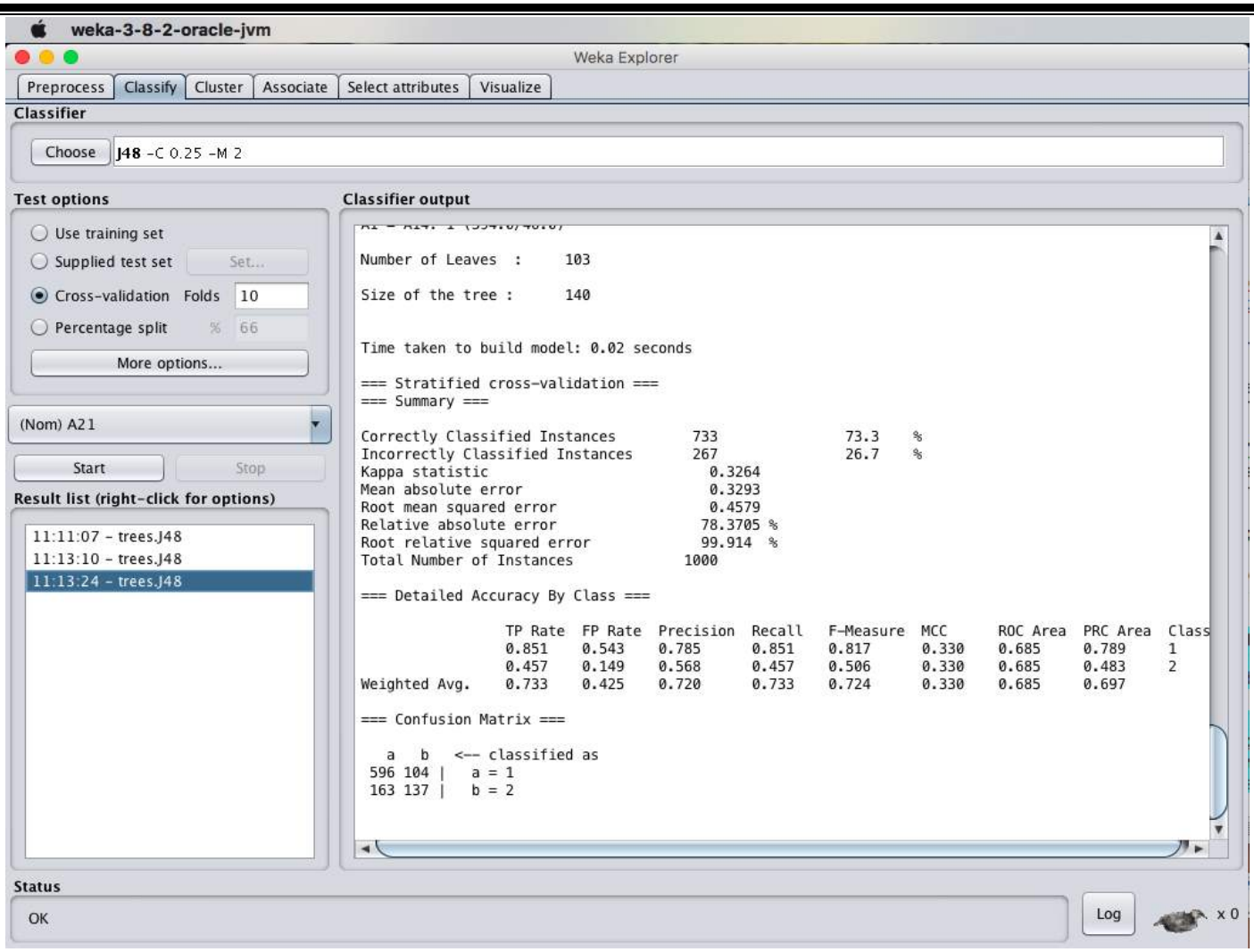
2. **Compare the results obtained by decision tree and SMO ?**

❖ **DECISION TREE :**

A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal, its also widely used in machine learning, which will be the main focus of this article.

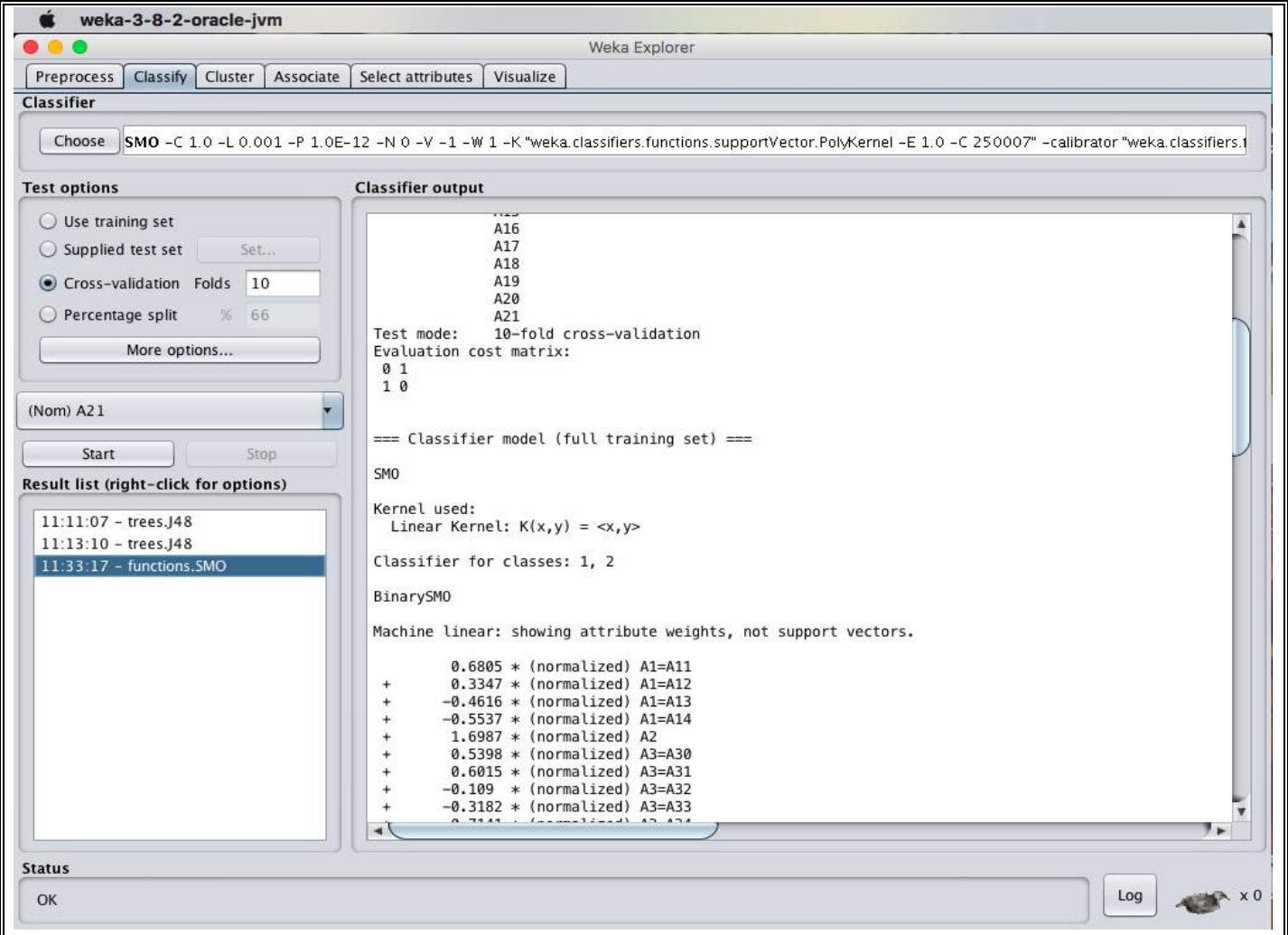

weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:13:24 - trees.J48
Classifier output
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: GermanCredit
Instances: 1000
Attributes: 21
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
Test mode: 5-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
Status
OK
Log
x 0

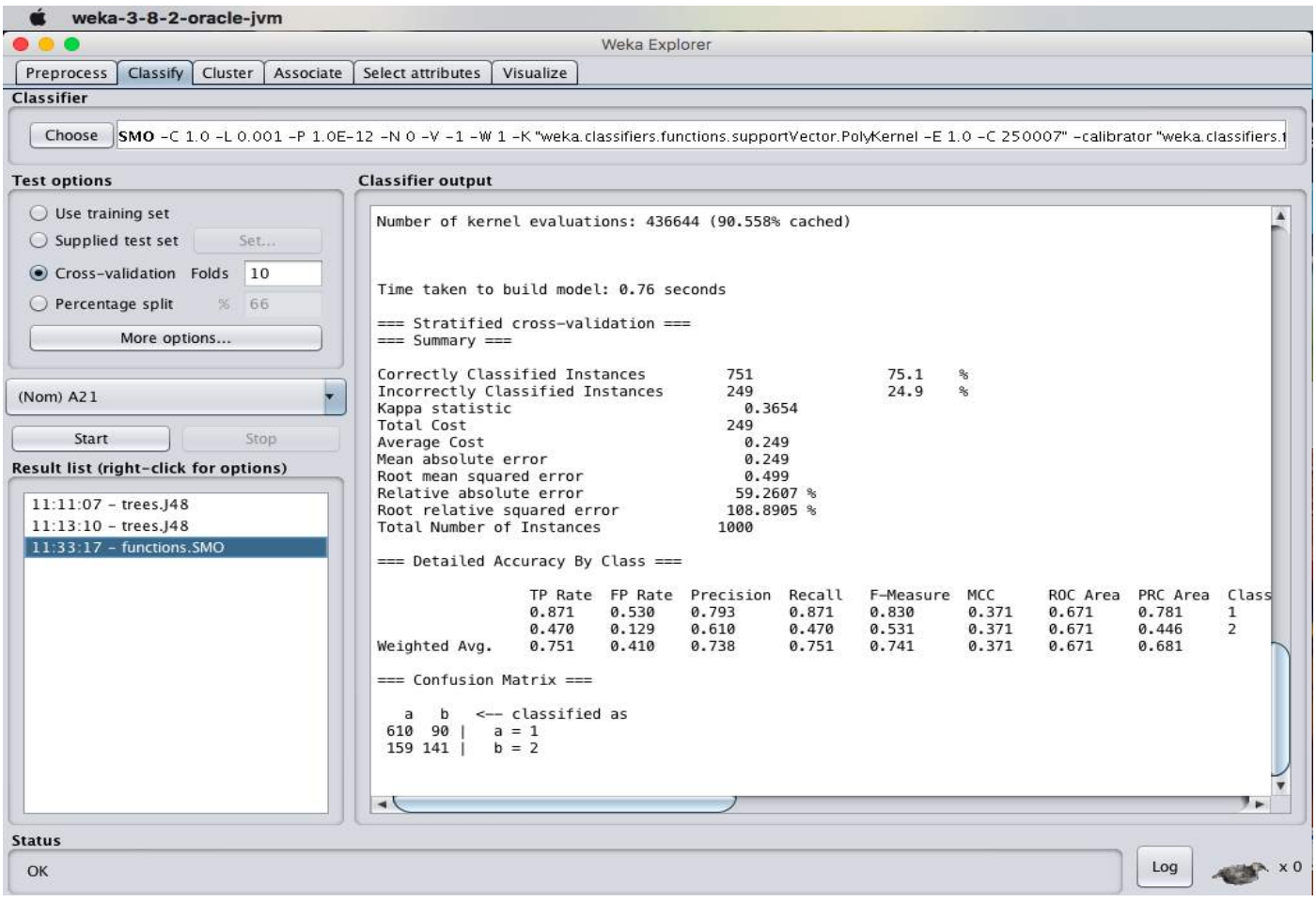

weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:13:24 - trees.J48
Classifier output
| | | | A4 = A46: 1 (0.0)
| | | | A4 = A47: 1 (0.0)
| | | | A4 = A48: 1 (0.0)
| | | | A4 = A49: 1 (0.0)
| | | | A4 = A410: 1 (0.0)
| | A6 = A62
| | | A4 = A40: 2 (15.0/5.0)
| | | A4 = A41: 1 (3.0)
| | | A4 = A42: 2 (4.0/1.0)
| | | A4 = A43: 2 (8.0/2.0)
| | | A4 = A44: 1 (0.0)
| | | A4 = A45: 1 (2.0)
| | | A4 = A46: 1 (0.0)
| | | A4 = A47: 1 (0.0)
| | | A4 = A48: 1 (0.0)
| | | A4 = A49
| | | | A15 = A151
| | | | | A16 <= 1: 1 (2.0)
| | | | | A16 > 1: 2 (2.0)
| | | | A15 = A152: 1 (6.0)
| | | | A15 = A153: 2 (1.0)
| | | A4 = A410: 1 (1.0)
| | A6 = A63: 1 (11.0/3.0)
| | A6 = A64: 1 (13.0/3.0)
| | A6 = A65: 1 (41.0/5.0)
| A5 > 9857: 2 (20.0/3.0)
A1 = A13: 1 (63.0/14.0)
A1 = A14: 1 (394.0/46.0)
Number of Leaves : 103
Size of the tree : 140
Status
OK
Log
x 0



❖ SMO ALGORITHM:

The iterative algorithm Sequential Minimal Optimization (SMO) is used for solving quadratic programming (QP) problems. One example where QP problems are relevant is during the training process of support vector machines (SVM). The SMO algorithm is used to solve in this example a constraint optimization problem. John Platt proposed this algorithm in 1998 and it was successfully used since then. We describe here the basics of the algorithm in the light of big data.


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.t
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:33:17 - functions.SMO
Classifier output
A16
A17
A18
A19
A20
A21
Test mode: 10-fold cross-validation
Evaluation cost matrix:
0 1
1 0
=== Classifier model (full training set) ===
SMO
Kernel used:
Linear Kernel: K(x,y) = <x,y>
Classifier for classes: 1, 2
BinarySMO
Machine linear: showing attribute weights, not support vectors.
0.6805 * (normalized) A1=A11
+ 0.3347 * (normalized) A1=A12
+ -0.4616 * (normalized) A1=A13
+ -0.5537 * (normalized) A1=A14
+ 1.6987 * (normalized) A2
+ 0.5398 * (normalized) A3=A30
+ 0.6015 * (normalized) A3=A31
+ -0.109 * (normalized) A3=A32
+ -0.3182 * (normalized) A3=A33
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.t
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:33:17 - functions.SMO
Classifier output
Number of kernel evaluations: 436644 (90.558% cached)
Time taken to build model: 0.76 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 751 75.1 %
Incorrectly Classified Instances 249 24.9 %
Kappa statistic 0.3654
Total Cost 249
Average Cost 0.249
Mean absolute error 0.249
Root mean squared error 0.499
Relative absolute error 59.2607 %
Root relative squared error 108.8905 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.871 0.530 0.793 0.871 0.830 0.371 0.671 0.781 1
0.470 0.129 0.610 0.470 0.531 0.371 0.671 0.446 2
Weighted Avg. 0.751 0.410 0.738 0.751 0.741 0.371 0.671 0.681
=== Confusion Matrix ===
a b <-- classified as
610 90 | a = 1
159 141 | b = 2
Status
OK
Log
x 0

When compared to both the Decision Tree and SMO. SMO is taking more time for building the model. Also there will be a change in instances.

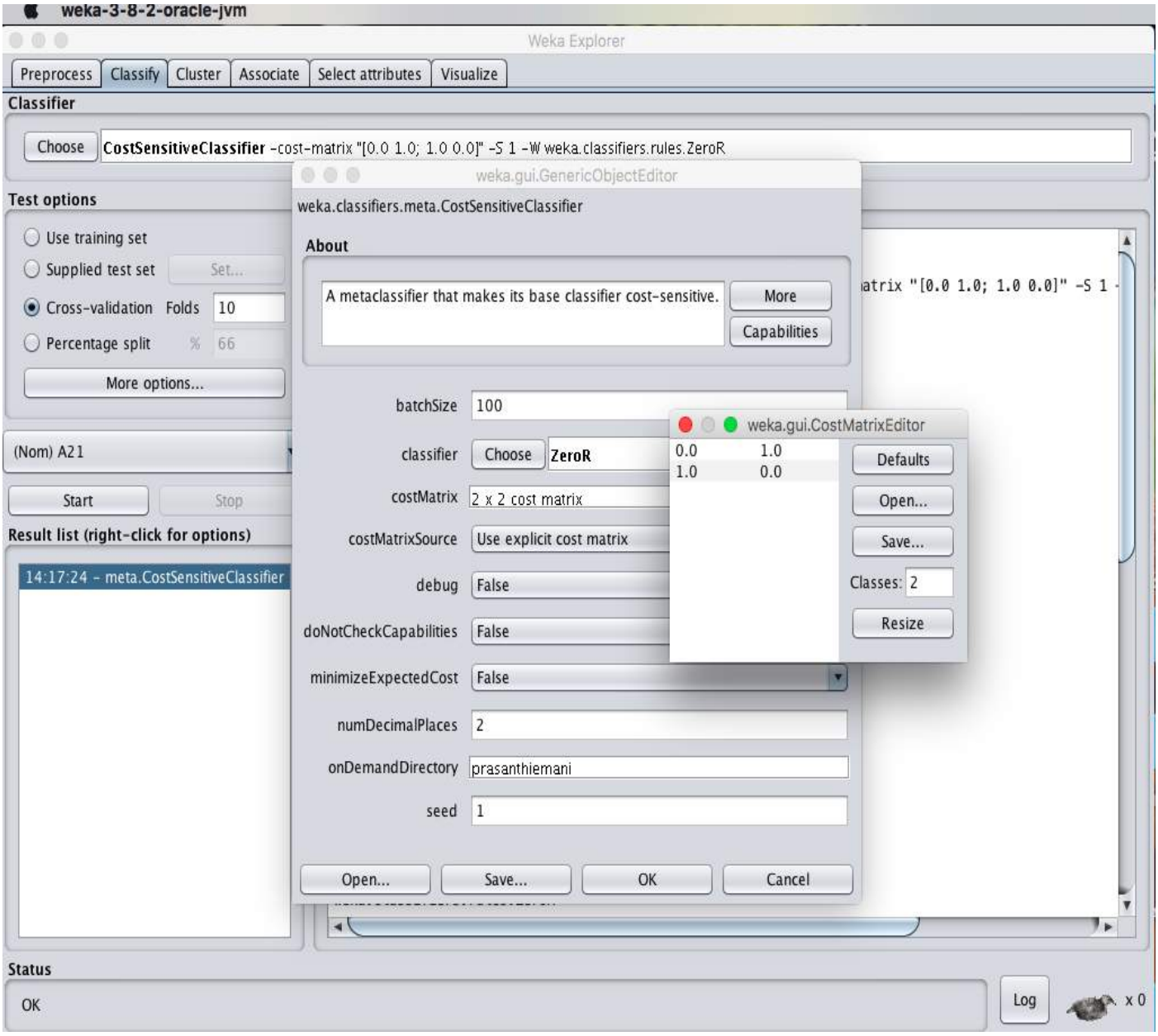
3. Set the cost sensitive evaluation and compare the obtained results.

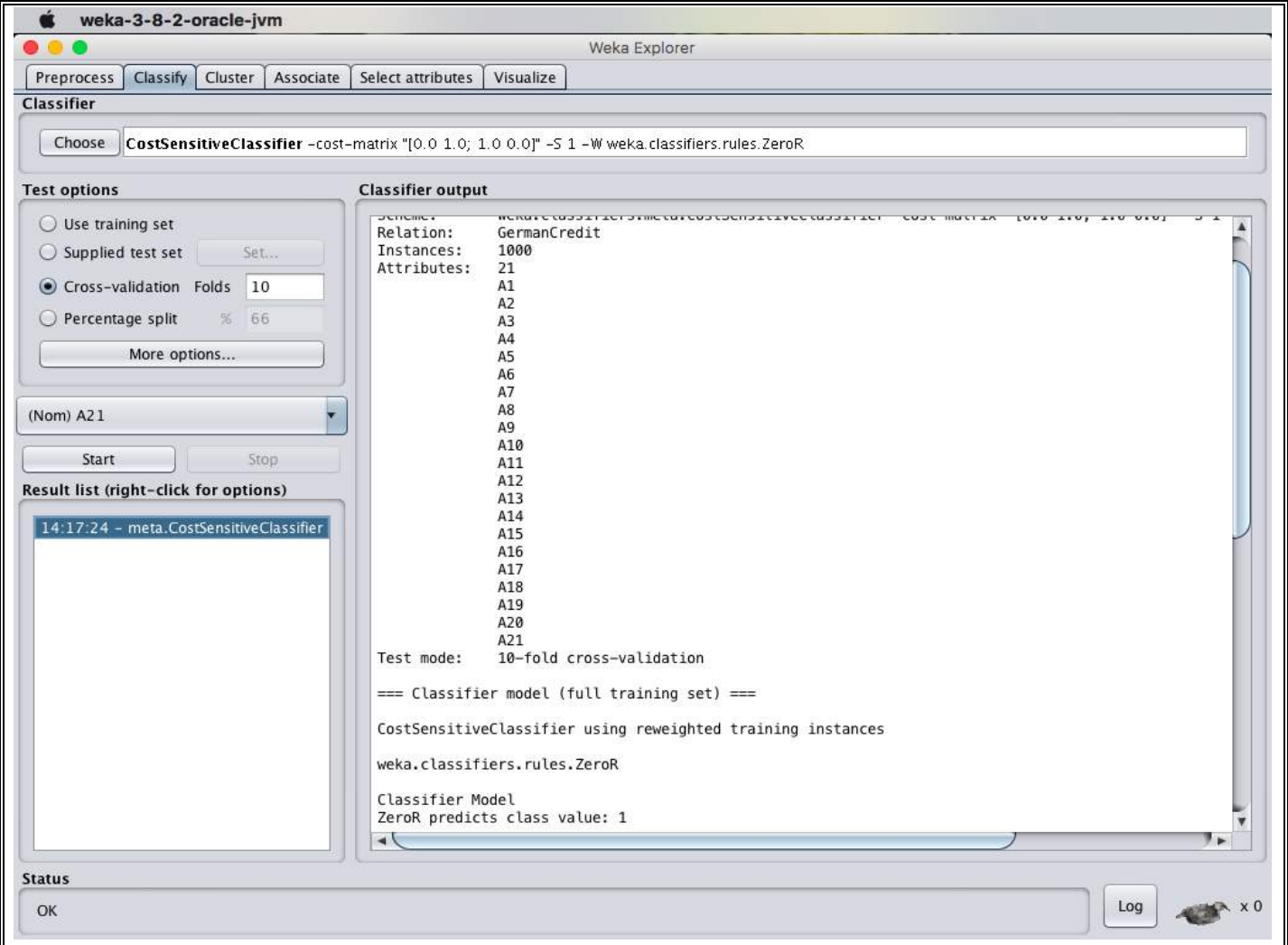
Cost-Sensitive Learning is a type of learning in data mining that takes the misclassification costs (and possibly other types of cost) into consideration. The goal of this type of learning is to minimize the total cost. The key difference between cost-sensitive learning and cost-insensitive learning is that cost-sensitive learning treats the different misclassifications differently. Costinsensitive learning does not take the misclassification costs into consideration. The goal of this type of learning is to pursue a high accuracy of classifying examples into a set of known classes.

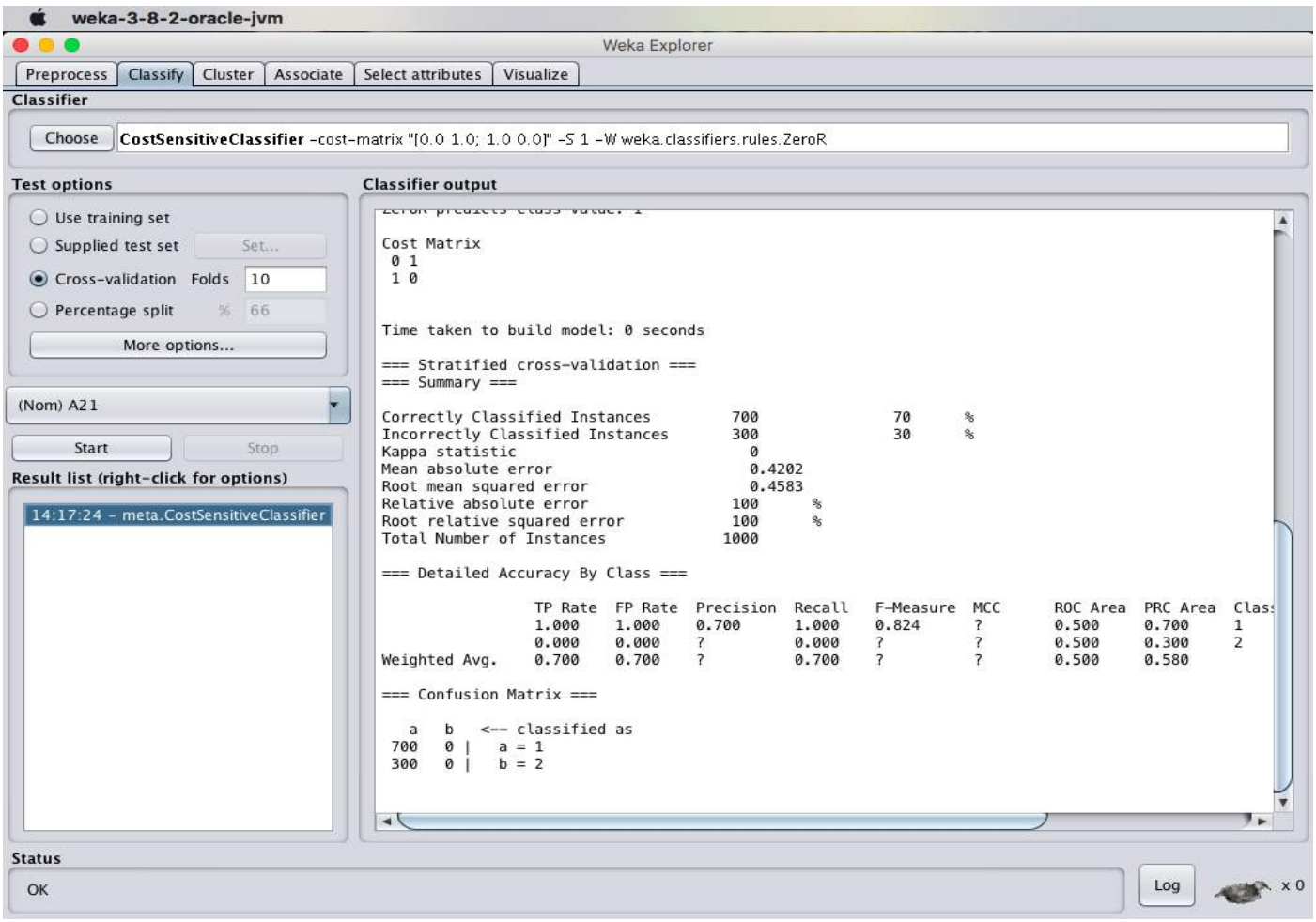
STEPS :

- Classifythe dataset with the cost sensitive classifier technique.
- Change the cost matrix to 2*2 matrix and execute.

ANALYSIS :




weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
CostSensitiveClassifier -cost-matrix "[0.0 1.0; 1.0 0.0]" -S 1 -W weka.classifiers.rules.ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
14:17:24 - meta.CostSensitiveClassifier
Classifier output
Relation: GermanCredit
Instances: 1000
Attributes: 21
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
CostSensitiveClassifier using reweighted training instances
weka.classifiers.rules.ZeroR
Classifier Model
ZeroR predicts class value: 1
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
CostSensitiveClassifier -cost-matrix "[0.0 1.0; 1.0 0.0]" -S 1 -W weka.classifiers.rules.ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
14:17:24 - meta.CostSensitiveClassifier
Classifier output
Cost Matrix
0 1
1 0
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 700 70 %
Incorrectly Classified Instances 300 30 %
Kappa statistic 0
Mean absolute error 0.4202
Root mean squared error 0.4583
Relative absolute error 100 %
Root relative squared error 100 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1.000 1.000 0.700 1.000 0.824 ? 0.500 0.700 1
0.000 0.000 ? 0.000 ? ? 0.500 0.300 2
Weighted Avg. 0.700 0.700 ? 0.700 ? ? 0.500 0.580
=== Confusion Matrix ===
a b <-- classified as
700 0 | a = 1
300 0 | b = 2
Status
OK
Log
x 0

4. **What is the significance of the following parameters :**

a) Mean Absolute Error :

Mean Absolute Error (MAE) is similar to the Mean Squared Error, but it uses absolute values instead of squaring. This measure is not as popular as MSE, though its meaning is more intuitive (the "average error").

b) Root Mean Square Error :

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

c) Relative Absolute Error :

The relative absolute error is very similar to the relative squared error in the sense that it is also relative to a simple predictor, which is just the average of the actual values. In this case, though, the error is just the total absolute error instead of the total squared error. Thus, the relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor.

d) Total Number of Instances :

The data present consists of various instances of the class. In the case of german_credit dataset, the total number of instances present in the german credit dataset are 1000 instances.

RESULT :

Thus, the observations and evaluations done on the german_credit dataset are analyzed. The comparison between decision tree and Sequential Minimal Optimization (SMO) has been successfully visualized. In addition to that cost sensitive classifier is been used to analyze few things.

EX.No: 13
Date :

EVALUATING ACCURACY OF THE CLASSIFIERS

PROBLEM STATEMENT :

Compare the confusion matrix generated using weka for the german_creditdataset(download from the UCI repository).

a) Logistic Regression
b) Naïve Bayes Algorithm
c) J48
d) K-Nearest Neighbor
e) SMO Algorithm

DESCRIPTION :

Consider the german credit dataset which can be downloaded from the UCI repository.

ANALYSIS :

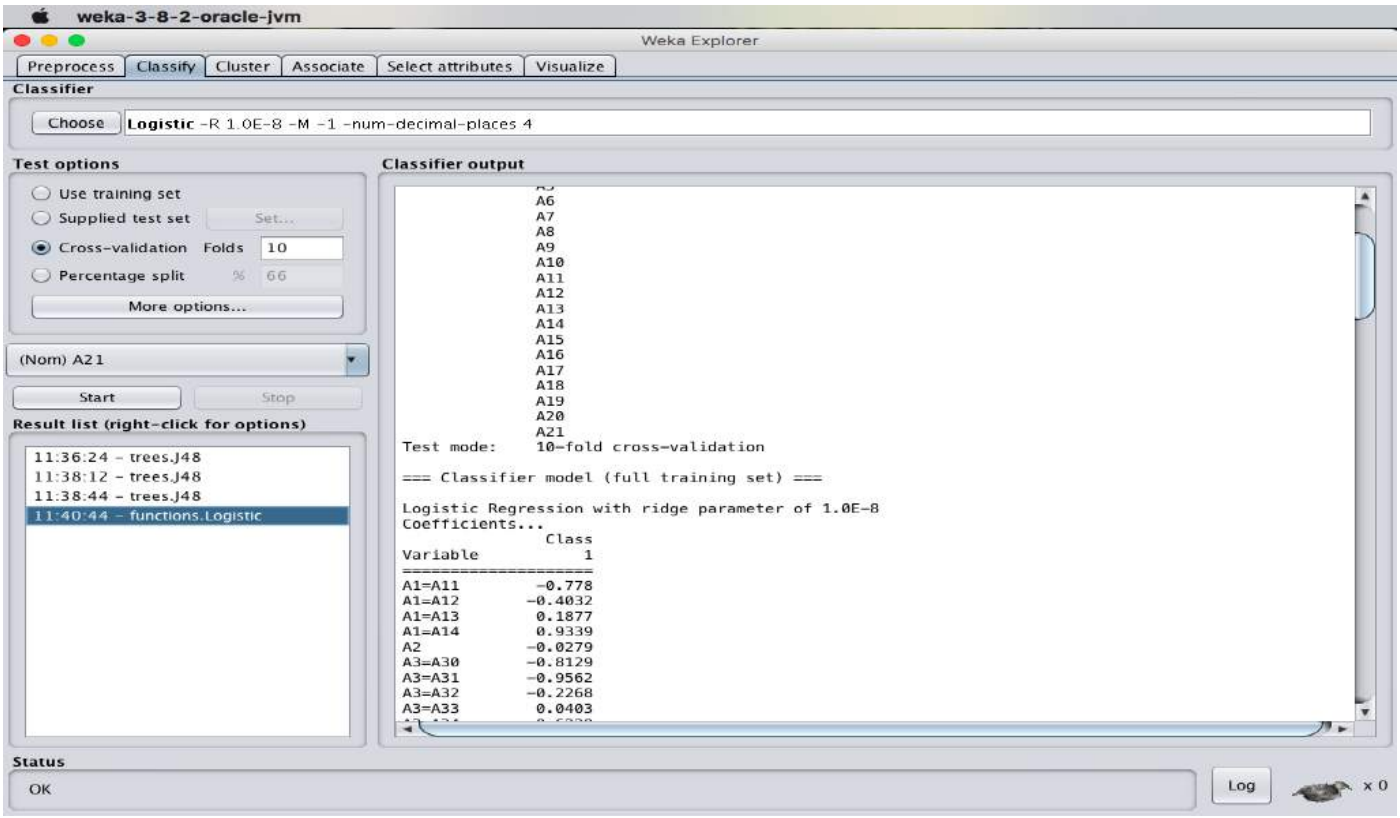
A) Logistic Regression :

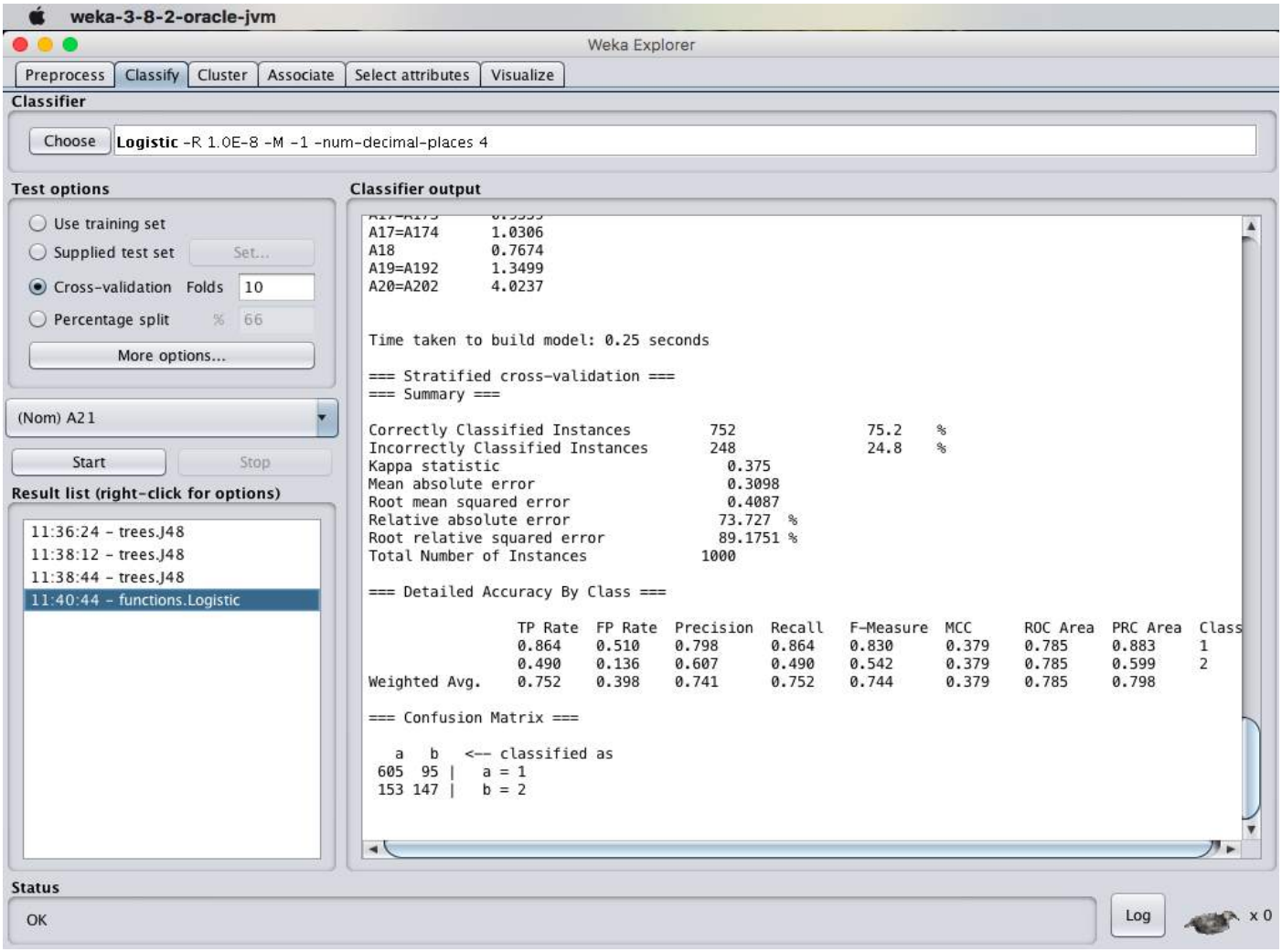
Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical).

Steps :

- Load the dataset into the weka tool and preprocess it.
- Apply the classification the logistic regression technique and execute for the result.

Output :





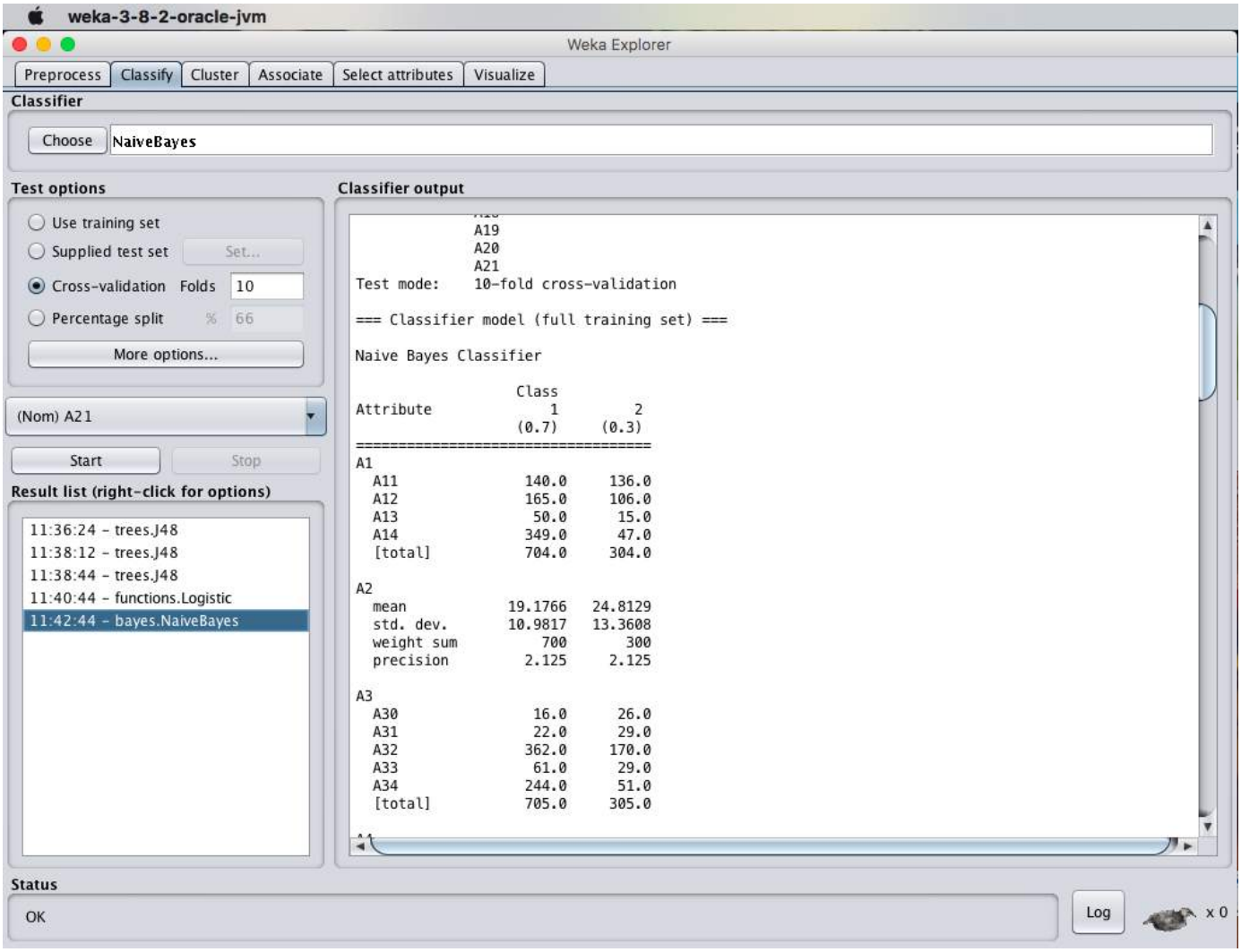
B) Naïve Bayes Algorithm :

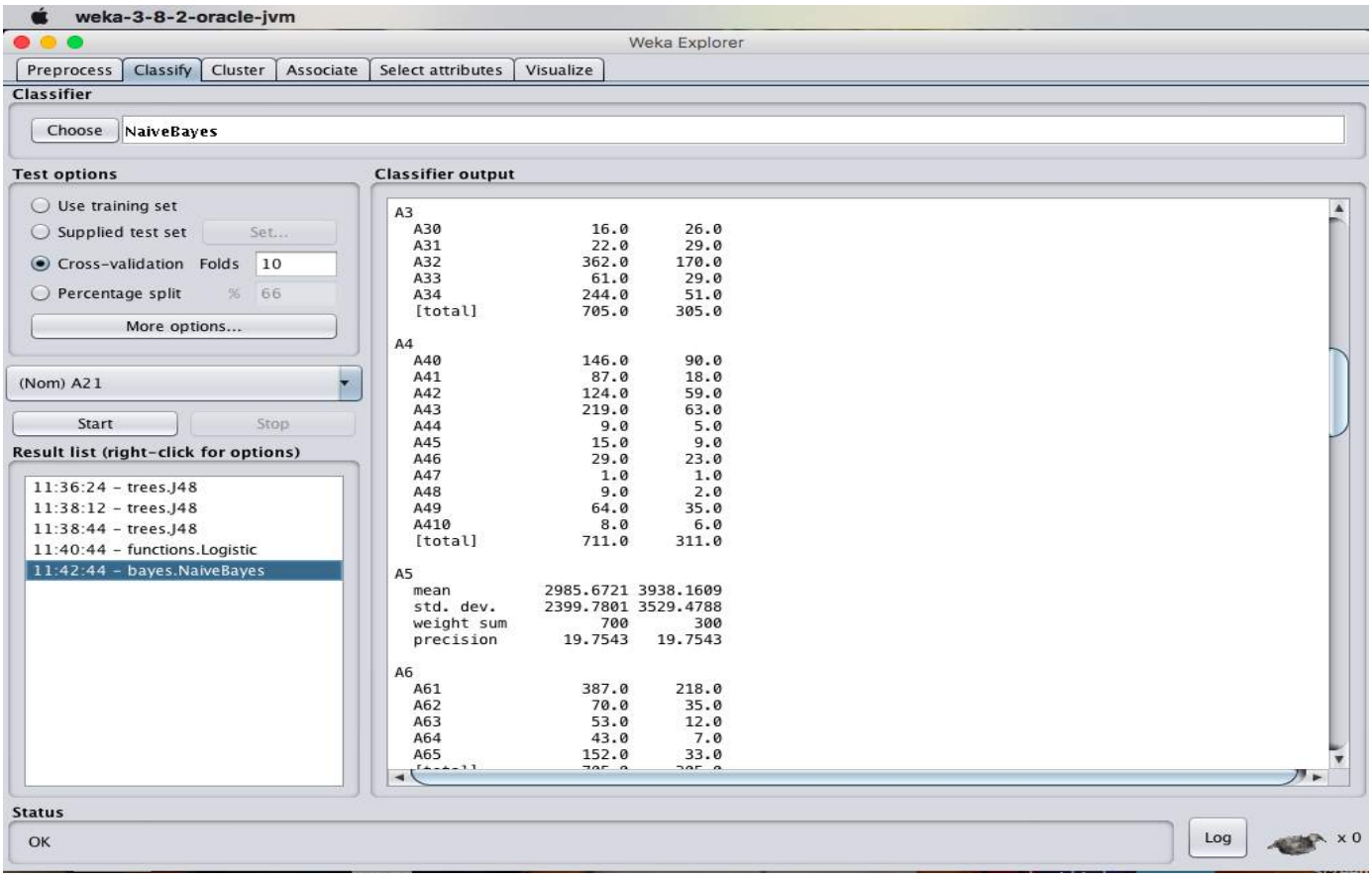
The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

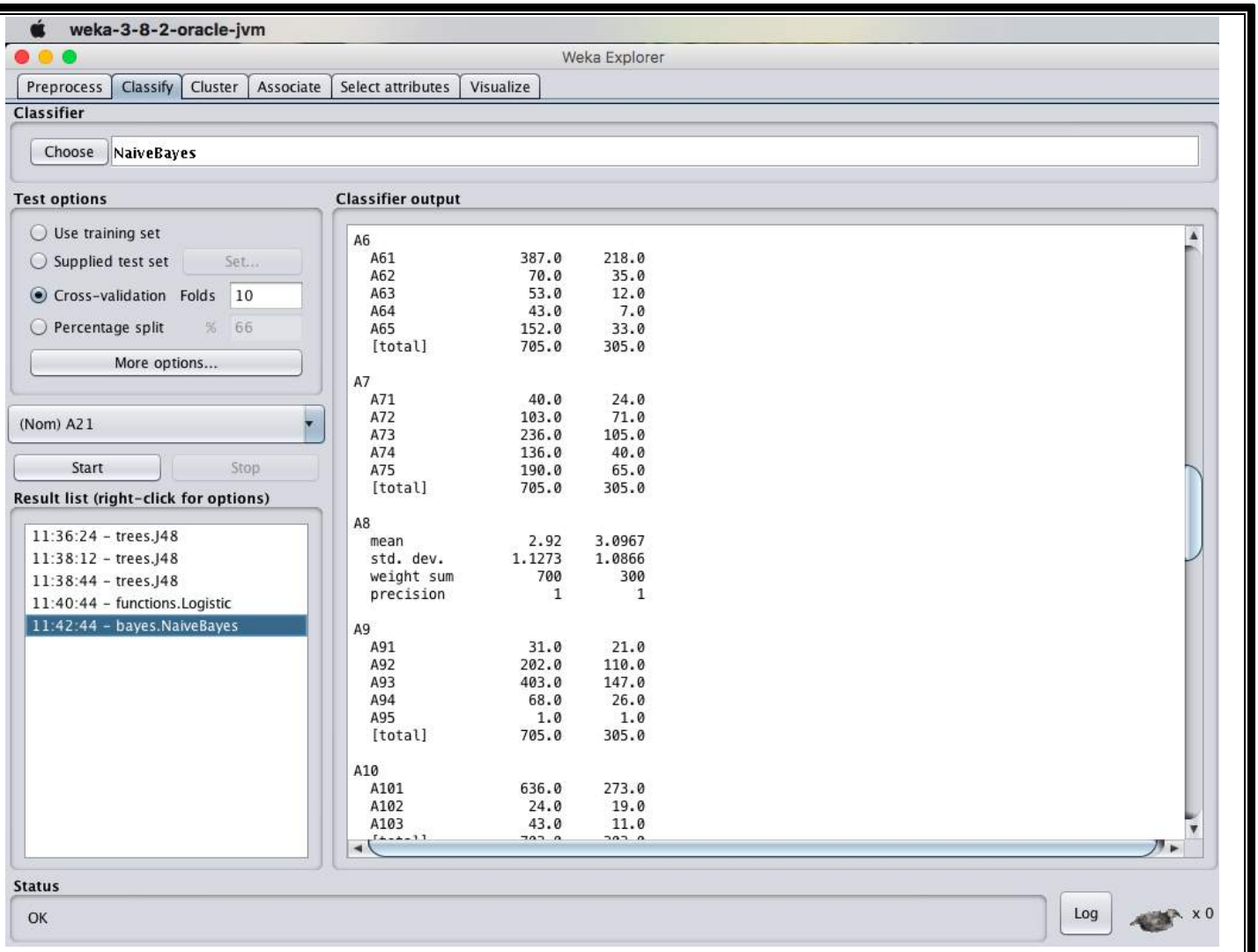
Steps :

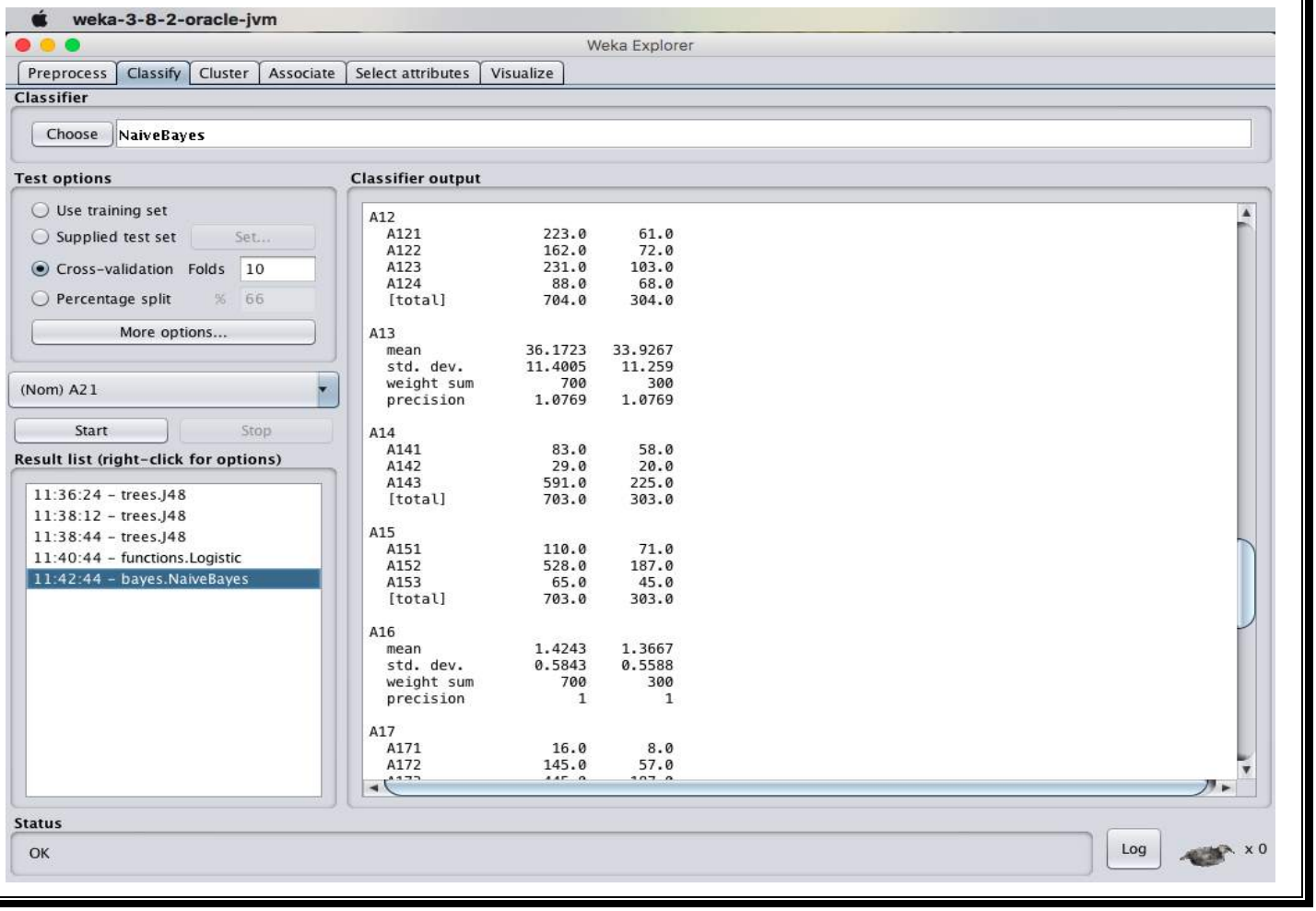
- Load the dataset into the weka tool and preprocess it.
- Apply the classification the Naïve bayes technique and execute for the result.

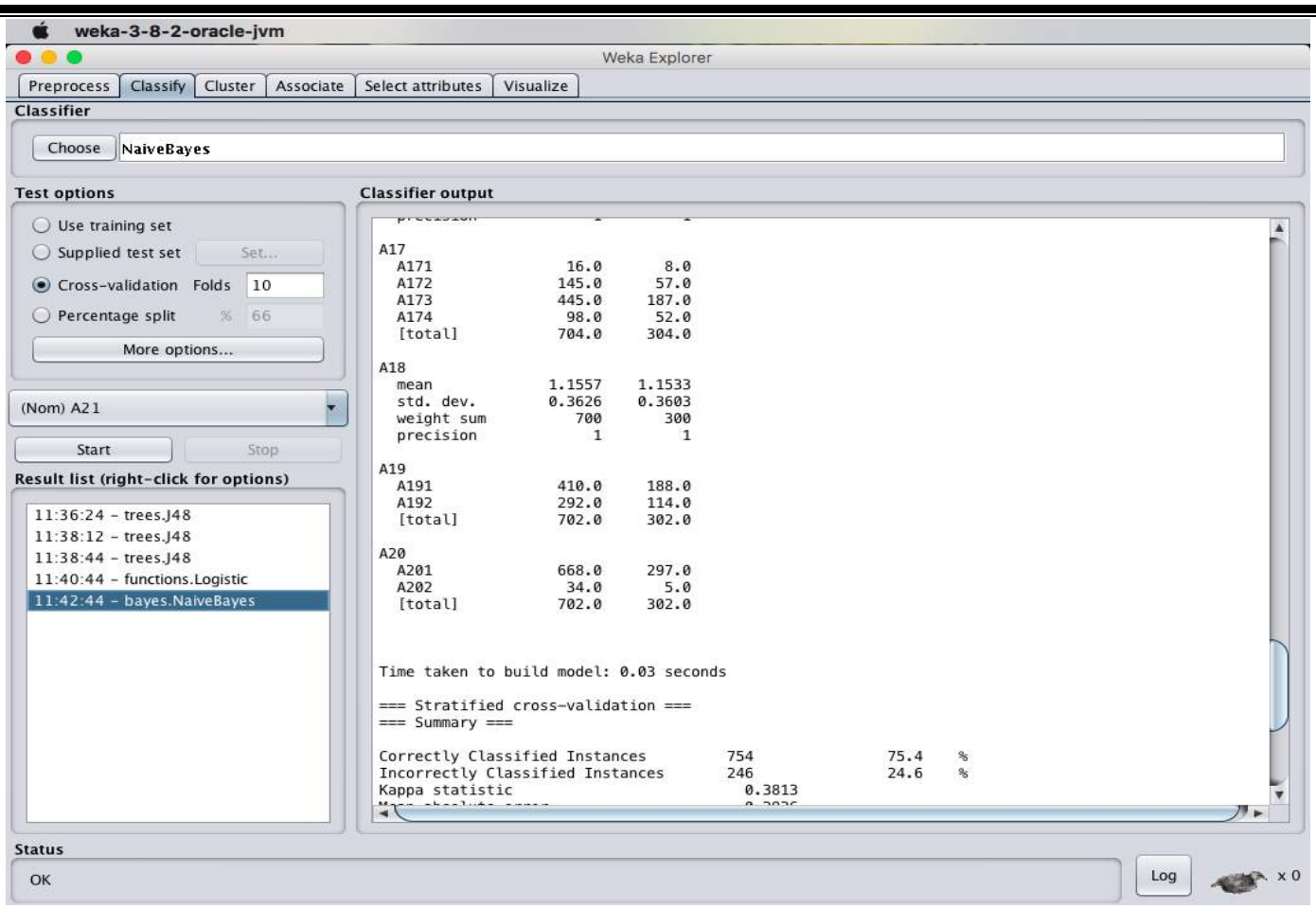
Output :

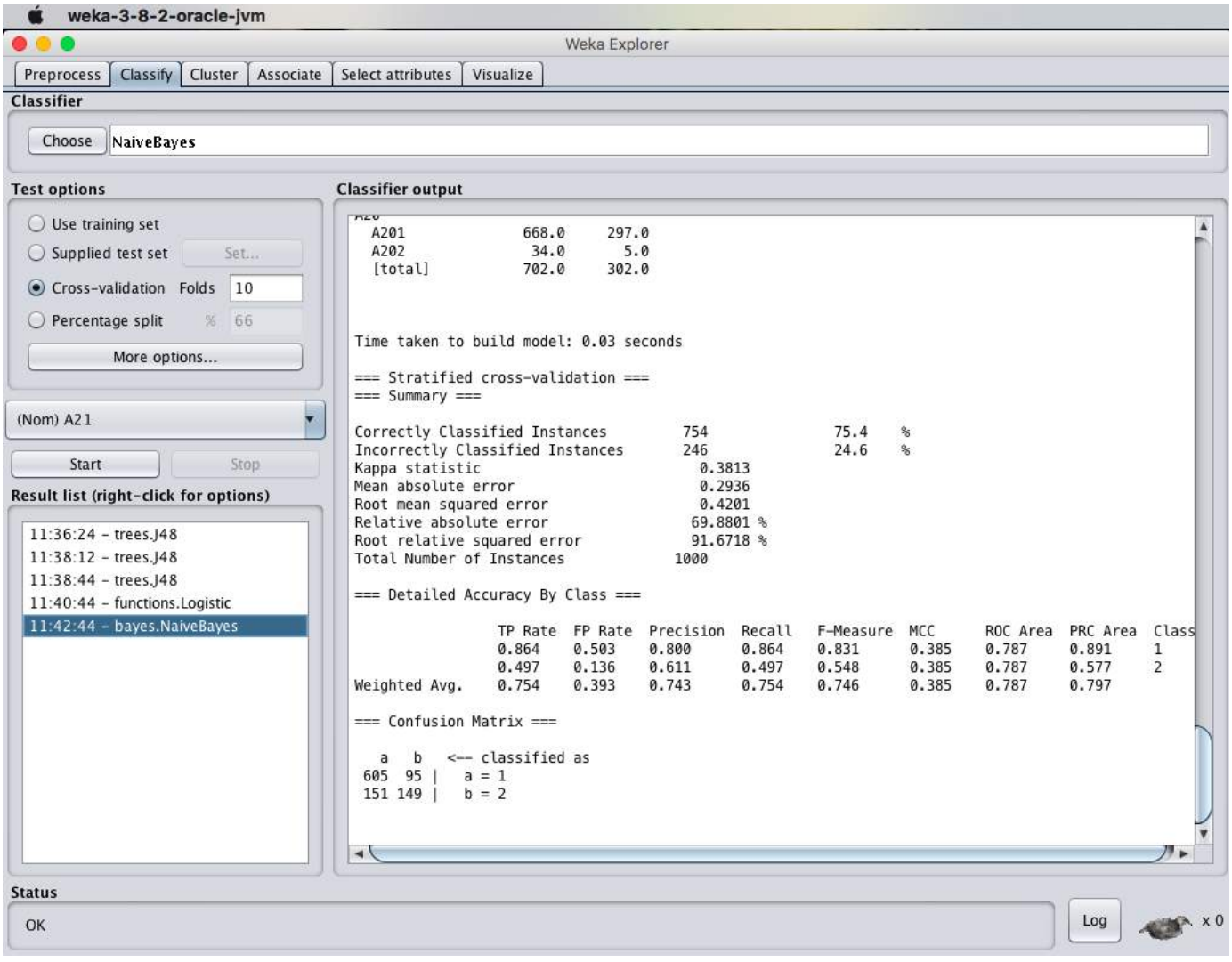

weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
A19
A20
A21
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
Naive Bayes Classifier
Class
Attribute 1 2
(0.7) (0.3)
A1
A11 140.0 136.0
A12 165.0 106.0
A13 50.0 15.0
A14 349.0 47.0
[total] 704.0 304.0
A2
mean 19.1766 24.8129
std. dev. 10.9817 13.3608
weight sum 700 300
precision 2.125 2.125
A3
A30 16.0 26.0
A31 22.0 29.0
A32 362.0 170.0
A33 61.0 29.0
A34 244.0 51.0
[total] 705.0 305.0
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
A3
A30 16.0 26.0
A31 22.0 29.0
A32 362.0 170.0
A33 61.0 29.0
A34 244.0 51.0
[total] 705.0 305.0
A4
A40 146.0 90.0
A41 87.0 18.0
A42 124.0 59.0
A43 219.0 63.0
A44 9.0 5.0
A45 15.0 9.0
A46 29.0 23.0
A47 1.0 1.0
A48 9.0 2.0
A49 64.0 35.0
A410 8.0 6.0
[total] 711.0 311.0
A5
mean 2985.6721 3938.1609
std. dev. 2399.7801 3529.4788
weight sum 700 300
precision 19.7543 19.7543
A6
A61 387.0 218.0
A62 70.0 35.0
A63 53.0 12.0
A64 43.0 7.0
A65 152.0 33.0
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A6
A61 387.0 218.0
A62 70.0 35.0
A63 53.0 12.0
A64 43.0 7.0
A65 152.0 33.0
[total] 705.0 305.0
A7
A71 40.0 24.0
A72 103.0 71.0
A73 236.0 105.0
A74 136.0 40.0
A75 190.0 65.0
[total] 705.0 305.0
A8
mean 2.92 3.0967
std. dev. 1.1273 1.0866
weight sum 700 300
precision 1 1
A9
A91 31.0 21.0
A92 202.0 110.0
A93 403.0 147.0
A94 68.0 26.0
A95 1.0 1.0
[total] 705.0 305.0
A10
A101 636.0 273.0
A102 24.0 19.0
A103 43.0 11.0
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A12
A121 223.0 61.0
A122 162.0 72.0
A123 231.0 103.0
A124 88.0 68.0
[total] 704.0 304.0
A13
mean 36.1723 33.9267
std. dev. 11.4005 11.259
weight sum 700 300
precision 1.0769 1.0769
A14
A141 83.0 58.0
A142 29.0 20.0
A143 591.0 225.0
[total] 703.0 303.0
A15
A151 110.0 71.0
A152 528.0 187.0
A153 65.0 45.0
[total] 703.0 303.0
A16
mean 1.4243 1.3667
std. dev. 0.5843 0.5588
weight sum 700 300
precision 1 1
A17
A171 16.0 8.0
A172 145.0 57.0
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A17
A171 16.0 8.0
A172 145.0 57.0
A173 445.0 187.0
A174 98.0 52.0
[total] 704.0 304.0
A18
mean 1.1557 1.1533
std. dev. 0.3626 0.3603
weight sum 700 300
precision 1 1
A19
A191 410.0 188.0
A192 292.0 114.0
[total] 702.0 302.0
A20
A201 668.0 297.0
A202 34.0 5.0
[total] 702.0 302.0
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 754 75.4 %
Incorrectly Classified Instances 246 24.6 %
Kappa statistic 0.3813
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A201 668.0 297.0
A202 34.0 5.0
[total] 702.0 302.0
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 754 75.4 %
Incorrectly Classified Instances 246 24.6 %
Kappa statistic 0.3813
Mean absolute error 0.2936
Root mean squared error 0.4201
Relative absolute error 69.8801 %
Root relative squared error 91.6718 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.864 0.503 0.800 0.864 0.831 0.385 0.787 0.891 1
0.497 0.136 0.611 0.497 0.548 0.385 0.787 0.577 2
Weighted Avg. 0.754 0.393 0.743 0.754 0.746 0.385 0.787 0.797
=== Confusion Matrix ===
a b <-- classified as
605 95 | a = 1
151 149 | b = 2
Status
OK
Log
x 0

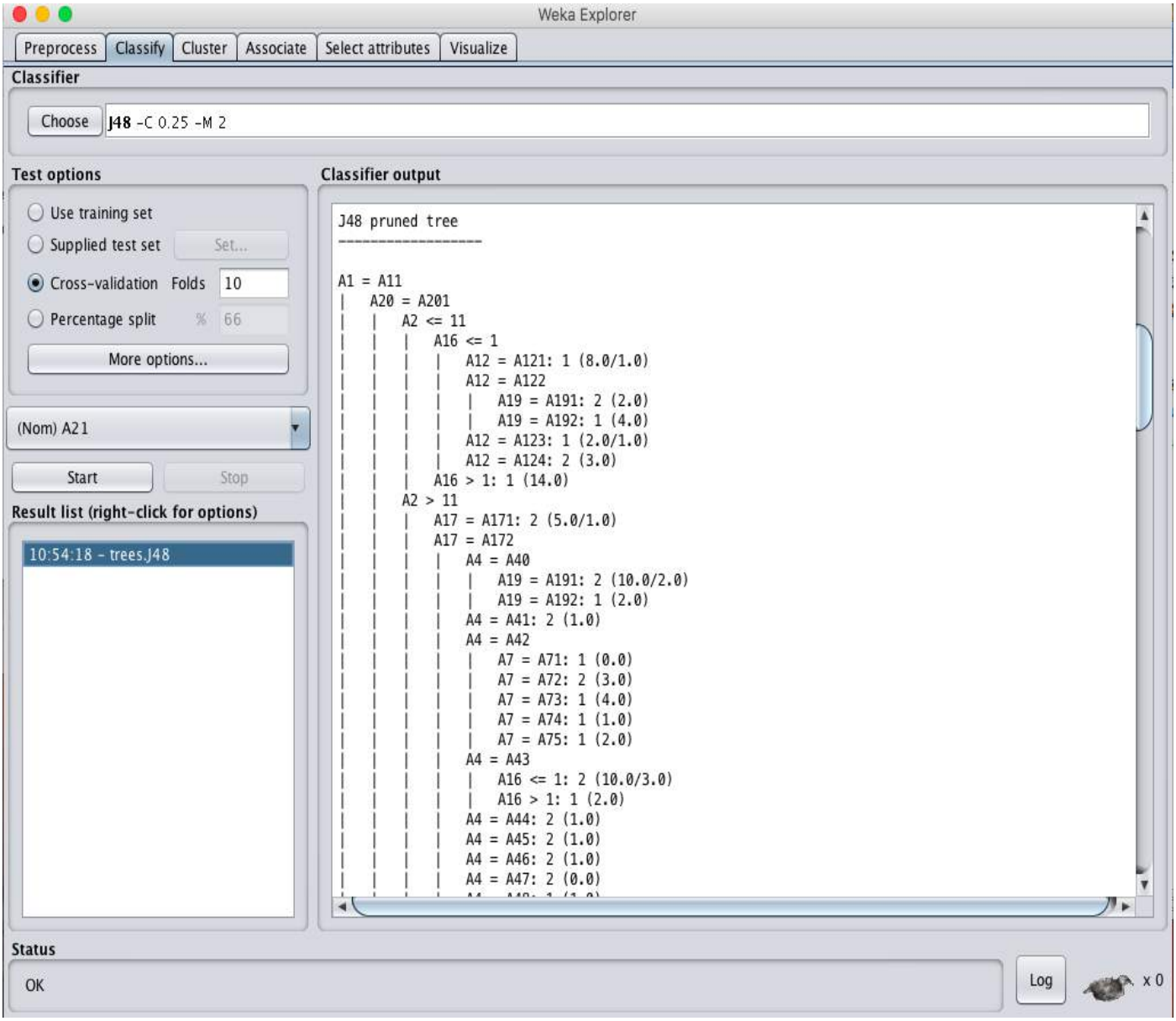
C) J48 Algorithm :

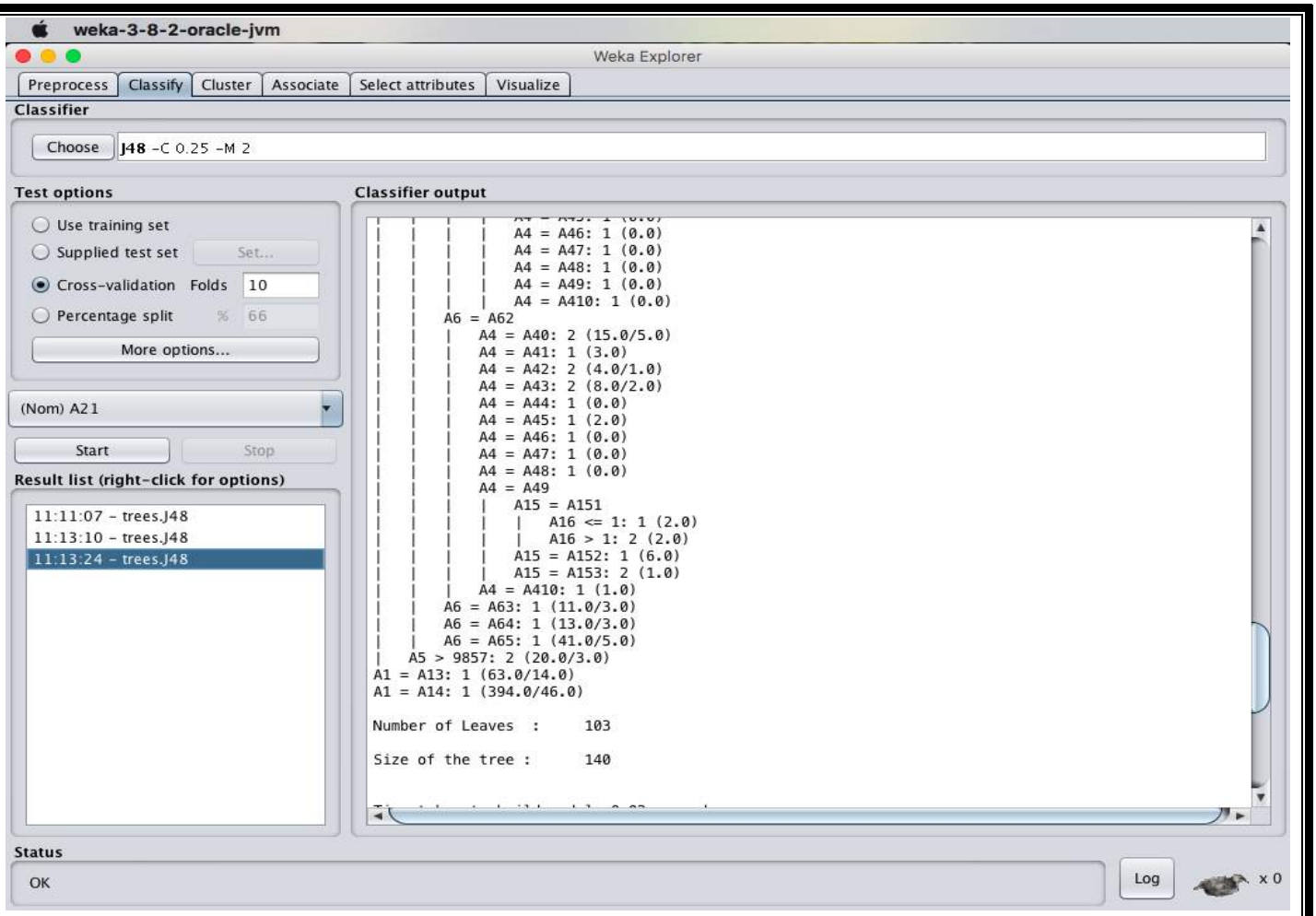
Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also on the bases of the training instances the classes for the newly generated instances are being found. This algorithm generates the rules for the prediction of the target variable. With the help of tree classification algorithm the critical distribution of the data is easily understandable.

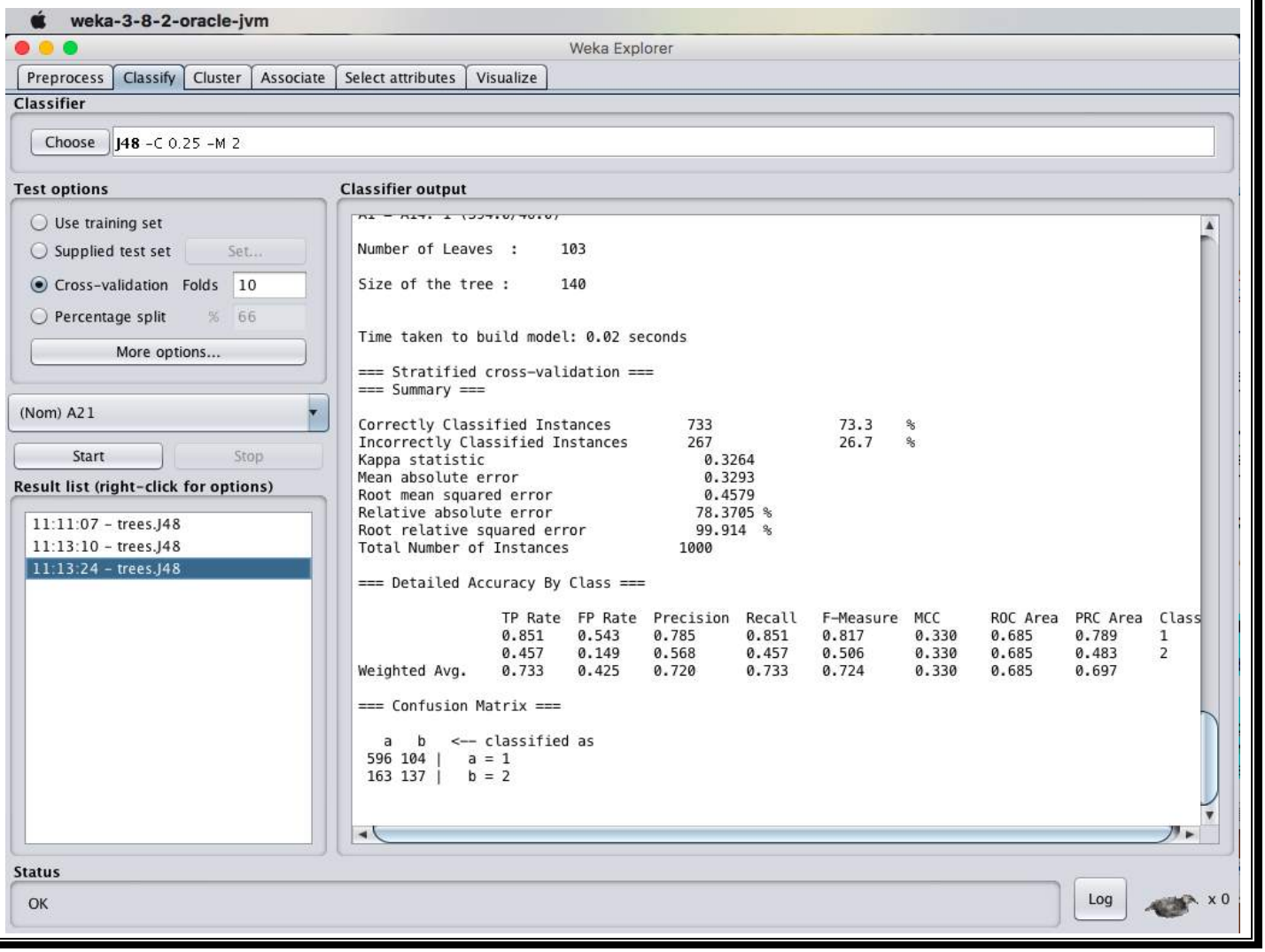
Steps :

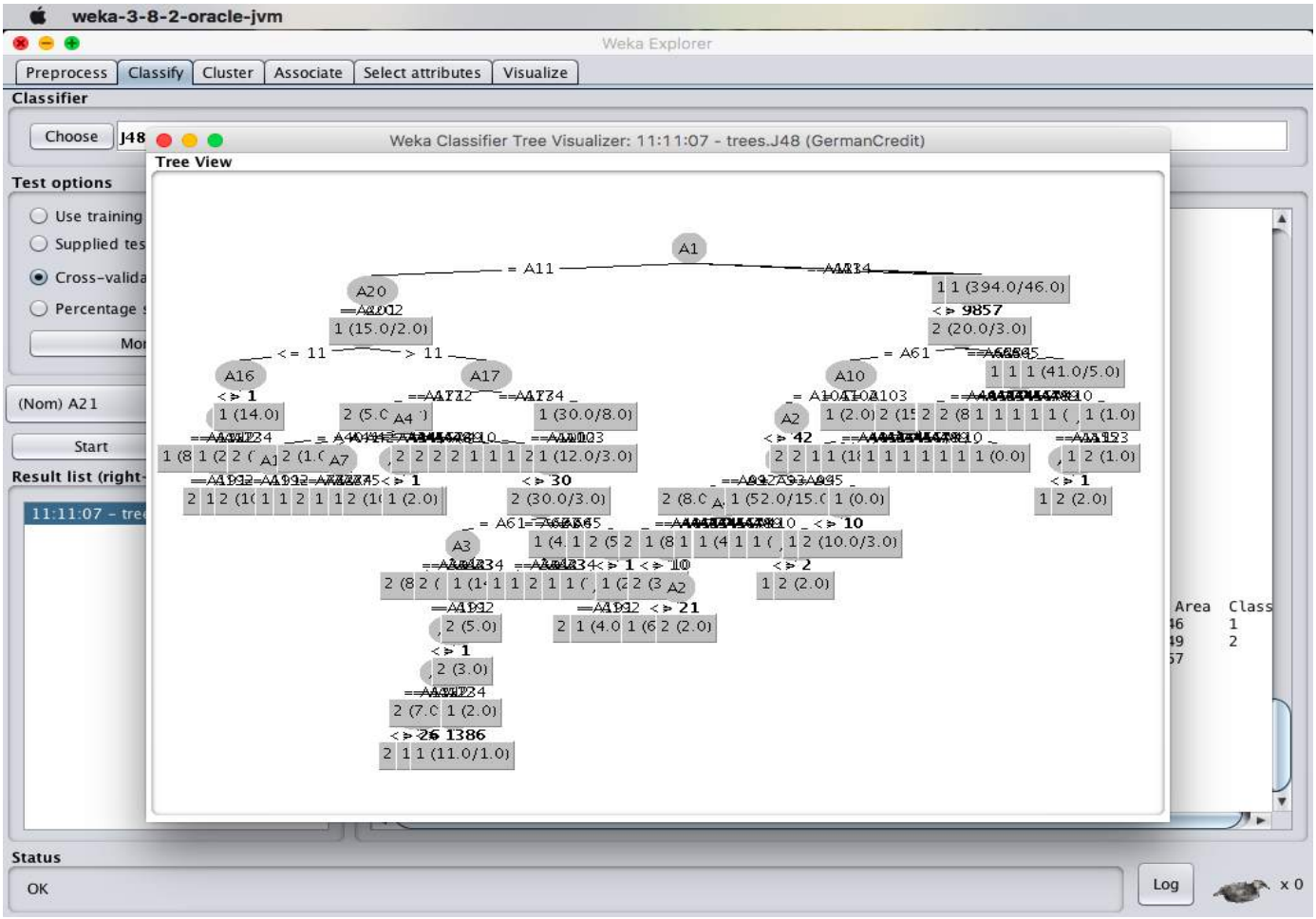
- Load the dataset into the weka tool and preprocess it.
- Apply the classification the J48 technique and execute for the result.

Output :




weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:13:24 - trees.J48
Classifier output
| | | | A4 = A46: 1 (0.0)
| | | | A4 = A47: 1 (0.0)
| | | | A4 = A48: 1 (0.0)
| | | | A4 = A49: 1 (0.0)
| | | | A4 = A410: 1 (0.0)
| | A6 = A62
| | | A4 = A40: 2 (15.0/5.0)
| | | A4 = A41: 1 (3.0)
| | | A4 = A42: 2 (4.0/1.0)
| | | A4 = A43: 2 (8.0/2.0)
| | | A4 = A44: 1 (0.0)
| | | A4 = A45: 1 (2.0)
| | | A4 = A46: 1 (0.0)
| | | A4 = A47: 1 (0.0)
| | | A4 = A48: 1 (0.0)
| | | A4 = A49
| | | | A15 = A151
| | | | | A16 <= 1: 1 (2.0)
| | | | | A16 > 1: 2 (2.0)
| | | | A15 = A152: 1 (6.0)
| | | | A15 = A153: 2 (1.0)
| | | A4 = A410: 1 (1.0)
| | A6 = A63: 1 (11.0/3.0)
| | A6 = A64: 1 (13.0/3.0)
| | A6 = A65: 1 (41.0/5.0)
| A5 > 9857: 2 (20.0/3.0)
A1 = A13: 1 (63.0/14.0)
A1 = A14: 1 (394.0/46.0)
Number of Leaves : 103
Size of the tree : 140
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:13:24 - trees.J48
Classifier output
Number of Leaves : 103
Size of the tree : 140
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 733 73.3 %
Incorrectly Classified Instances 267 26.7 %
Kappa statistic 0.3264
Mean absolute error 0.3293
Root mean squared error 0.4579
Relative absolute error 78.3705 %
Root relative squared error 99.914 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.851 0.543 0.785 0.851 0.817 0.330 0.685 0.789 1
0.457 0.149 0.568 0.457 0.506 0.330 0.685 0.483 2
Weighted Avg. 0.733 0.425 0.720 0.733 0.724 0.330 0.685 0.697
=== Confusion Matrix ===
a b <-- classified as
596 104 | a = 1
163 137 | b = 2
Status
OK
Log
x 0



D) K-Nearest Neighbor :

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

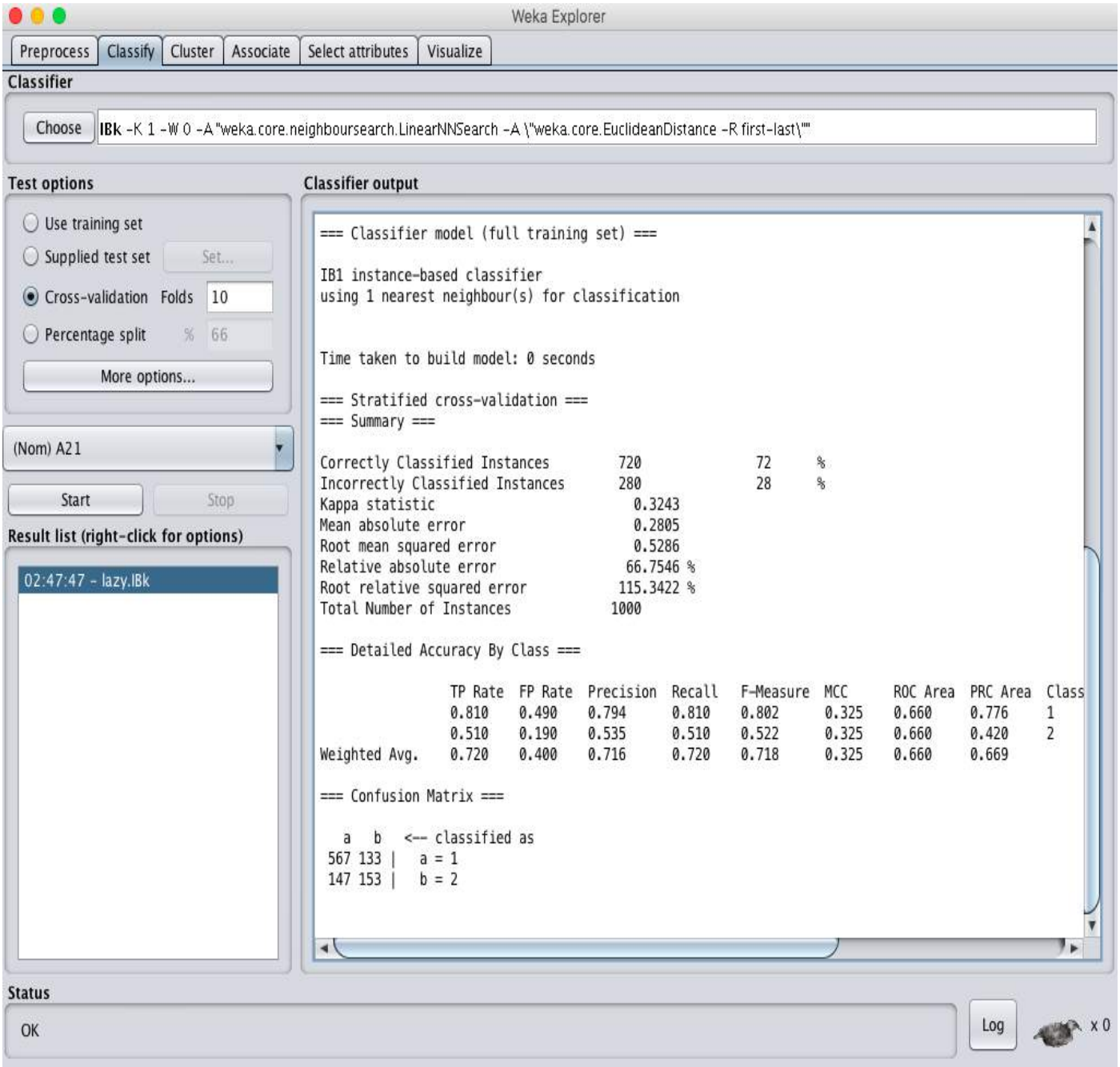
It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

Steps :

- Load the dataset into the weka tool and preprocess it.
- Apply the classification the K- Nearest Neighbor technique and execute for the result.

Output :

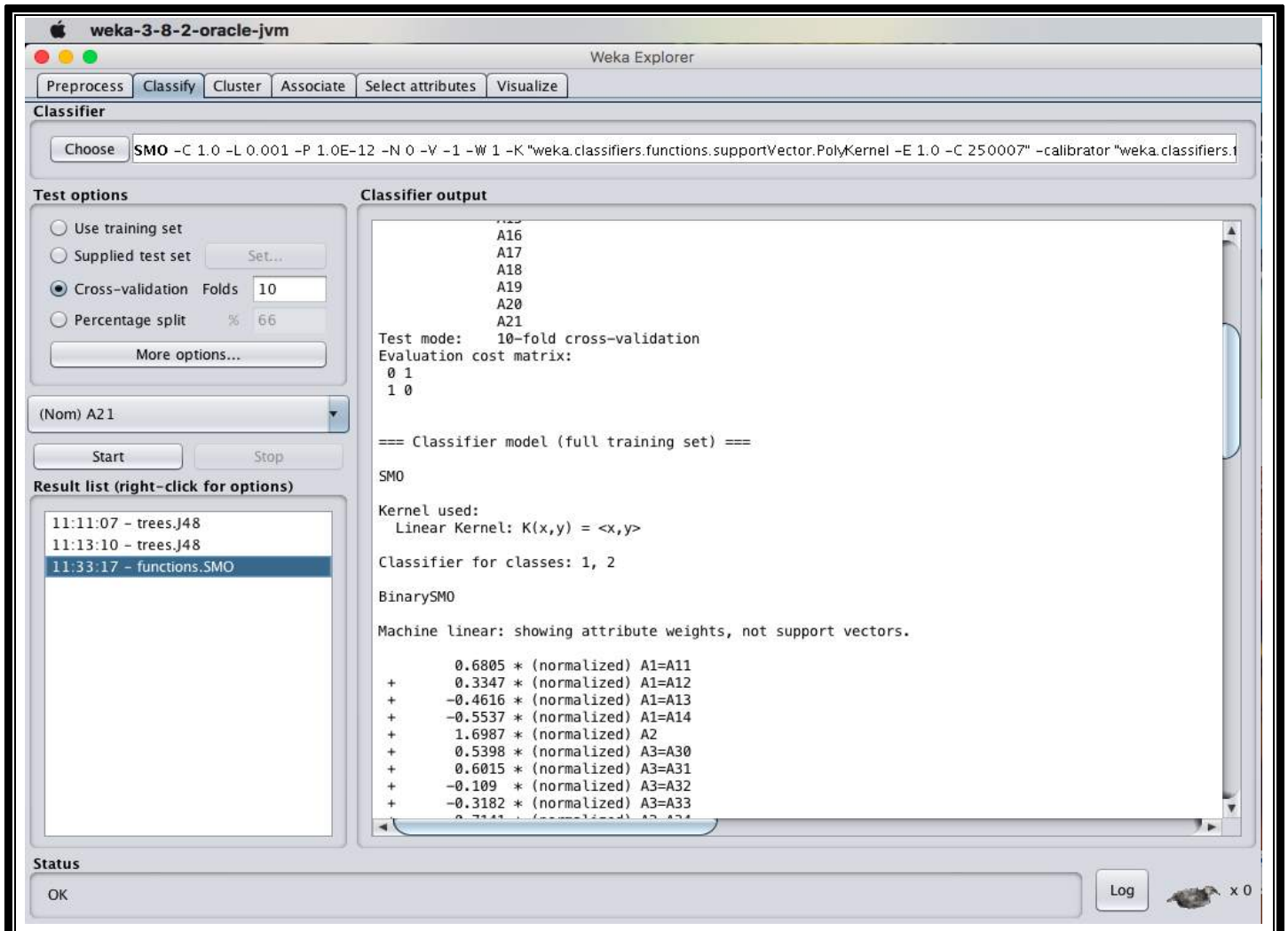


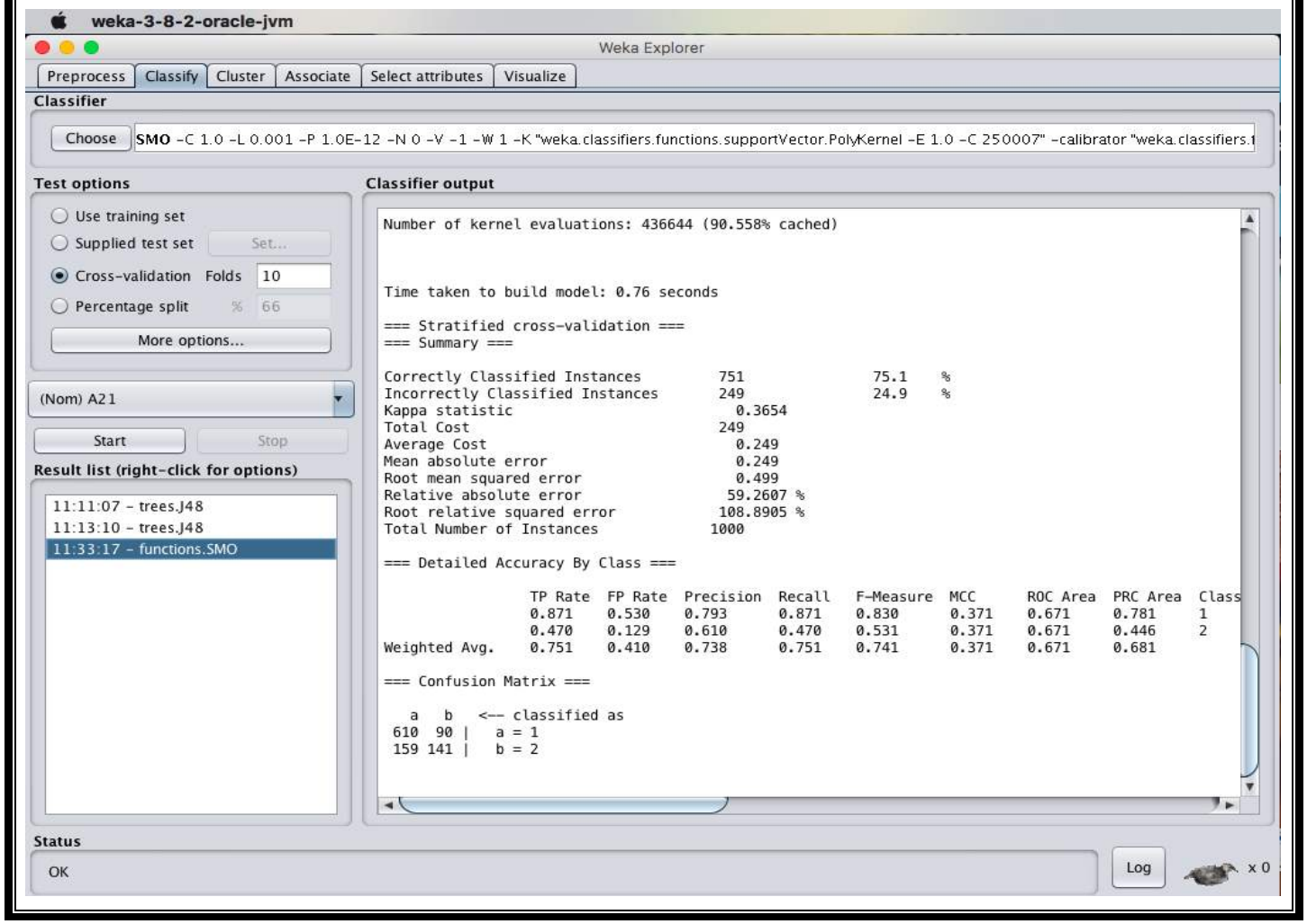
E) SMO Algorithm :

The iterative algorithm Sequential Minimal Optimization (SMO) is used for solving quadratic programming (QP) problems. One example where QP problems are relevant is during the training process of support vector machines (SVM). The SMO algorithm is used to solve in this example a constraint optimization problem. John Platt proposed this algorithm in 1998 and it was successfully used since then. We describe here the basics of the algorithm in the light of big data.

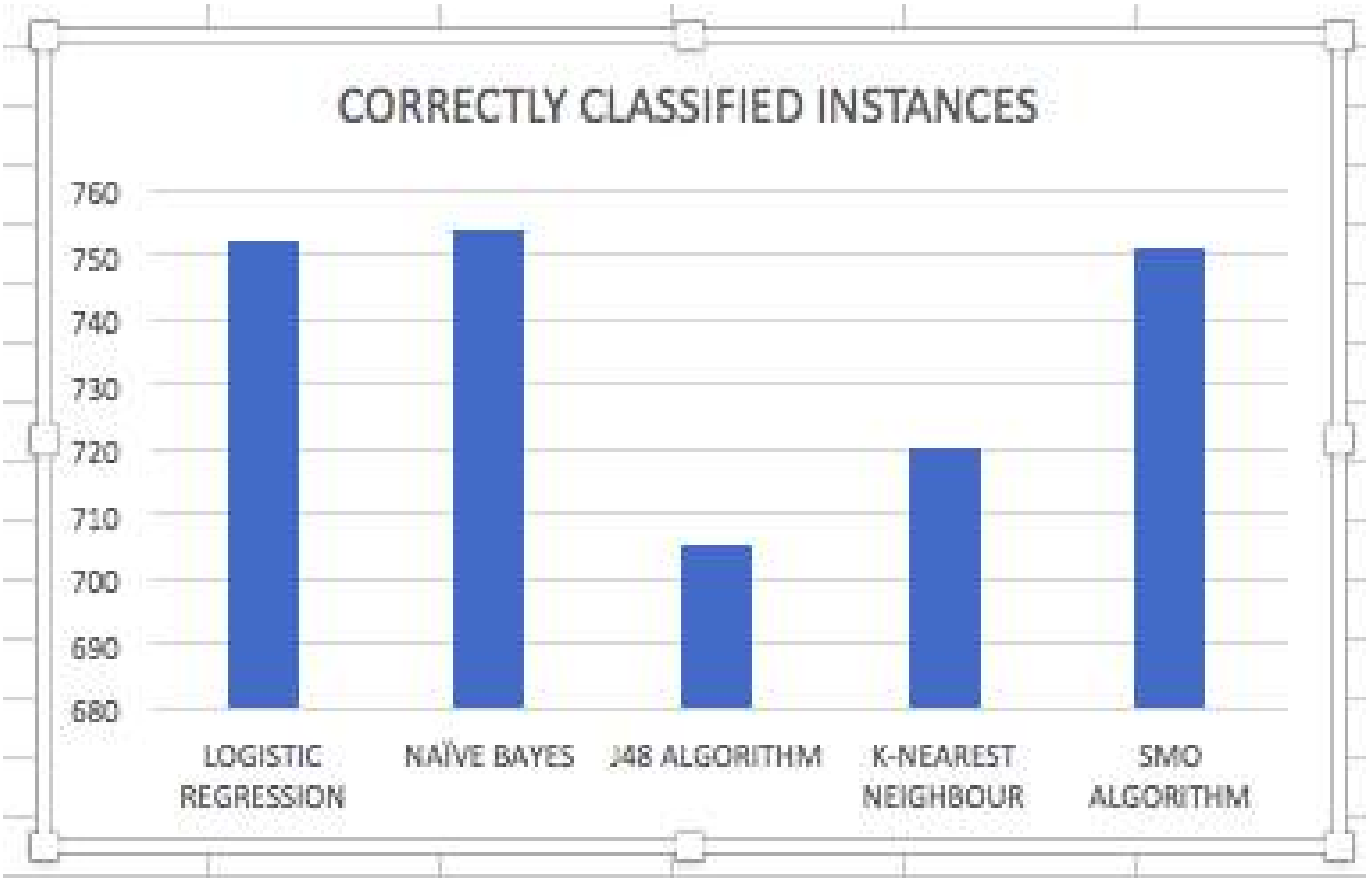
Steps :

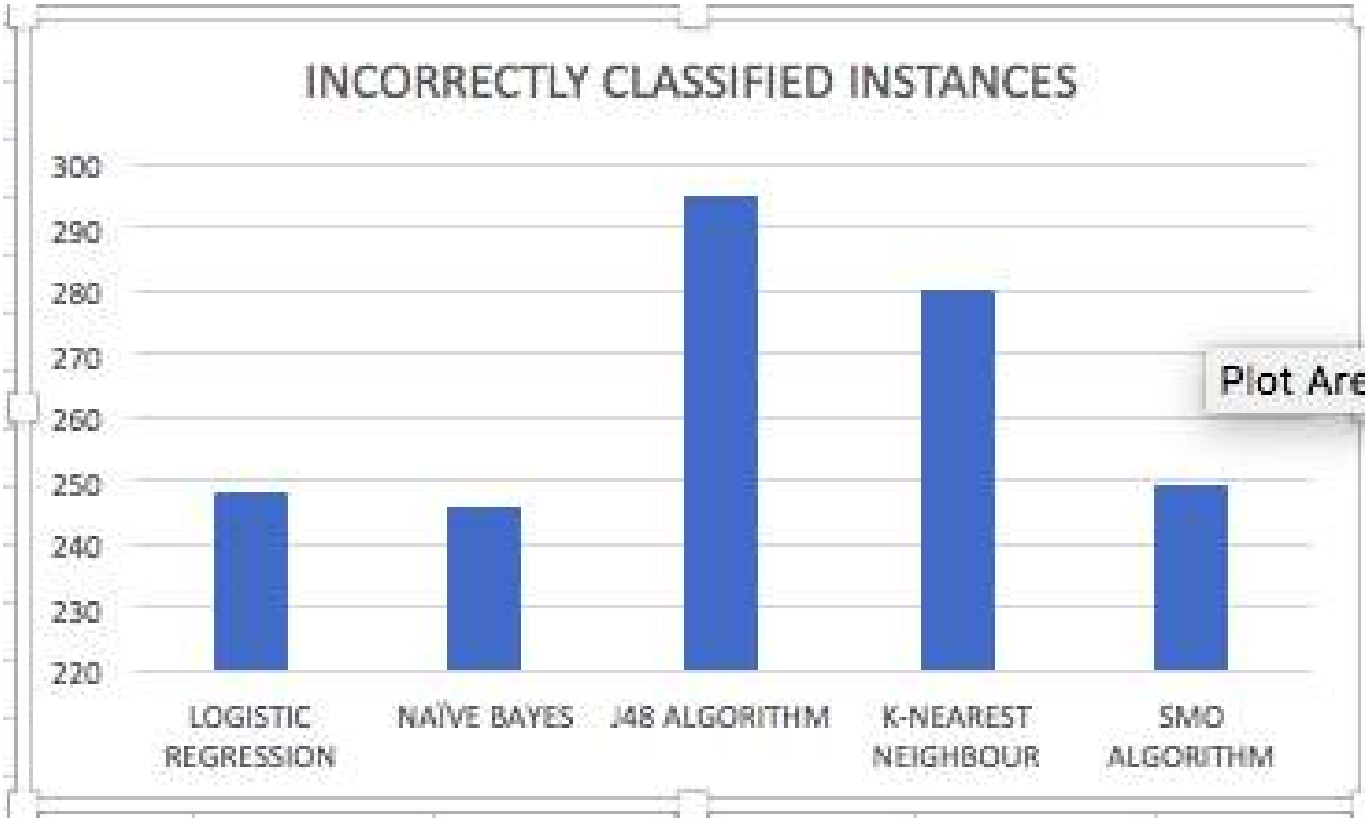
- Load the dataset into the weka tool and preprocess it.
- Apply the classification the Sequential Minimal Optimization (SMO)technique and execute for the result.

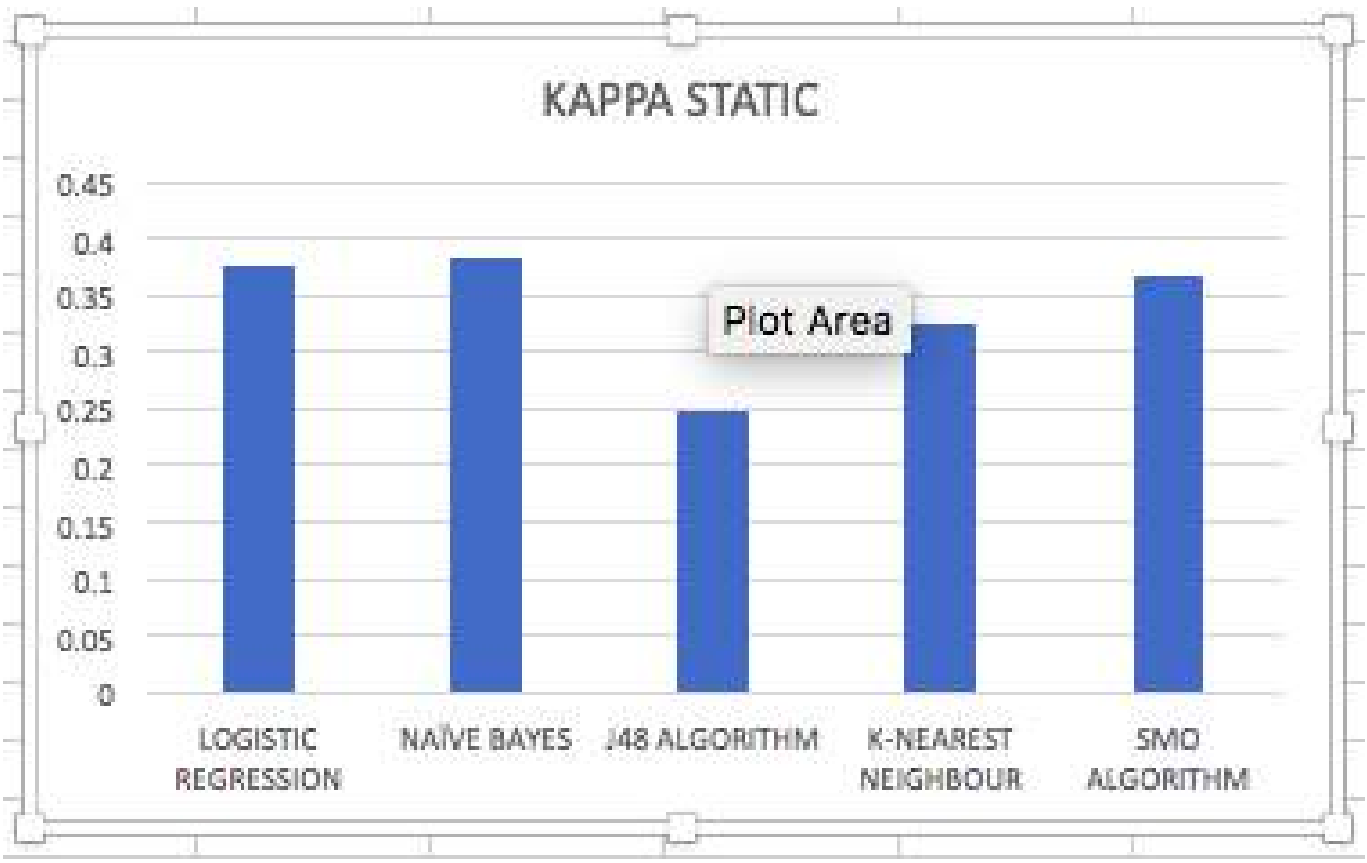

weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.t
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:33:17 - functions.SMO
Classifier output
A16
A17
A18
A19
A20
A21
Test mode: 10-fold cross-validation
Evaluation cost matrix:
0 1
1 0
=== Classifier model (full training set) ===
SMO
Kernel used:
Linear Kernel: K(x,y) = <x,y>
Classifier for classes: 1, 2
BinarySMO
Machine linear: showing attribute weights, not support vectors.
0.6805 * (normalized) A1=A11
+ 0.3347 * (normalized) A1=A12
+ -0.4616 * (normalized) A1=A13
+ -0.5537 * (normalized) A1=A14
+ 1.6987 * (normalized) A2
+ 0.5398 * (normalized) A3=A30
+ 0.6015 * (normalized) A3=A31
+ -0.109 * (normalized) A3=A32
+ -0.3182 * (normalized) A3=A33
Status
OK
Log
x 0

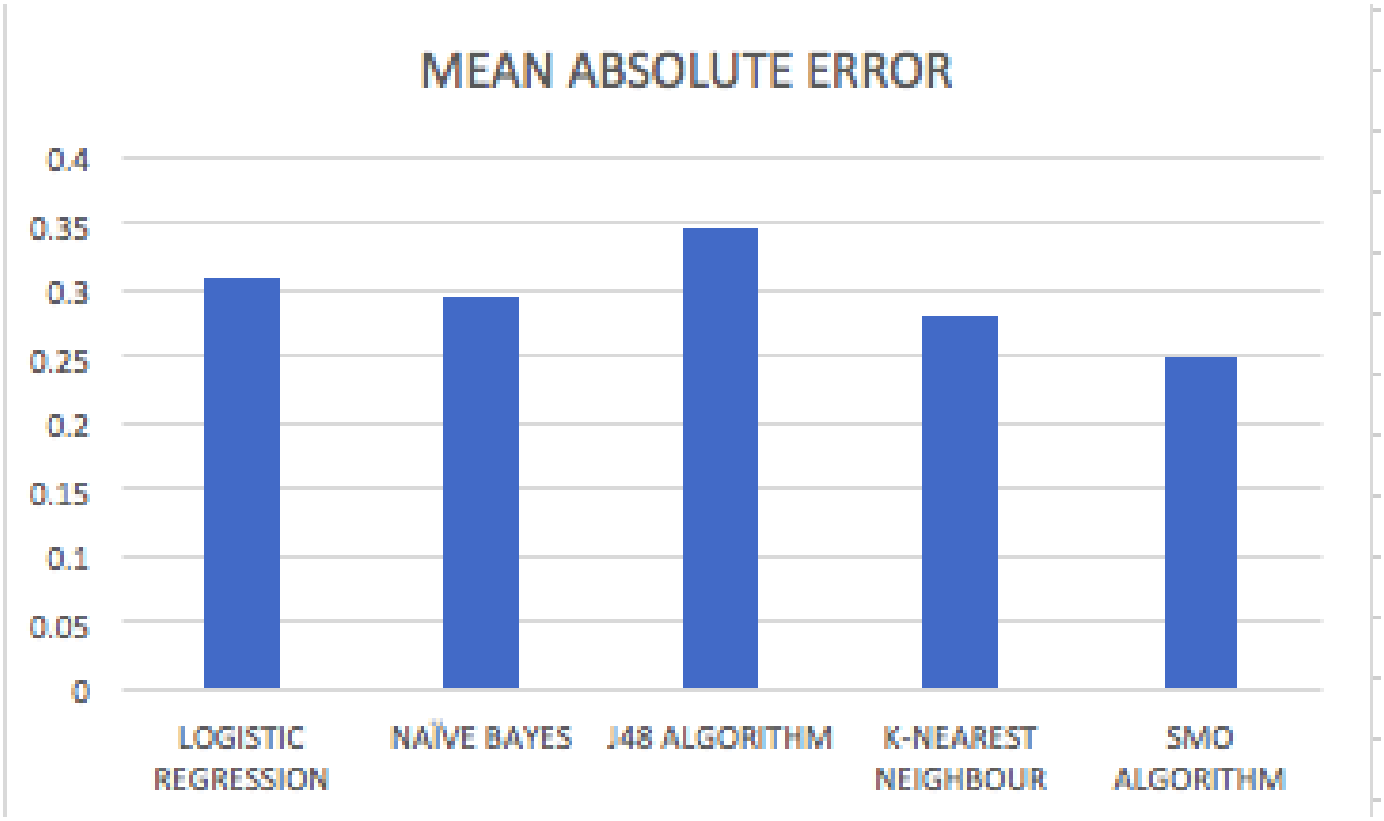

weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.t
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:33:17 - functions.SMO
Classifier output
Number of kernel evaluations: 436644 (90.558% cached)
Time taken to build model: 0.76 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 751 75.1 %
Incorrectly Classified Instances 249 24.9 %
Kappa statistic 0.3654
Total Cost 249
Average Cost 0.249
Mean absolute error 0.249
Root mean squared error 0.499
Relative absolute error 59.2607 %
Root relative squared error 108.8905 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.871 0.530 0.793 0.871 0.830 0.371 0.671 0.781 1
0.470 0.129 0.610 0.470 0.531 0.371 0.671 0.446 2
Weighted Avg. 0.751 0.410 0.738 0.751 0.741 0.371 0.671 0.681
=== Confusion Matrix ===
a b <-- classified as
610 90 | a = 1
159 141 | b = 2
Status
OK
Log
x 0

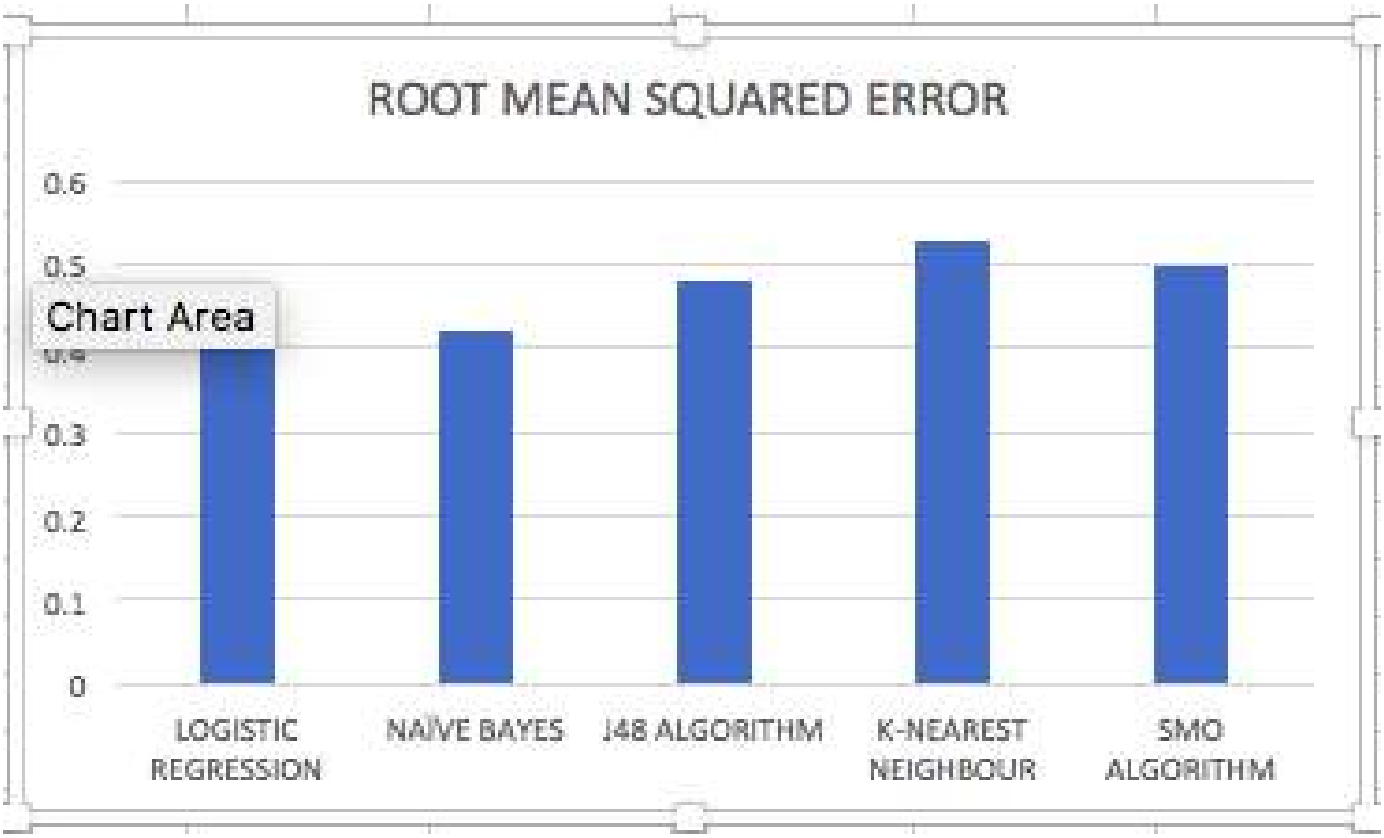
COMPARISION OF VALUES :

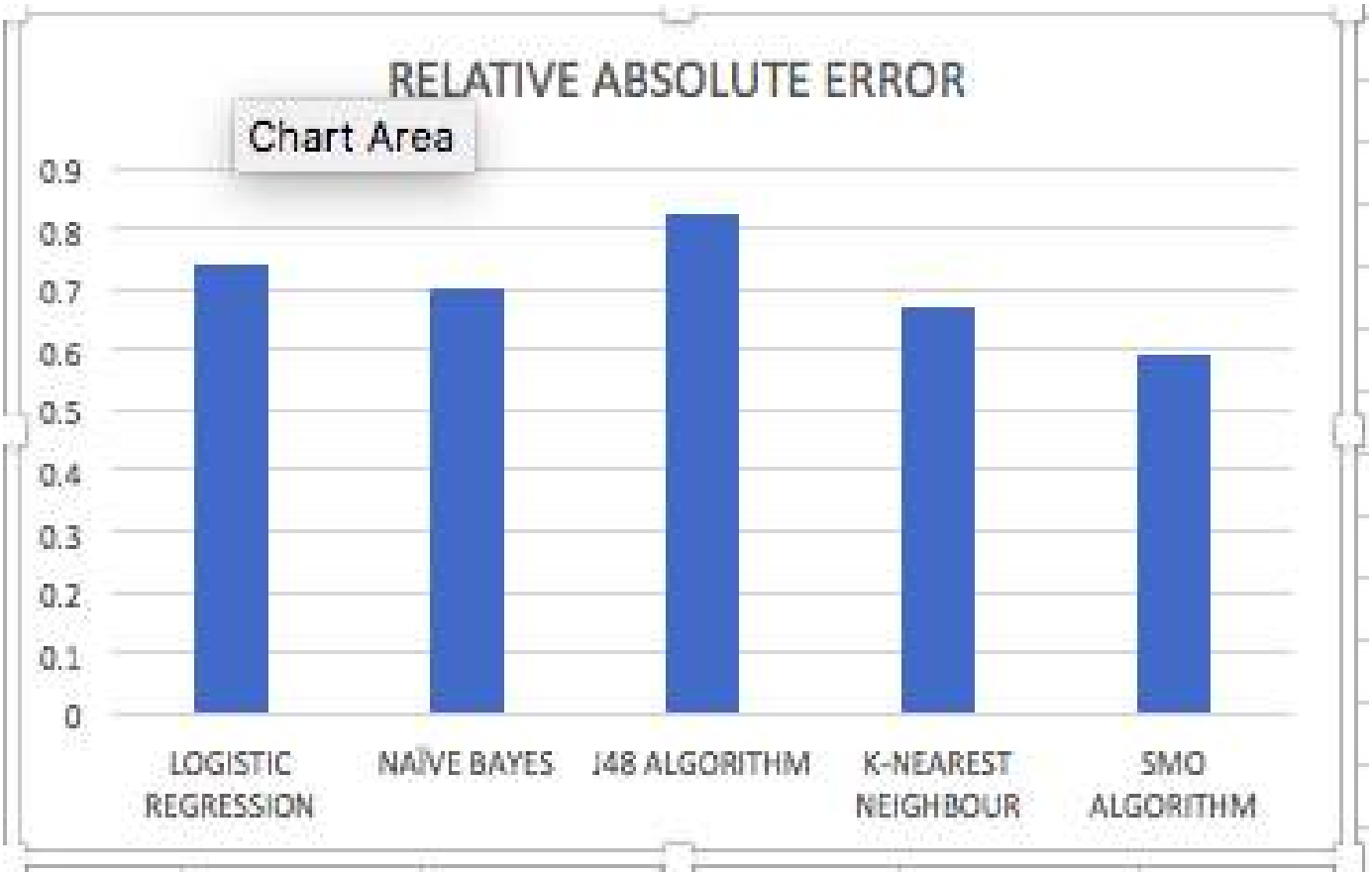


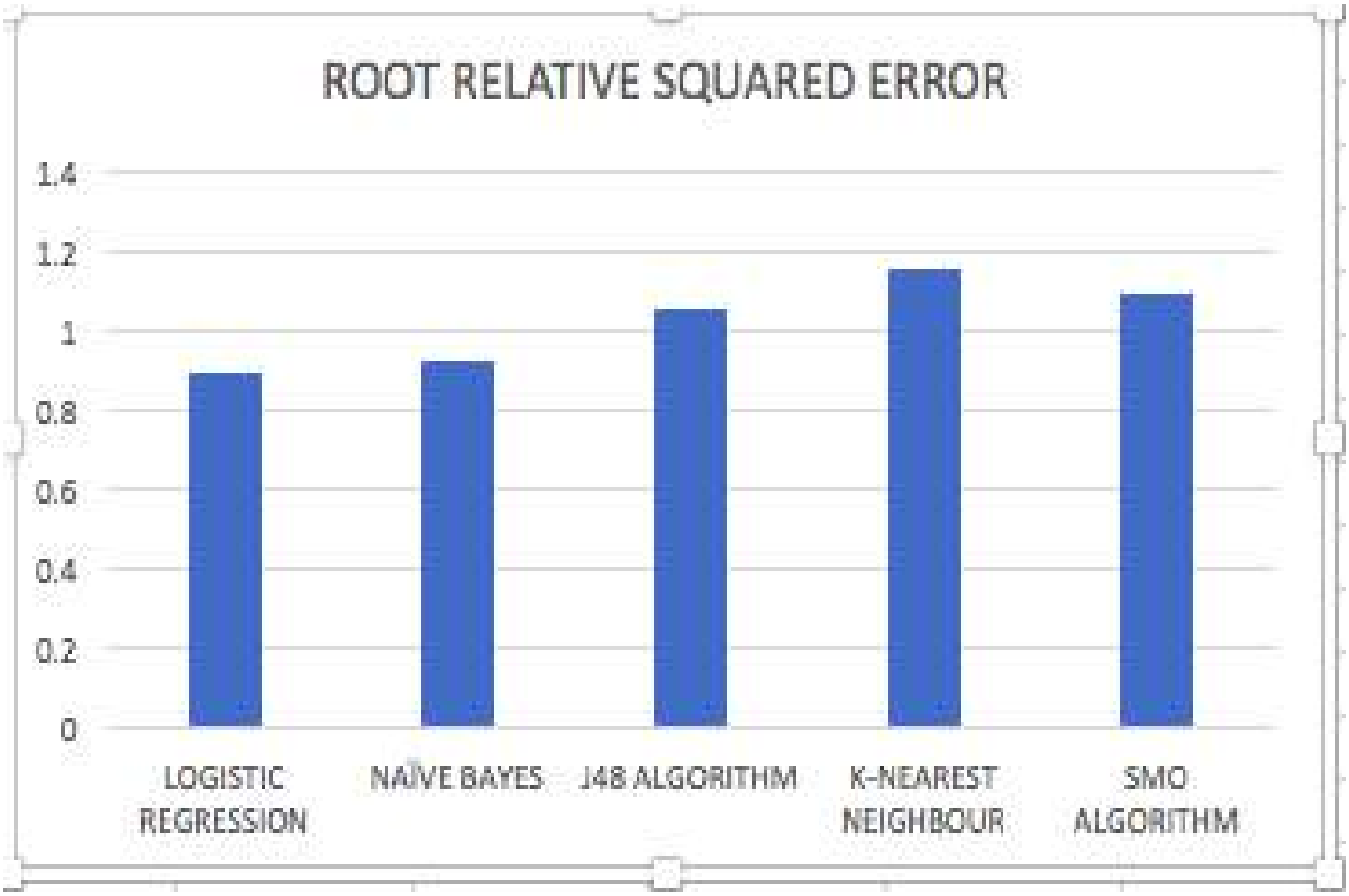



KAPPA STATIC
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Plot Area
LOGISTIC REGRESSION
NAÏVE BAYES
J48 ALGORITHM
K-NEAREST NEIGHBOUR
SMO ALGORITHM


MEAN ABSOLUTE ERROR
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
LOGISTIC REGRESSION
NAÏVE BAYES
J48 ALGORITHM
K-NEAREST NEIGHBOUR
SMO ALGORITHM


ROOT MEAN SQUARED ERROR
Chart Area
0.6
0.5
0.3
0.2
0.1
0
LOGISTIC REGRESSION
NAÏVE BAYES
J48 ALGORITHM
K-NEAREST NEIGHBOUR
SMO ALGORITHM


RELATIVE ABSOLUTE ERROR
Chart Area
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
LOGISTIC REGRESSION
NAÏVE BAYES
J48 ALGORITHM
K-NEAREST NEIGHBOUR
SMO ALGORITHM


ROOT RELATIVE SQUARED ERROR
1.4
1.2
1
0.8
0.6
0.4
0.2
0
LOGISTIC REGRESSION
NAÏVE BAYES
J48 ALGORITHM
K-NEAREST NEIGHBOUR
SMO ALGORITHM

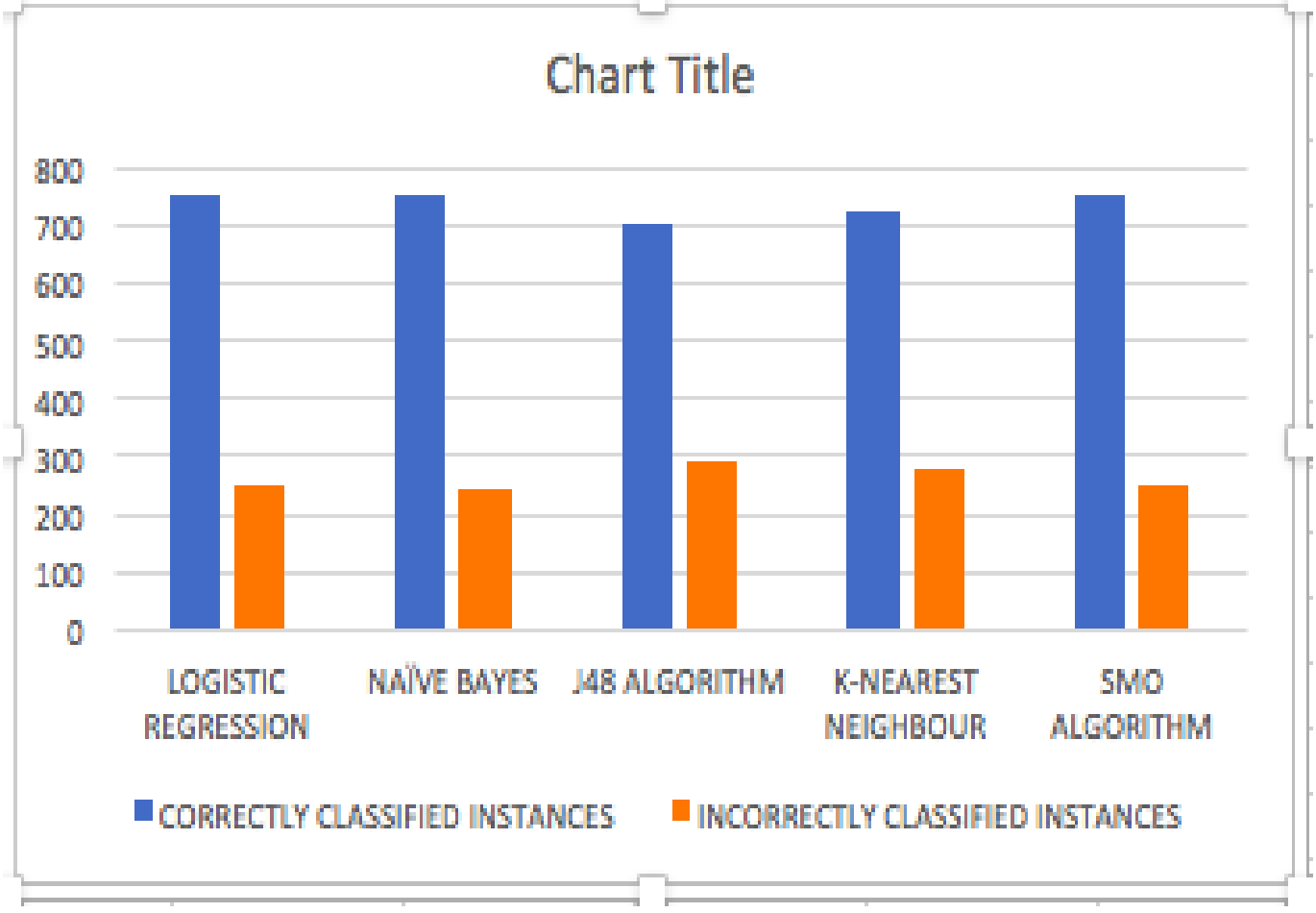


Chart Title
800
700
600
500
400
300
200
100
0
LOGISTIC REGRESSION
NAÏVE BAYES
J48 ALGORITHM
K-NEAREST NEIGHBOUR
SMO ALGORITHM
CORRECTLY CLASSIFIED INSTANCES
INCORRECTLY CLASSIFIED INSTANCES

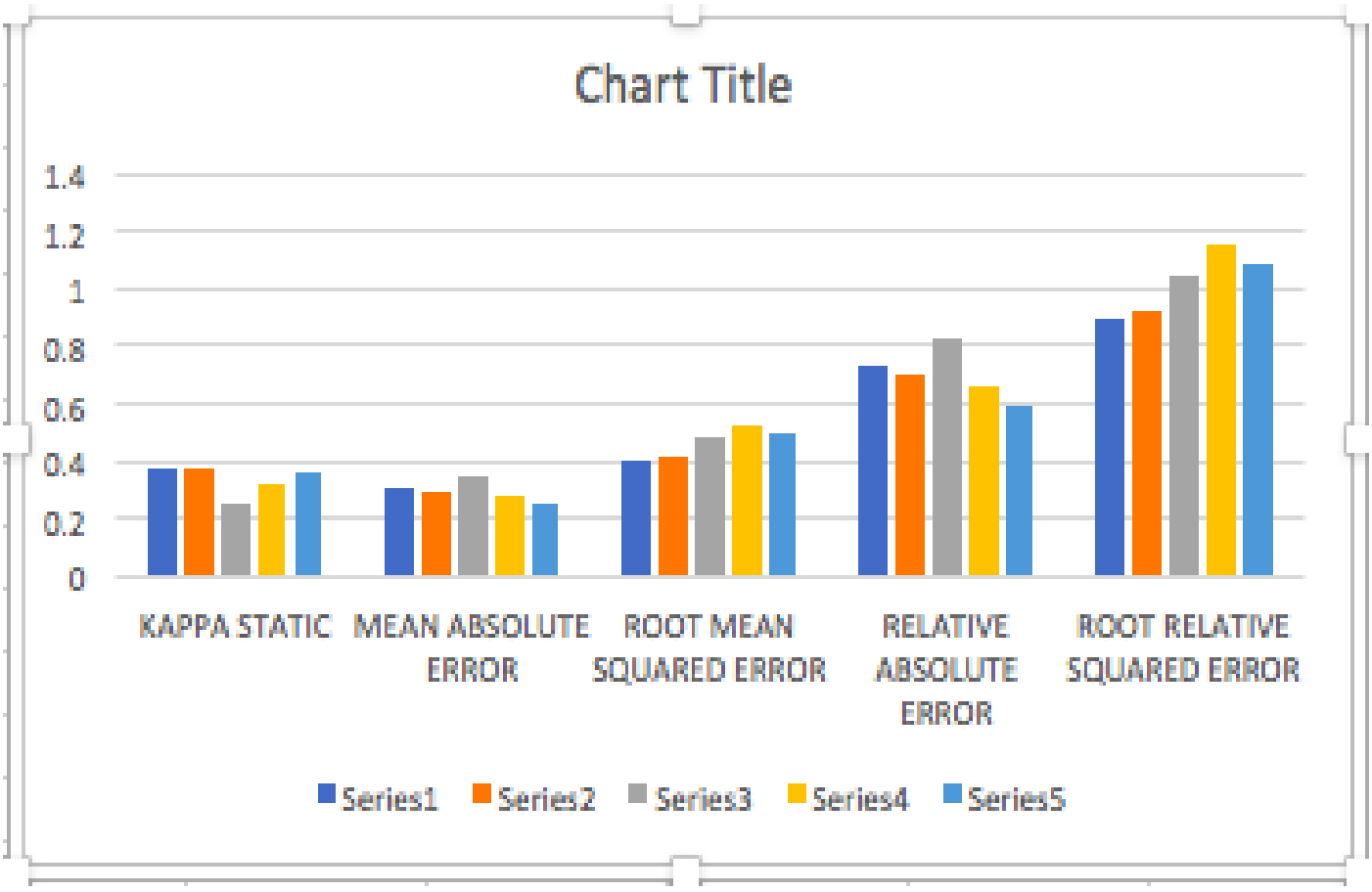


Chart Title
1.4
1.2
1
0.8
0.6
0.4
0.2
0
KAPPA STATIC
MEAN ABSOLUTE ERROR
ROOT MEAN SQUARED ERROR
RELATIVE ABSOLUTE ERROR
ROOT RELATIVE SQUARED ERROR
Series1
Series2
Series3
Series4
Series5

RESULT :

Thus, the comparison of the confusion matrix for all the methods and techniques. Out of the comparing matrix with all the techniques there is a change in instances. Naïve bayes has more number of correct instances than other but when compared to time K-nearest neighbor is best. The above graphs will show the variations of values in the parameters.

EX.No: 14
Date :

NUMERICAL PREDICTION ANALYSIS USING LINEAR REGRESSION THROUGH WEKA

PROBLEM STATEMENT :

Using regression analysis create a model to calculate the price of the house. Create the model based on other comparable houses in the neighborhood and how much they sold for.

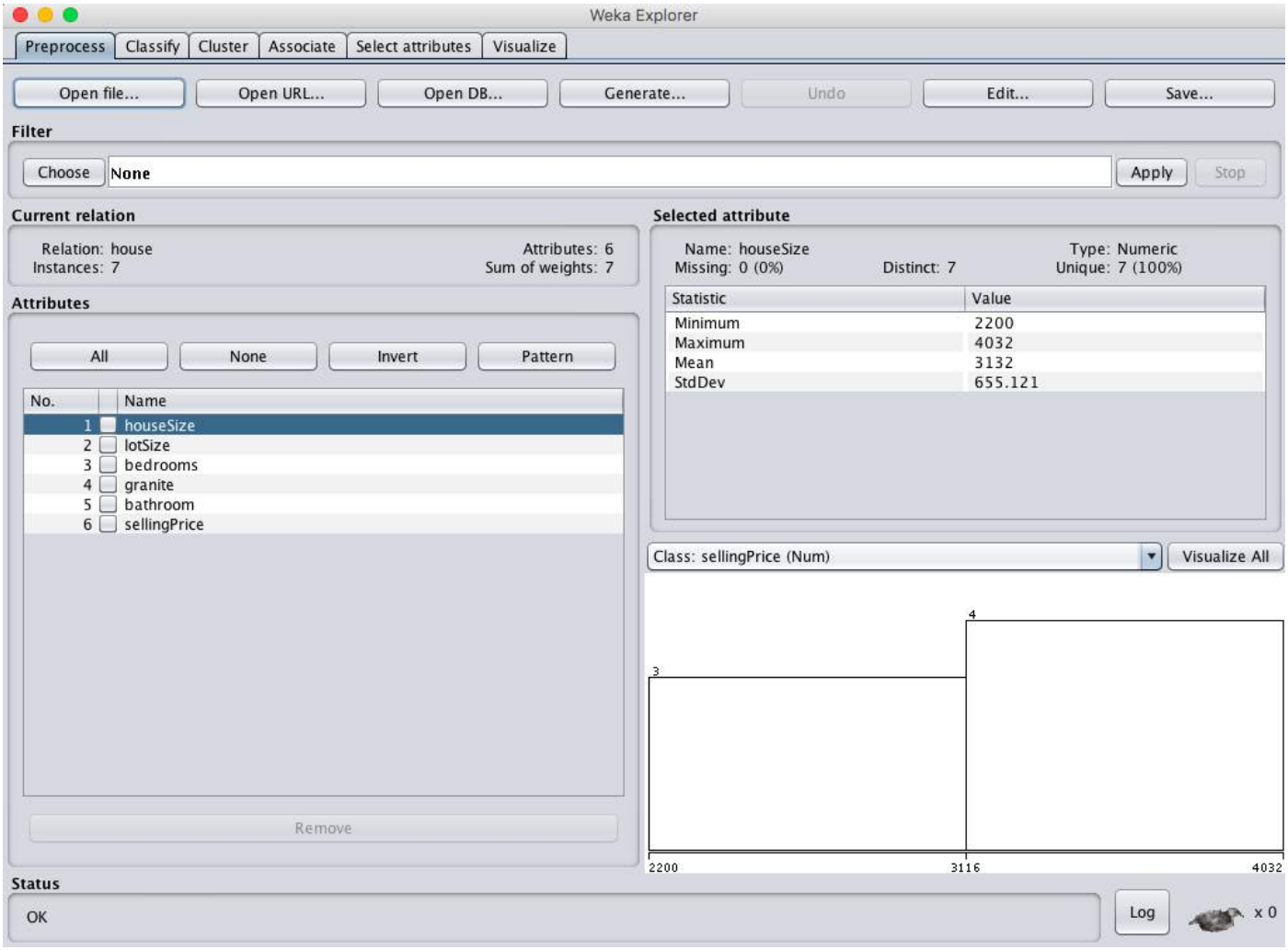
Build the dataset for weka in arff file format and load the dataset into weka and finally create the regression model with weka.

DESCRIPTION :

Consider a dataset of house.arff where it contains the attributes as house size, lot size, bedrooms, granite, bathroom and the selling price.

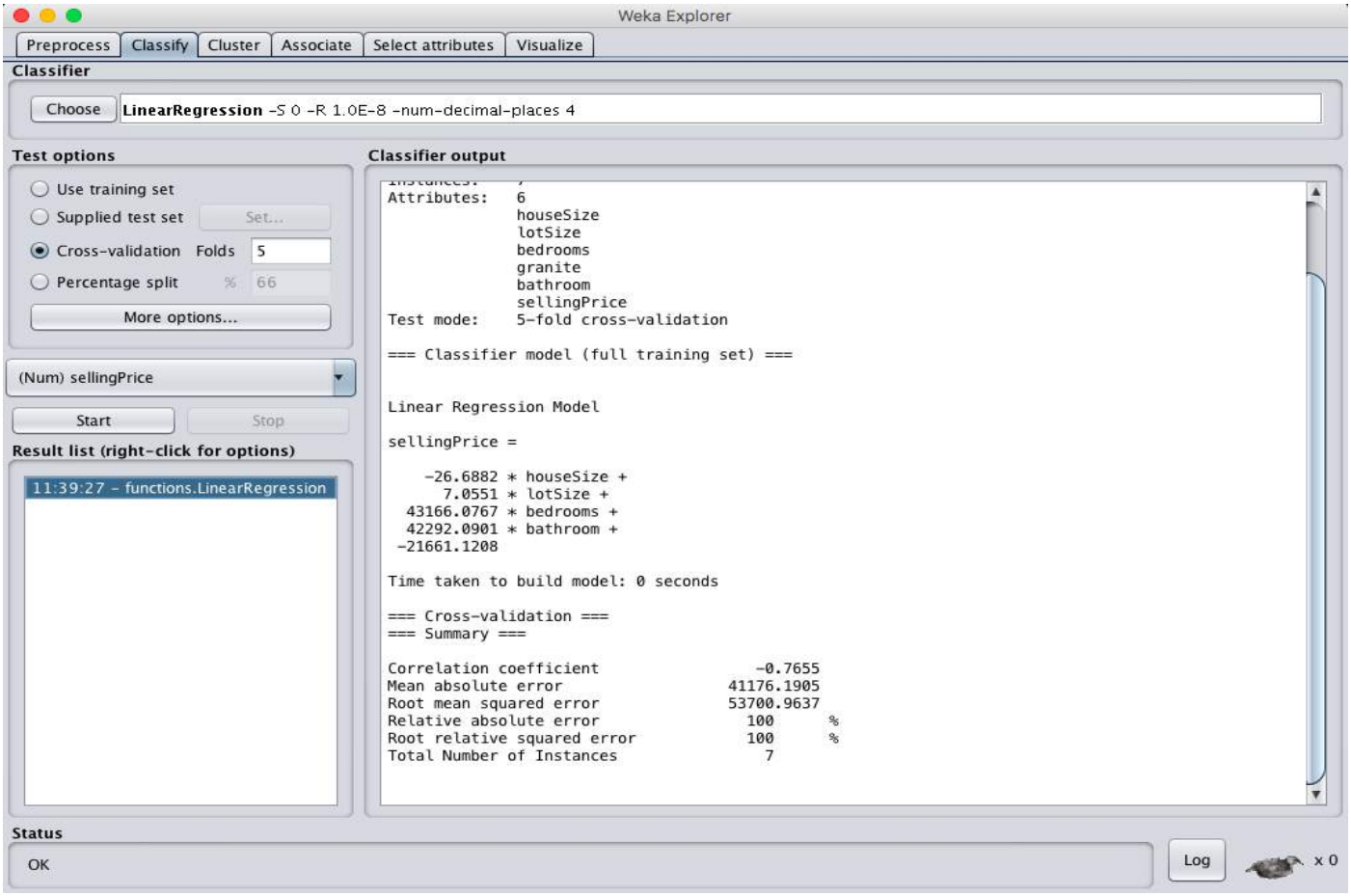
Steps :

- Load the dataset into the weka tool and check for the attributes.
- Classify the data using linear regression analysis method (or) technique.
- Check for the cross-validation folds where the value of the folds should be less than the value of the instances present in the dataset.
- Observe the cross validation summary after applying the linear regression technique for the price of the house.

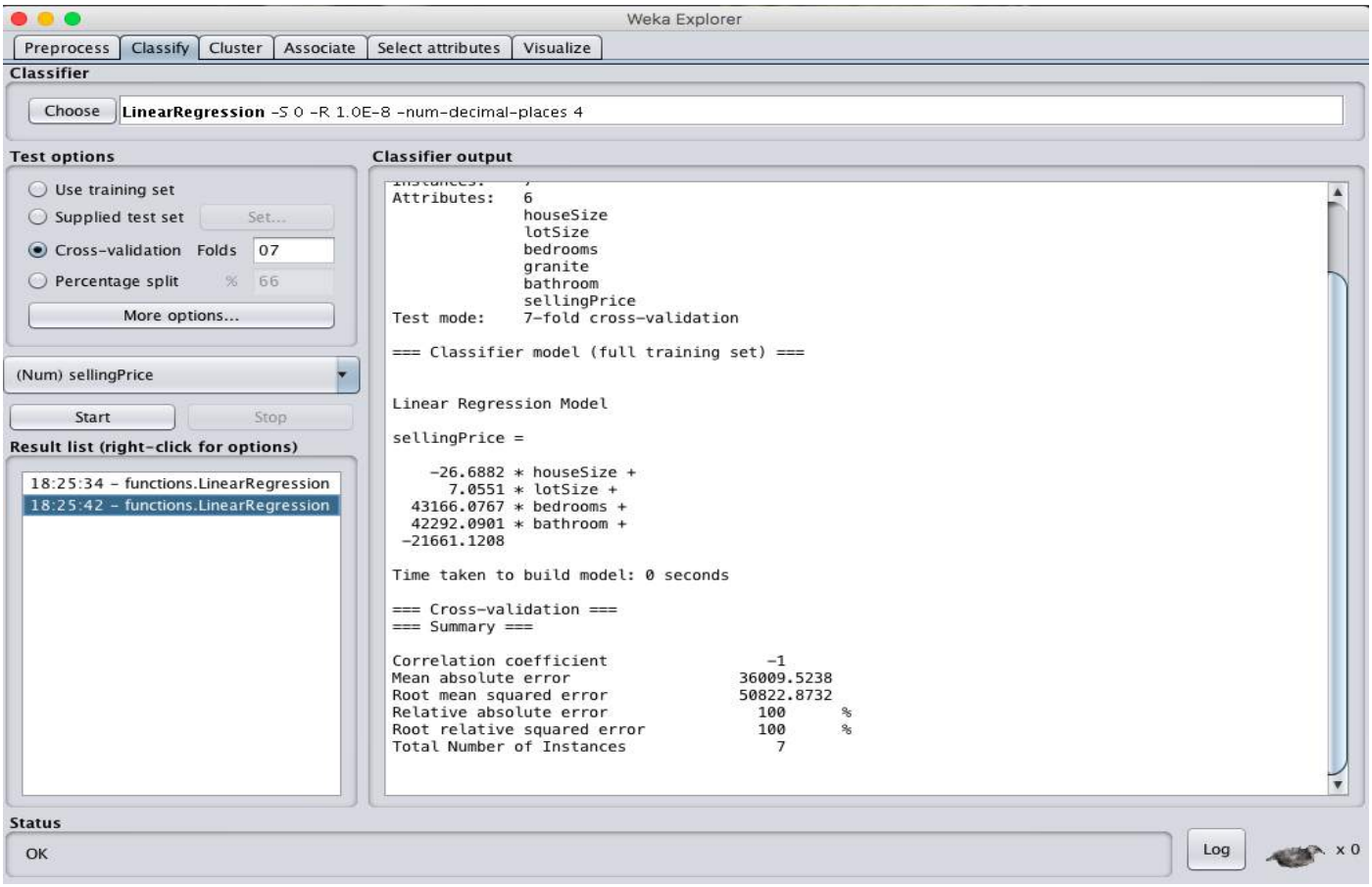


OBSERVATION :

- **When cross validation folds = 05 :**



- **When cross validation folds = 10 :**



RESULT :

Thus, the house selling price has been observed using linear regression model. If the value of cross validation folds decreases time for creating model will be less than when folds value high, and the mean absolute error and Root mean square error values decreases with increase in the cross validation folds value.

EX.No: 15
Date :

EXTRACT TRANSFORM LOAD (ETL) AND OLAP OPERATION USING KNIME TOOL

PROBLEM STATEMENT :

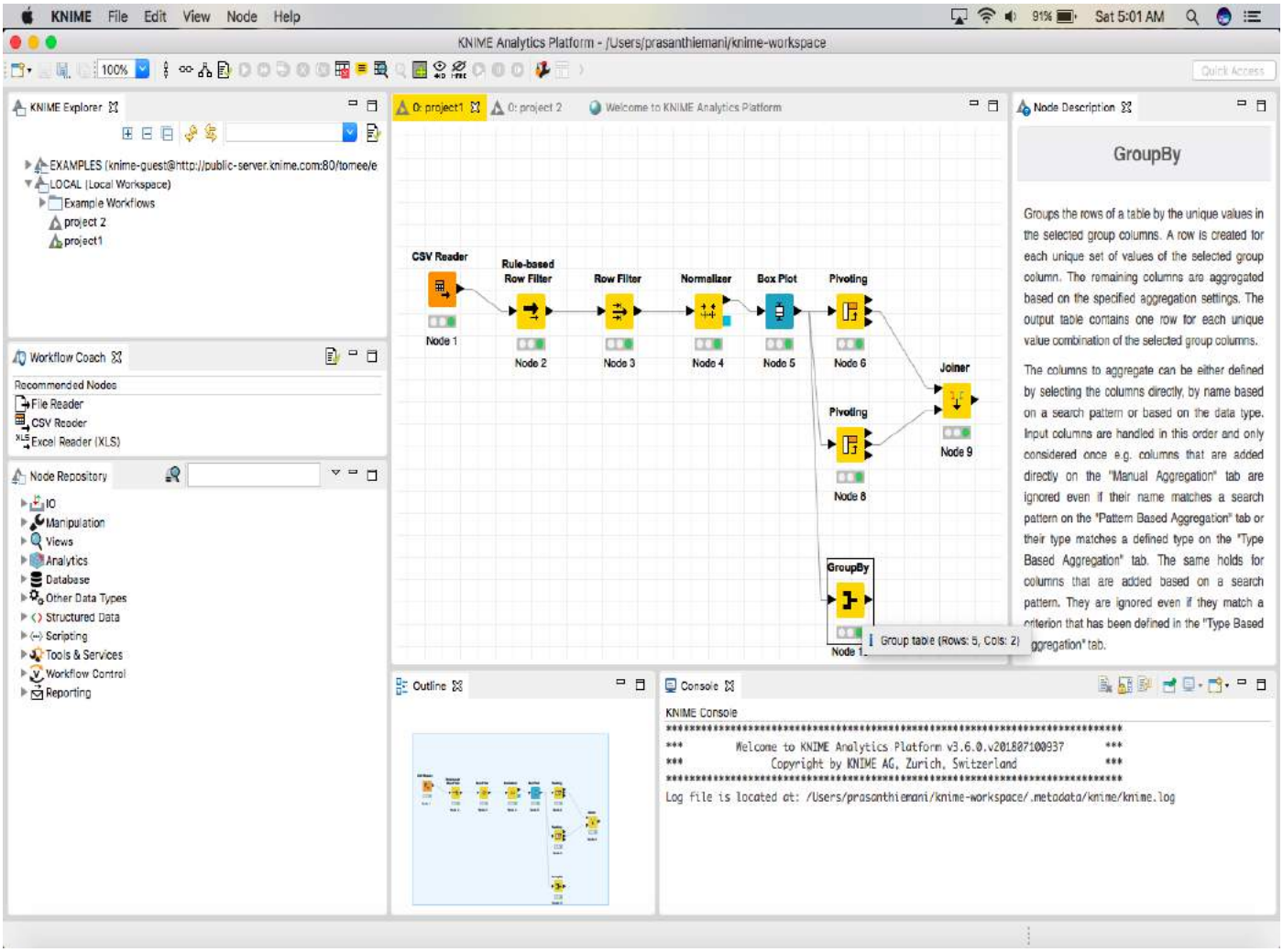
Extract the data from csv file, transform[use row filter and rule based filter] use pivot and group by] load the data for reporting (Visualization).

DESCRIPTION :

Consider a dataset movies.csv where it contains the attributes ad title, genre, director, year, duration, gross, budget, cast_facebook_likes, votes, reviews, rating.

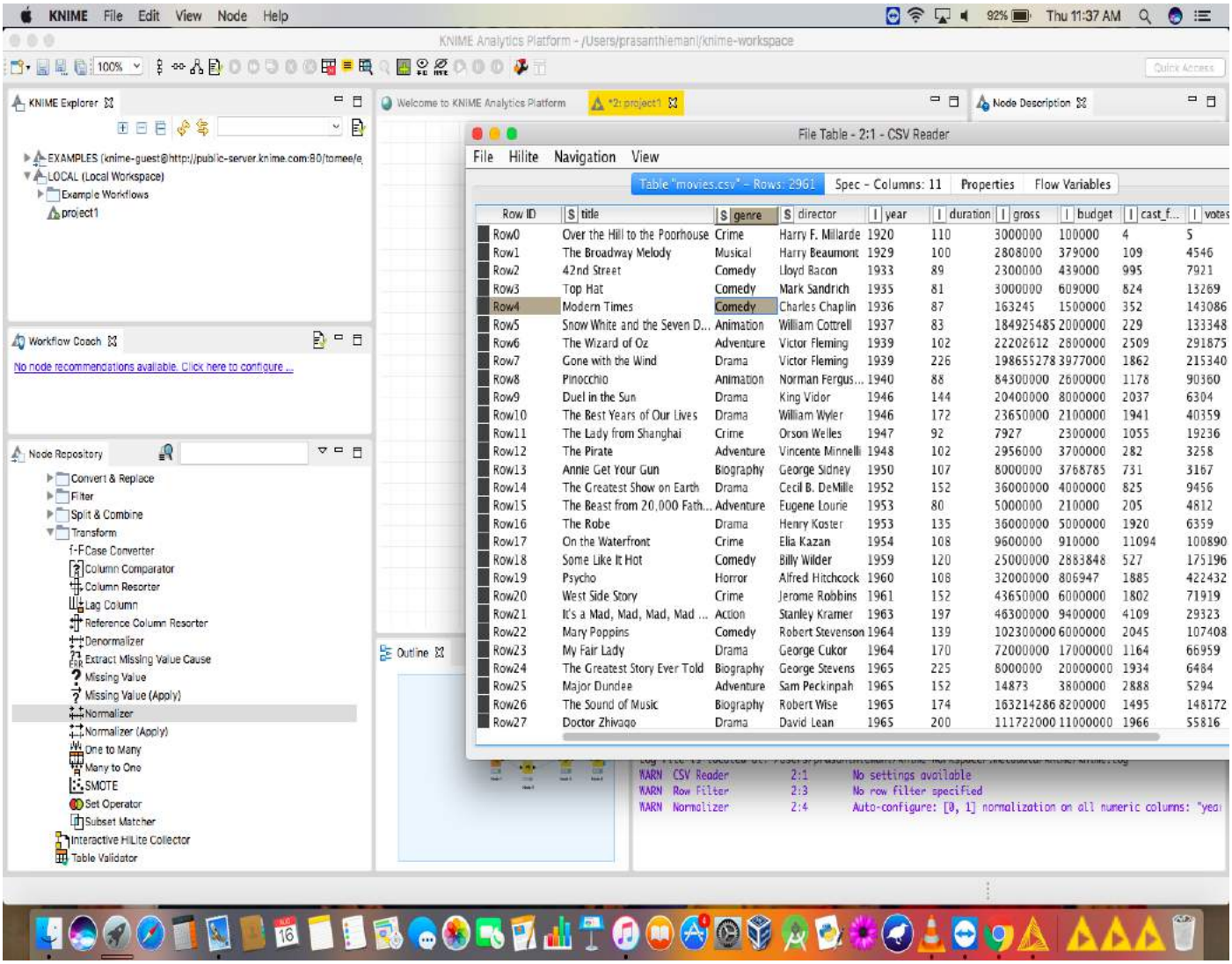
STEPS :

- Import the dataset into the knime tool using csv reader.



❖ **CSV Reader :**

Execute the CSV reader,look at the table of the loaded dataset.



❖ **Rule-Based Row Filter :**

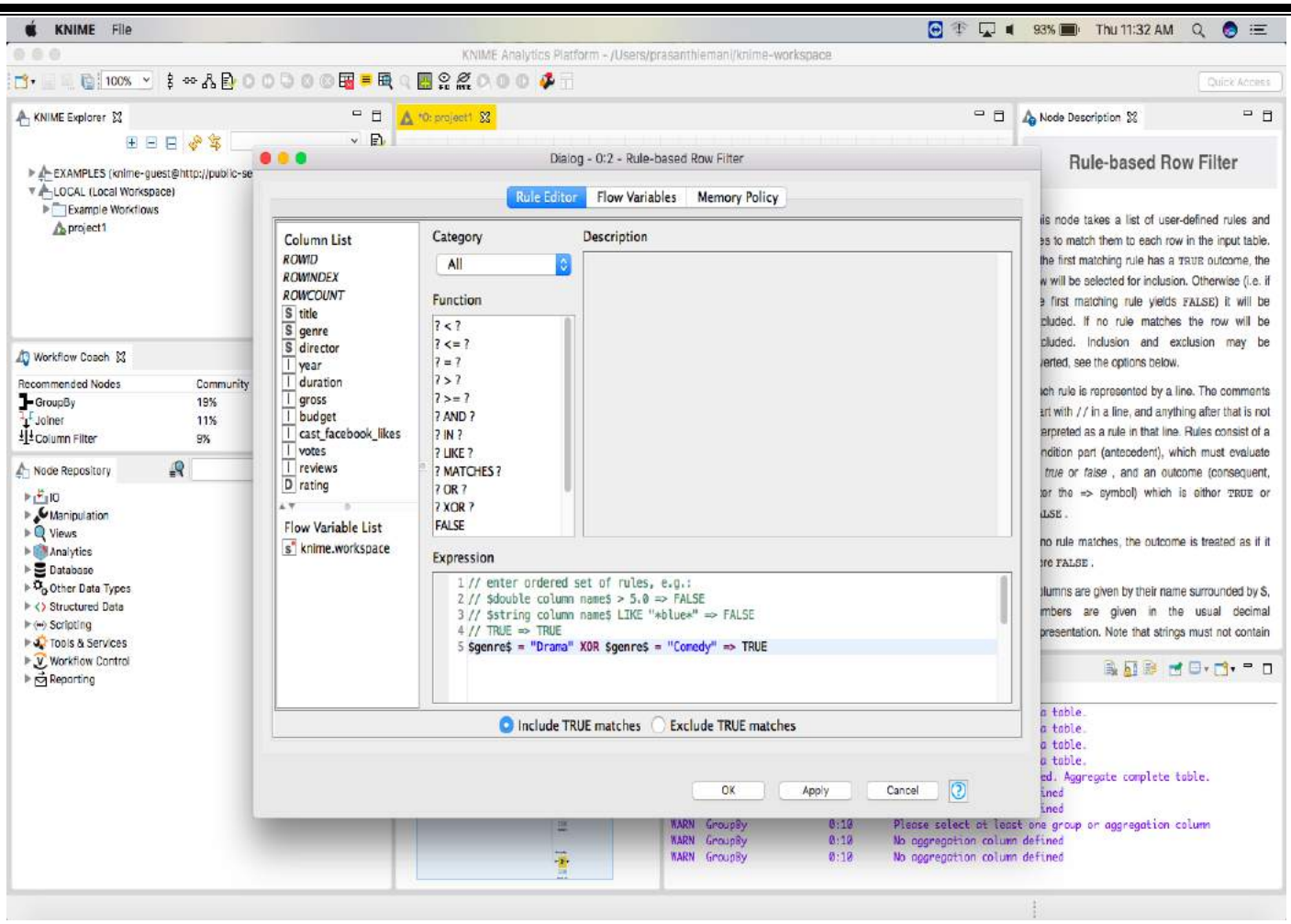
This node takes a list of user-defined rules and tries to match them to each row in the input table. If the first matching rule has a TRUE outcome, the row will be selected for inclusion. Otherwise (i.e. if the first matching rule yields FALSE) it will be excluded. If no rule matches the row will be excluded.

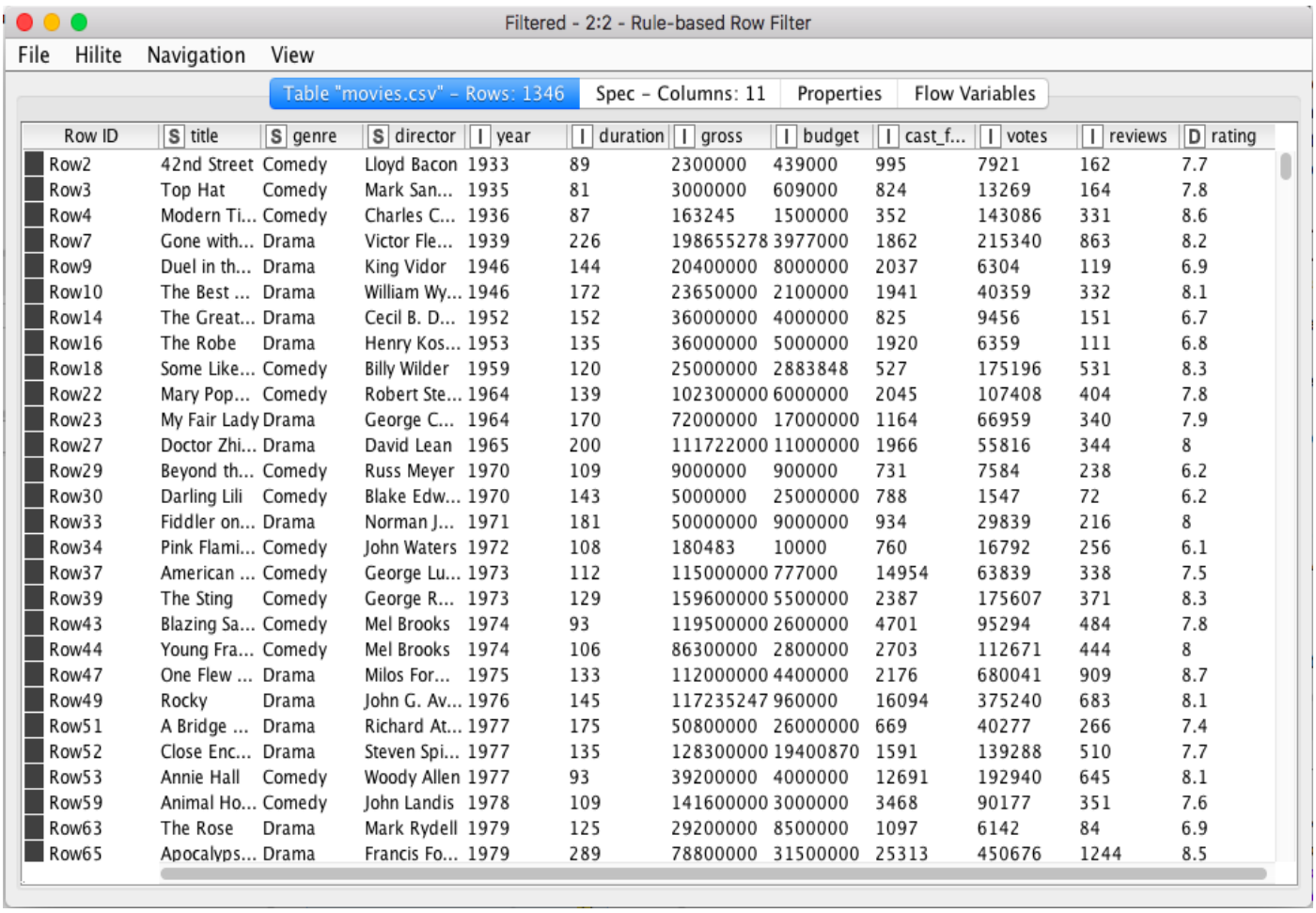
Steps :

- Make a connection between CSV reader and rule based row filter.
- Configure rule based row filter.
- Execute and Check out for the table after applying rule based row filter.

Code :

$genre$ = “Drama” XOR $genre$ = “Comedy” => TRUE





Row ID	title	genre	director	year	duration	gross	budget	cast_f...	votes	reviews	rating
Row2	42nd Street	Comedy	Lloyd Bacon	1933	89	2300000	439000	995	7921	162	7.7
Row3	Top Hat	Comedy	Mark San...	1935	81	3000000	609000	824	13269	164	7.8
Row4	Modern Ti...	Comedy	Charles C...	1936	87	163245	1500000	352	143086	331	8.6
Row7	Gone with...	Drama	Victor Fle...	1939	226	198655278	3977000	1862	215340	863	8.2
Row9	Duel in th...	Drama	King Vidor	1946	144	20400000	8000000	2037	6304	119	6.9
Row10	The Best ...	Drama	William Wy...	1946	172	23650000	2100000	1941	40359	332	8.1
Row14	The Great...	Drama	Cecil B. D...	1952	152	36000000	4000000	825	9456	151	6.7
Row16	The Robe	Drama	Henry Kos...	1953	135	36000000	5000000	1920	6359	111	6.8
Row18	Some Like...	Comedy	Billy Wilder	1959	120	25000000	2883848	527	175196	531	8.3
Row22	Mary Pop...	Comedy	Robert Ste...	1964	139	102300000	6000000	2045	107408	404	7.8
Row23	My Fair Lady	Drama	George C...	1964	170	72000000	17000000	1164	66959	340	7.9
Row27	Doctor Zhi...	Drama	David Lean	1965	200	111722000	11000000	1966	55816	344	8
Row29	Beyond th...	Comedy	Russ Meyer	1970	109	9000000	900000	731	7584	238	6.2
Row30	Darling Lili	Comedy	Blake Edw...	1970	143	5000000	25000000	788	1547	72	6.2
Row33	Fiddler on...	Drama	Norman J...	1971	181	50000000	9000000	934	29839	216	8
Row34	Pink Flami...	Comedy	John Waters	1972	108	180483	10000	760	16792	256	6.1
Row37	American ...	Comedy	George Lu...	1973	112	115000000	777000	14954	63839	338	7.5
Row39	The Sting	Comedy	George R...	1973	129	159600000	5500000	2387	175607	371	8.3
Row43	Blazing Sa...	Comedy	Mel Brooks	1974	93	119500000	2600000	4701	95294	484	7.8
Row44	Young Fra...	Comedy	Mel Brooks	1974	106	86300000	2800000	2703	112671	444	8
Row47	One Flew ...	Drama	Milos For...	1975	133	112000000	4400000	2176	680041	909	8.7
Row49	Rocky	Drama	John G. Av...	1976	145	117235247	960000	16094	375240	683	8.1
Row51	A Bridge ...	Drama	Richard At...	1977	175	50800000	26000000	669	40277	266	7.4
Row52	Close Enc...	Drama	Steven Spi...	1977	135	128300000	19400870	1591	139288	510	7.7
Row53	Annie Hall	Comedy	Woody Allen	1977	93	39200000	4000000	12691	192940	645	8.1
Row59	Animal Ho...	Comedy	John Landis	1978	109	141600000	3000000	3468	90177	351	7.6
Row63	The Rose	Drama	Mark Rydell	1979	125	29200000	8500000	1097	6142	84	6.9
Row65	Apocalyps...	Drama	Francis Fo...	1979	289	78800000	31500000	25313	450676	1244	8.5

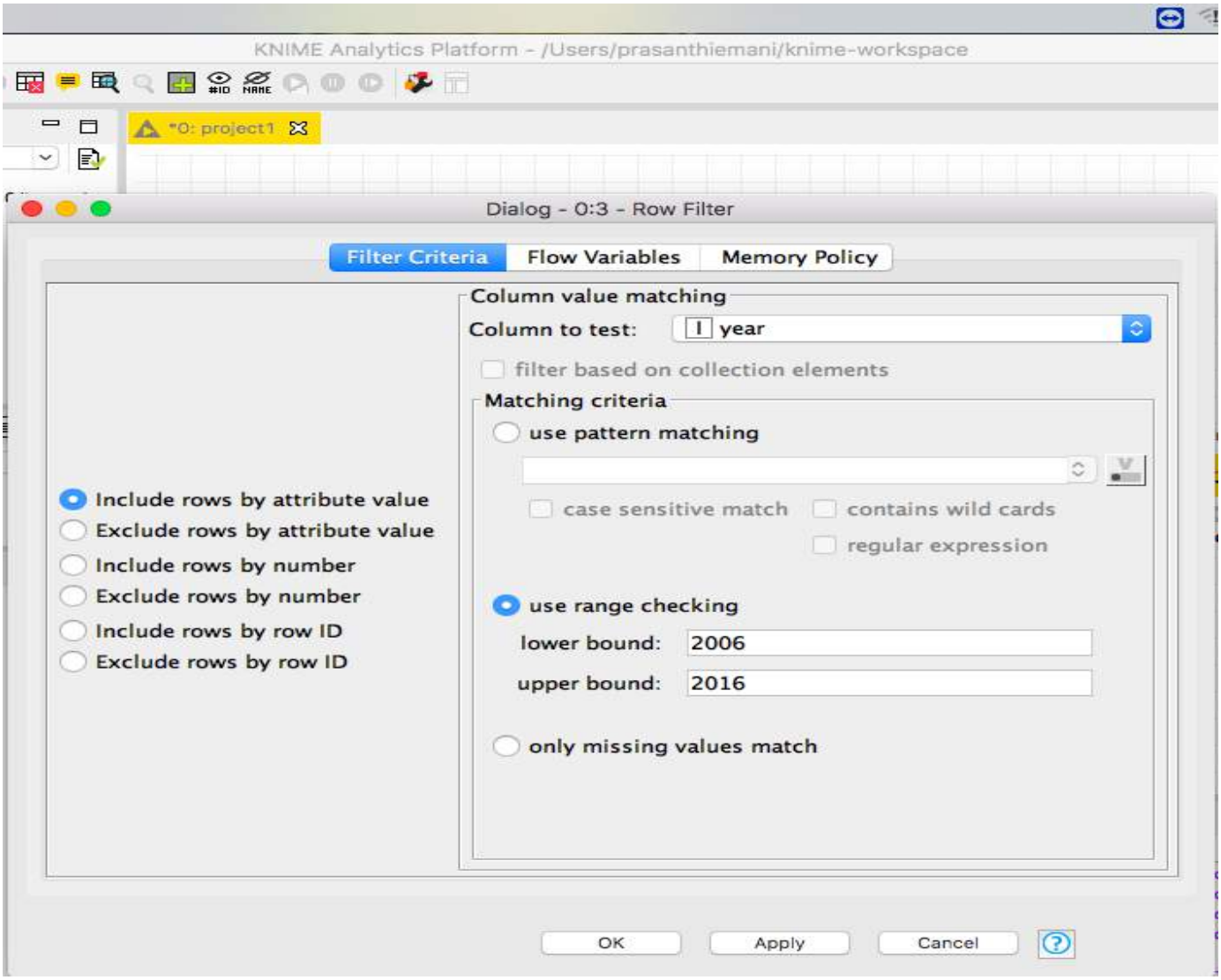
❖ **Row Filter :**

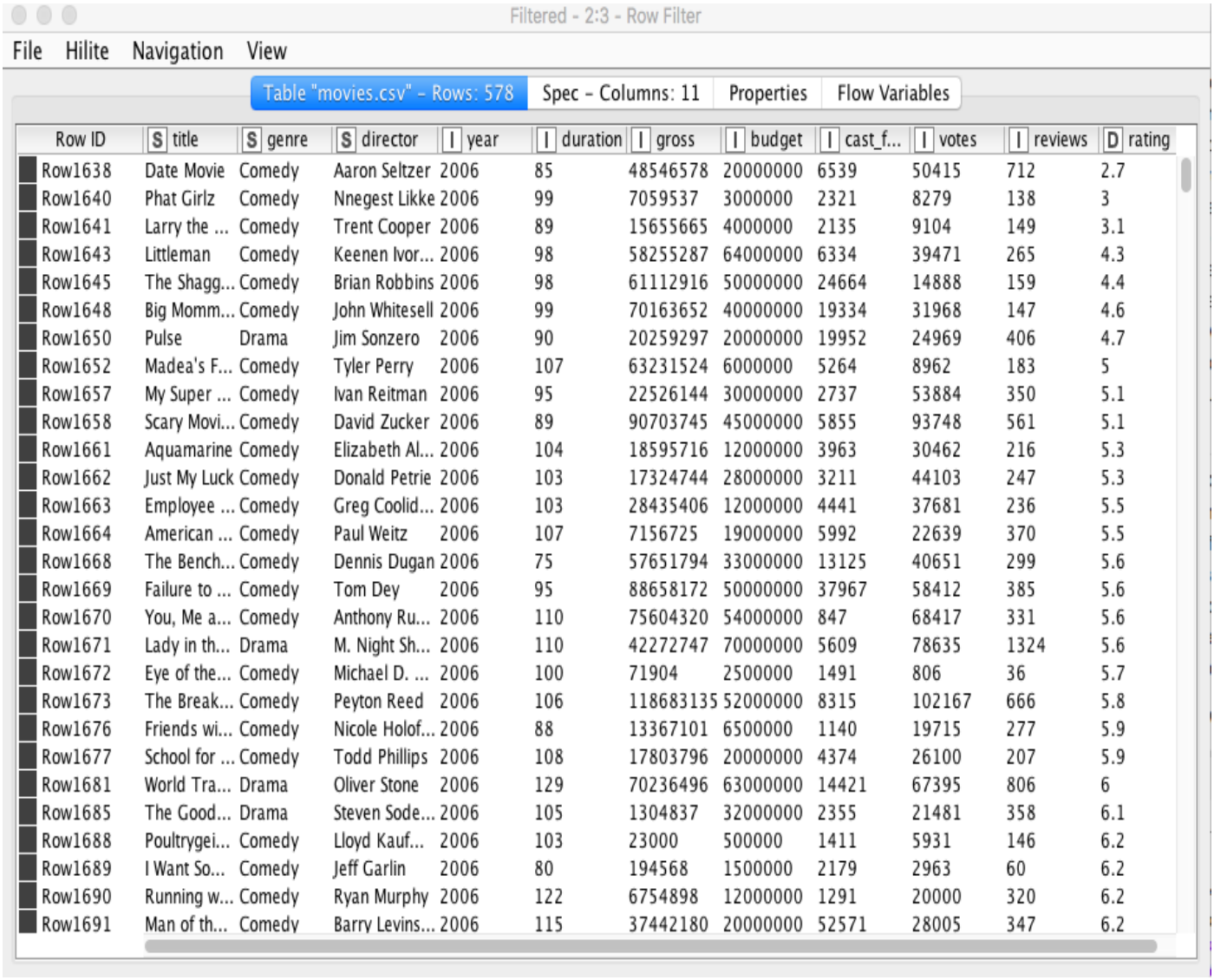
3 matching criteria on data colums: on String by full or partial pattern matching, on numbers by range, on missing values, all of them also on collection columns. 1 matching criterion on row numbers: from row number to row number. 1 matching criterion on RowID: full and partial patterm matching. Partial pattern matching is obtained through wild cards and RegEx. All matching criteria can be used in Include or Exclude mode. Include keeps the match results. Exclude excludes it.

Steps :

- Make a connection between rule based row filter and row filter.
- Configure row filter.
- Execute and Check out for the table after applying row filter.

➢ Use range checking :
Lower Bound : 2006
Upper Bound : 2016



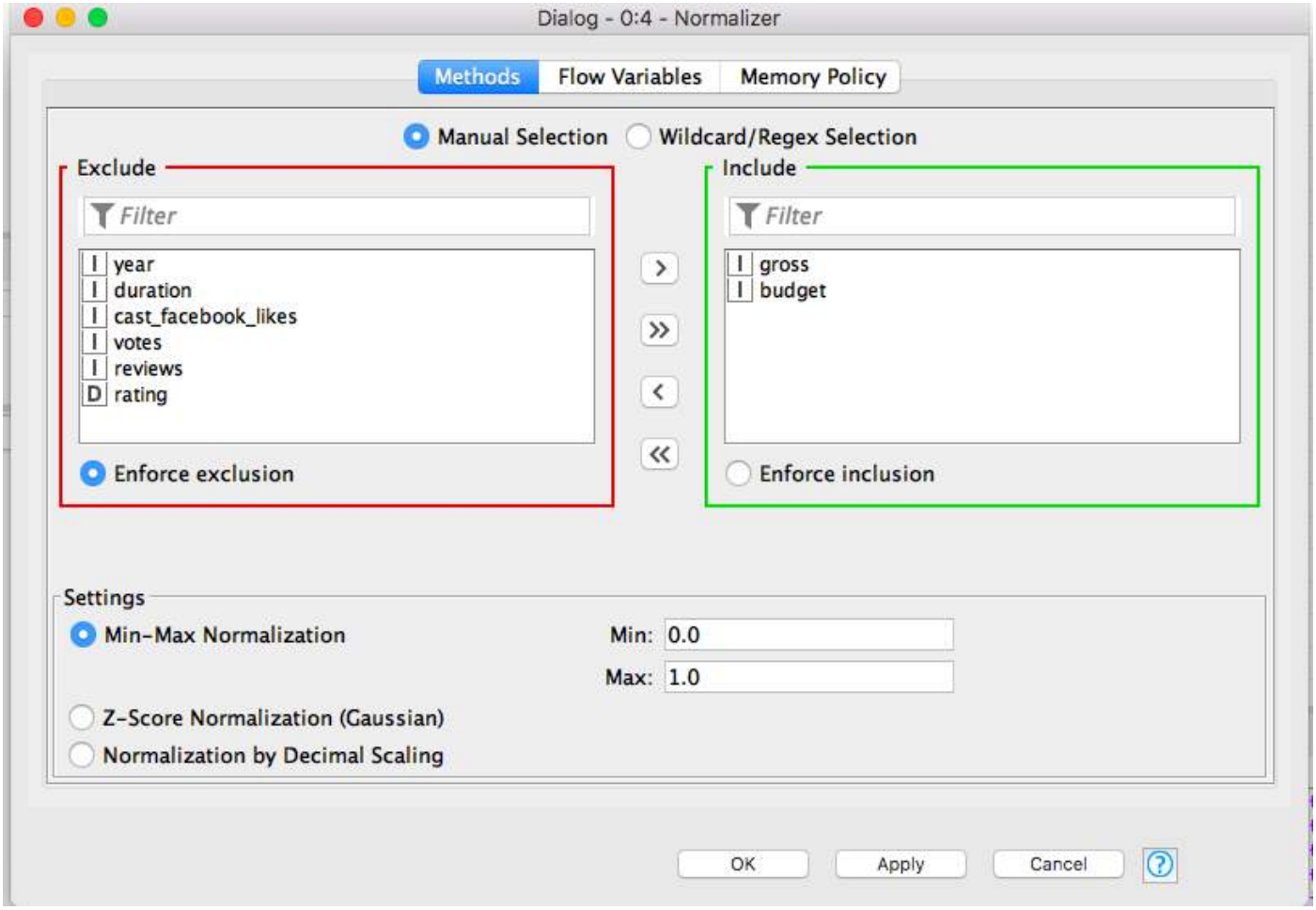


Row ID	title	genre	director	year	duration	gross	budget	cast_f...	votes	reviews	rating
Row1638	Date Movie	Comedy	Aaron Seltzer	2006	85	48546578	20000000	6539	50415	712	2.7
Row1640	Phat Girlz	Comedy	Nnegest Likke	2006	99	7059537	3000000	2321	8279	138	3
Row1641	Larry the ...	Comedy	Trent Cooper	2006	89	15655665	4000000	2135	9104	149	3.1
Row1643	Littleman	Comedy	Keenen Ivor...	2006	98	58255287	64000000	6334	39471	265	4.3
Row1645	The Shagg...	Comedy	Brian Robbins	2006	98	61112916	50000000	24664	14888	159	4.4
Row1648	Big Momm...	Comedy	John Whitesell	2006	99	70163652	40000000	19334	31968	147	4.6
Row1650	Pulse	Drama	Jim Sonzero	2006	90	20259297	20000000	19952	24969	406	4.7
Row1652	Madea's F...	Comedy	Tyler Perry	2006	107	63231524	6000000	5264	8962	183	5
Row1657	My Super ...	Comedy	Ivan Reitman	2006	95	22526144	30000000	2737	53884	350	5.1
Row1658	Scary Movi...	Comedy	David Zucker	2006	89	90703745	45000000	5855	93748	561	5.1
Row1661	Aquamarine	Comedy	Elizabeth Al...	2006	104	18595716	12000000	3963	30462	216	5.3
Row1662	Just My Luck	Comedy	Donald Petrie	2006	103	17324744	28000000	3211	44103	247	5.3
Row1663	Employee ...	Comedy	Greg Coolid...	2006	103	28435406	12000000	4441	37681	236	5.5
Row1664	American ...	Comedy	Paul Weitz	2006	107	7156725	19000000	5992	22639	370	5.5
Row1668	The Bench...	Comedy	Dennis Dugan	2006	75	57651794	33000000	13125	40651	299	5.6
Row1669	Failure to ...	Comedy	Tom Dey	2006	95	88658172	50000000	37967	58412	385	5.6
Row1670	You, Me a...	Comedy	Anthony Ru...	2006	110	75604320	54000000	847	68417	331	5.6
Row1671	Lady in th...	Drama	M. Night Sh...	2006	110	42272747	70000000	5609	78635	1324	5.6
Row1672	Eye of the...	Comedy	Michael D. ...	2006	100	71904	2500000	1491	806	36	5.7
Row1673	The Break...	Comedy	Peyton Reed	2006	106	118683135	52000000	8315	102167	666	5.8
Row1676	Friends wi...	Comedy	Nicole Holof...	2006	88	13367101	6500000	1140	19715	277	5.9
Row1677	School for ...	Comedy	Todd Phillips	2006	108	17803796	20000000	4374	26100	207	5.9
Row1681	World Tra...	Drama	Oliver Stone	2006	129	70236496	63000000	14421	67395	806	6
Row1685	The Good...	Drama	Steven Sode...	2006	105	1304837	32000000	2355	21481	358	6.1
Row1688	Poultrygei...	Comedy	Lloyd Kauf...	2006	103	23000	500000	1411	5931	146	6.2
Row1689	I Want So...	Comedy	Jeff Garlin	2006	80	194568	1500000	2179	2963	60	6.2
Row1690	Running w...	Comedy	Ryan Murphy	2006	122	6754898	12000000	1291	20000	320	6.2
Row1691	Man of th...	Comedy	Barry Levins...	2006	115	37442180	20000000	52571	28005	347	6.2

❖ **Normalizer :**

Steps :

- Connect normalizer with the row filter.
- Configure normalizer as methods which are to be included for normalization technique and set min and max values.
- Include:
  a) Gross
  b) Budget
  c) Min : 0.0
  d) Max : 0.1
- Execute the normalizer and check for the values in the table where you will find the normalized values of the table.



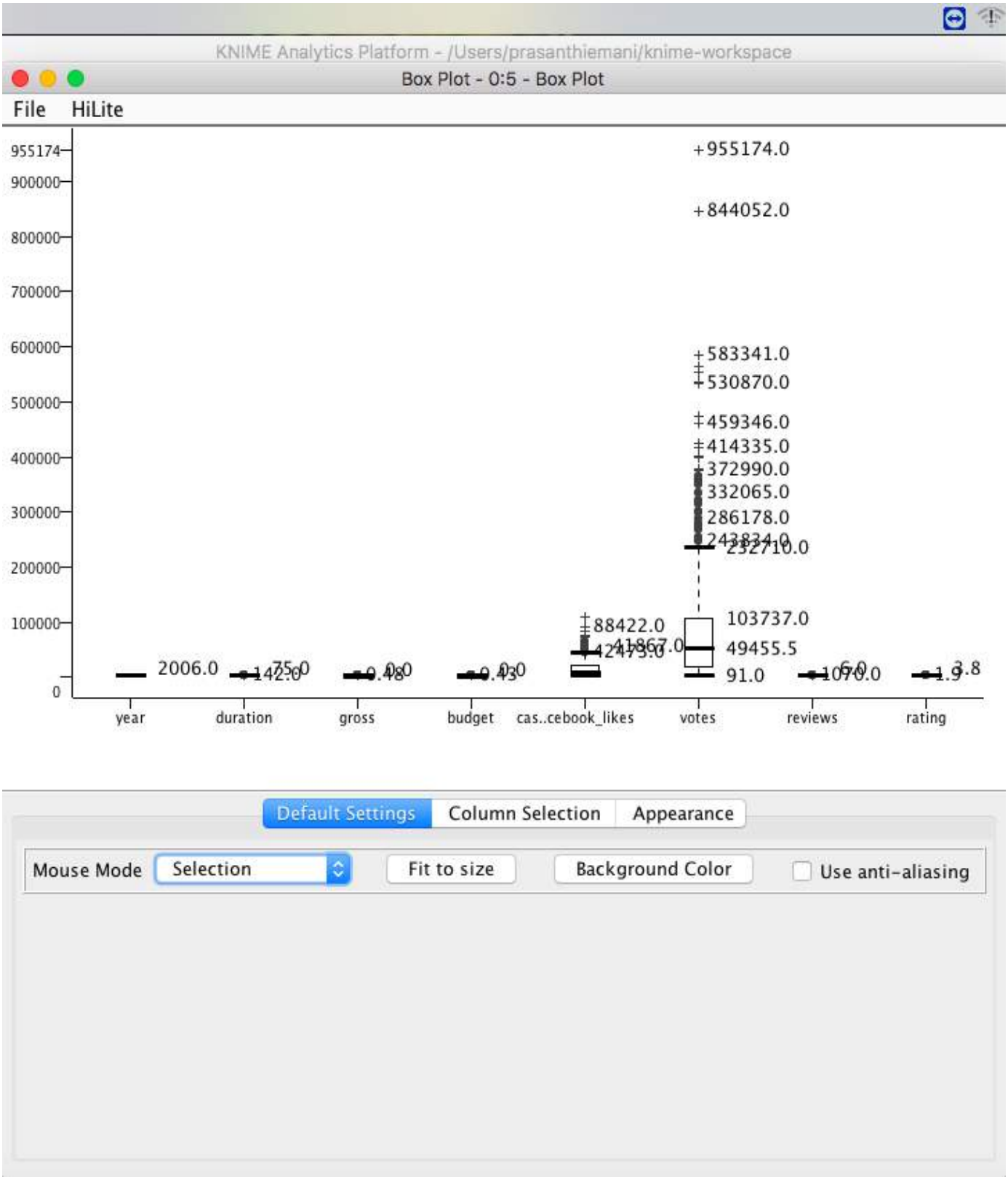
❖ **Boxplot :**

A box plot displays robust statistical parameters: minimum, lower quartile, median, upper quartile, and maximum. A box plot for one numerical attribute is constructed in the following way: The box itself goes from lower quartile (Q1) to upper quartile (Q3). Median is drawn as horizontal bar inside box. Distance between Q1 and Q3 is called interquartile range (IQR). Above and below box are so-called whiskers. They are drawn at minimum and maximum value as horizontal bars and are connected with the box by a dotted line.

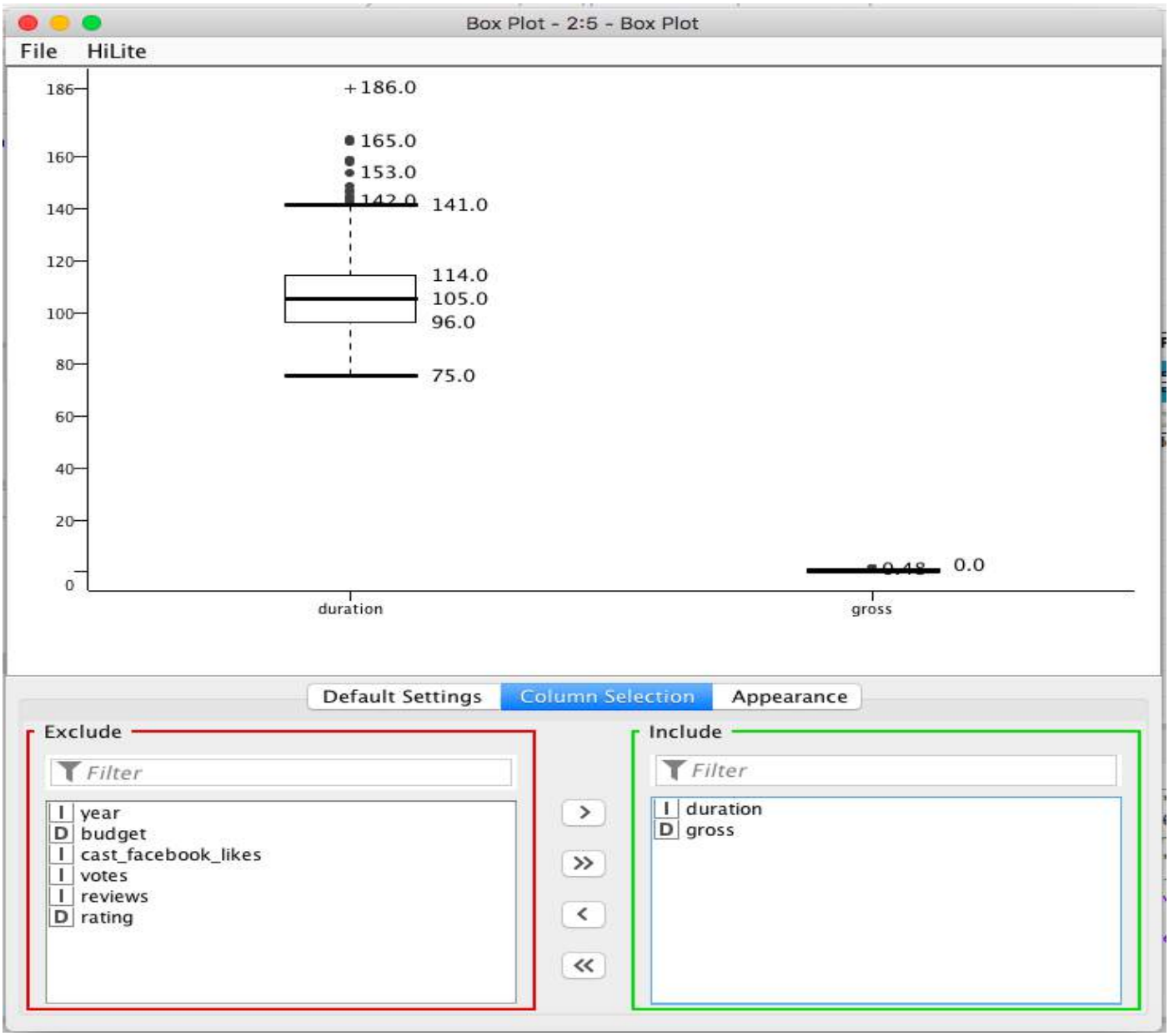
Steps :

- Make connection between normalizer and boxplot.
- View for the boxplot directly.
- We can select the specific columns for the individual boxplot through column selection.

- **Boxplot for the whole of dataset :**



➢ **Boxplot for the two of the attributes : gross and duration :**

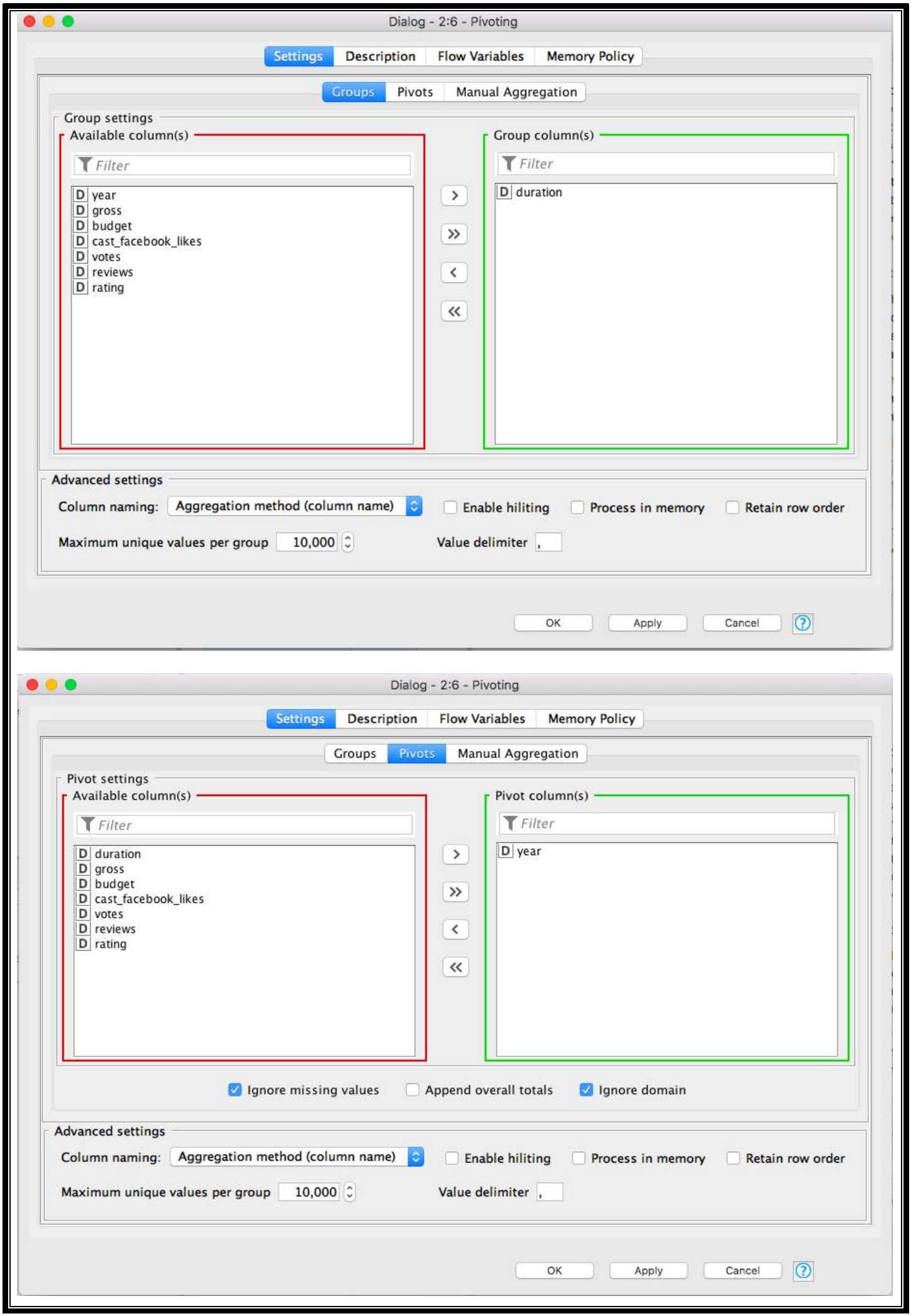


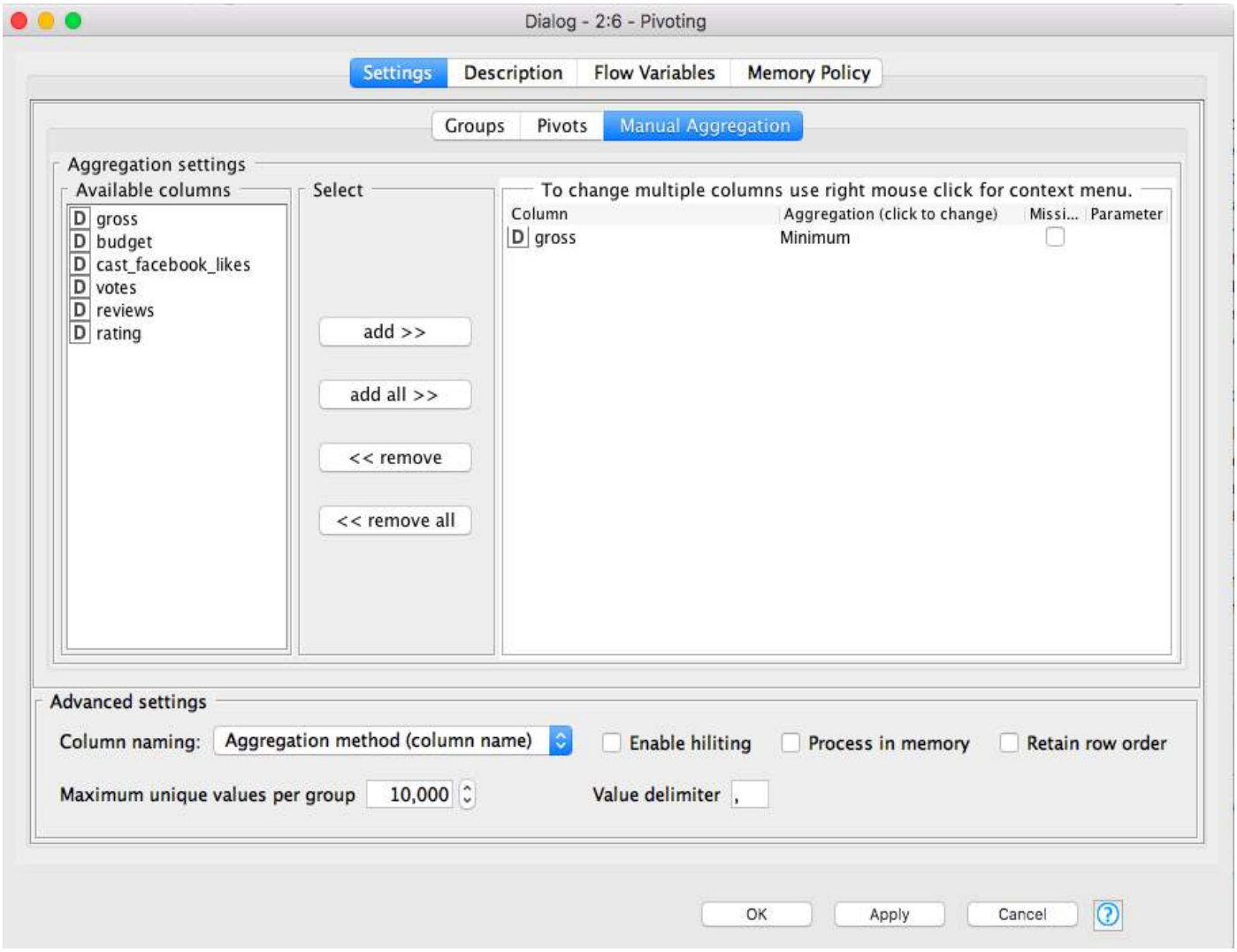
❖ **Pivoting :**

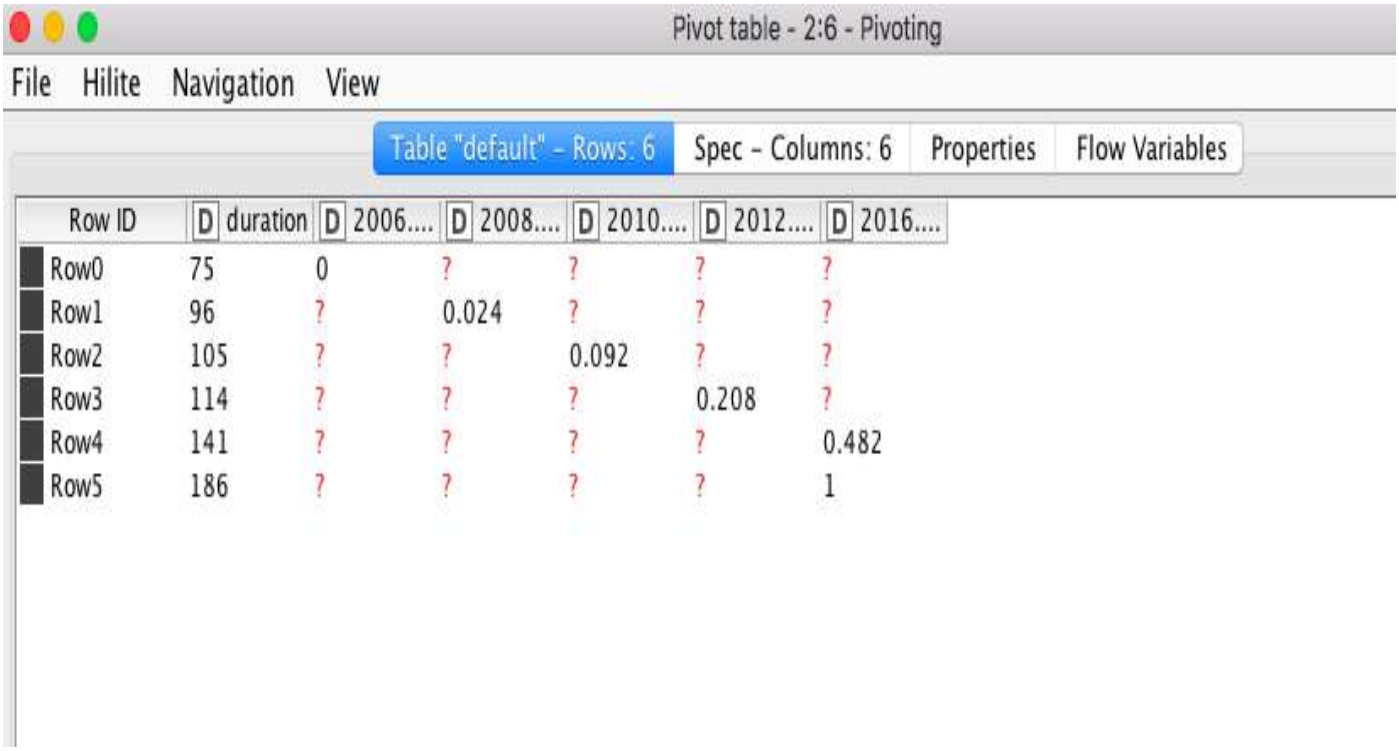
Performs a pivoting on the given input table using a selected number of columns for grouping and pivoting. The group columns will result into unique rows, whereby the pivot values turned into columns for each set of column combinations together with each aggregation method. In addition, the node returns the total aggregation (a) based on only the group columns and (b) based on only the pivoted columns resulting in a single row; optionally, with the total aggregation without pivoting.

Steps :

- Connect pivot with boxplot and have the connection between them.
- Configure pivoting with 3 different columns for same data type for :
  Groups – duration
  Pivots - year
  Manual Aggregation – gross
- Execute pivoting and checkout for the changes in the table.


Dialog - 2:6 - Pivoting
Settings
Description
Flow Variables
Memory Policy
Groups
Pivots
Manual Aggregation
Group settings
Available column(s)
Filter
year
gross
budget
cast_facebook_likes
votes
reviews
rating
Group column(s)
Filter
duration
Advanced settings
Column naming:
Aggregation method (column name)
Enable hiliting
Process in memory
Retain row order
Maximum unique values per group
10,000
Value delimiter
,
OK
Apply
Cancel
Dialog - 2:6 - Pivoting
Settings
Description
Flow Variables
Memory Policy
Groups
Pivots
Manual Aggregation
Pivot settings
Available column(s)
Filter
duration
gross
budget
cast_facebook_likes
votes
reviews
rating
Pivot column(s)
Filter
year
Ignore missing values
Append overall totals
Ignore domain
Advanced settings
Column naming:
Aggregation method (column name)
Enable hiliting
Process in memory
Retain row order
Maximum unique values per group
10,000
Value delimiter
,
OK
Apply
Cancel





Row ID	D duration	D 2006....	D 2008....	D 2010....	D 2012....	D 2016....
Row0	75	0	?	?	?	?
Row1	96	?	0.024	?	?	?
Row2	105	?	?	0.092	?	?
Row3	114	?	?	?	0.208	?
Row4	141	?	?	?	?	0.482
Row5	186	?	?	?	?	1

❖ **Joiner :**

A Joiner node joins two tables together on one or more common key values. Possible join modes: inner join, left outer join, right outer join, full outer join. Two tabs: "Joiner Settings" and "Column Selection". "Joiner Settings" defines the parameters for the join operation: join mode and column keys. "Column Selection" sets which columns to keep and/or drop and strategies to deal with duplicate columns.

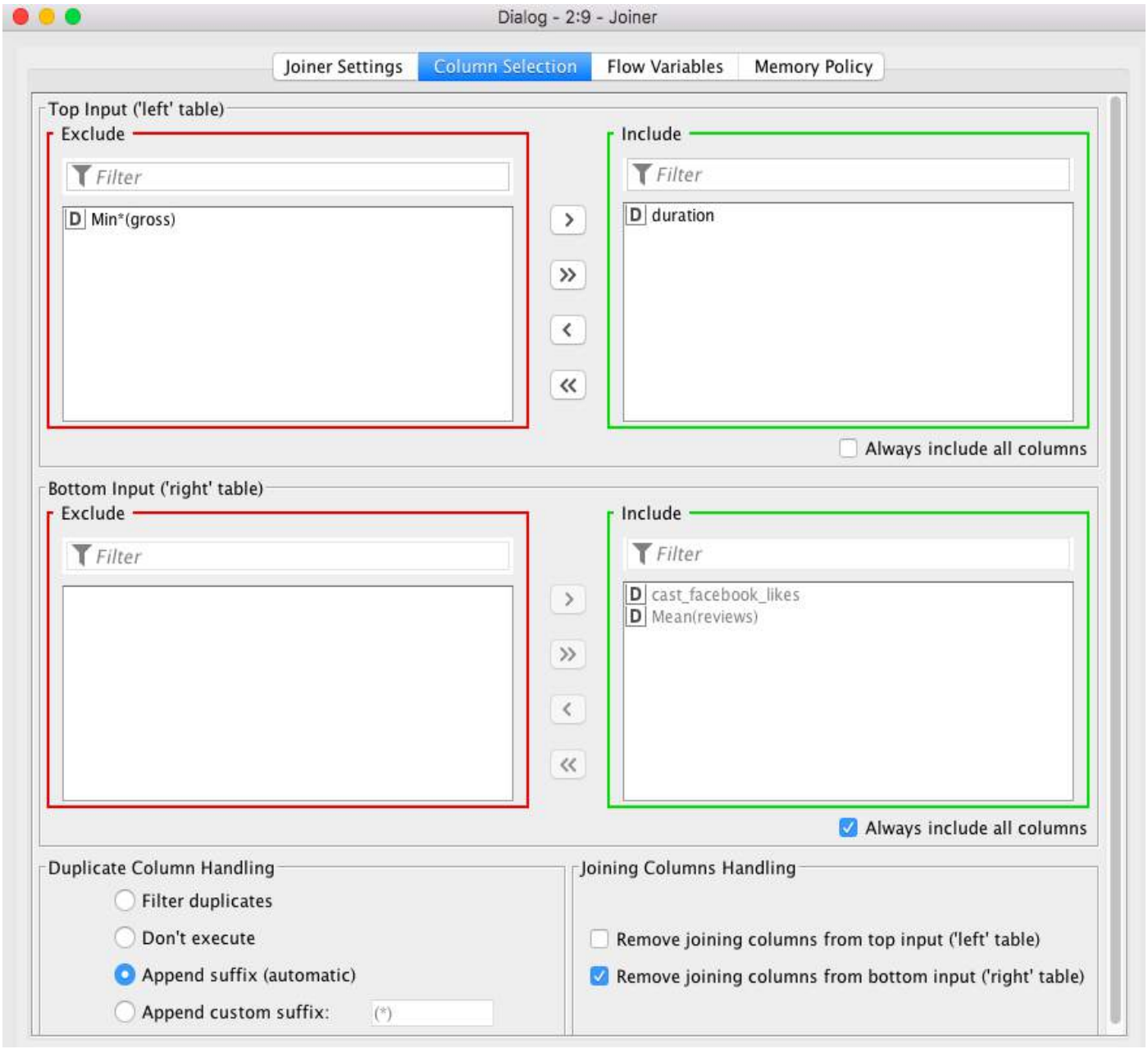
Steps :

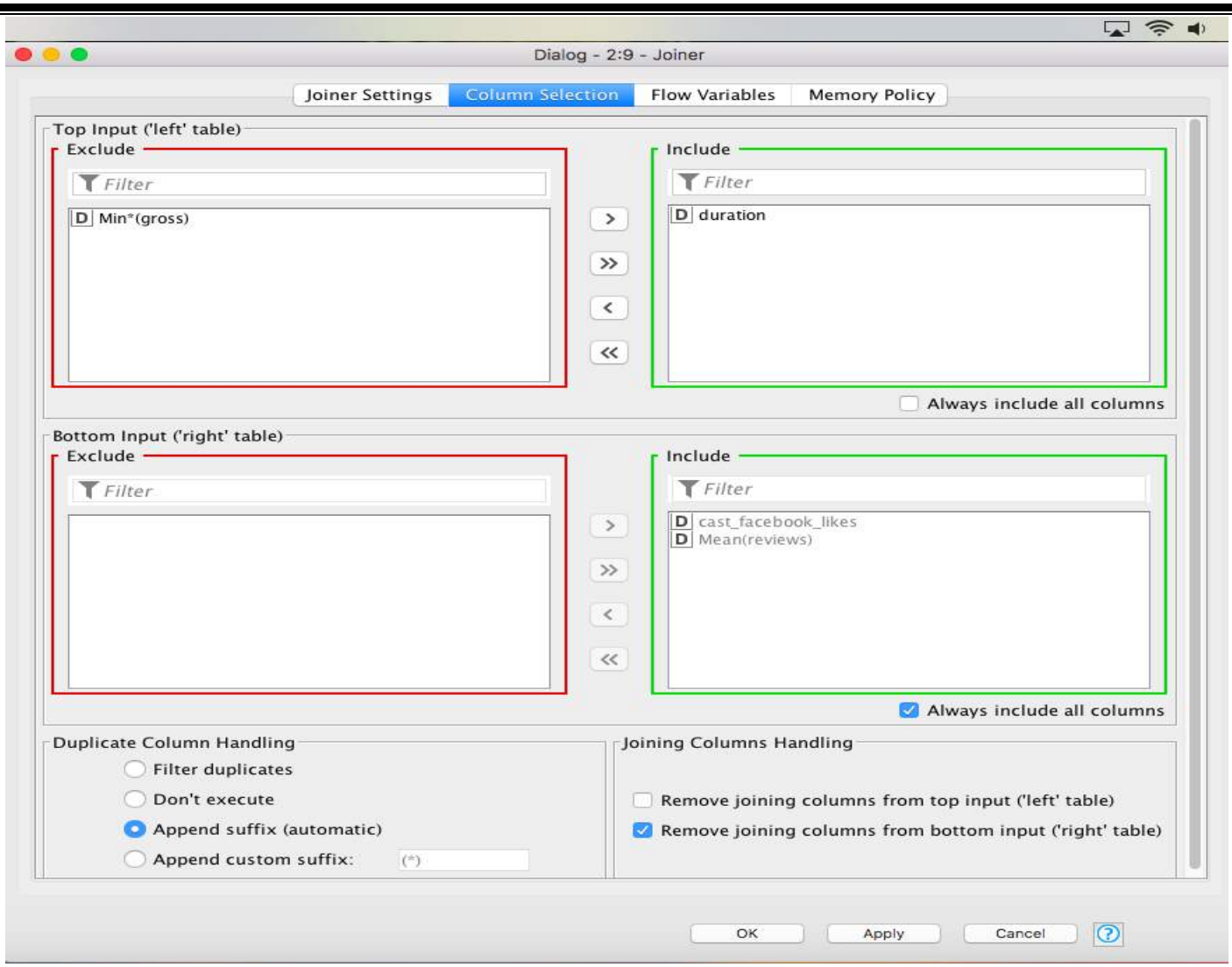
- Have the connection between joiner and pivot so that it is easy to analyse.
- Configure joiner with columns if necessary as :

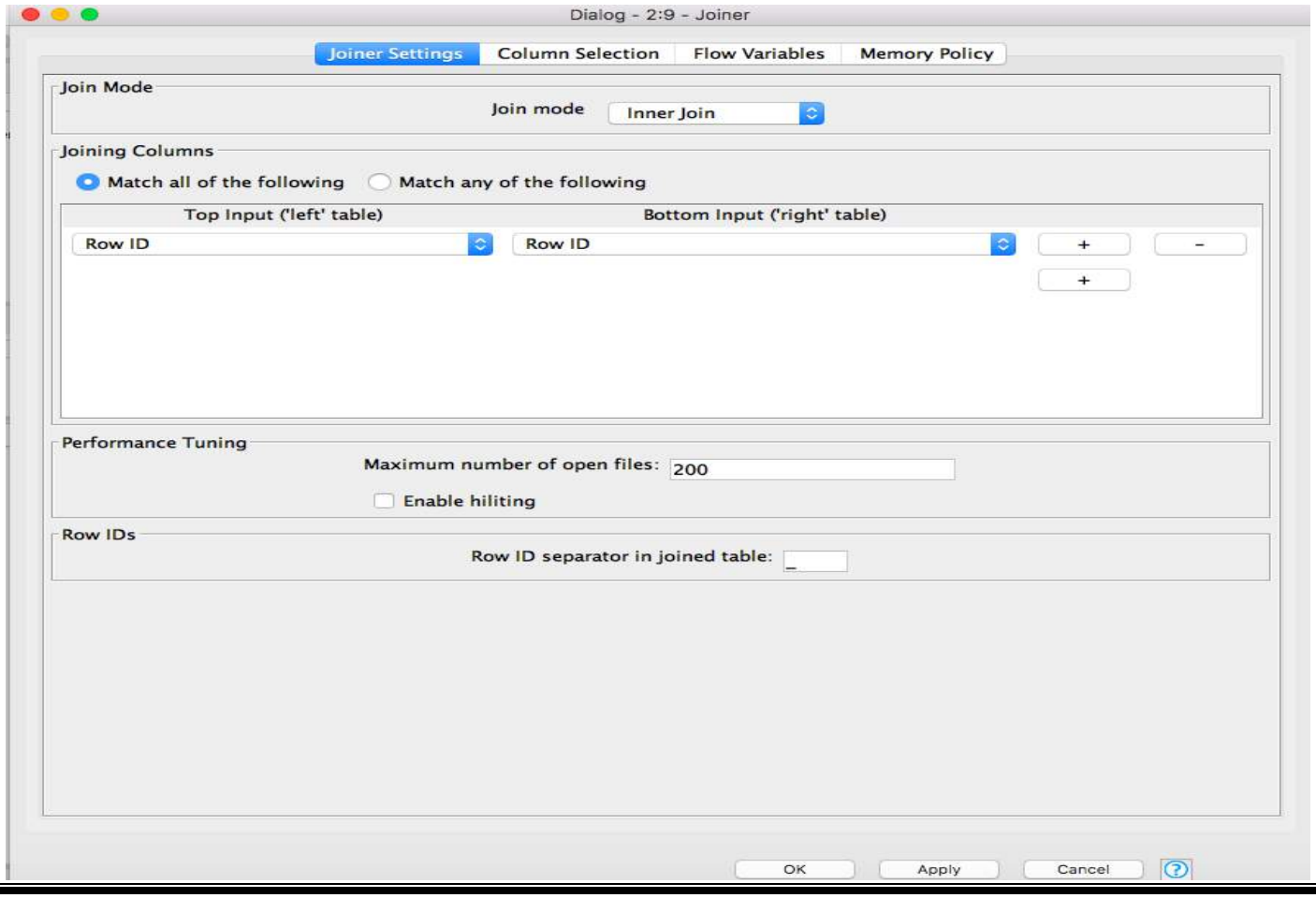
➢ Top Input (left table):
Include : duration

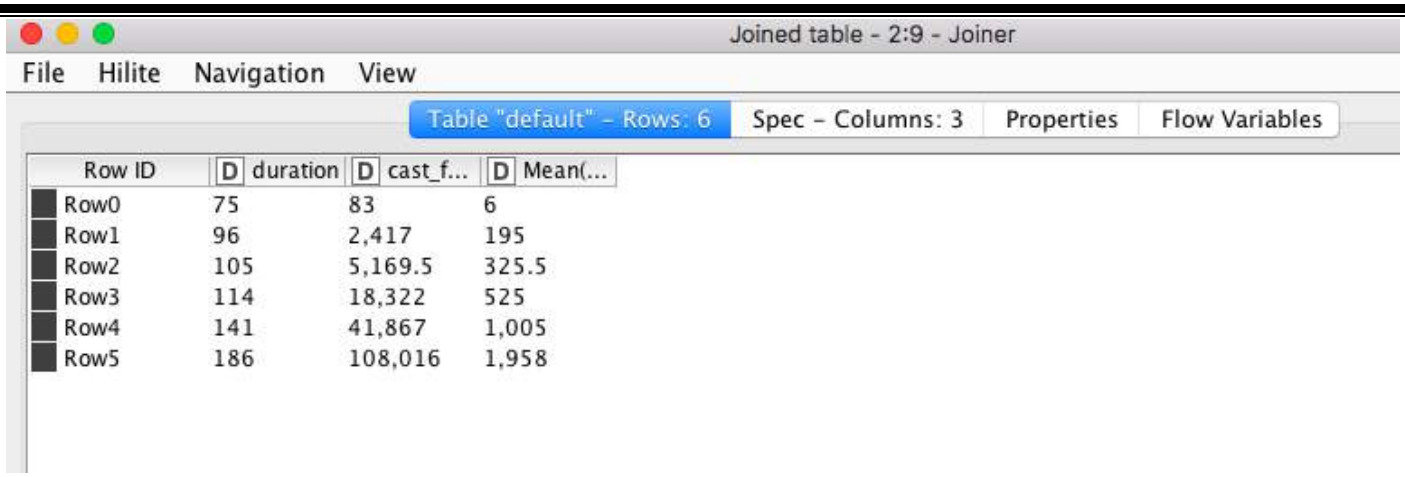
➢ Bottom Input(right table):
Include :
→cast_facebook_likes
→ Mean

- Execute the joiner and check for the final table.




Dialog - 2:9 - Joiner
Joiner Settings
Column Selection
Flow Variables
Memory Policy
Top Input ('left' table)
Exclude
Filter
D Min*(gross)
Include
Filter
D duration
Always include all columns
Bottom Input ('right' table)
Exclude
Filter
Include
Filter
D cast_facebook_likes
D Mean(reviews)
Always include all columns
Duplicate Column Handling
Filter duplicates
Don't execute
Append suffix (automatic)
Append custom suffix:
(*)
Joining Columns Handling
Remove joining columns from top input ('left' table)
Remove joining columns from bottom input ('right' table)
OK
Apply
Cancel


Dialog - 2:9 - Joiner
Joiner Settings
Column Selection
Flow Variables
Memory Policy
Join Mode
Join mode
Inner Join
Joining Columns
Match all of the following
Match any of the following
Top Input ('left' table)
Bottom Input ('right' table)
Row ID
Row ID
+
-
+
Performance Tuning
Maximum number of open files:
200
Enable hiliting
Row IDs
Row ID separator in joined table:
_
OK
Apply
Cancel



Row ID	duration	cast_f...	Mean(...
Row0	75	83	6
Row1	96	2,417	195
Row2	105	5,169.5	325.5
Row3	114	18,322	525
Row4	141	41,867	1,005
Row5	186	108,016	1,958

❖ **Group By :**

Groups the rows of a table by the unique values in the selected group columns. A row is created for each unique set of values of the selected group column. The remaining columns are aggregated based on the specified aggregation settings. The output table contains one row for each unique value combination of the selected group columns.

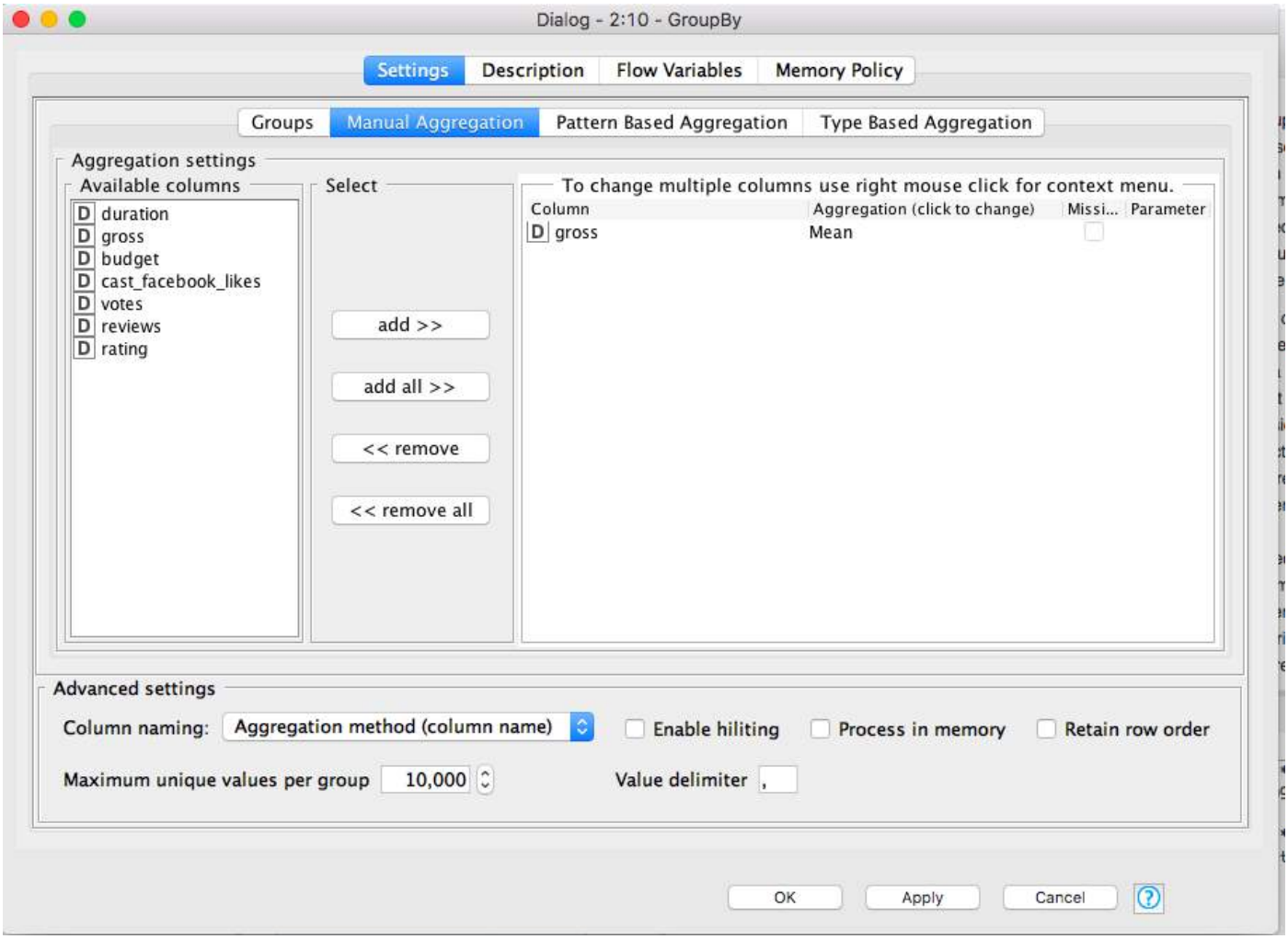
Steps :

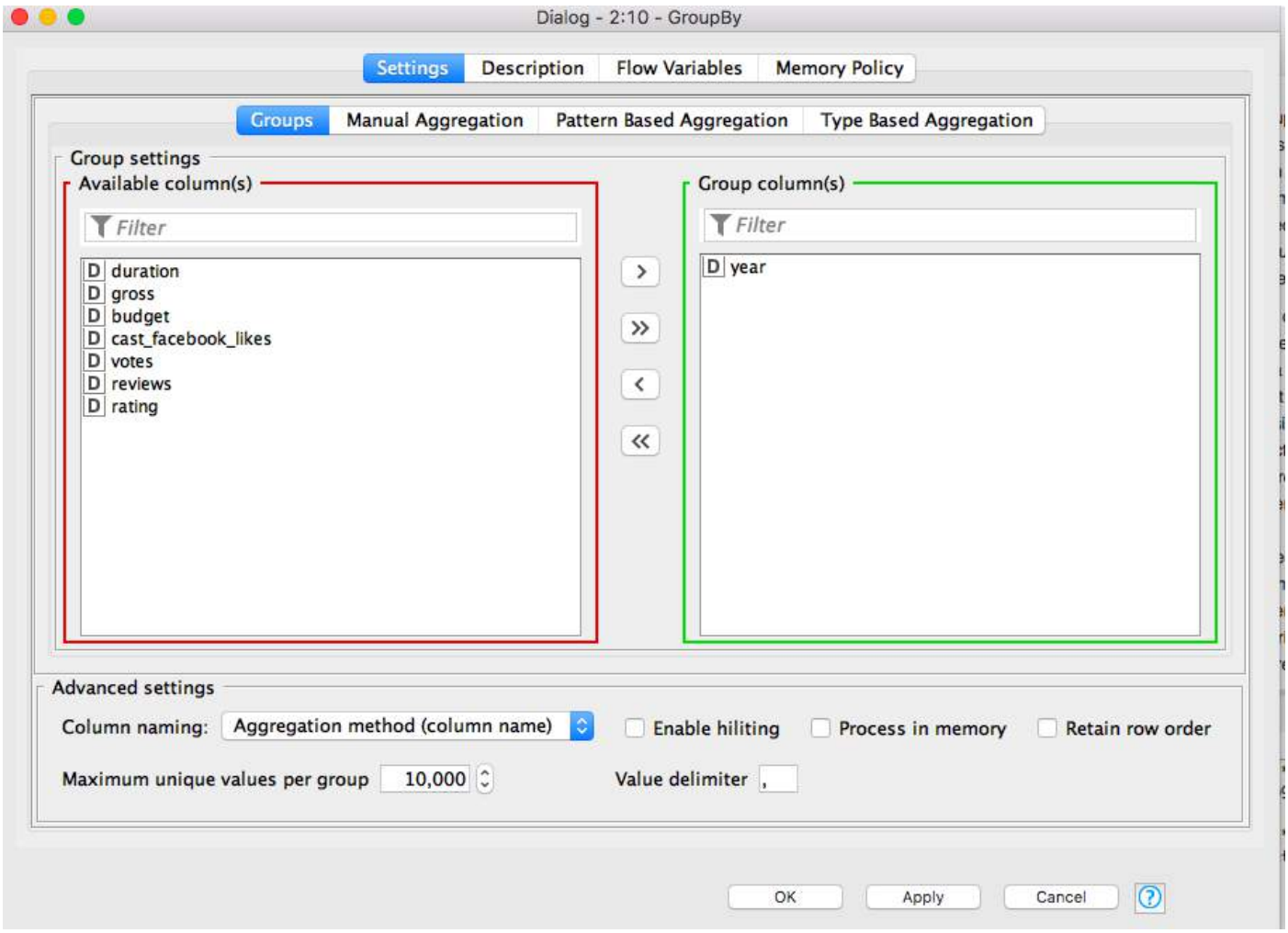
- Make the connection to group by with the boxplot directly.
- Configure group by as the following :

➢ Groups : year

➢ Manual Aggregation : gross

- Execute the group by and check for the analysed table.





Row ID	D year	D Mean(...
Row0	2,006	0
Row1	2,008	0.024
Row2	2,010	0.092
Row3	2,012	0.208
Row4	2,016	0.741

RESULT :

Thus, the operations that will be done on the table for the better access of the data will be done in this way where by applying normalization, pivoting and group by techniques.

❖ MULTIPLE REGRESSION :

Multiple regression is a statistical tool used to derive the value of a criterion from several other independent, or predictor, variables. It is the simultaneous combination of multiple factors to assess how and to what extent they affect a certain outcome.

Queries

```
a=diabetes$Age
b=diabetes$Glucose
c=diabetes$Bloodpressure
model<- lm(a~b+c)
print(model)
```