INDUS INSTITUTE OF TECHNOLOGY & ENGINEERING

DATA WAREHOUSE AND MINING

LAB PRACTICALS

Name: Ahmed Shaikh
Enrollment Number: IU19410050002
CLASS: CE-A
Semester : 7th

PRACTICAL - 1

AIM: Study Practical: Introduction to Weka

WEKA - an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems.

Explorer :

On the top, there are several tabs as listed here −

1) ***Preprocess:*** This allows us to choose the data file.

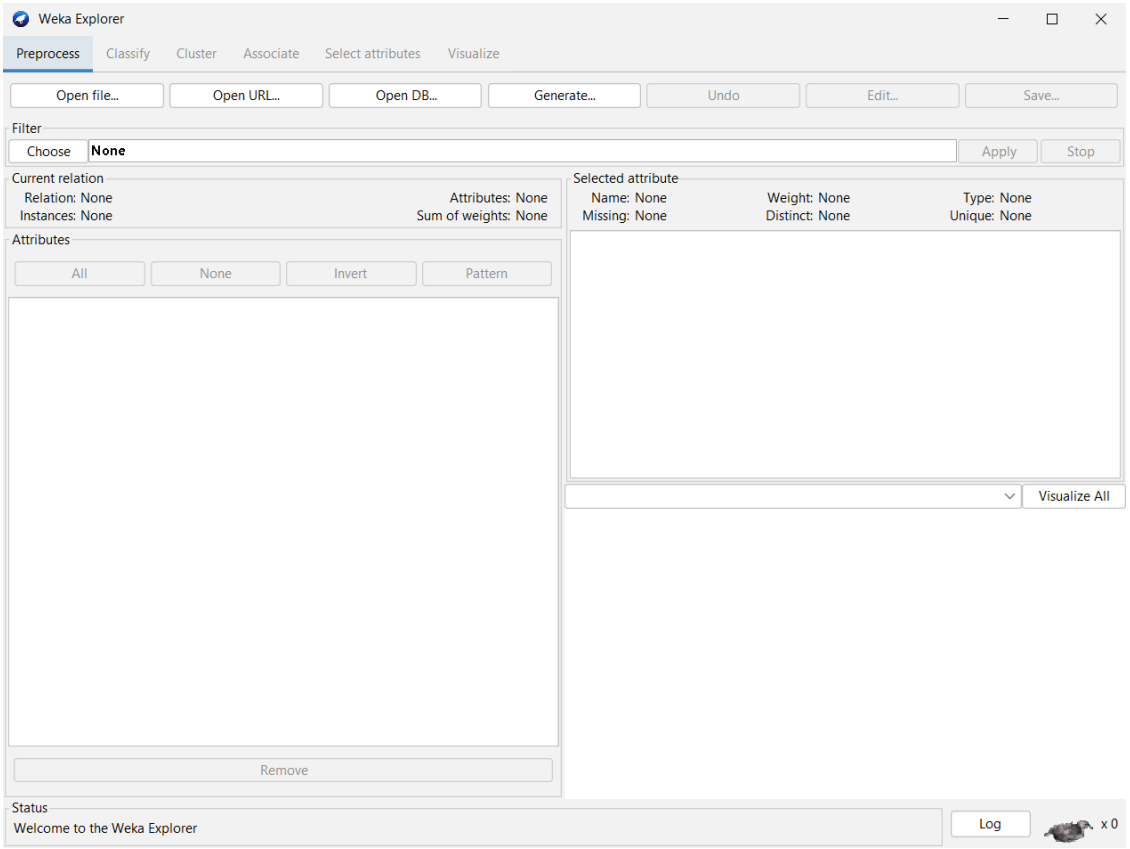
2) ***Classify:*** This allows us to apply and experiment with different algorithms on preprocesseddata files.

3) ***Cluster:*** This allows us to apply different clustering tools, which identify clusters within thedata file.

4) ***Association:*** This allows us to apply association rules, which identify the association within thedata.
5) ***Select attributes:*** These allow us to see the changes on the inclusion and exclusion ofattributes from the experiment.

6) ***Visualize:*** This allows us to see the possible visualisation produced on the data set in a 2Dformat, in scatter plot and bar graph output.

The user cannot move between the different tabs until the initial preprocessing of the data set hasbeen completed.

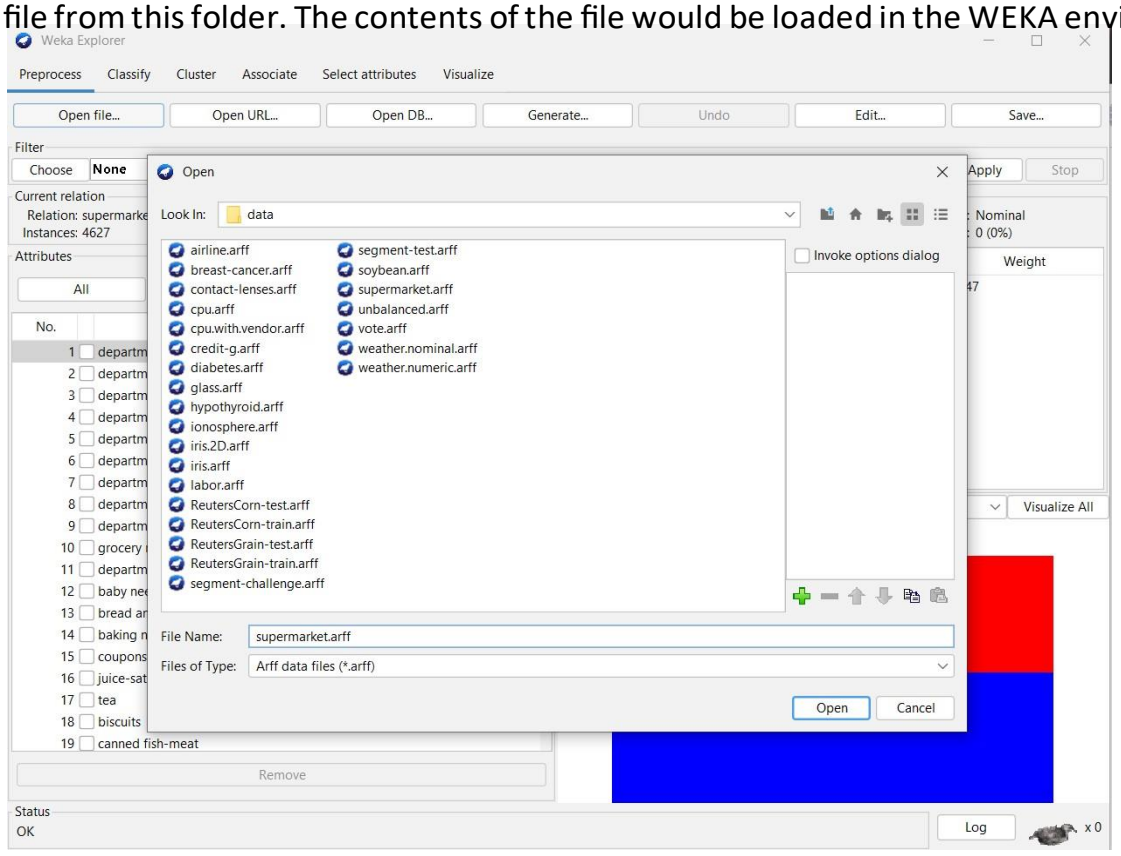


The data can be loaded from the following sources −

- Local file system
- Web
- Database

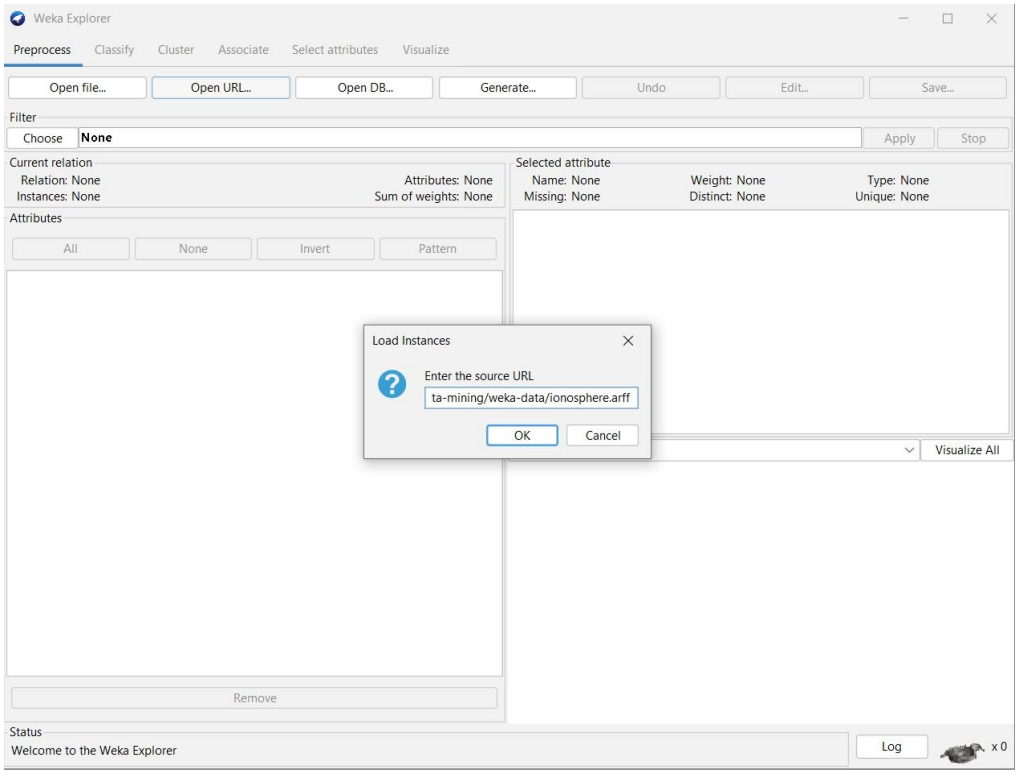
Loading Data from Local File System

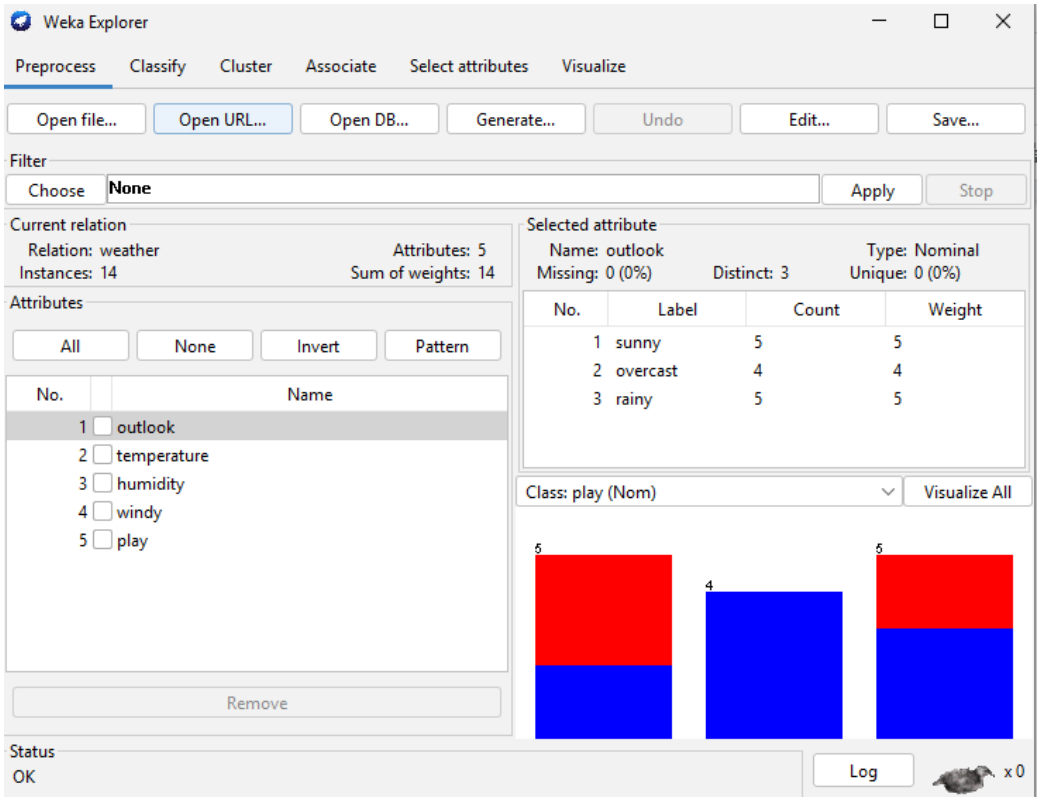
Click on the Open file button.Now, navigate to the folder where your data files are stored.select any data file from this folder. The contents of the file would be loaded in the WEKA environment.

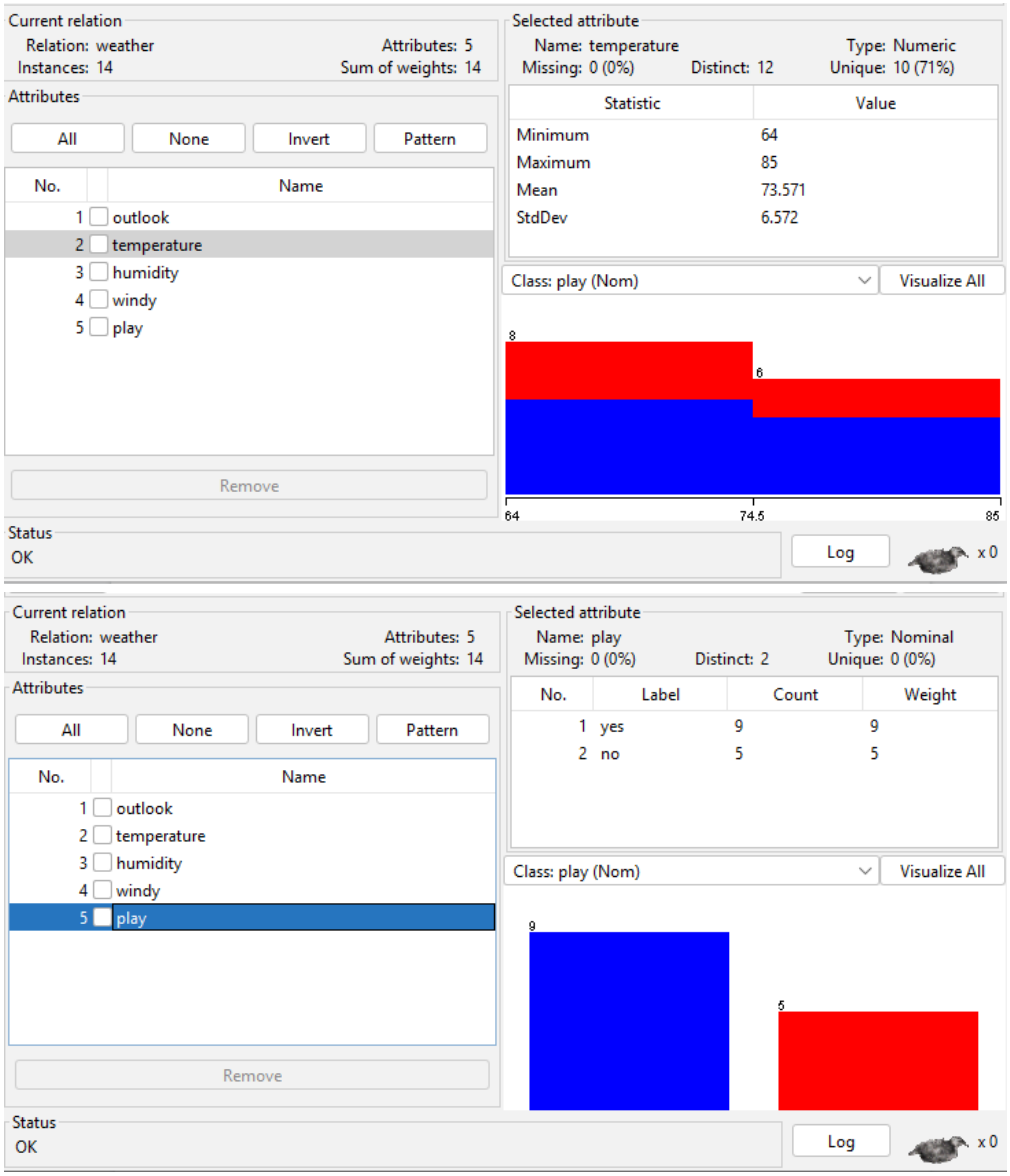


Loading Data from Web

Once you click on the Open URL button, you can see a window as follows −

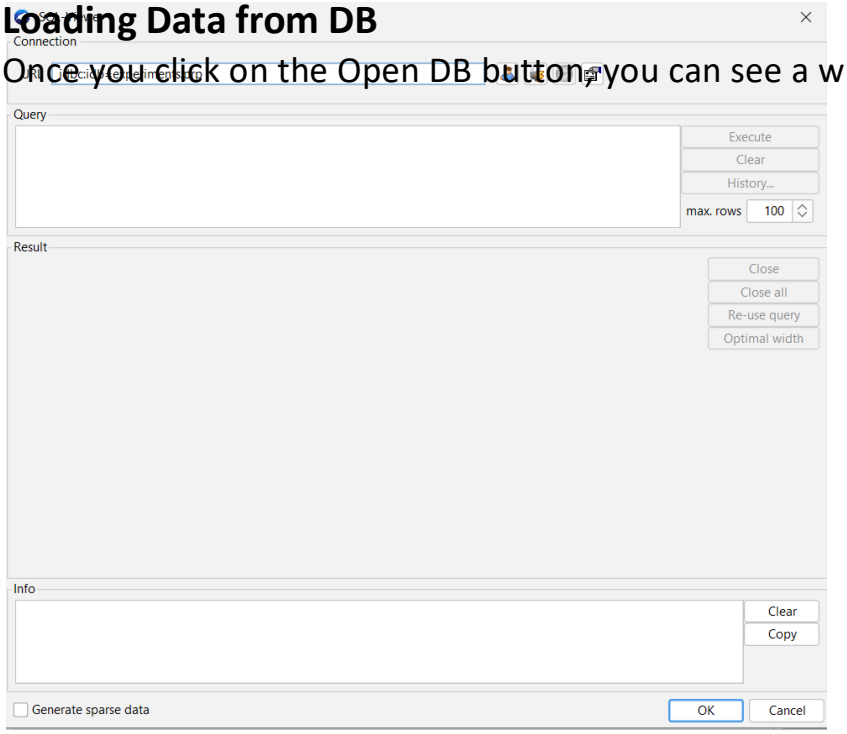






Loading Data from DB

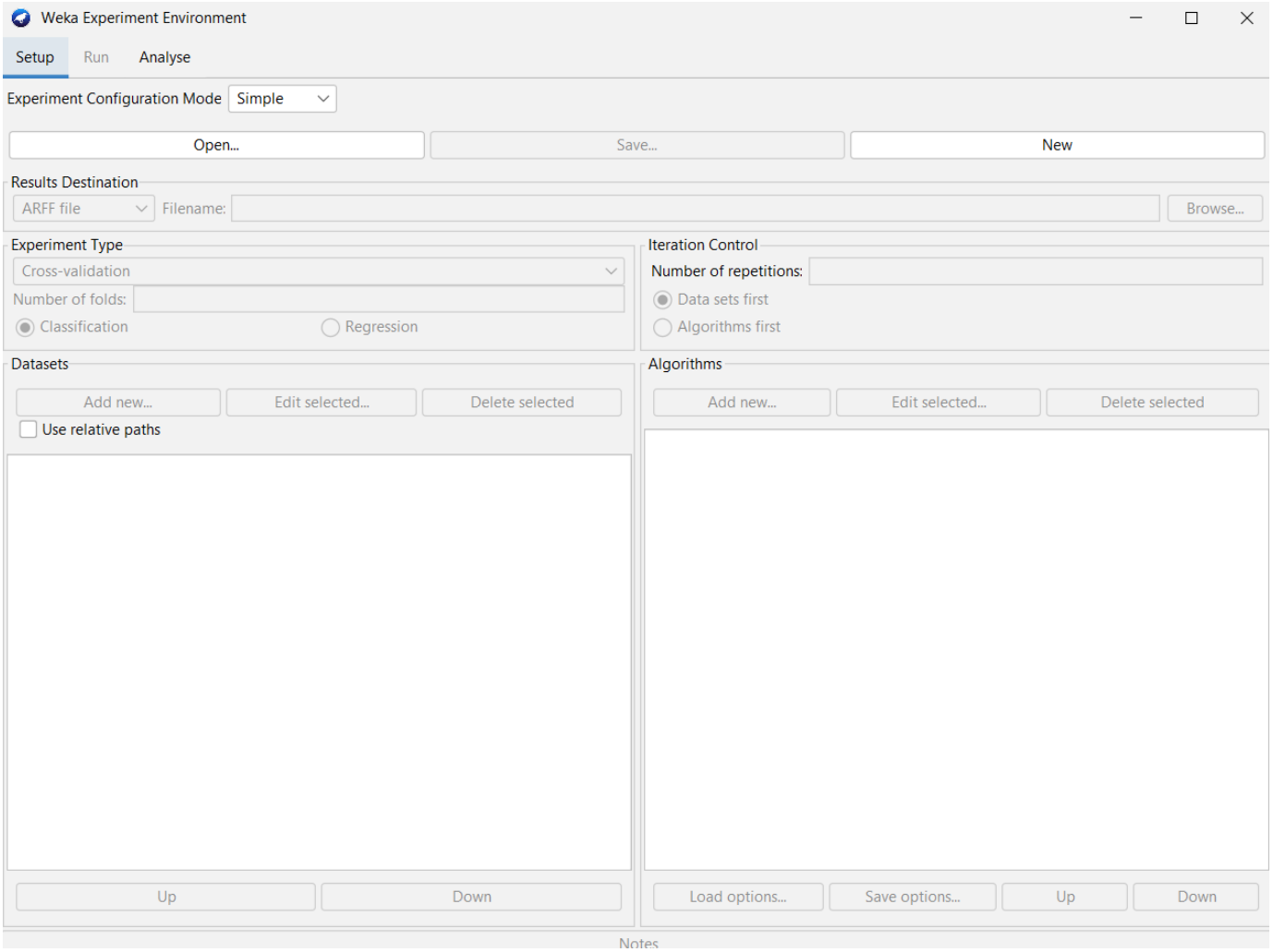
Once you click on the Open DB button, you can see a window as follows −



Set the connection string to your database, set up the query for data selection, process thequery and load the selected records in WEKA.

Experimenter :

The Experimenter option available in Weka enables the user to perform some experiments on thedata set by choosing different algorithms and analysing the output. It has the following components.



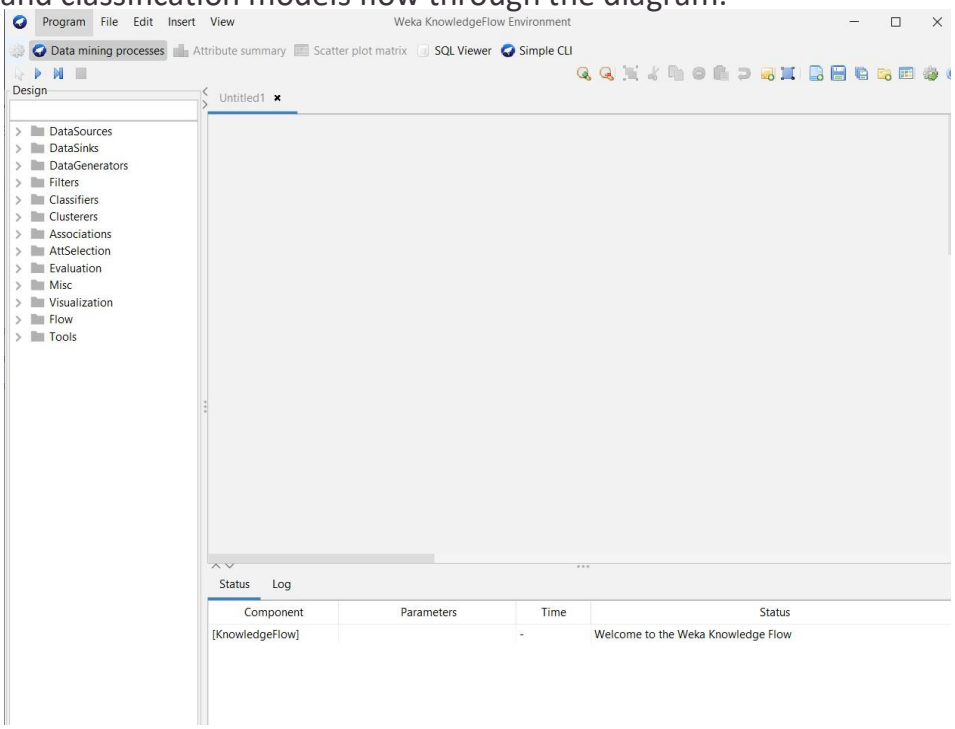
Setup: The first one is to set up the data sets, algorithms output destination, etc. Figure 4 shows an example of comparing the J4.8 decision tree with ZeroR on the IRIS data set. We can add moredata sets and compare the outcome using more algorithms, if required.

Run: You can use this tab to run the experiment.

Analyse: This tab can be used to analyse the result.

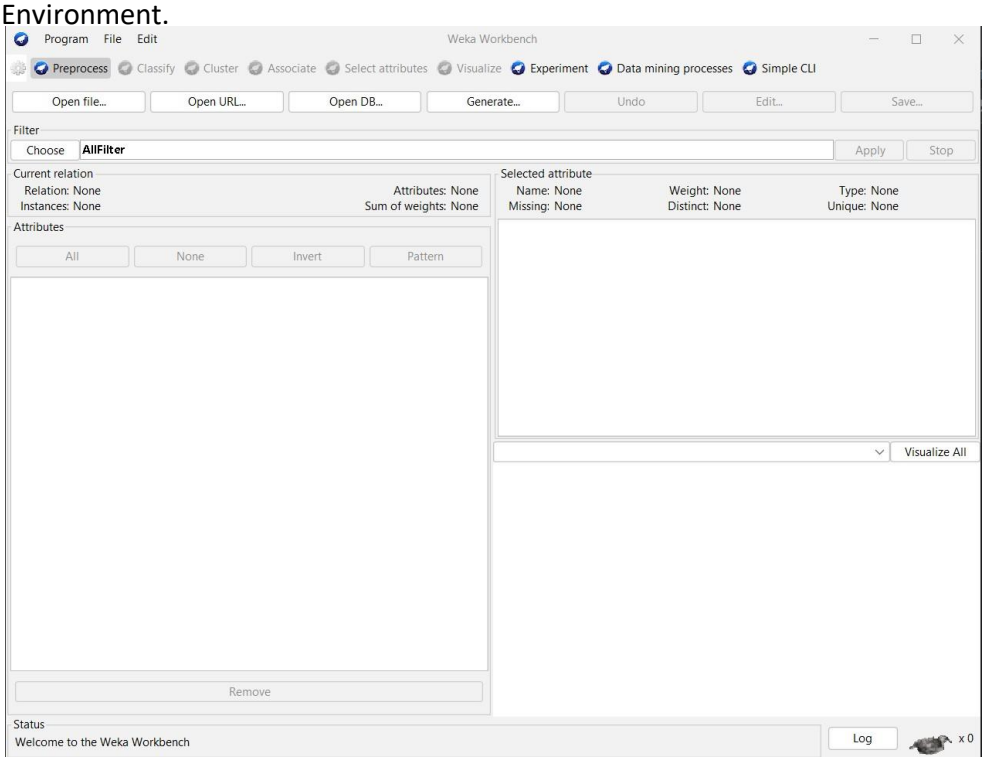
Knowledge Flow:

The Knowledge Flow interface is an alternative to the Explorer. You lay out filters, classifiers, evaluators, and visualizers interactively on a 2D canvas and connect them together with different kinds of connector. Data and classification models flow through the diagram.



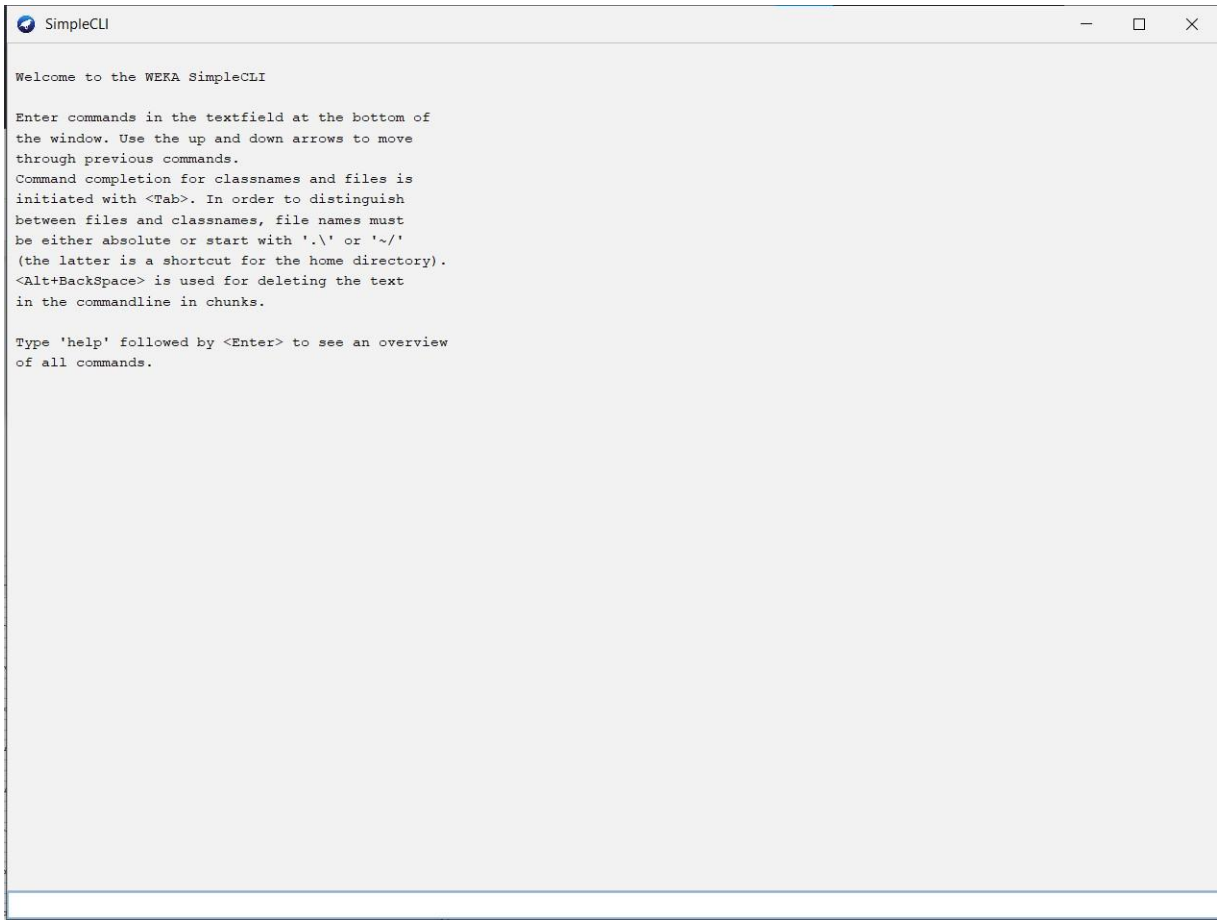
Workbench :

The Weka Workbench is an environment that combines all of the GUI interfaces into a single interface.It is useful if you find yourself jumping a lot between two or more different interfaces, such as between the Explorer and the Experiment Environment.



Simple CLI:

Weka can be used from a simple Command Line Interface (CLI).This is powerful because you can write shellscripts to use the full API from command line calls with parameters, allowing you to build models, run experiments and make predictions without a graphical user interface.The SimpleCLI provides an environment where you can quickly and easily experiment with the Weka command line interface commands.



PRACTICAL - 2

AIM: Study Practical: Introduction to RStudio

1. Statistical Operations

CODE:

```
x <- c(2,7,3,4.2,2,2,54,-21,8,-5,2)

#Mean
mean(x)

#Median
median(x)

#Minimum
min(x)

#Maximum
max(x)

#Standard Deviation
sd(x)

#Length
length(x)

#Quantile
quantile(x)

#Range
range(x)

#Interquantile range
IQR(x)

#Variance
var(x)

#Coefficient of Variance
cv <- sd(x) / mean(x) * 100
```

OUTPUT

:

```
[1] 5.290909
[1] 2
[1] -21
[1] 54
[1] 17.97139
[1] 11
   0%   25%   50%   75%  100%
-21.0   2.0   2.0   5.6  54.0
[1] -21  54
[1] 3.6
[1] 322.9709
```

2. Charts

a. Density Plot

CODE:

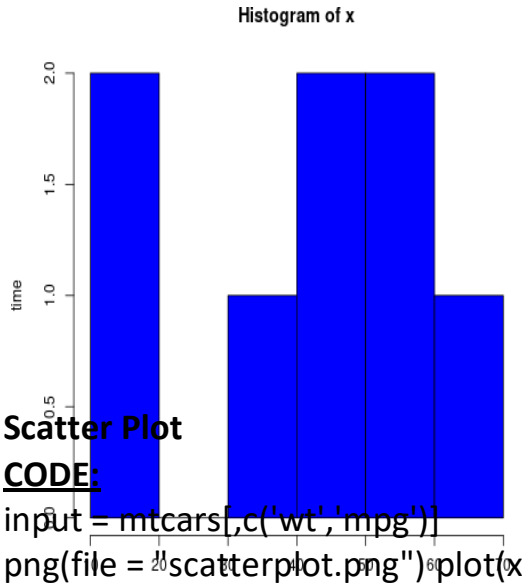
OUTPUT:

b. Histogram

CODE:

```
x=c(10,12,54,67,45,34,56,43)
png(file = "histogram.png")
hist(x,xlab="freq",ylab="time",col="blue",border="black",breaks=5)dev.off()
```

OUTPUT:

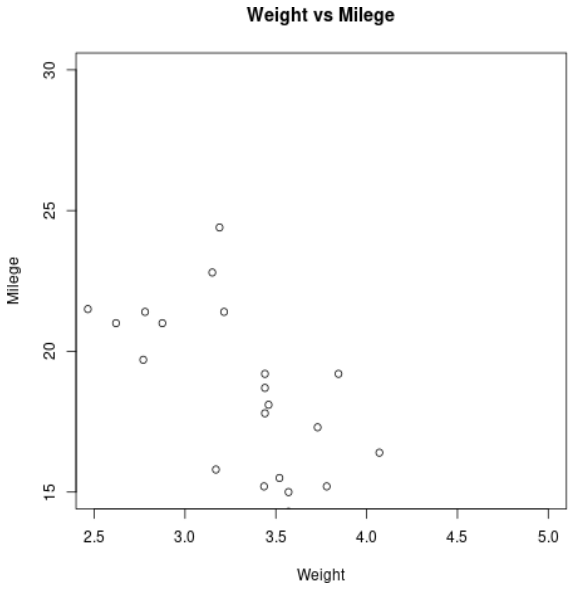


c. Scatter Plot

CODE:

```
input = mtcars[,c('wt','mpg')]
png(file = "scatterplot.png")plot(x
= input$wt,y = input$mpg,
   xlab = "Weight",
   ylab = "Milege",
   xlim = c(2.5,5),
   ylim = c(15,30),
   main = "Weight vs Milege" )
```

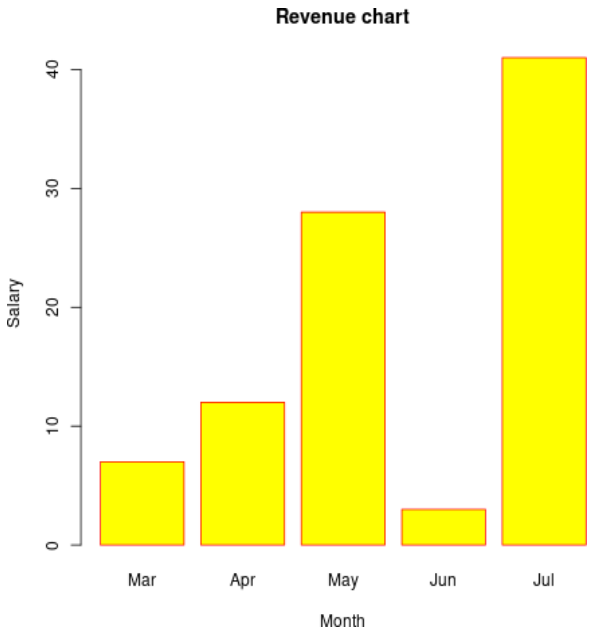
OUTPUT:



d. Bar Chart

CODE:

```
H = c(7,12,28,3,41)
M = c("Mar","Apr","May","Jun","Jul") png(file
= "barchart_months_revenue.png")
barplot(H,names.arg=M,xlab="Month",ylab="Salary",col="yellow",main="Revenue
chart",border="red")
```

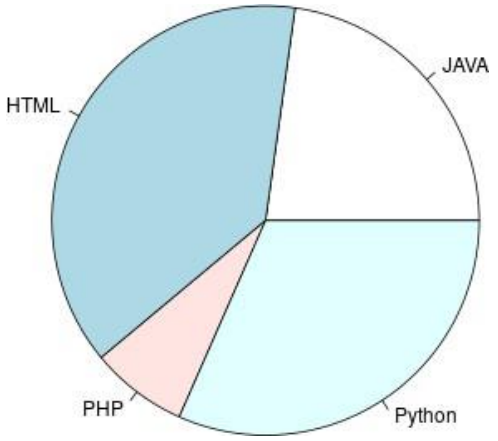


e. Pie Chart

CODE:

```
x = c(31, 52, 10, 43)
labels = c("JAVA", "HTML", "PHP", "Python")png(file =
"language.png")
pie(x,labels)
```

OUTPUT:

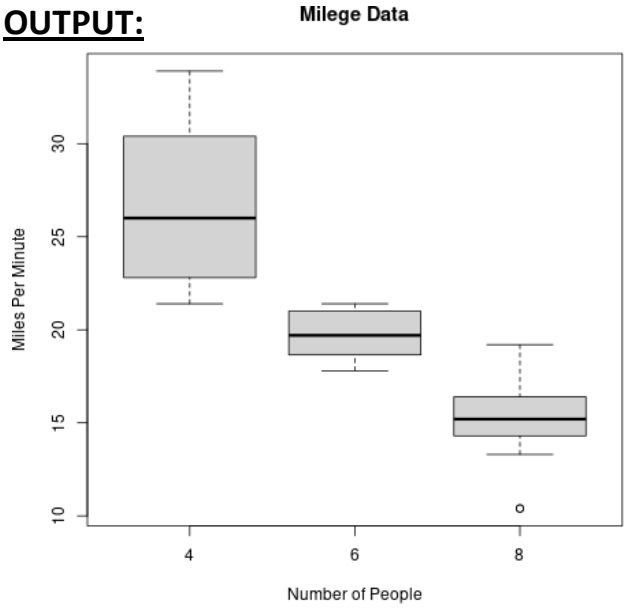


f. Box Plot

CODE:

```
png(file = "boxplot.png")
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of People",ylab =
  "Miles Per Minute", main = "Milege Data")
```

OUTPUT:



PRACTICAL - 3

AIM: To perform Linear Regression in RStudio

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

y=ax+b
y response variablex is
predictor variable
a & b are constants called as coefficients

lm() function lm(formula=y~x, data)
predict() function predict(object,newdata)

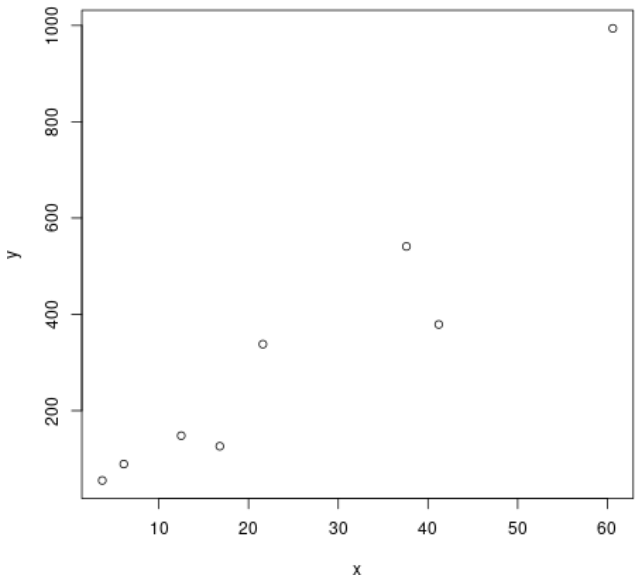
Problem 1: Develop the equation of simple regression line to predict sales(y) from advertising(x) using the given data

CODE:

```
x = c(12.5,3.7,21.6,60.6,37.6,6.1,16.8,41.2)y =
c(148,55,338,994,541,89,126,379)
y.lm = lm(y~x)
coeffs = coefficients(y.lm)
coeffs
newdata = data.frame(x=50)
predict(y.lm,newdata)
print(summary(y.lm))
summary(y.lm)$r.squared
#plotting
png(file = "scatterplot.png")
plot(x,y)
```

OUTPUT:

```
(Intercept)           x
   -45.149      15.14840
712.2706

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-199.97  -17.99   29.16   47.06  121.16

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -45.149     63.654  -0.709 0.504761
x             15.148      2.047   7.402 0.000312 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 107 on 6 degrees of freedom
Multiple R-squared:  0.9013,    Adjusted R-squared:  0.8848
F-statistic: 54.79 on 1 and 6 DF,  p-value: 0.0003123

[1] 0.9012992
```



Problem 2: Use a computer to develop the equation of the regression model for the following data. Comment on the regression coefficients. determine the predicted value of y for x1=33, x2=29,x3=13

CODE:

```
x1 = c(21,43,56,19,29,34,40,32,16,18,27,31)
x2 = c(6,25,42,27,20,45,33,14,4,31,12,3)
x3 = c(5,8,25,9,12,21,14,11,7,16,10,8)
y = c(114,94,87,98,101,85,94,107,119,93,108,117)
y.lm = lm(y~x1+x2+x3)
coeffs = coefficients(y.lm)
coeffs
newdata = data.frame(x1=33, x2=29,x3=13)
predict(y.lm,newdata)
print(summary(y.lm))
summary(y.lm)$r.squared
```

OUTPUT:

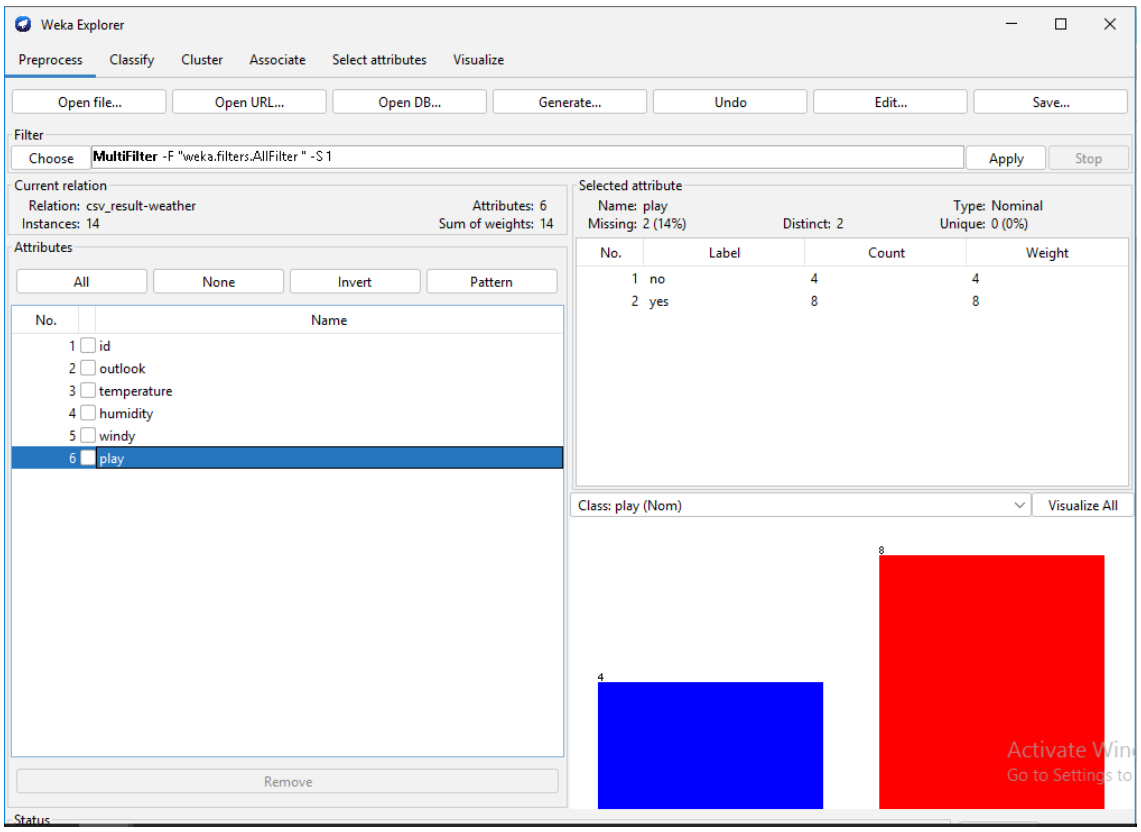
```
 (Intercept)           x1           x2           x3
118.55951024  -0.07940245  -0.88428115   0.37690982
       1
95.1949

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7481 -1.6934  0.5343  1.1214  2.6097

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 118.55951    1.85798  63.811 4.05e-12 ***
x1           -0.07940    0.06848  -1.159    0.280
x2           -0.88428    0.08631 -10.245 7.08e-06 ***
x3            0.37691    0.21973   1.715    0.125
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.134 on 8 degrees of freedom
Multiple R-squared:  0.975,     Adjusted R-squared:  0.9656
F-statistic: 103.8 on 3 and 8 DF,  p-value: 9.582e-07

[1] 0.9749568
```

PRACTICAL - 4

AIM: Implementation of Preprocessing or Data Cleaning(Missing Values) in Weka.

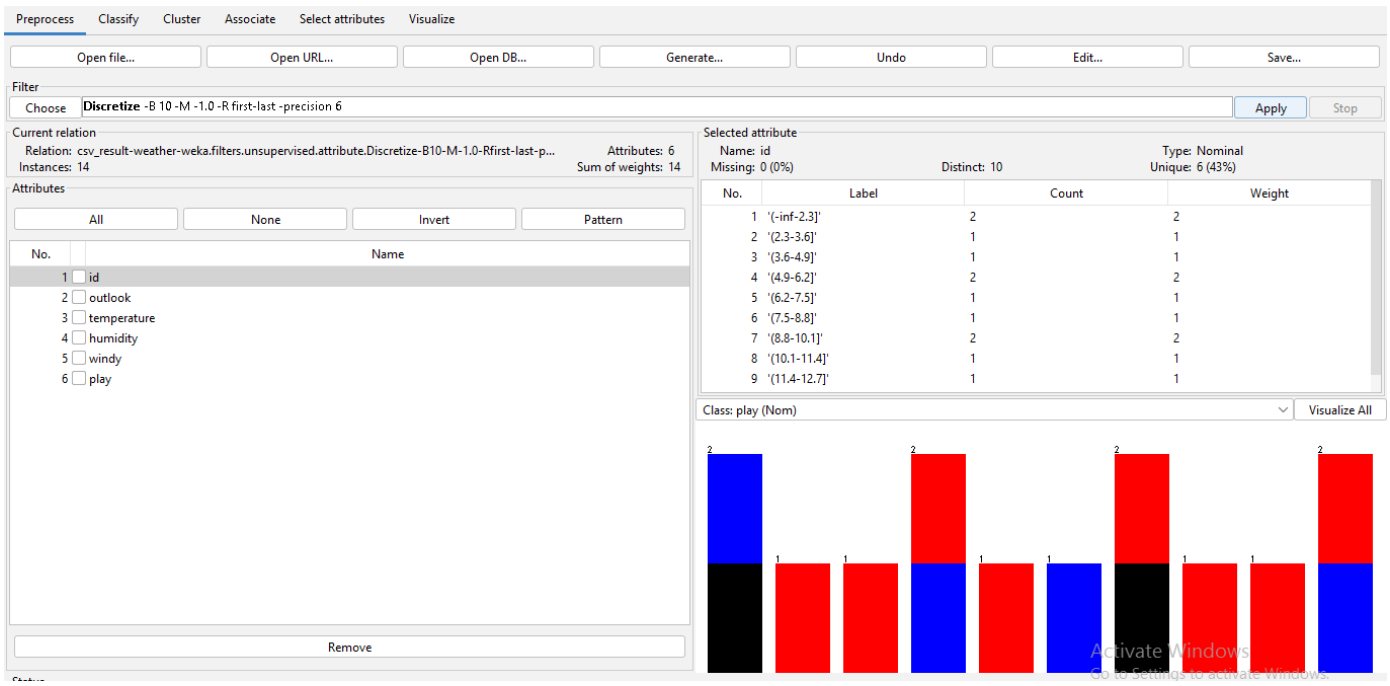
1) REPLACE MISSING VALUES:

Replaces all missing values for nominal and numeric attributes in a dataset with range.



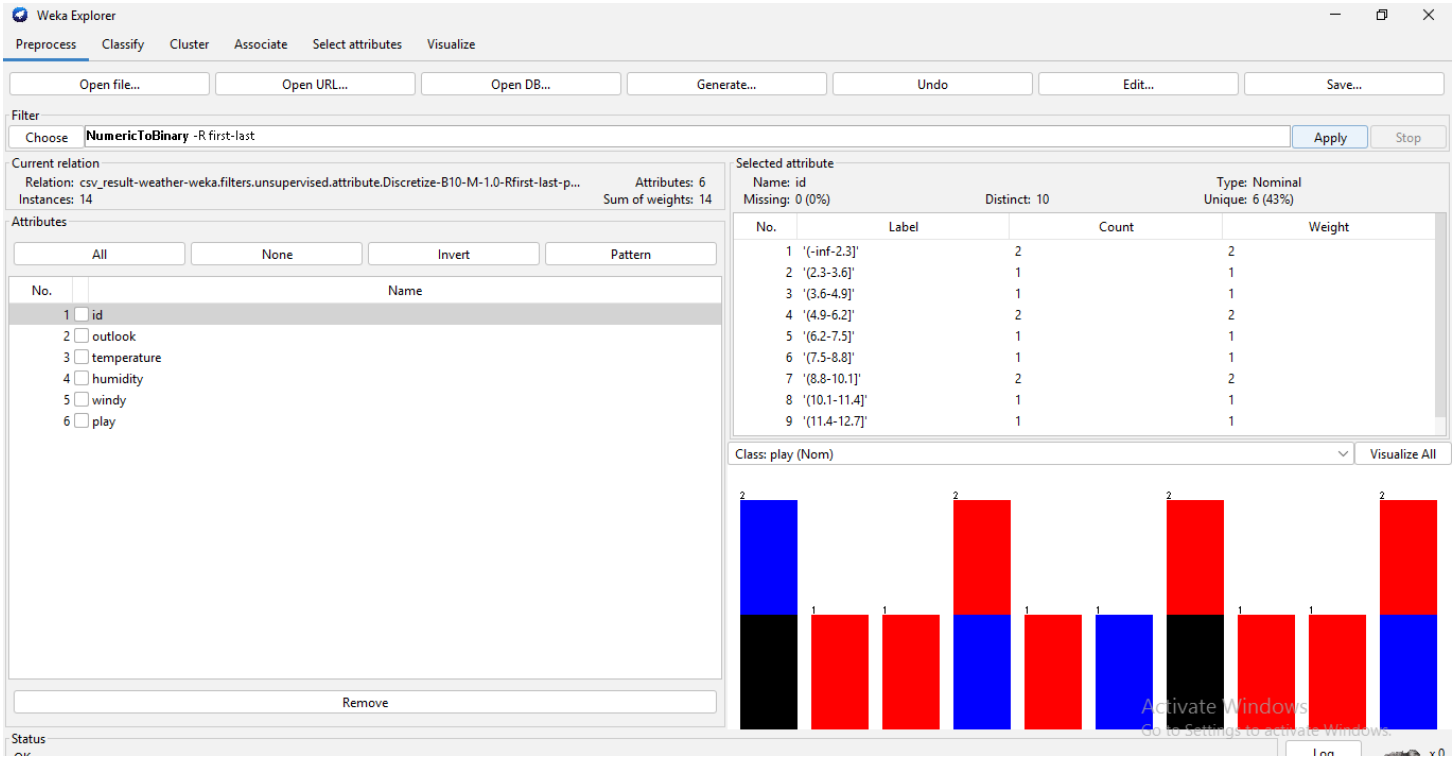
2) DISCRETIZE:

An instance filter that discretizes a range of numeric attributes in the dataset into nominalattributes



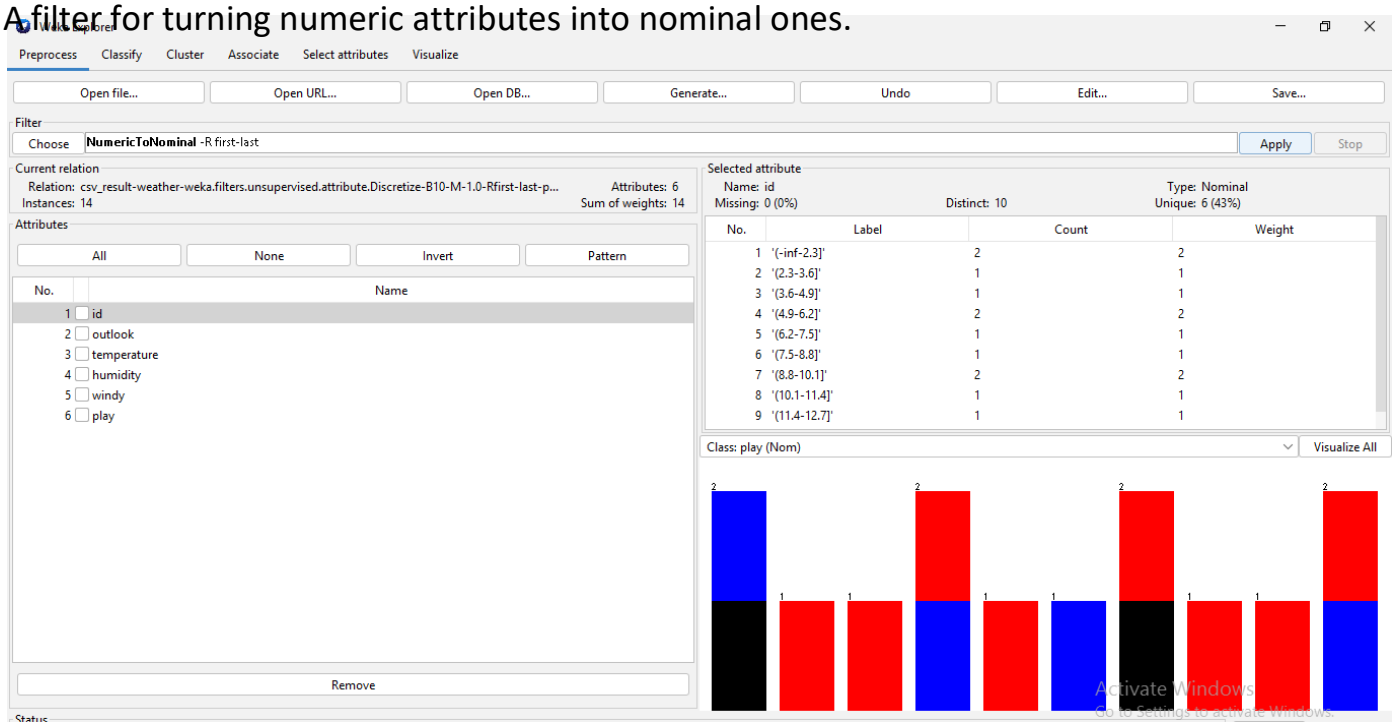
3) NUMERIC TO BINARY:

Converts all numeric attributes into binary attributes (apart from the class attribute, if set): if the value of the numeric attribute is exactly zero, the value of the new attribute will be zeros.



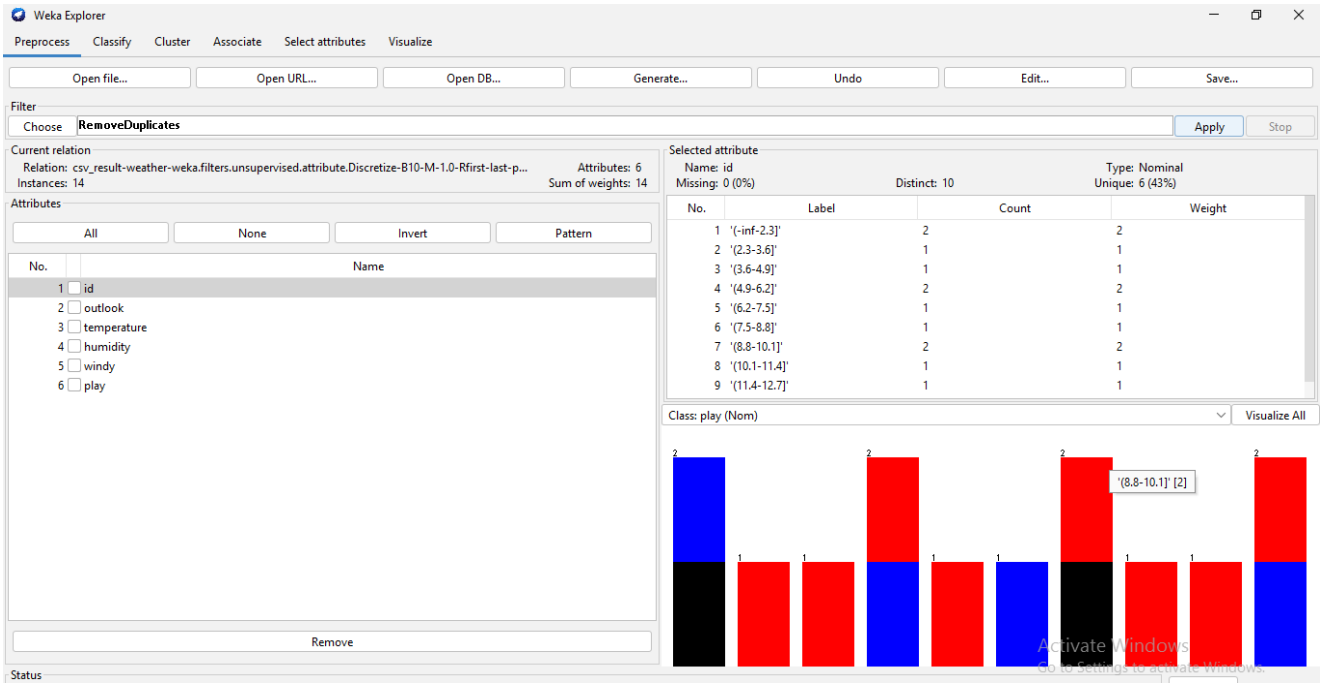
4) NUMERIC TO NOMINAL:

A filter for turning numeric attributes into nominal ones.



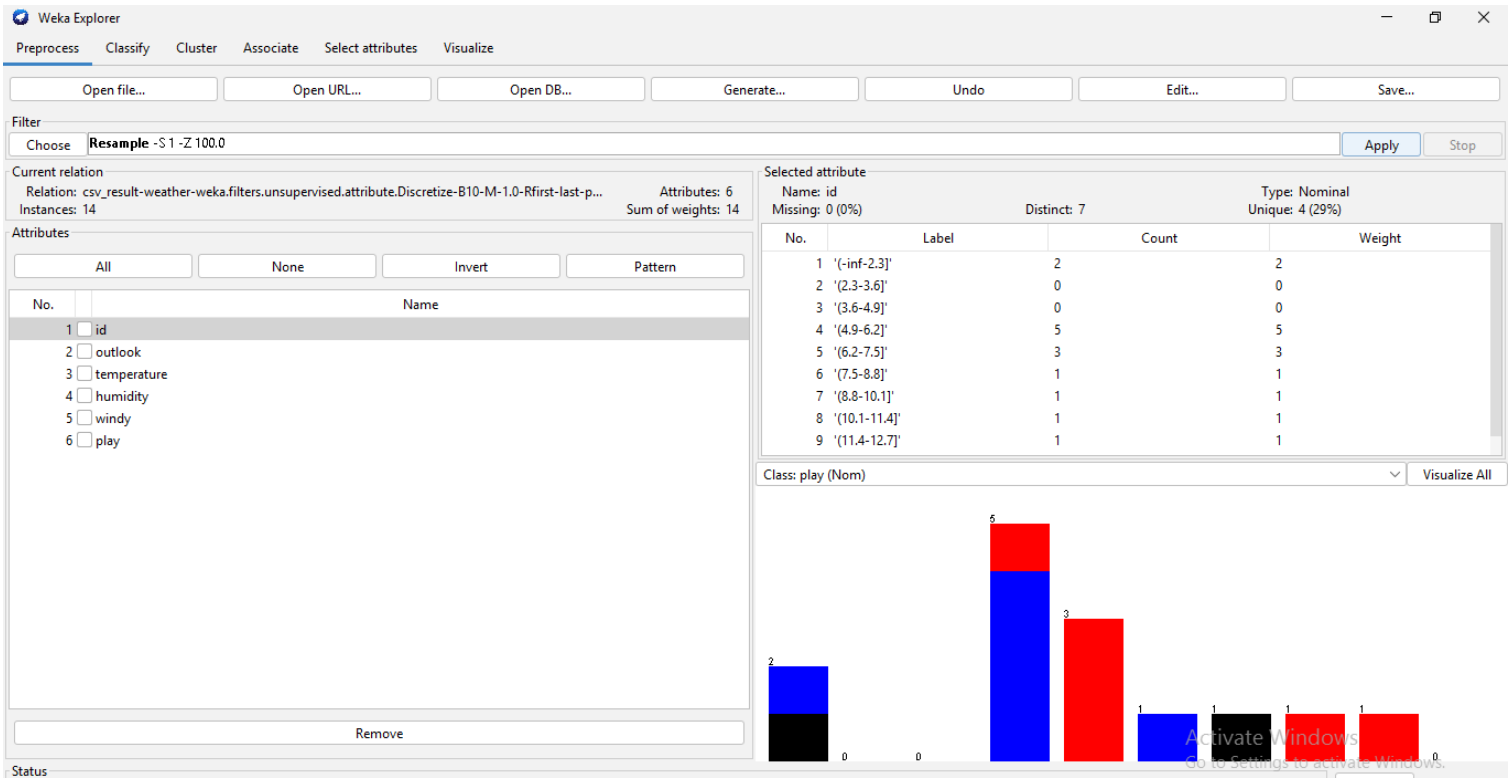
5) REMOVE DUPLICATES:

Removes all duplicate instances from the first batch of data it receives



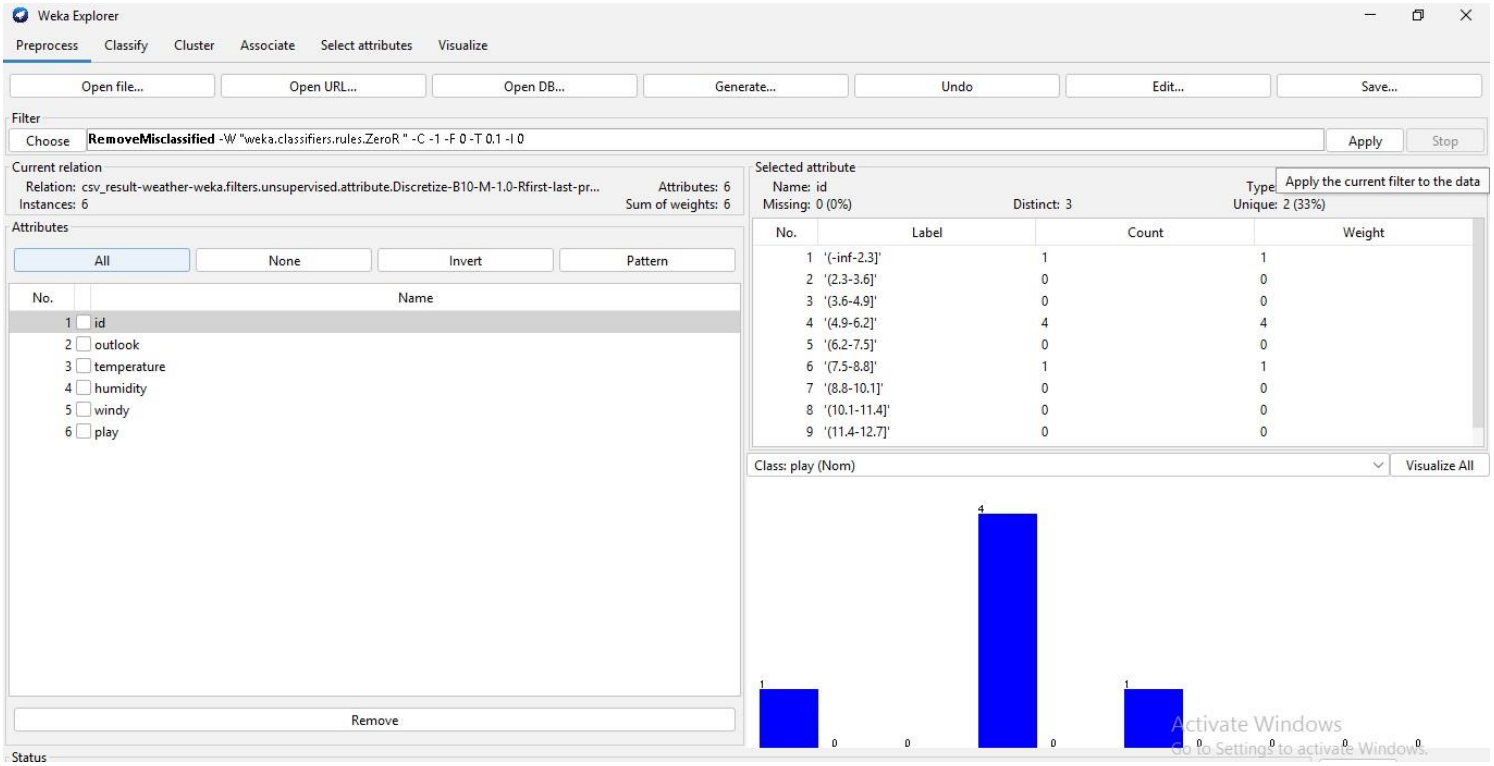
6) RESAMPLE:

Produces a random subsample of a dataset using either sampling with replacement orwithout replacement



7) REMOVE MISCLASSIFIED:

Afilterthatremovesinstanceswhichareincorrectlyclassified.



PRACTICAL - 5

AIM: To Perform Frequent Pattern Mining using Apriori Algorithm over anydataset

Apriori algorithm refers to the algorithm which is used to calculate the association rules betweenobjects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule learning that analyzes that people who bought product A also bought product B.

The given three components comprise the apriori algorithm.

1. Support
2. Confidence
3. Lift

Support

Support (Item 1) = (Transactions relating Item 1) / (Total transactions)

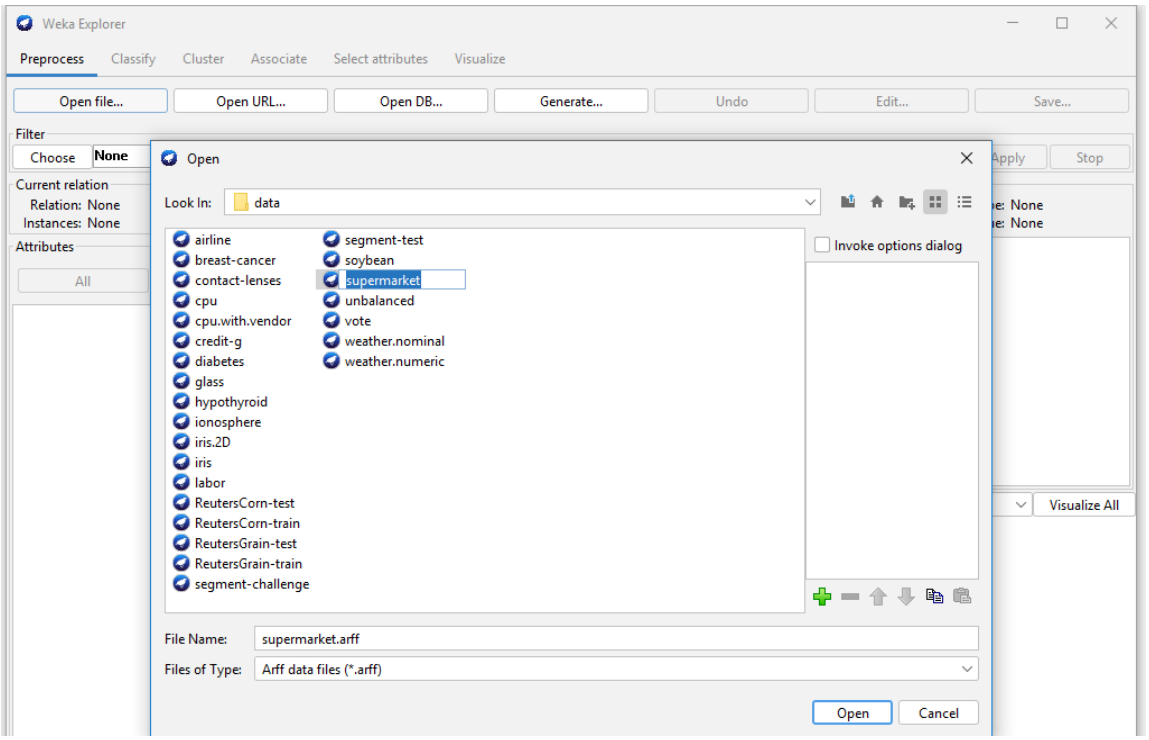
Confidence

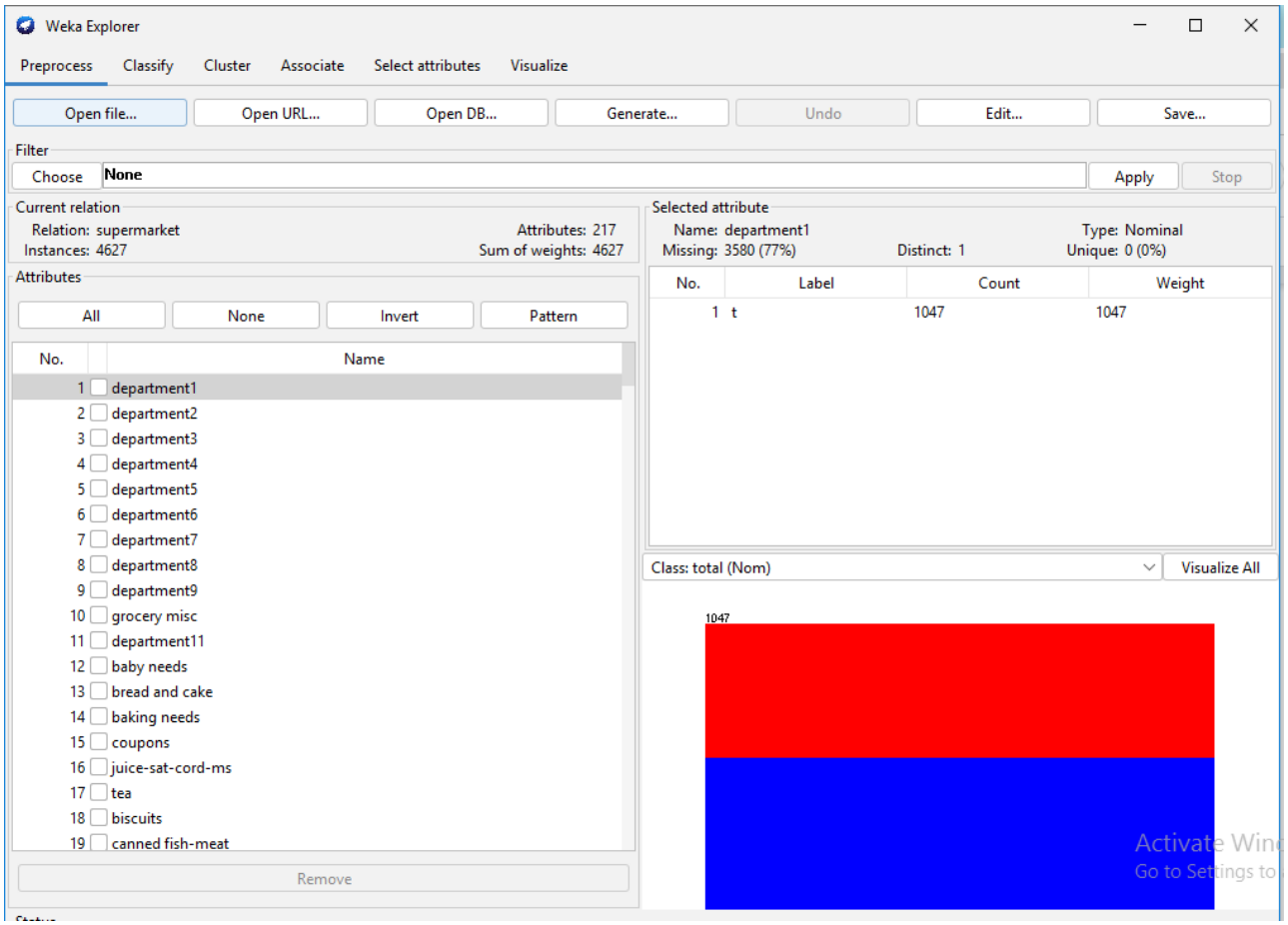
Confidence = (Transactions relating both Item 1 and Item 2) / (Total transactions involving Item1)

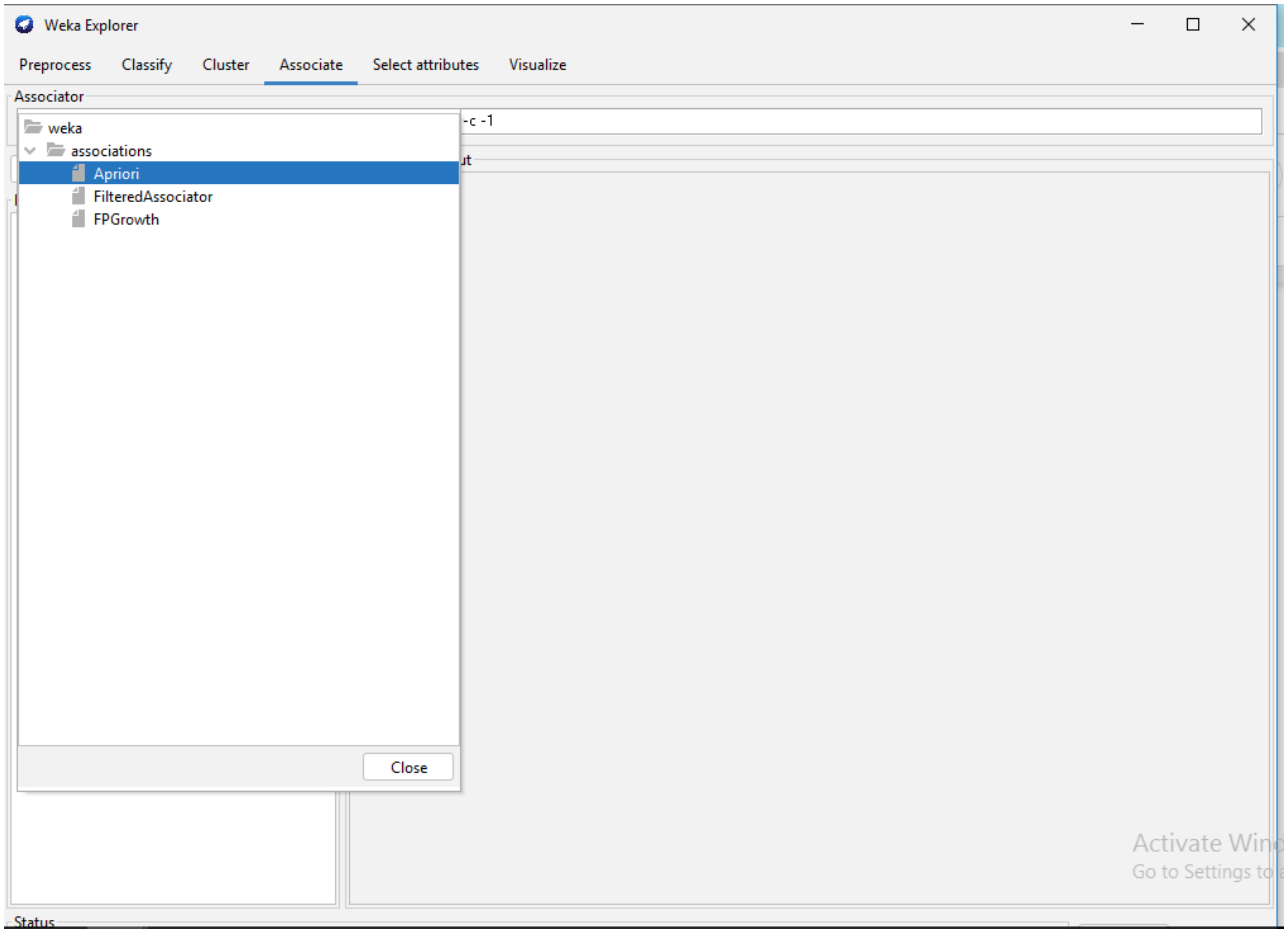
Lift

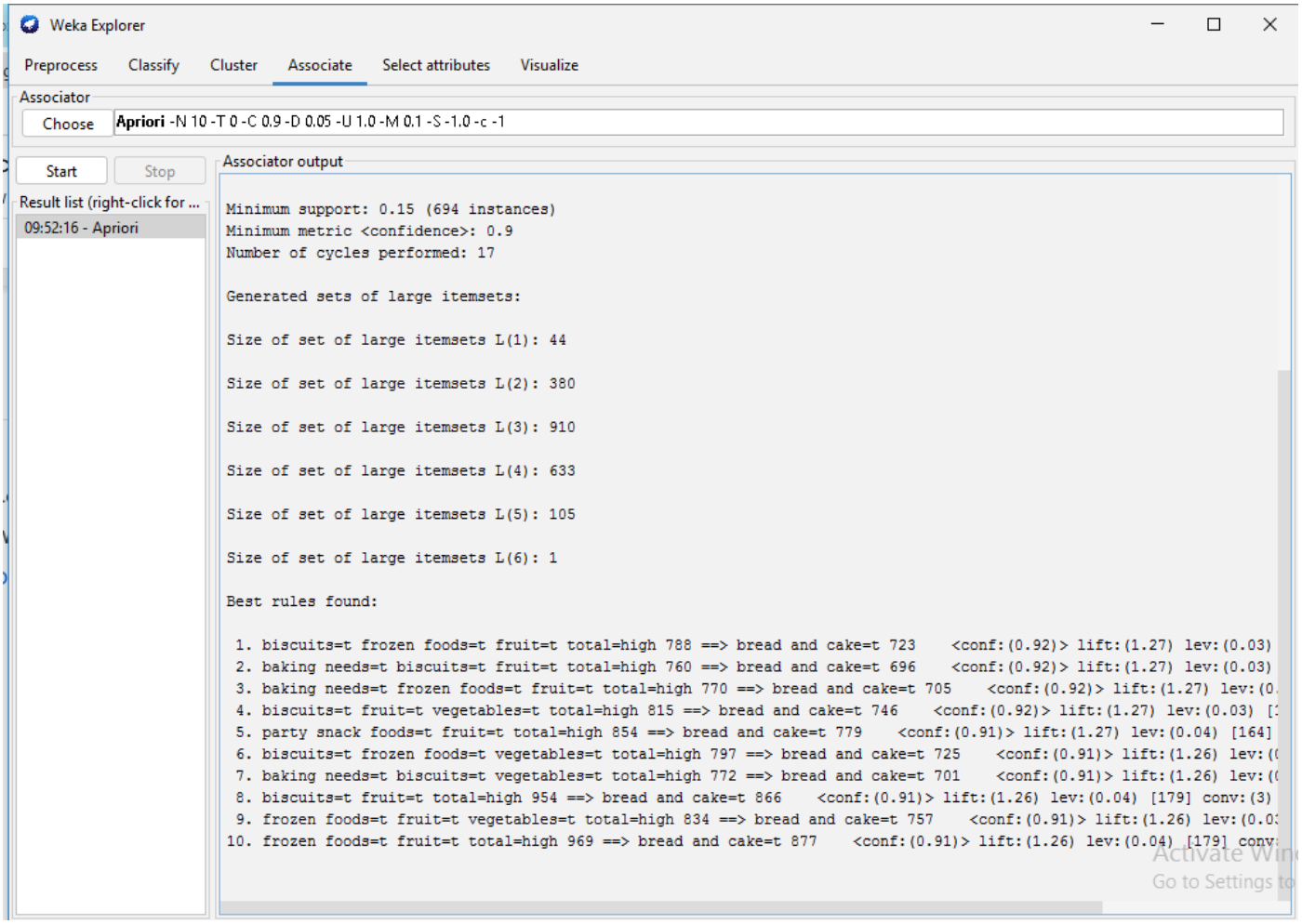
Lift = Confidence/ Support

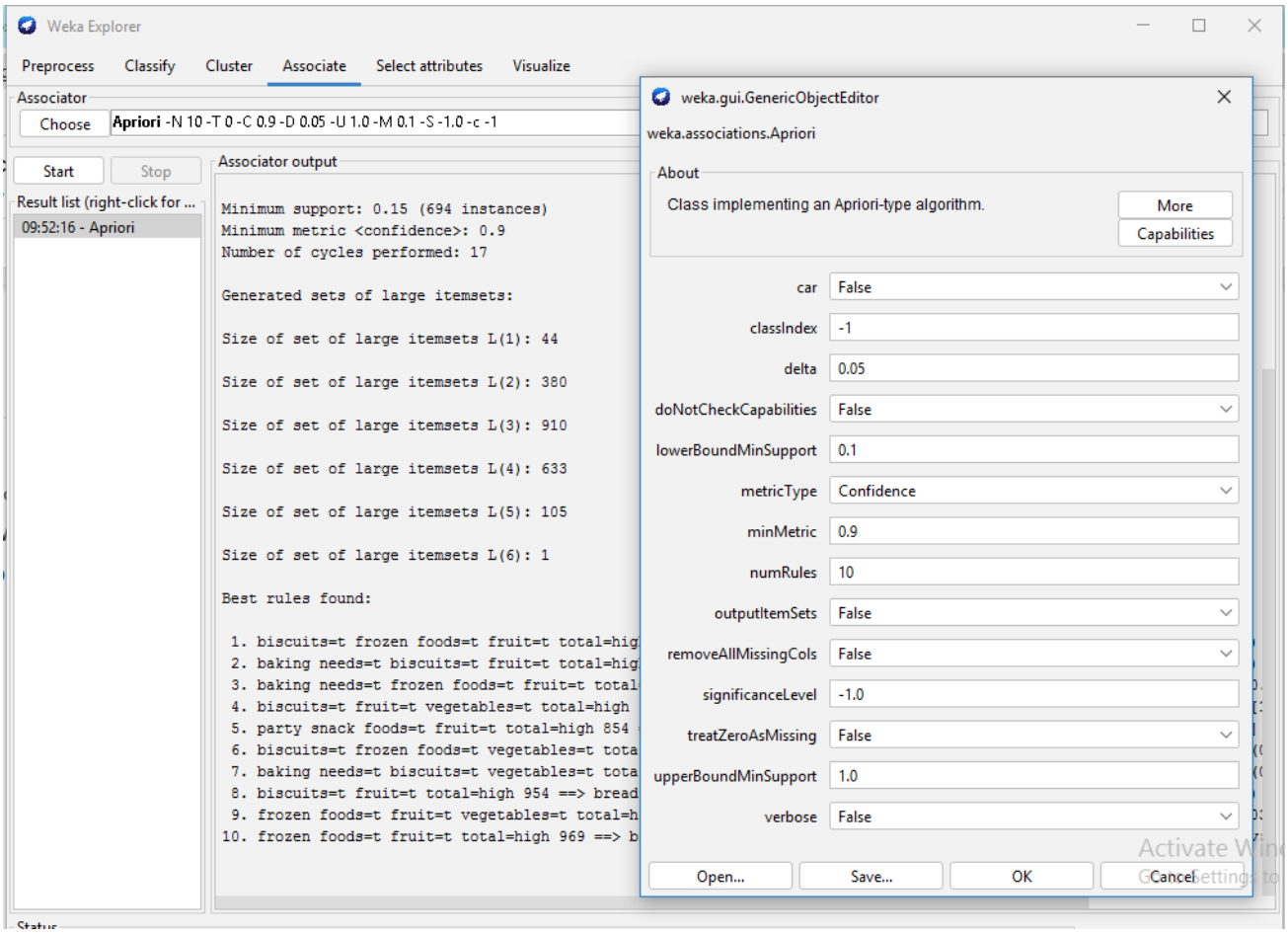
Uploading Data set




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: supermarket
Instances: 4627
Attributes: 217
Sum of weights: 4627
Selected attribute
Name: department1
Missing: 3580 (77%)
Distinct: 1
Type: Nominal
Unique: 0 (0%)
Attributes
All
None
Invert
Pattern
No.
Name
1 department1
2 department2
3 department3
4 department4
5 department5
6 department6
7 department7
8 department8
9 department9
10 grocery misc
11 department11
12 baby needs
13 bread and cake
14 baking needs
15 coupons
16 juice-sat-cord-ms
17 tea
18 biscuits
19 canned fish-meat
Remove
No.
Label
Count
Weight
1 t 1047 1047
Class: total (Nom)
Visualize All
1047


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
weka
associations
Apriori
FilteredAssociator
FPGrowth
Close
-c -1
Status


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -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-click for ...
09:52:16 - Apriori
Associator output
Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17
Generated sets of large itemsets:
Size of set of large itemsets L(1): 44
Size of set of large itemsets L(2): 380
Size of set of large itemsets L(3): 910
Size of set of large itemsets L(4): 633
Size of set of large itemsets L(5): 105
Size of set of large itemsets L(6): 1
Best rules found:
1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723 <conf:(0.92)> lift:(1.27) lev:(0.03)
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696 <conf:(0.92)> lift:(1.27) lev:(0.03)
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705 <conf:(0.92)> lift:(1.27) lev:(0.
4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746 <conf:(0.92)> lift:(1.27) lev:(0.03) [
5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779 <conf:(0.91)> lift:(1.27) lev:(0.04) [164]
6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725 <conf:(0.91)> lift:(1.26) lev:(
7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701 <conf:(0.91)> lift:(1.26) lev:(
8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)
9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757 <conf:(0.91)> lift:(1.26) lev:(0.0
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877 <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:
Activate Win
Go to Settings to


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -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-click for ...
09:52:16 - Apriori
Associator output
Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17
Generated sets of large itemsets:
Size of set of large itemsets L(1): 44
Size of set of large itemsets L(2): 380
Size of set of large itemsets L(3): 910
Size of set of large itemsets L(4): 633
Size of set of large itemsets L(5): 105
Size of set of large itemsets L(6): 1
Best rules found:
1. biscuits=t frozen foods=t fruit=t total=hig
2. baking needs=t biscuits=t fruit=t total=hig
3. baking needs=t frozen foods=t fruit=t total
4. biscuits=t fruit=t vegetables=t total=high
5. party snack foods=t fruit=t total=high 854
6. biscuits=t frozen foods=t vegetables=t tota
7. baking needs=t biscuits=t vegetables=t tota
8. biscuits=t fruit=t total=high 954 ==> bread
9. frozen foods=t fruit=t vegetables=t total=h
10. frozen foods=t fruit=t total=high 969 ==> b
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 0.1
metricType Confidence
minMetric 0.9
numRules 10
outputItemSets False
removeAllMissingCols False
significanceLevel -1.0
treatZeroAsMissing False
upperBoundMinSupport 1.0
verbose False
Open...
Save...
OK
Cancel
Activate Win
Go to Settings to

PRACTICAL - 6

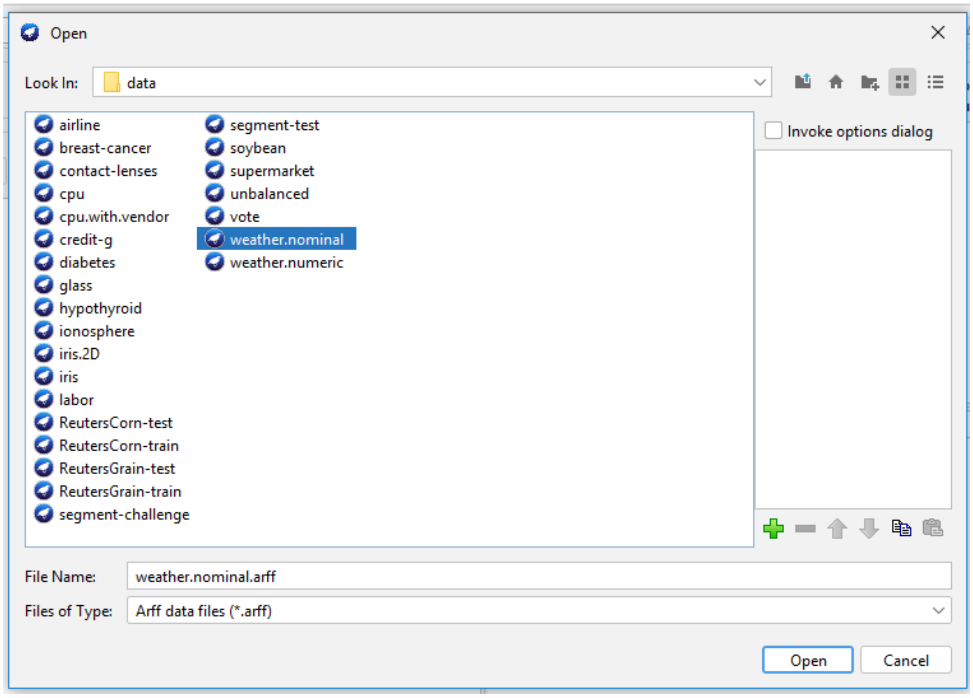
AIM: Implementation of any one classifier using JAVA (Bayesianclassifier) and verify results with WEKA.

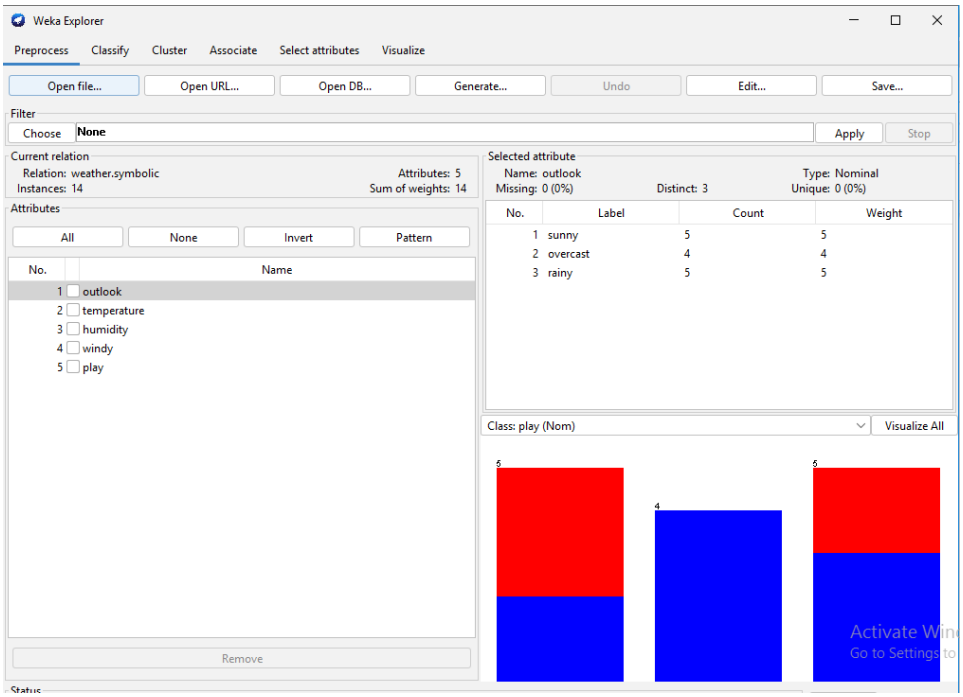
Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It isnot a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

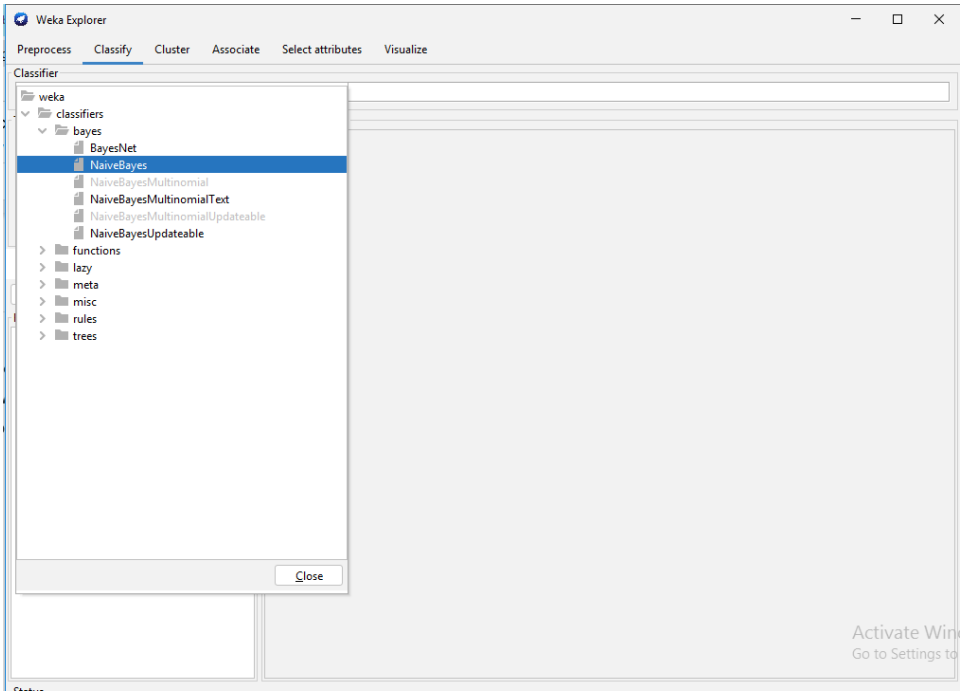
Bayes' Theorem:

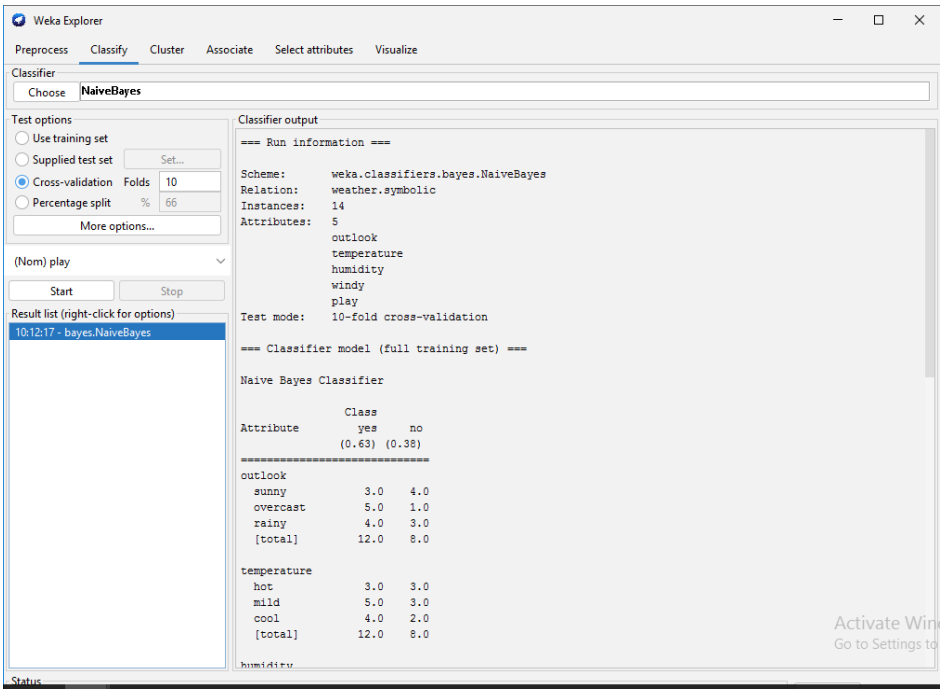
- The formula for Bayes' theorem is given as:

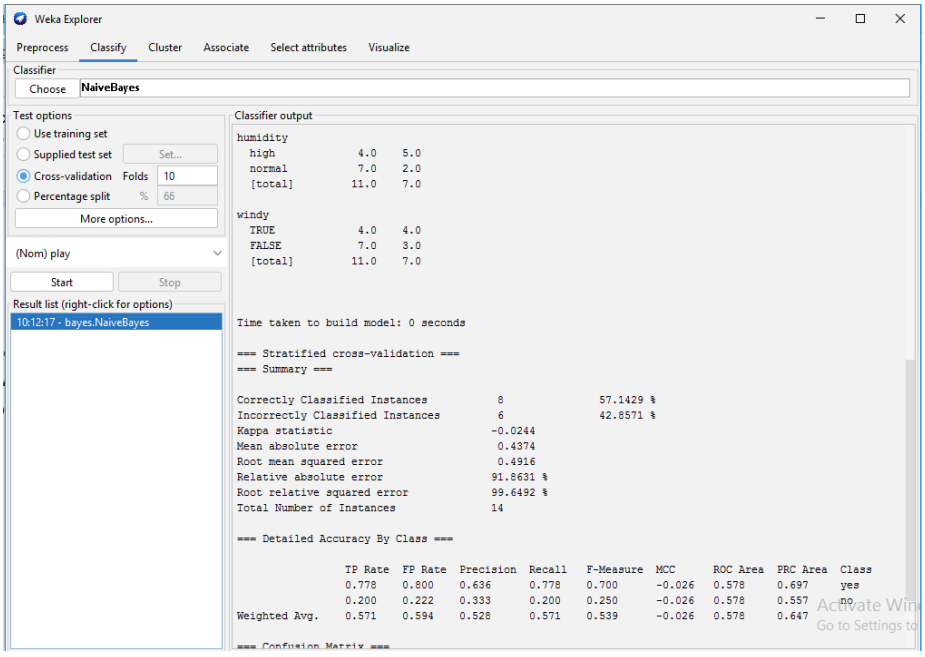
$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$






Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
BayesNet
NaiveBayes
NaiveBayesMultinomial
NaiveBayesMultinomialText
NaiveBayesMultinomialUpdateable
NaiveBayesUpdateable
functions
lazy
meta
misc
rules
trees
Close


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) play
Start
Stop
Result list (right-click for options)
10:12:17 - bayes.NaiveBayes
Classifier output
=== Run information ===
Scheme: weka.classifiers.bayes.NaiveBayes
Relation: weather.symbolic
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
play
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
Naive Bayes Classifier
Class
Attribute yes no
(0.63) (0.38)
outlook
sunny 3.0 4.0
overcast 5.0 1.0
rainy 4.0 3.0
[total] 12.0 8.0
temperature
hot 3.0 3.0
mild 5.0 3.0
cool 4.0 2.0
[total] 12.0 8.0
Status


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) play
Start
Stop
Result list (right-click for options)
10:12:17 - bayes.NaiveBayes
Classifier output
humidity
high 4.0 5.0
normal 7.0 2.0
[total] 11.0 7.0
windy
TRUE 4.0 4.0
FALSE 7.0 3.0
[total] 11.0 7.0
Time taken to build model: 0 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 8 57.1429 %
Incorrectly Classified Instances 6 42.8571 %
Kappa statistic -0.0244
Mean absolute error 0.4374
Root mean squared error 0.4916
Relative absolute error 91.8631 %
Root relative squared error 99.6492 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.778 0.800 0.636 0.778 0.700 -0.026 0.578 0.697 yes
0.200 0.222 0.333 0.200 0.250 -0.026 0.578 0.557 no
Weighted Avg. 0.571 0.594 0.528 0.571 0.539 -0.026 0.578 0.647
=== Confusion Matrix ===