**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

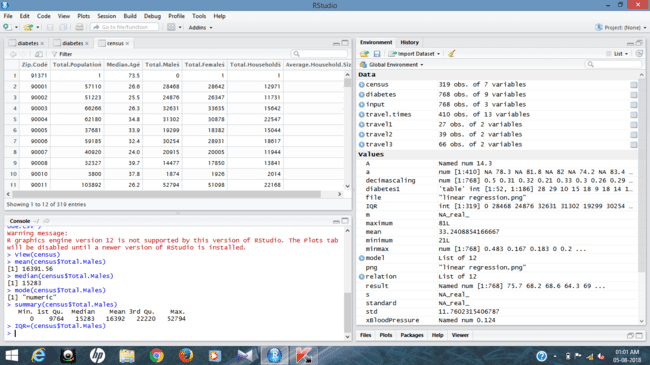
Data dispersion technique including summary.

**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.

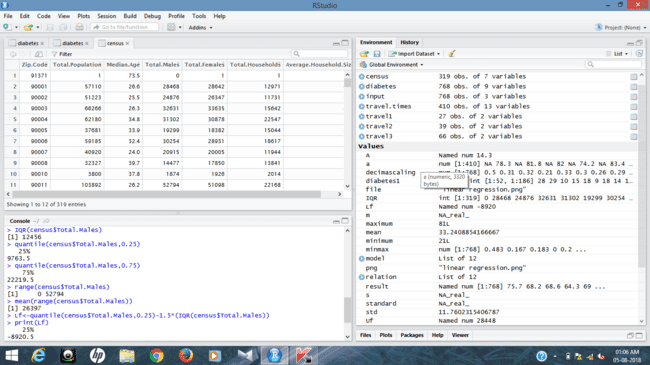
**CENTRAL TENDENCY:**

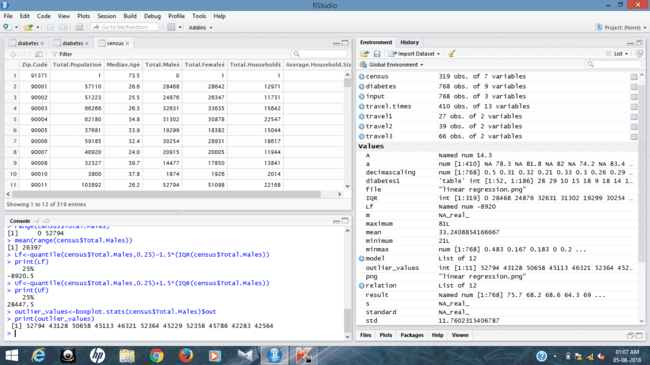
1. **Mean** :The mean is the average of the numbers: a calculated "central" value of a set of numbers.
2. **Median :**The median is a statistical term that is one way of finding the 'average' of a set of data points.
3. **Mode :**The mode of a set of data values is the value that appears most often.
4. **Summary :**A summary table stores data that has been aggregated in a way that answers a meancommon (or resource-intensive) business query.



**MEASURES OF DISPERSION:**

1. **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.
2. **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.
3. **Mid Range :**The arithmetic mean of the largest and the smallest values in a sample or other group.
4. **Outlier :**An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.
5. Lower Fence : Q1 - 1.5\*IQ
6. Upper Fence : Q3 + 1.5\*IQ
7. Outlier Values





**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.

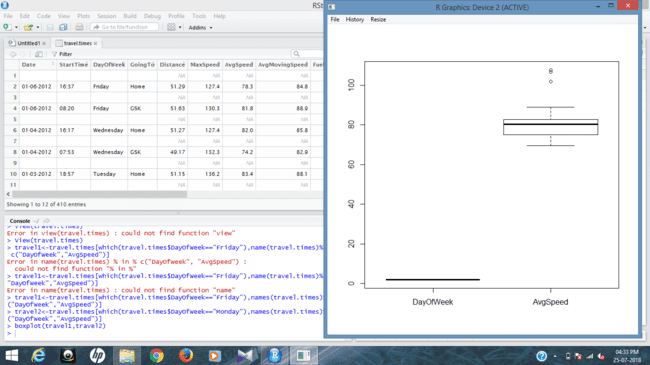
**IMPLEMENTATION:**

1. BoxPlot
2. Histogram
3. ScatterPlot
4. **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:**

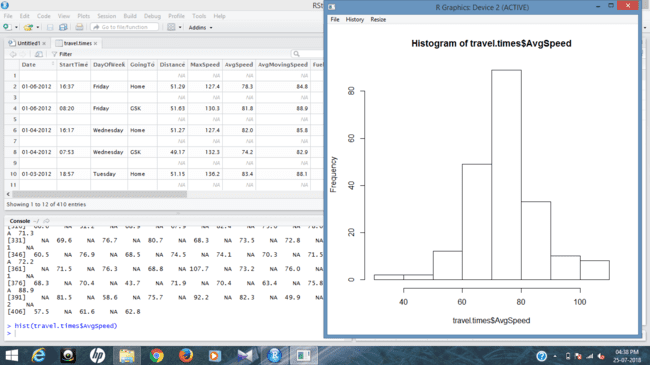


1. **HISTOGRAM:**

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](https://whatis.techtarget.com/definition/independent-variable) is plotted along the horizontal axis and the [dependent variable](https://whatis.techtarget.com/definition/dependent-variable) is plotted along the vertical axis. The data appears as coloured or shaded rectangles of variable area.

* Hist(travel.times$AvgSpeed)

**OUTPUT:**

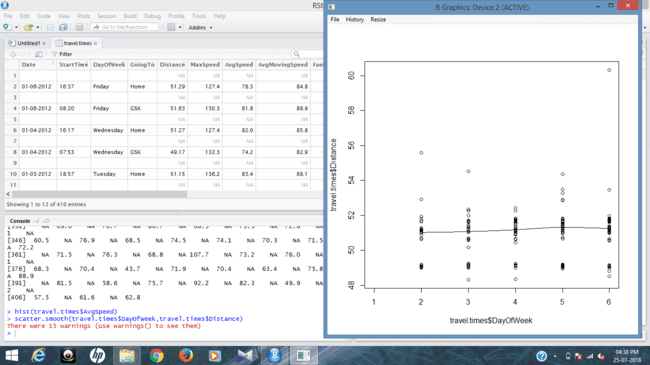
****

1. **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:**

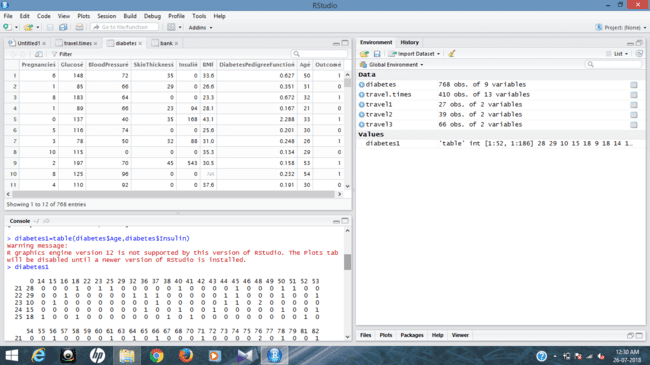
**STEPS INVOLVED:**

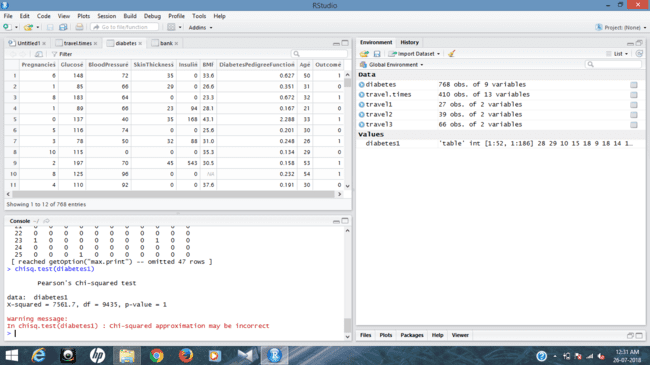
1. Create a new table with required dataframes.
2. After that apply the formula or query for the chi-square test.

**QUERIES:**

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

**OUTPUT:**



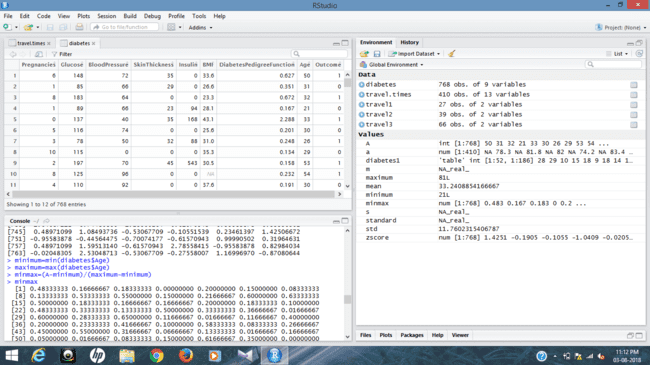


* **NORMALIZATION :**

1. **MIN MAX NORMALIZATION:**

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

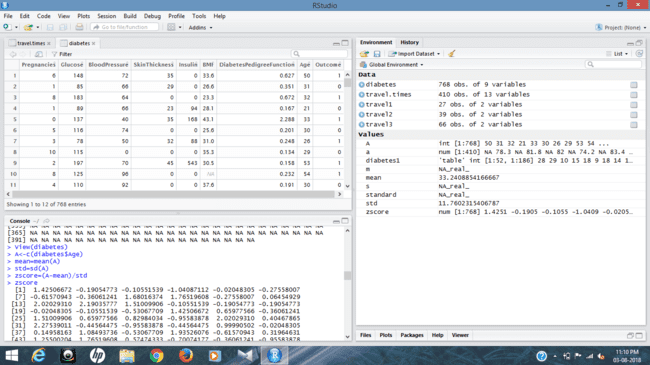
**OUTPUT:**

****

1. **Z SCORE NORMALIZATION :**

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

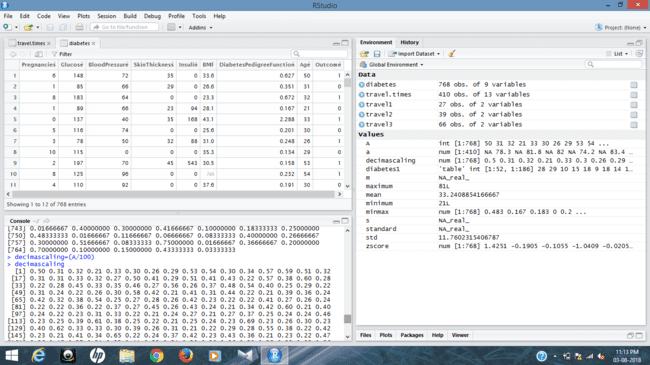
**OUTPUT:**



1. **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.

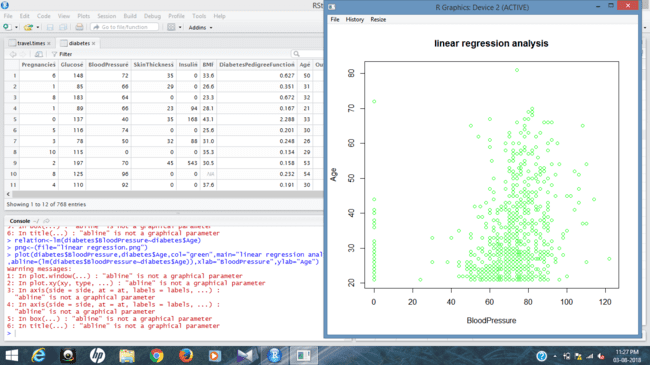
* **LINEAR REGRESSION :**

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 :**

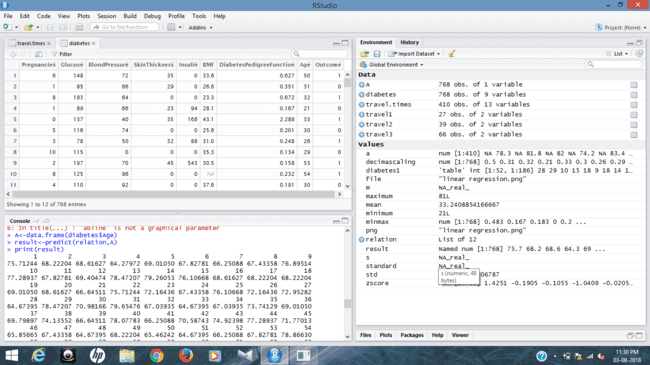
* Relation <- lm(diabetes$BloodPressure~diabetes$Age)
* Png<- (file=”linear regression.png”)
* Plot(diabetes$Age, diabetes$BloodPressure, col=”green”, main= “ Linear Regression Analysis” , abline= (lm(diabetes$BloodPressure~ diabetes$Age)), xlab = “BloodPressure”, ylanb= “Age”)

**OUTPUT:**

****

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

**OUTPUT :**

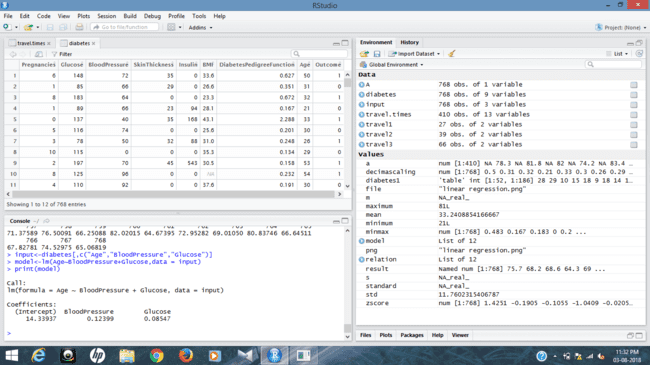
****

* **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.

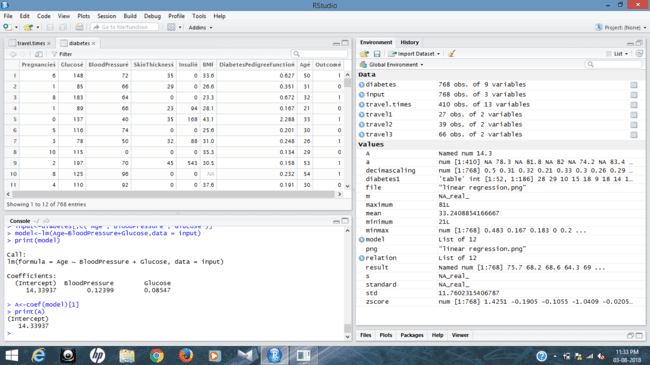
* **QUERIES :**
* Input <- diabetes[,c(“Age”, “BloodPressure”, “Glucose”)]
* Model <- lm(Age~ BloodPressure+Glucose,data=input)
* Print(model)

**OUTPUT:**

****

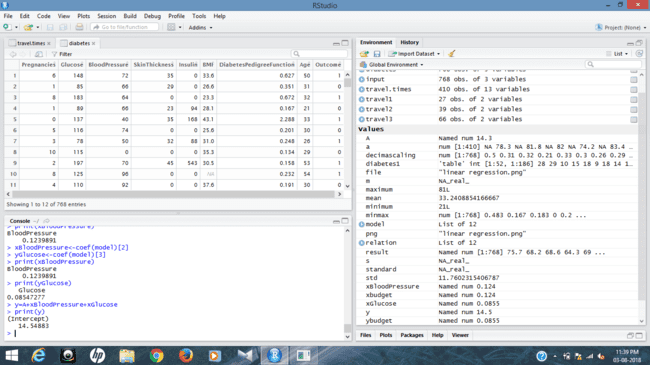
* A<- coef(model)[1]
* Print(A)

**OUTPUT:**

****

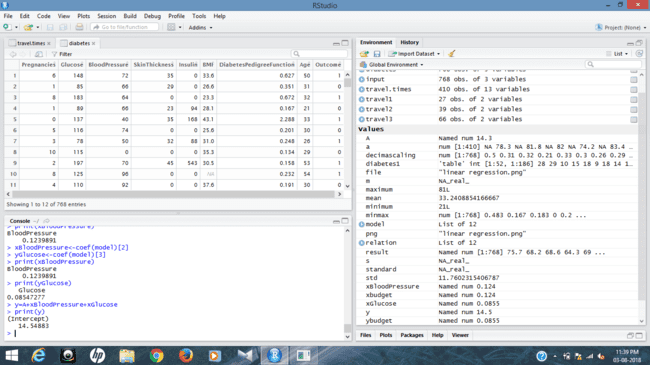
* xBloodPressure<- coef(model)[2]
* yGlucose<- coef(model)[3]
* print(xBloodPressure)
* print(yGlucose)

**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 :**

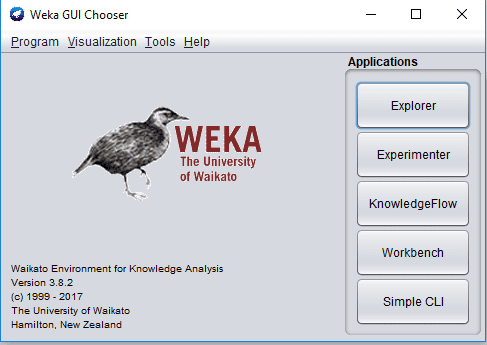
For each attribute in the dataset find the following information:

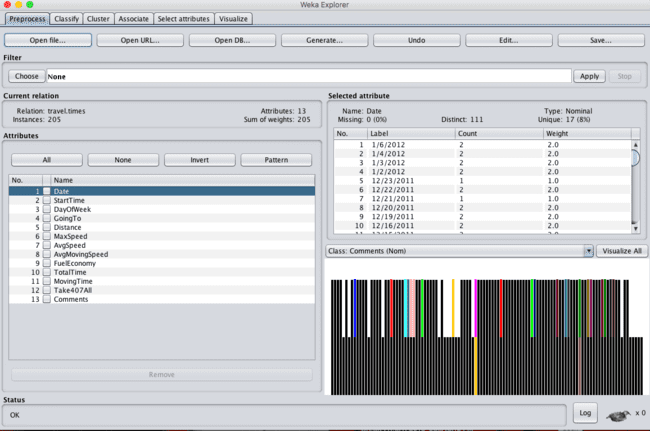
1. Attribute type
2. Percentage of missing values
3. Find the minimum, maximum, mean, Standard Deviation with numerical attributes.
4. Are there any records that have a values that no other record has ?
5. Write a note on class distribution for each of the attributes.
6. Apply attribute selection measures 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 :**

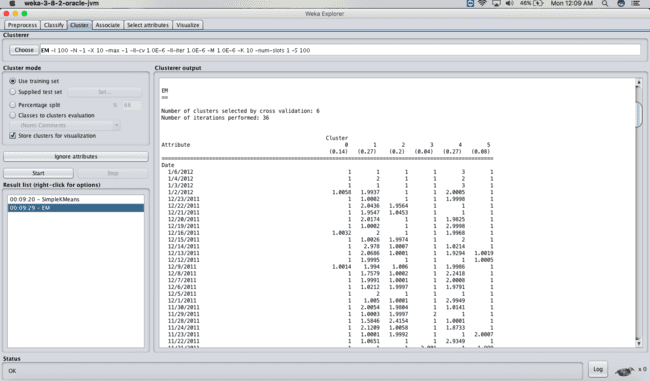
1. **ATTRIBUTE TYPE :**

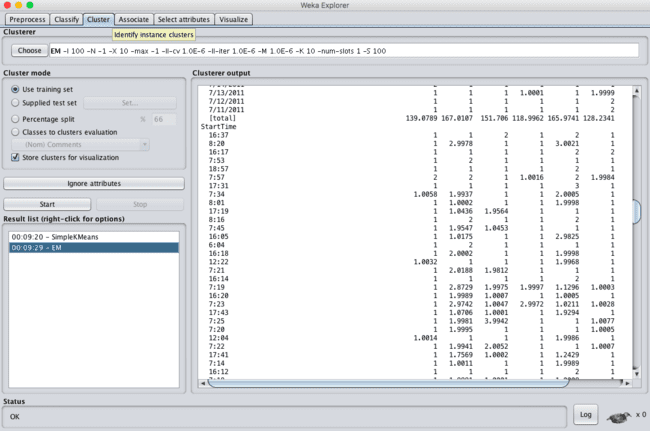
|  |  |  |
| --- | --- | --- |
| **S.NO** | **ATTRIBUTE** | **TYPE** |
|  | Date | Nominal |
|  | Start Time | Nominal |
|  | Day Of Week | Nominal |
|  | Going To | Nominal |
|  | Distance | Numeric |
|  | Max Speed | Numeric |
|  | Avg Speed | Numeric |
|  | Avg Moving Speed | Numeric |
|  | Fuel Economy | Nominal |
|  | Total Time | Numeric |
|  | Moving Time | Numeric |
|  | Comments | Nominal |
|  | Take 407 All | Nominal |

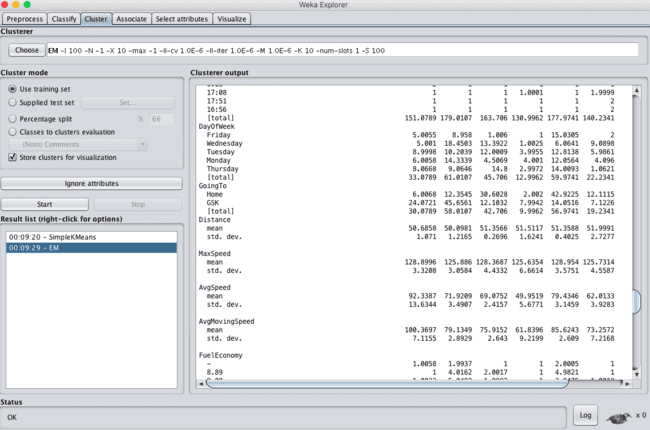
1. **PERCENTAGE OF MISSING VALUES :**

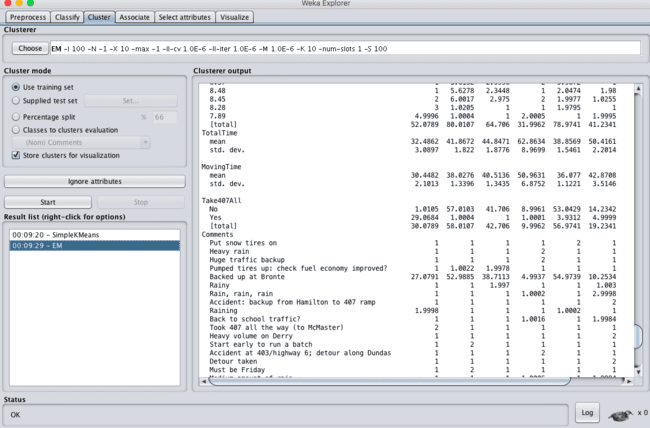
|  |  |  |
| --- | --- | --- |
| **S.NO** | **ATTRIBUTE** | **Percentage Of Missing Values** |
|  | Date | 0 % |
|  | Start Time | 0 % |
|  | Day Of Week | 0 % |
|  | Going To | 0 % |
|  | Distance | 0 % |
|  | Max Speed | 0 % |
|  | Avg Speed | 0 % |
|  | Avg Moving Speed | 0 % |
|  | Fuel Economy | 8 % |
|  | Total Time | 0 % |
|  | Moving Time | 0 % |
|  | Comments | 88 % |
|  | Take 407 All | 0 % |

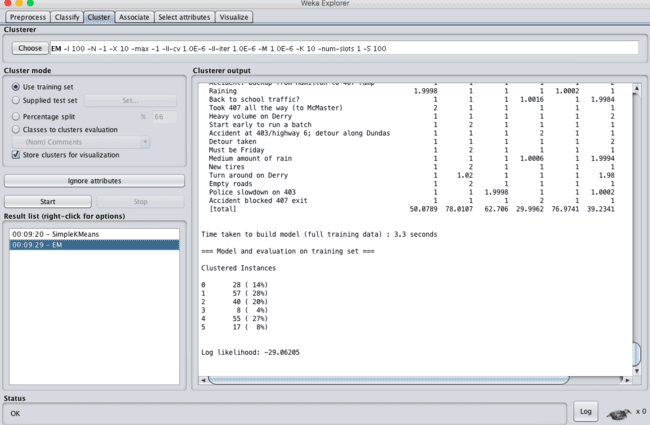
1. **MIN, MAX, MEAN, STANDARD DEVIATION :**



****

****

****

****

1. **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.

1. **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.

1. **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 :**

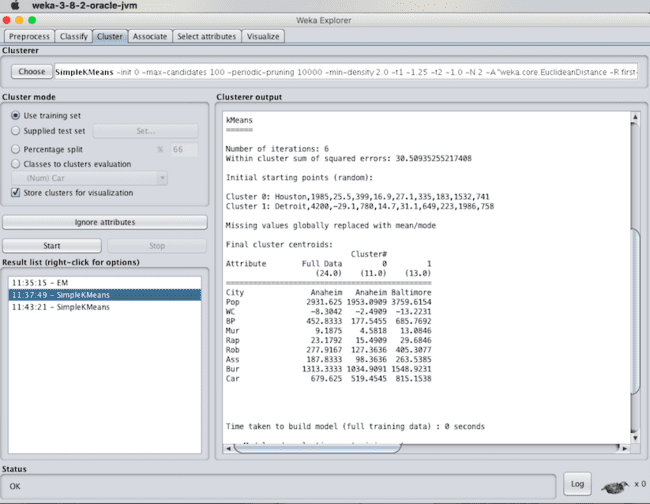
Consider a dataset of citycrimes.csv file of which it contains the attributes are City, Pop, WC, BP, Mur, Rap, Rob, Ass, Bus and car for the performance of the dataset by applying the K-means algorithm in weka and as well using R- tool.

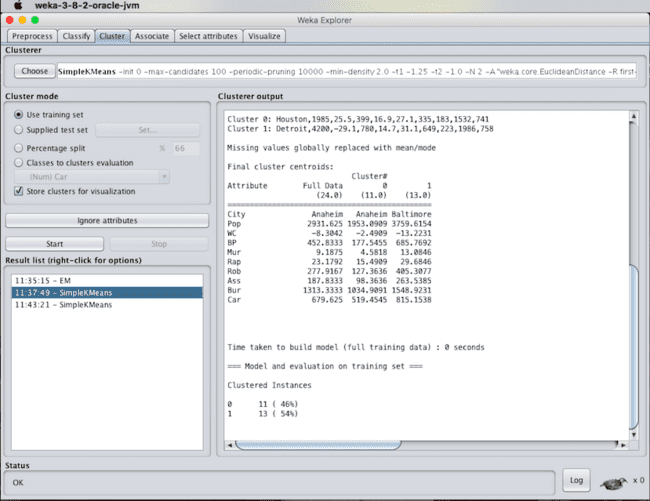
* **USING WEKA TOOL :**

1. **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.

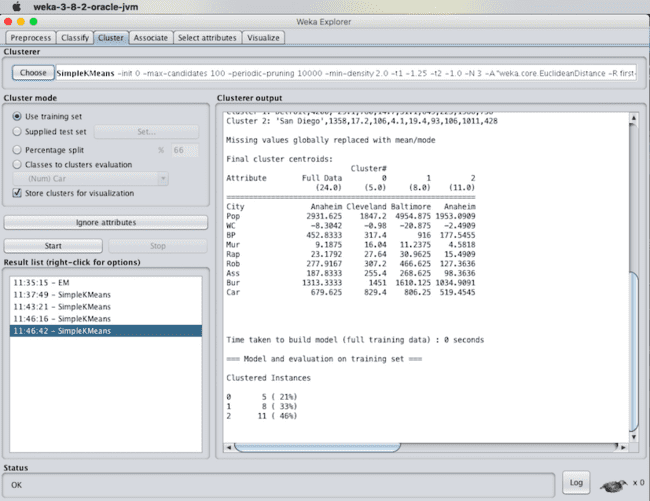
****

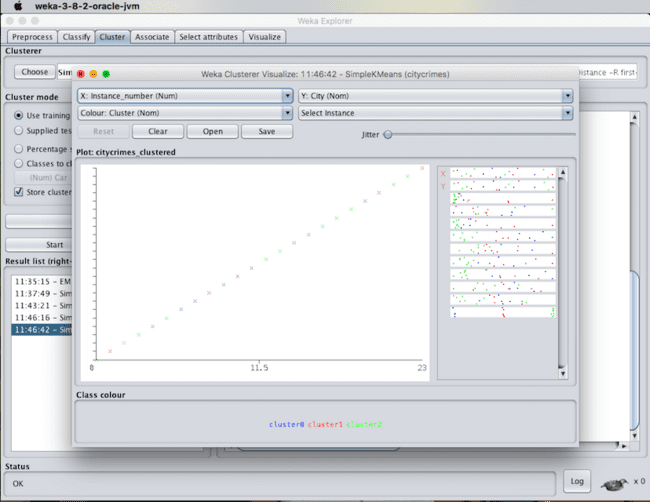
****

1. **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.





1. **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.

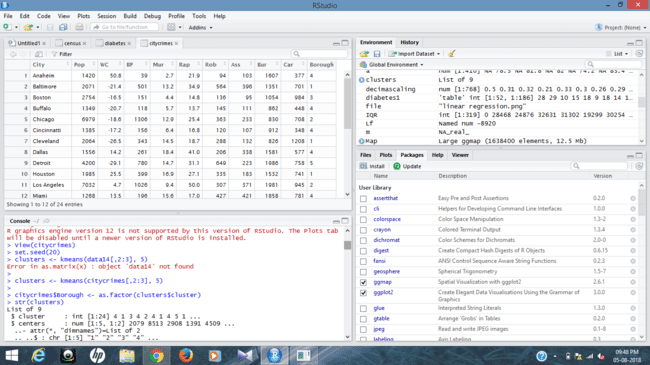
* **USING R-TOOL :**

**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 :**

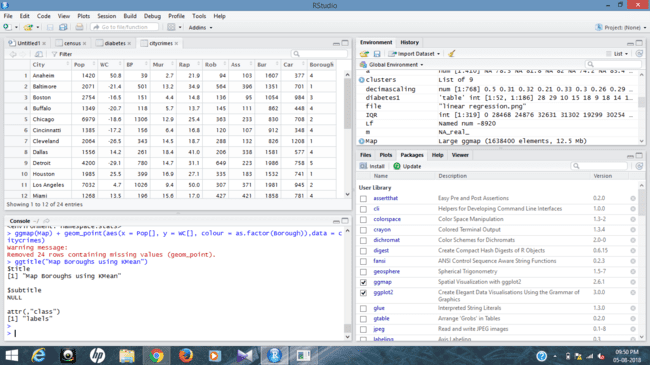
* set.seed(20)
* clusters <- kmeans(citycrimes[,2:3], 5)
* citycrimes$Borough<- as.factor(clusters$cluster)
* str(clusters)

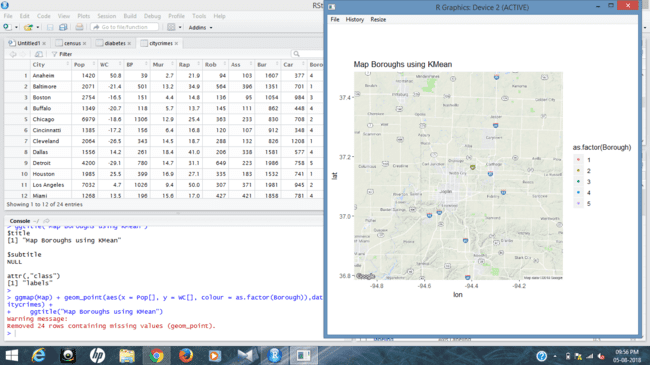




* library(ggmap)
* Map <- get\_map(“citycrimes:,zoom=10)
* ggmap(Map) + geom\_point(aes(x = Pop[] , y = WC[] , colour = as.factor(Borough)) , data = citycrimes ) +

ggtitle(“Map Boroughs using KMean”)





**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 :**

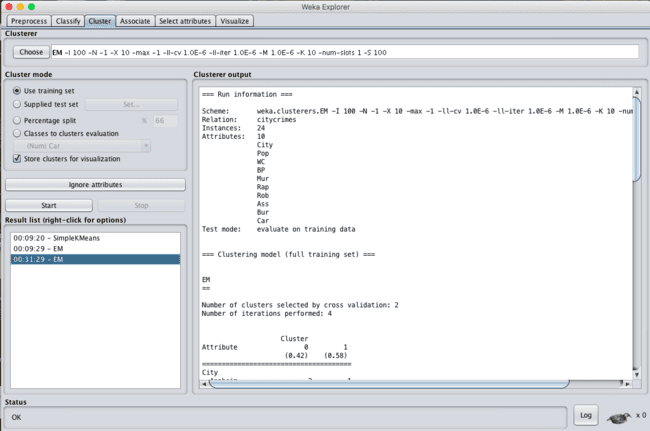
Analyze the dataset using Expectation Maximization algorithm(EM) by setting the minimum standard deviation for normal density calculation and compare the results with the simple K-means algorithm.

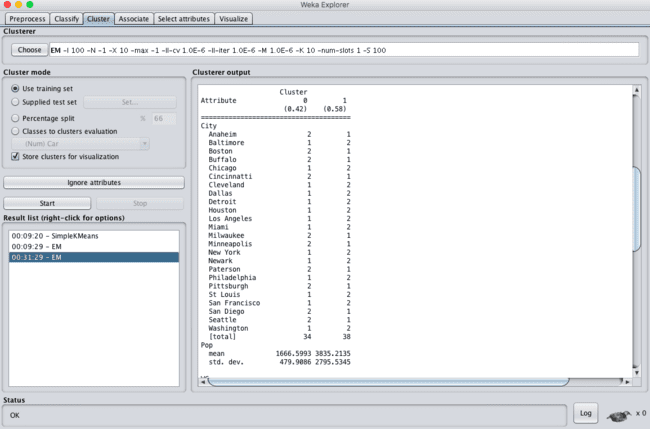
**DESCRIPTION :**

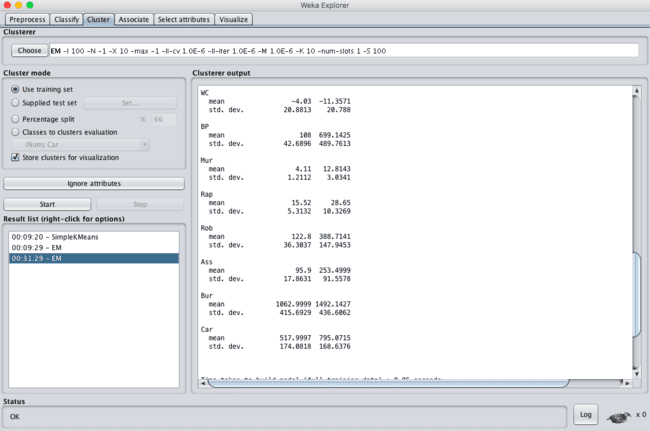
Consider a dataset of citycrimes.csv file of which it contains the attributes are City, Pop, WC, BP, Mur, Rap, Rob, Ass, Bus and car for the performance of the dataset by applying the K-means algorithm in weka and as well using R- tool.

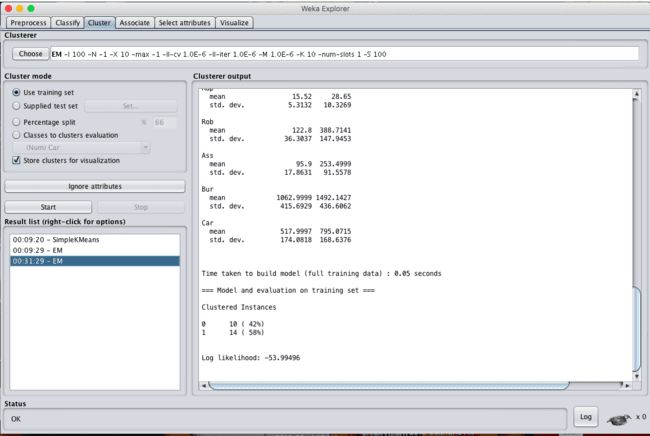
When the clustering is been made through the expectation maximization algorithm by setting minimum standard deviation values then the results will be of the following :

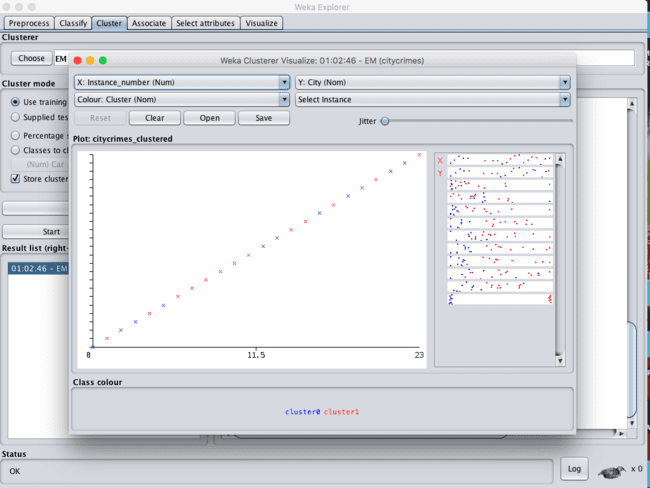
* **Steps Involved :**
* Initially, load the dataset into the weka tool and check for all the attributes present in the dataset.
* Then move to cluster panel and apply the EM algorithm technique for the datasheet.
* Finally, Observe the results that are obtained.



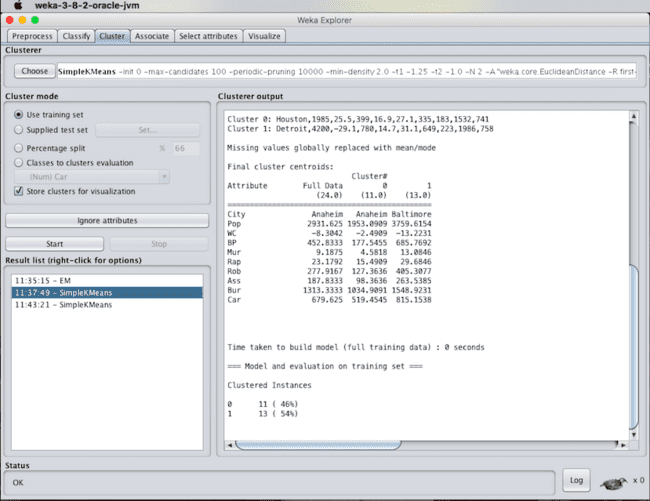






****

* **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:

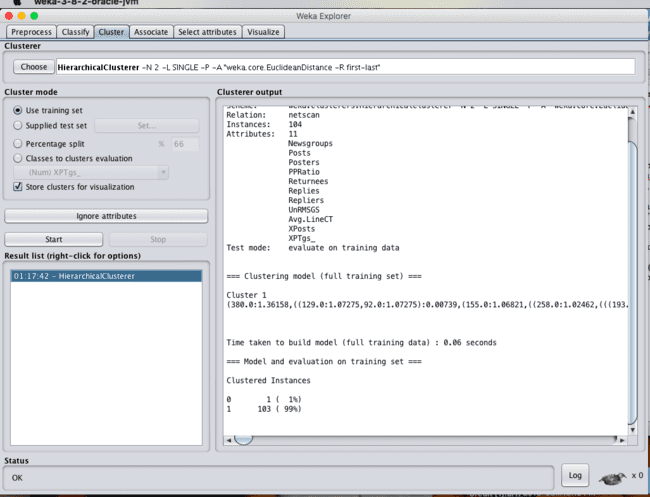
1. what are the most alive groups in terms of number of people involved, cluster together.
2. what are the most active community.

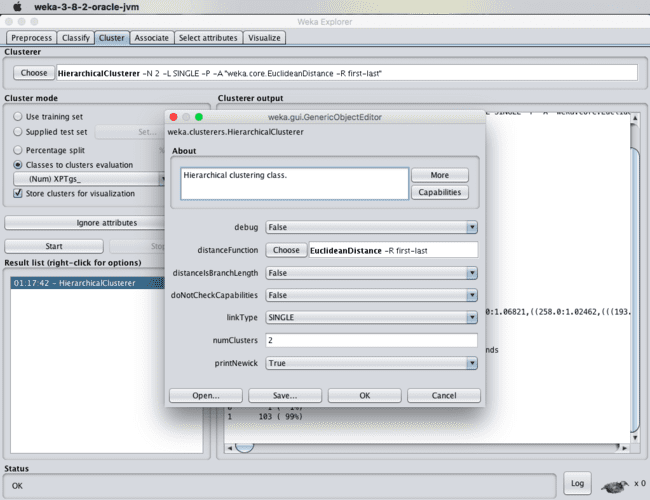
**DESCRIPTION:**

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 :**

1. 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.
2. 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.





**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:

1. Upper bound min\_sup
2. Lower bound min\_sup
3. Metric type
4. Output itemsets

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

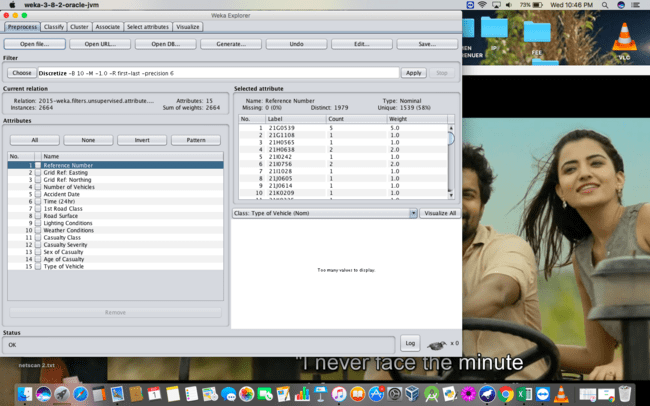
**DESCRIPTION :**

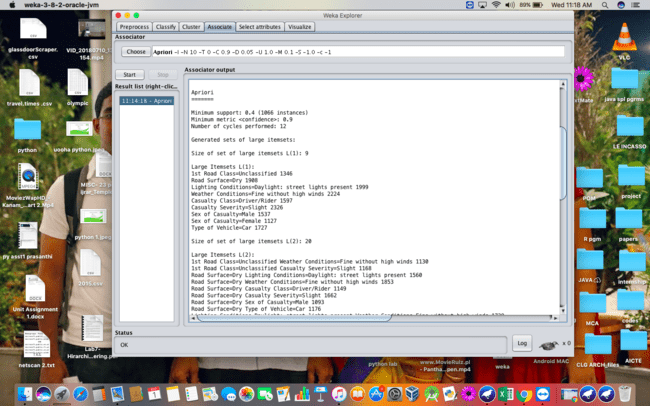
Consider a dataset of 2015.csv file of which it contains the attributes are Reference Number, Grid ref: Easting, Grid Ref: Northing, Number of vehicles, Accident date, Time(24 hr), 1st Road class, Road Surface, Lighting conditions, Weather conditions, casuality class, Sex of casuality, Age of casuality,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

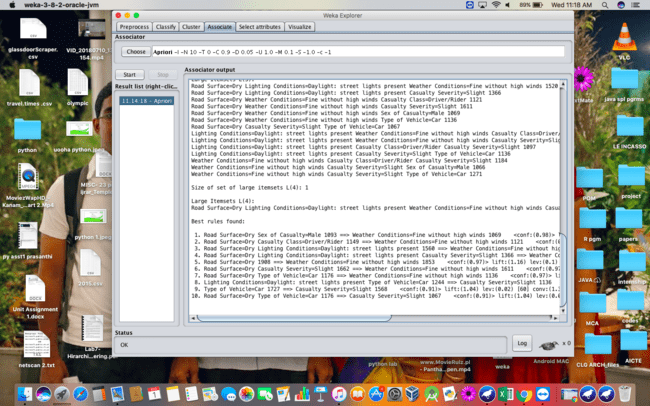
* **USING WEKA TOOL :**

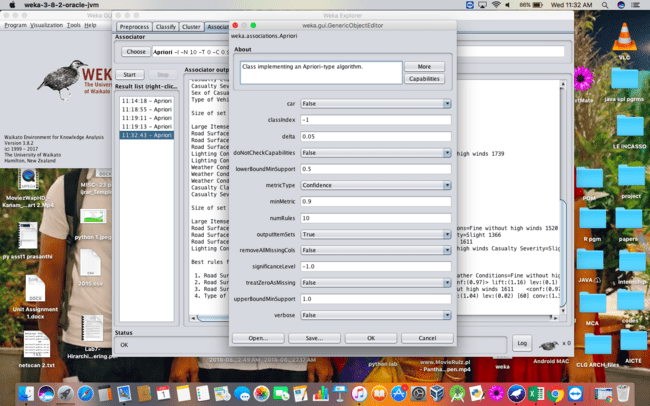
**STEPS INVOLVED :**

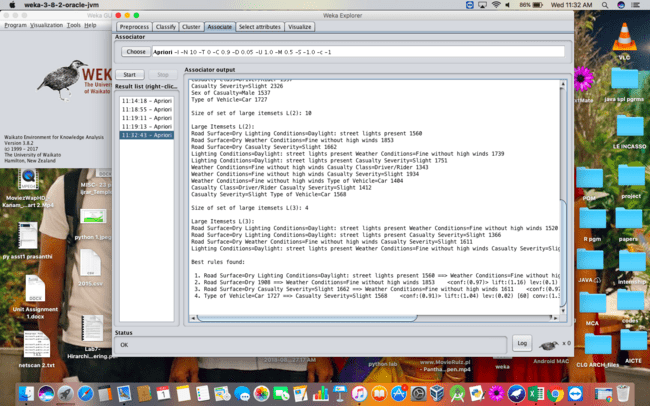
* Choose a set of attributes for clustering and for giving a motivation.
* Choose the dataset and import the dataset into Weka tool.
* Discretize the attributes from numeric to nominal to perform the algorithm.
* Cluster the dataset and choose simple Apriori algorithm.
* Set the Upper bound min\_sup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

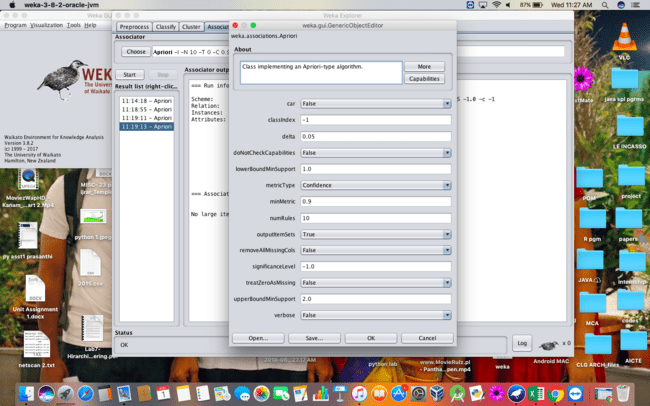


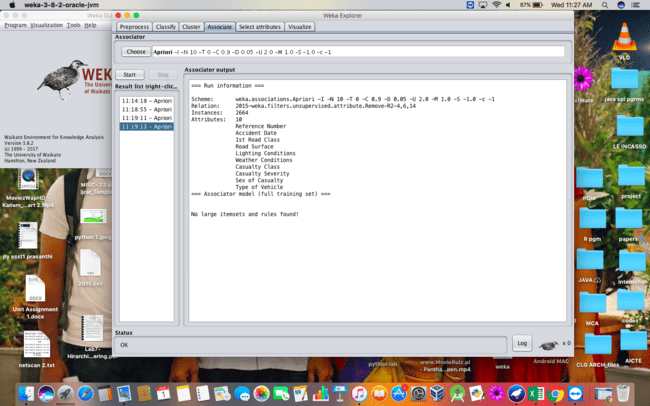












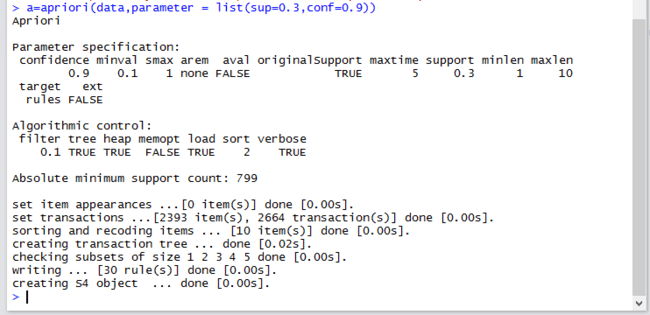
* **USING R-TOOL :**

**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)



**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:

1. Upper bound min\_sup
2. Lower bound min\_sup
3. 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.

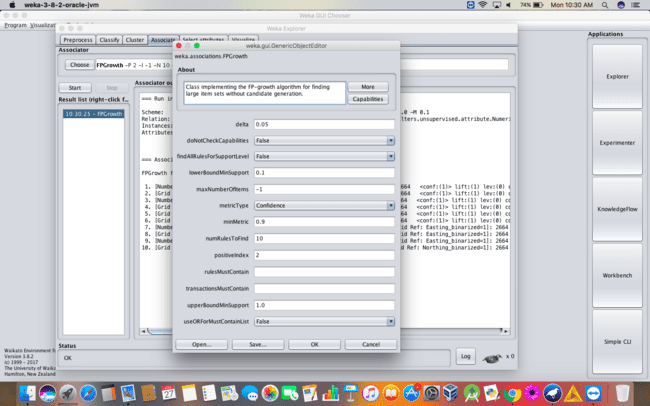
* **USING WEKA TOOL :**

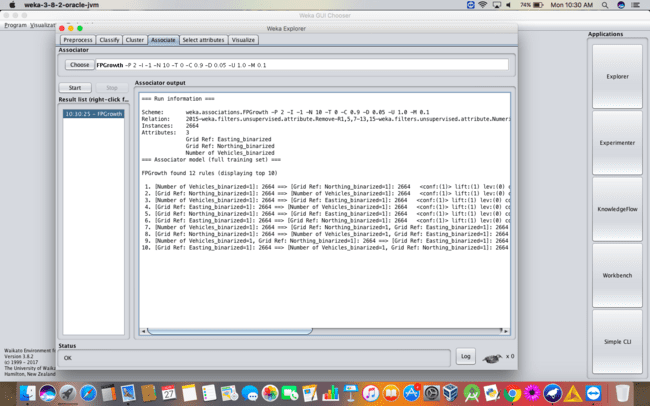
**STEPS INVOLVED :**

* 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.

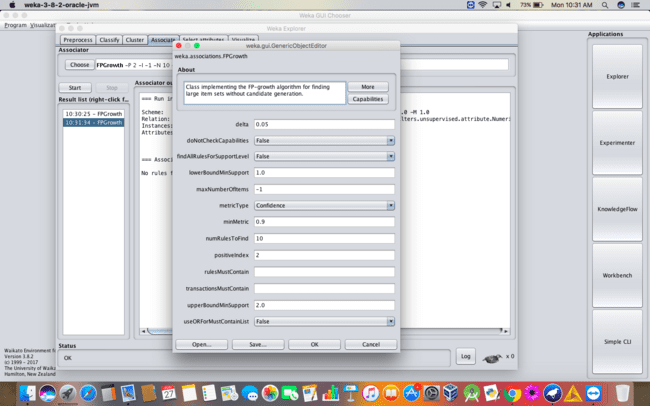
**OBSERVATIONS :**

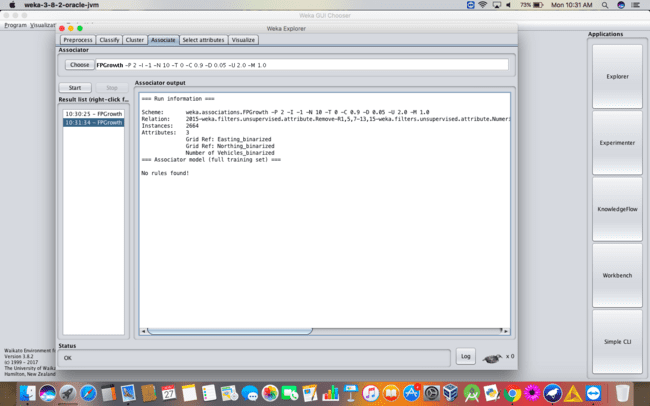
1. When the association rules are of values:
2. Upper bound min\_sup = 1.0
3. Lower bound min\_sup = 0.1
4. Metric type = confidence.

****

****

1. When the association rules are of values:
2. Upper bound min\_sup = 2.0
3. Lower bound min\_sup = 1.0
4. 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)](file:///G:\Data%20Mining\ML_MSIT\projects\datasets\credit-g.arff) Excel [spreadsheet](file:///G:\Data%20Mining\ML_MSIT\projects\datasets\credit-g2.xls) 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.**

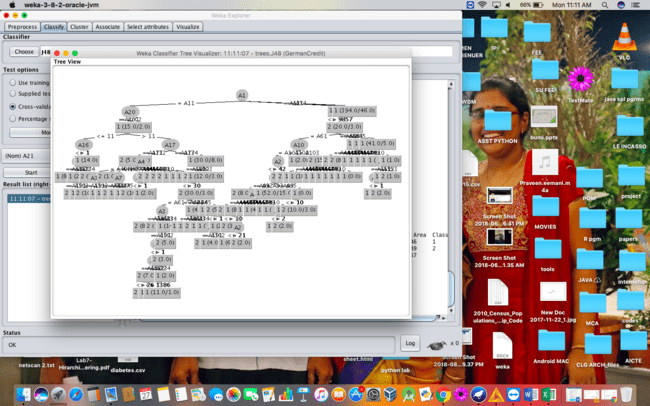
The attributes that might be crucial in making the credit assessment are :

* Numerical attributes
* Nominal attributes
* Nominal and numeric attributes are the capabilities of the given dataset.

1. **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.**

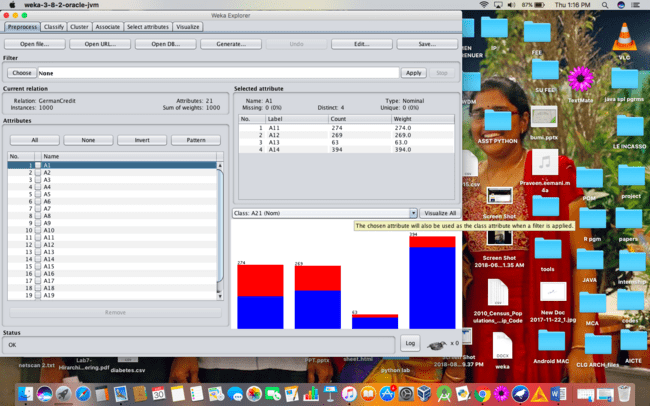
* **Decision Tree :**

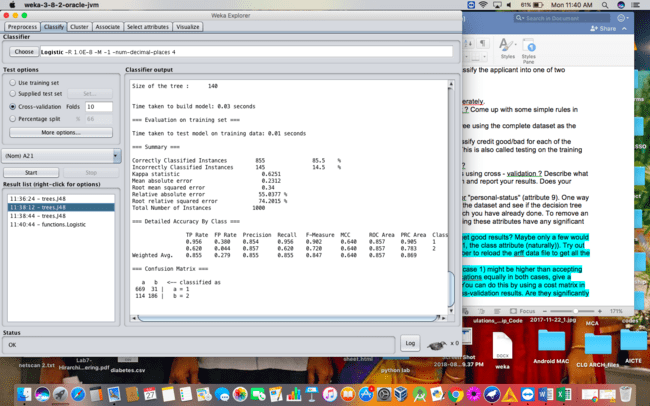
Visualize the decision tree for the given dataset.



1. **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.





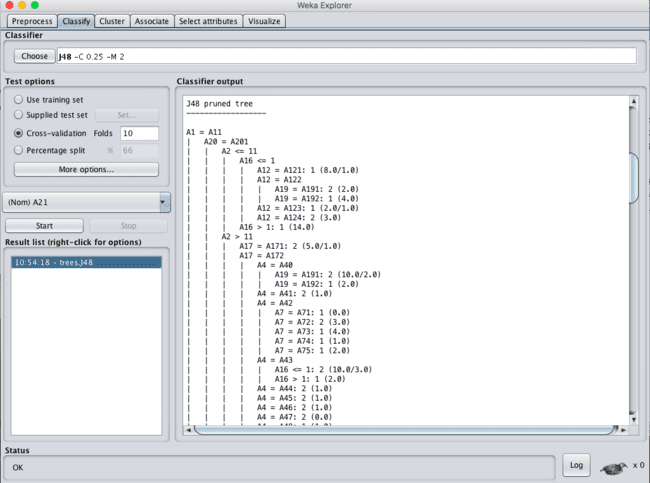


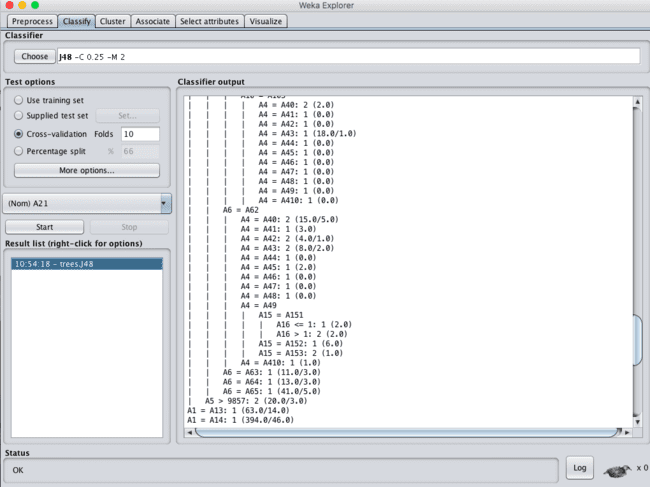
1. **Is testing on the training set as you did above a good idea ? Why or Why not ?**

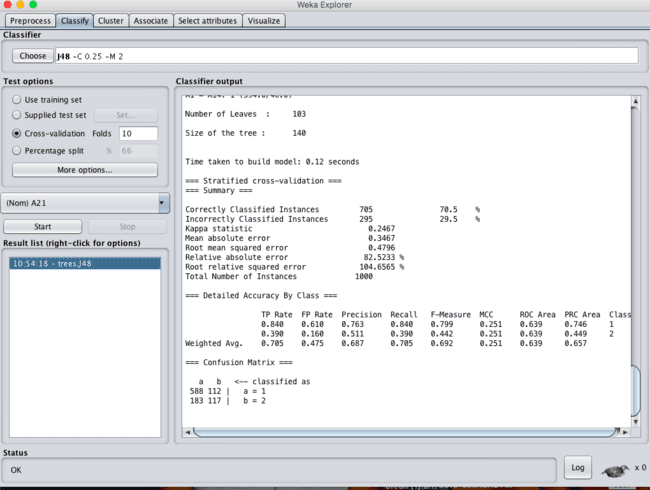
Testing on the training dataset is a good idea where it helps to increase the accuracy of your model by decreasing its complexity. In this case of the dataset, you can prune tree after training. This will decrease the amount of specification in the specific training dataset and increase generalisation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

1. **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 ?**

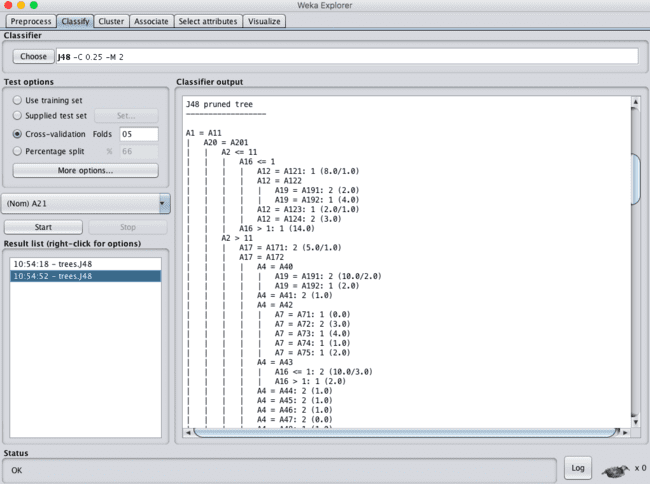
* **CROSS VALIDATION ANALYSIS :**
* **When cross validation folds are 10 :**

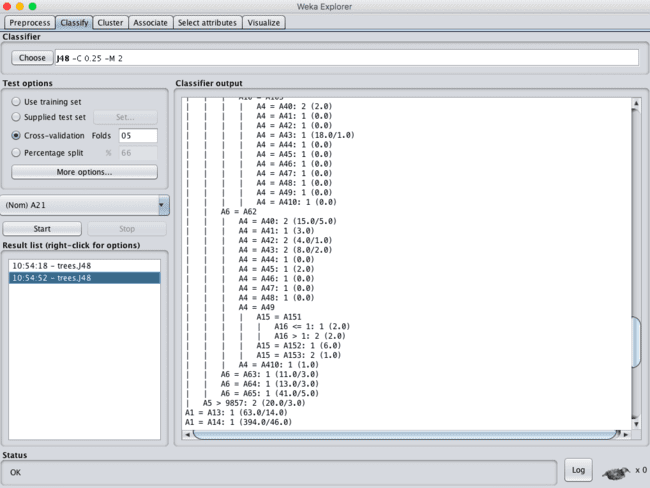


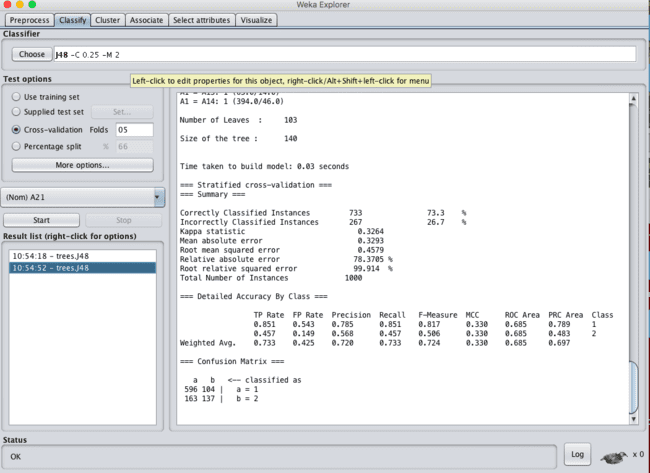




* **When cross validation folds are : 05 :-**

****





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 :
5. Mean Absolute Error
6. Root Mean Square Error
7. Relative Absolute Error
8. Total Number of Instances

**DESCRIPTION :**

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

**ANALYSIS :**

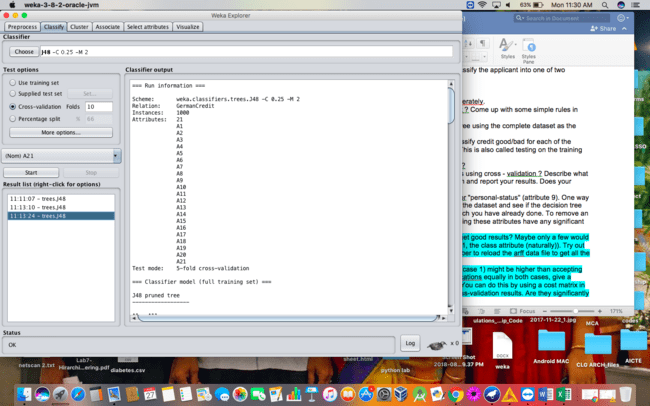
1. **Do you really need to input so many attributes to get good results?**

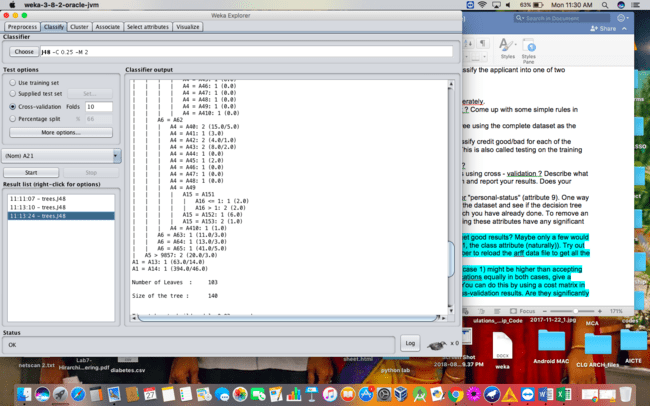
Yes, Having many attributes as the input leads to the good results regarding the dataset. Having more attributes leads to have more and different types of the evaluations.

1. **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.

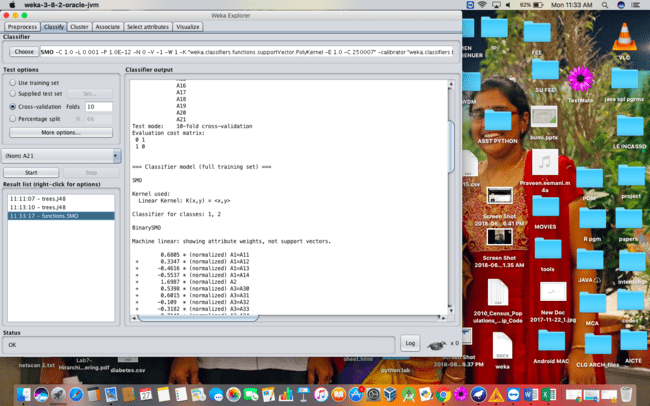


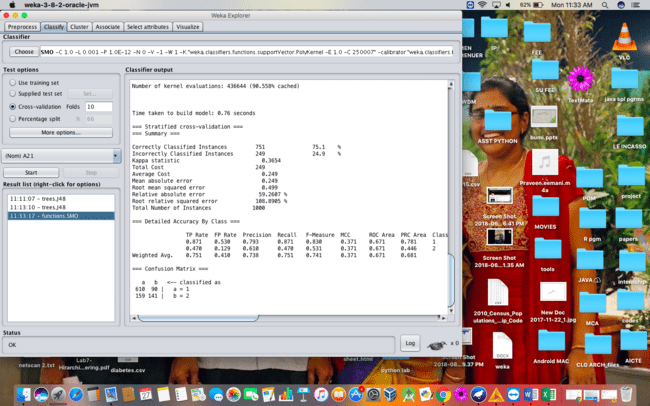




* **SMO ALGORITHM:**

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.





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.

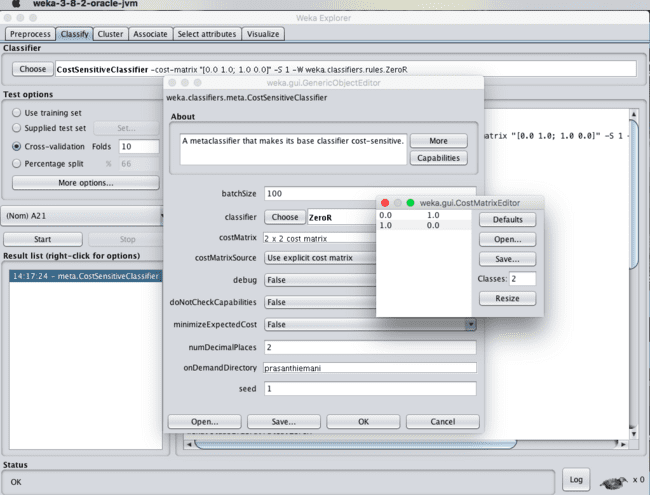
1. **Set the cost sensitive evaluation and compare the obtained results.**

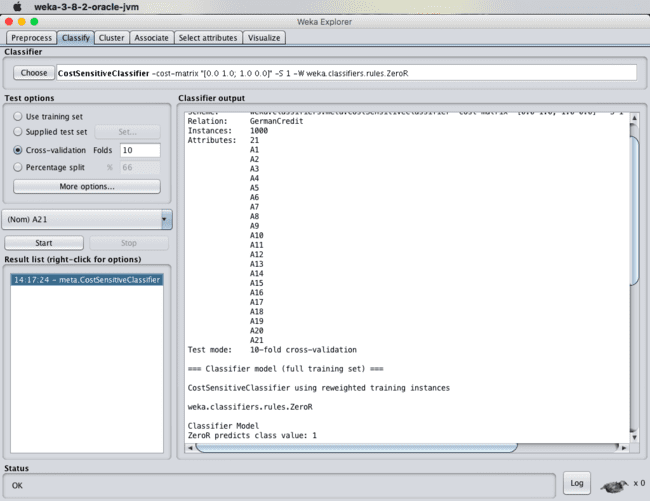
Cost-Sensitive Learning is a type of learning in data mining that takes the misclassification costs (and possibly other types of cost) into consideration. The goal of this type of learning is to minimize the total cost. The key difference between cost-sensitive learning and cost-insensitive learning 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.

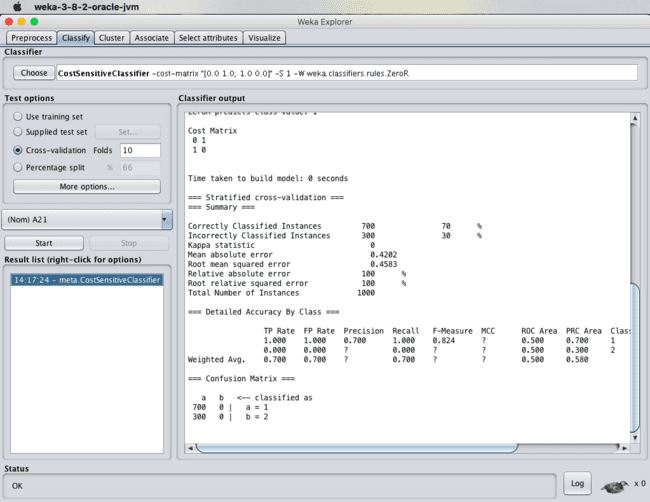
**STEPS :**

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

**ANALYSIS :**







1. **What is the significance of the following parameters :**
2. **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").

1. **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.

1. **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.

1. **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).

1. Logistic Regression
2. Naïve Bayes Algorithm
3. J48
4. K-Nearest Neighbor
5. SMO Algorithm

**DESCRIPTION :**

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

**ANALYSIS :**

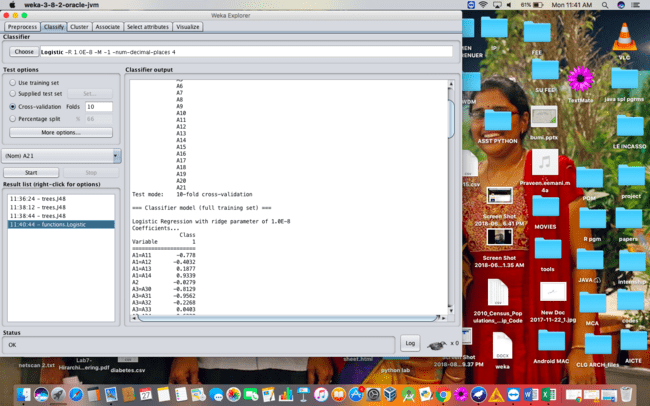
1. **Logistic Regression :**

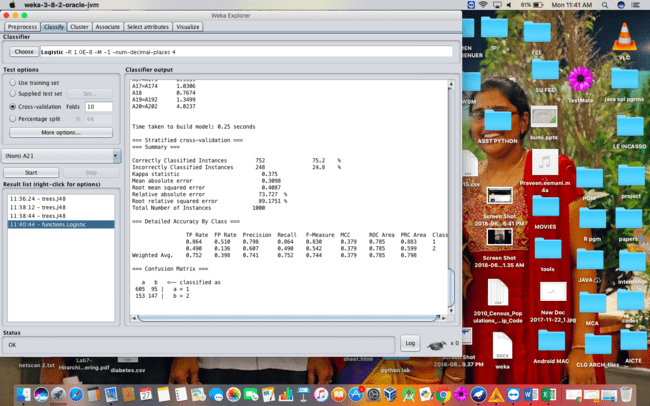
Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical).

**Steps :**

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

**Output :**





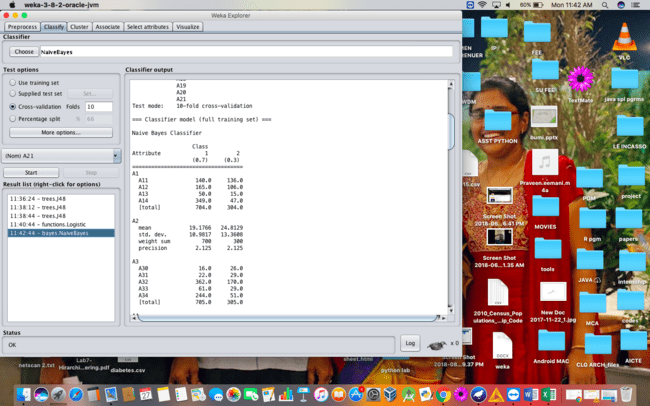
1. **Naïve Bayes Algorithm :**

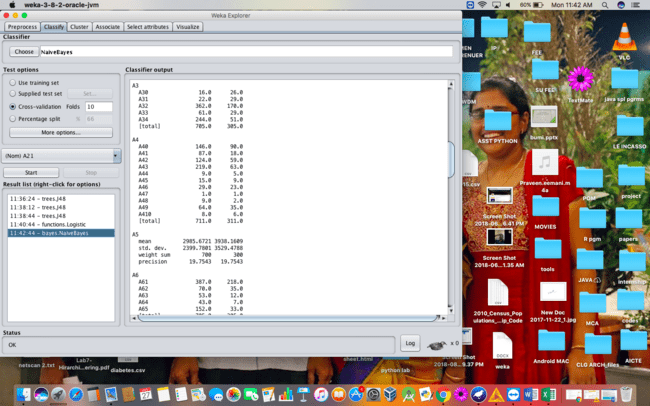
The Naive Bayesian classifier is based on Bayes’ theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian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

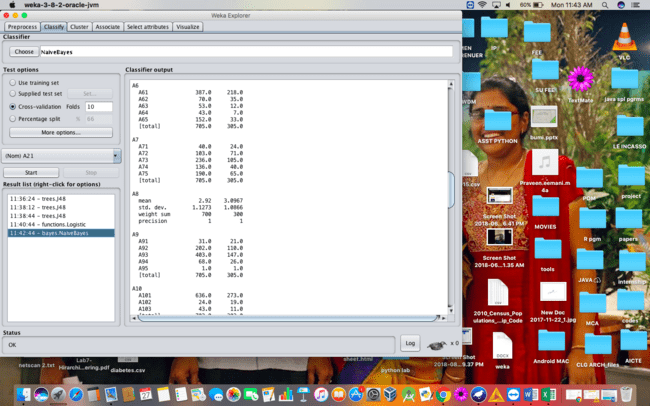
**Steps :**

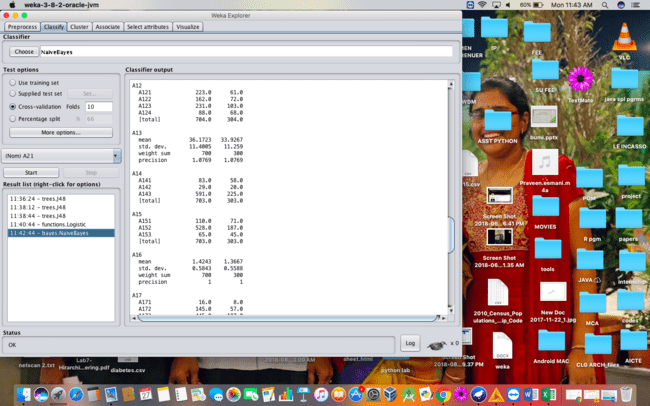
* Load the dataset into the weka tool and preprocess it.
* Apply the classification the Naïve bayes technique and execute for the result.

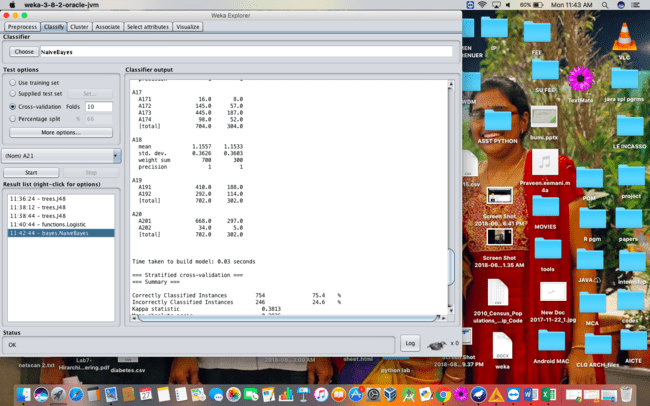
**Output :**

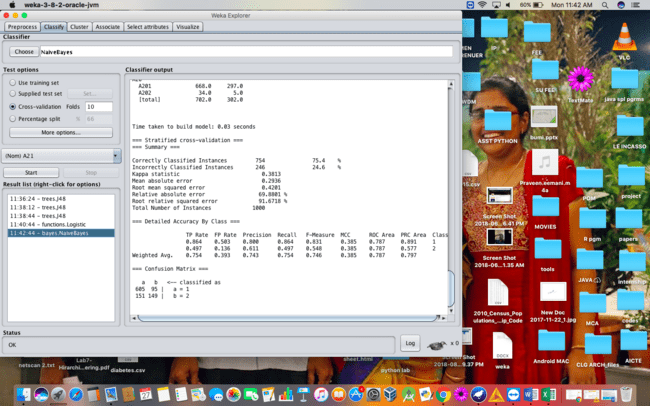
****

****

****

****

****

****

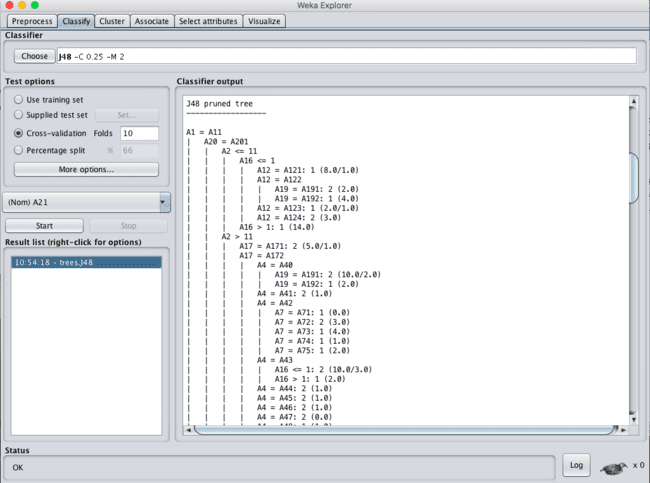
1. **J48 Algorithm :**

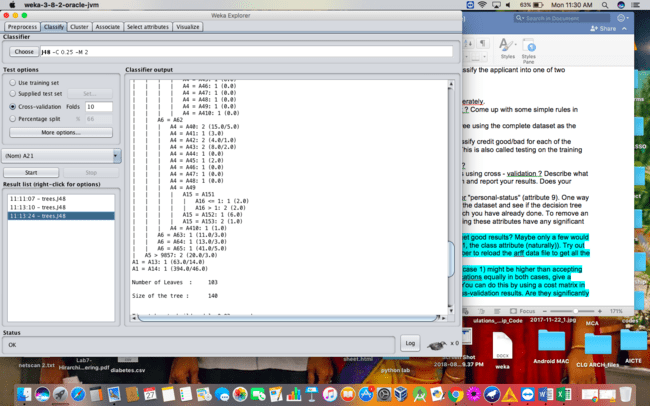
Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also on the bases of the training instances the classes for the newly generated instances 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 :**

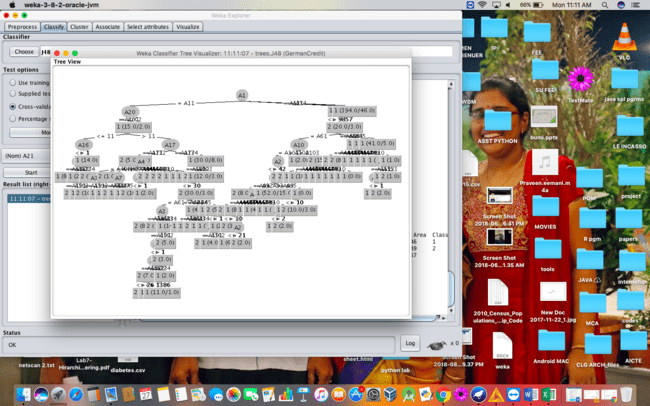
* Load the dataset into the weka tool and preprocess it.
* Apply the classification the J48 technique and execute for the result.

**Output :**









1. **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.

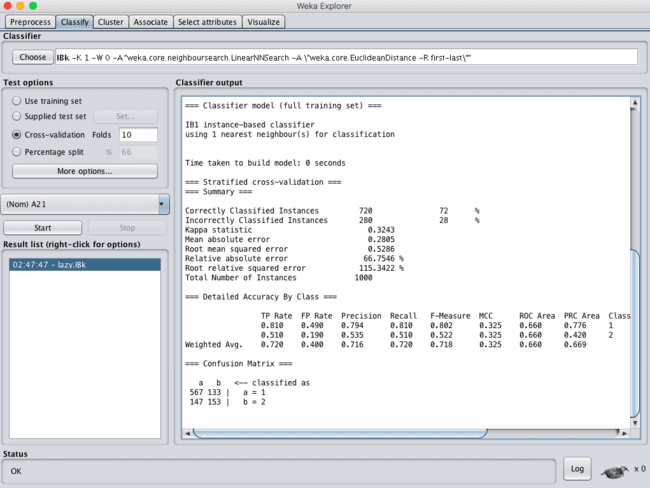
It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

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 :**

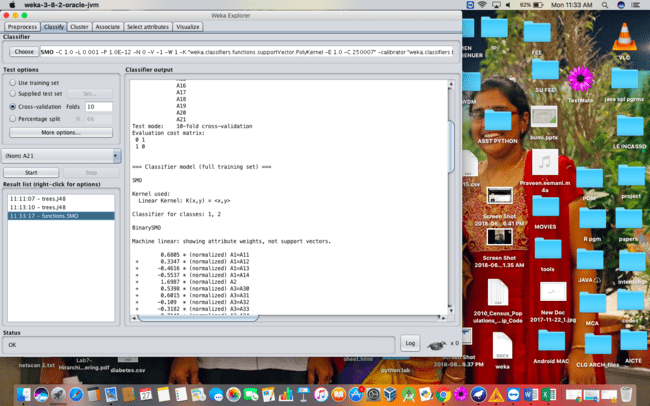


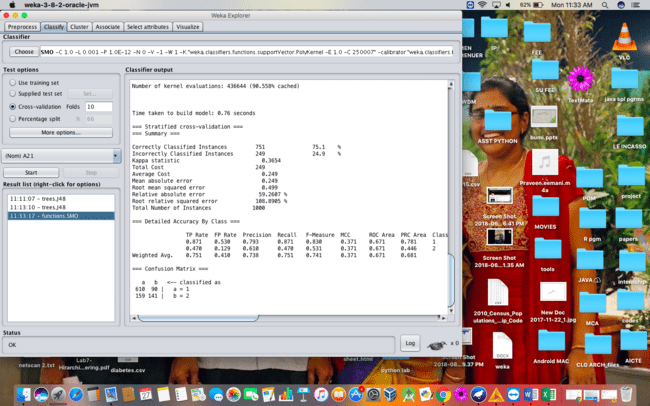
1. **SMO Algorithm :**

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.

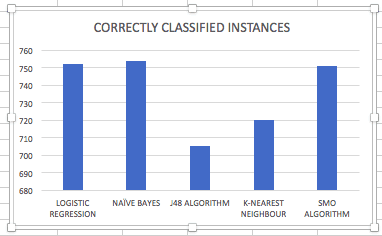
**Steps :**

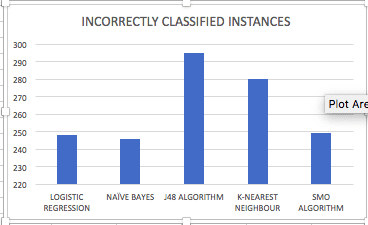
* Load the dataset into the weka tool and preprocess it.
* Apply the classification the Sequential Minimal Optimization (SMO)technique and execute for the result.



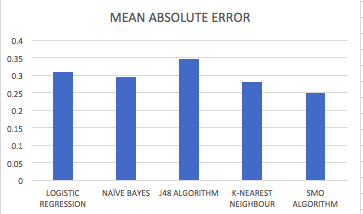


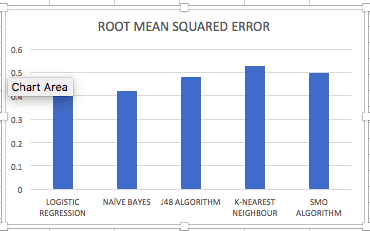
**COMPARISION OF VALUES :**

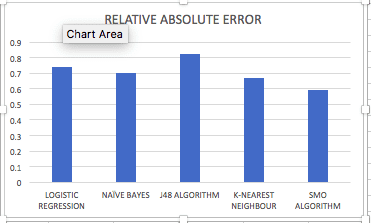
****

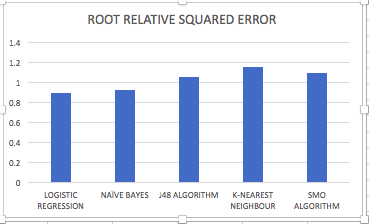
****

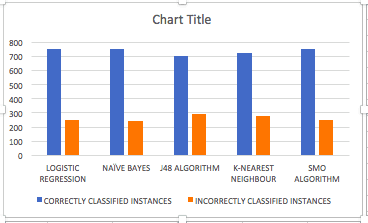
****

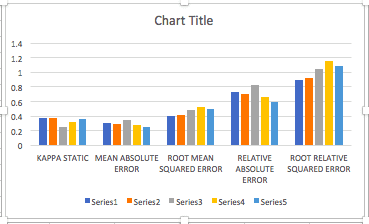
****

****

****

****

****

****

**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.

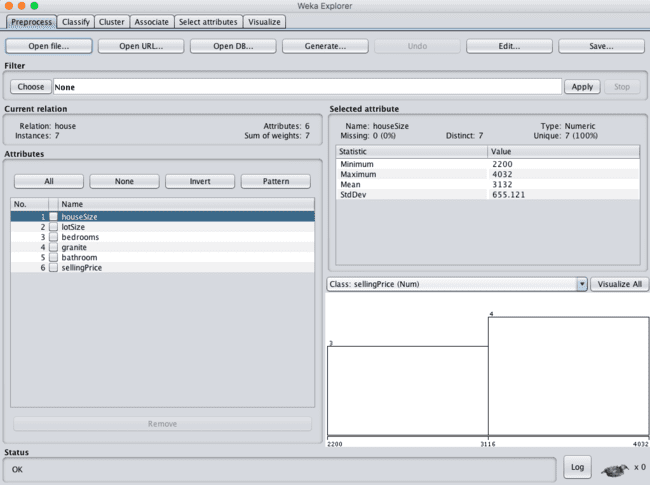
Build the dataset for weka in arff file format and load the dataset into weka and finally create the regression model with weka.

**DESCRIPTION :**

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

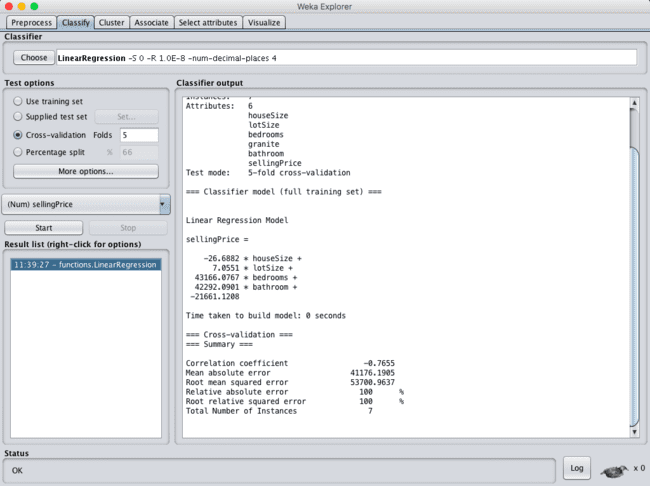
**Steps :**

* 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.

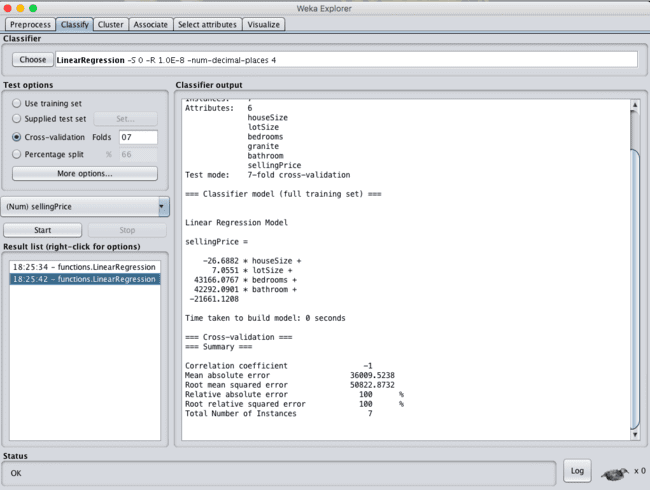
****

**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 :**

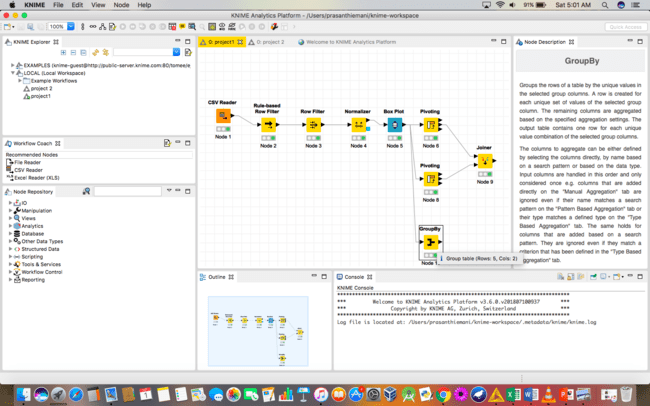
Extract the data from csv file, transform[use row filter and rule based filter ] use pivot and group by] load the data for reporting (Visualization).

**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.

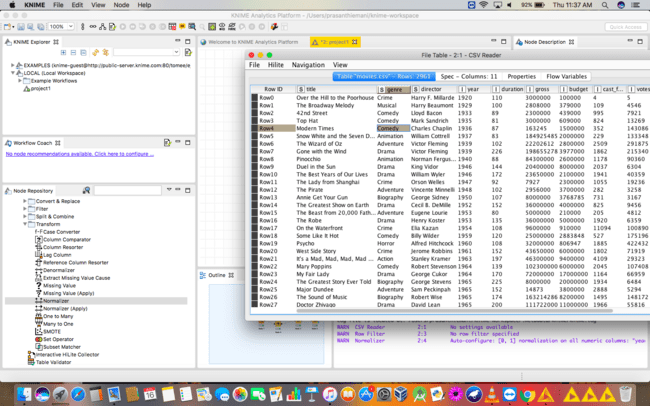
**STEPS :**

* 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 :**

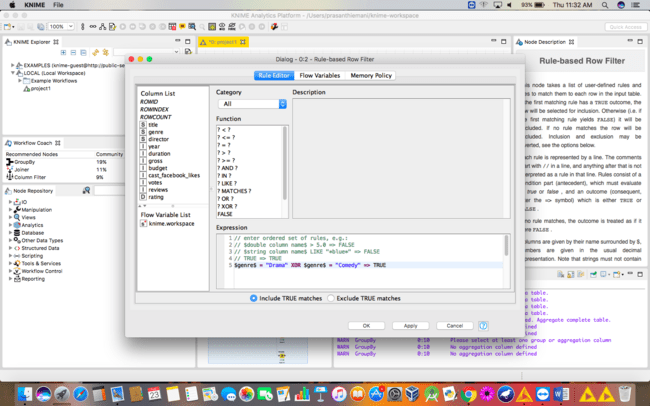
This node takes a list of user-defined rules and tries to match them to each row in the input table. If the first matching rule has a TRUE outcome, the row will be selected for inclusion. Otherwise (i.e. if the first matching rule yields FALSE) it will be excluded. If no rule matches the row will be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

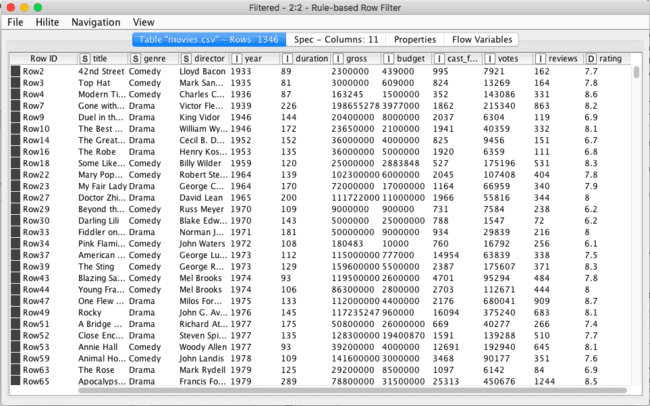
**Steps :**

* 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 Filter :**

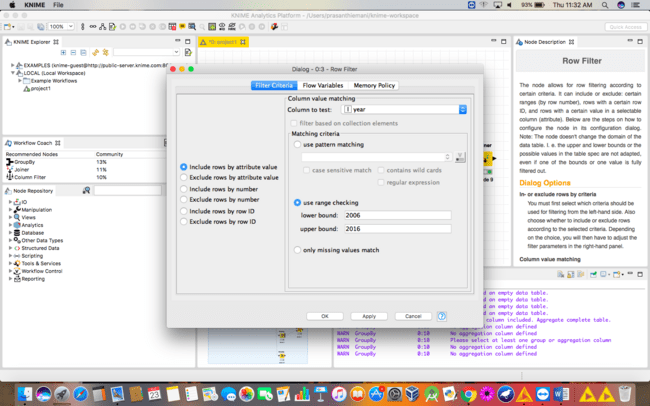
3 matching criteria on data colums: on String by full or partial pattern matching, on numbers by range, on missing values, all of them also on collection columns. 1 matching criterion on row numbers: from row number to row number. 1 matching criterion on RowID: full and partial patterm matching. Partial 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.

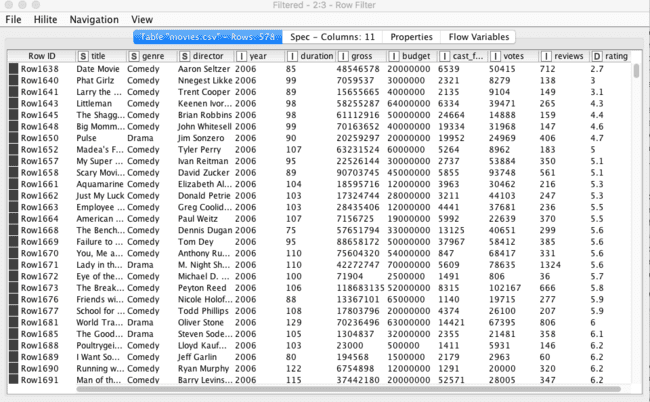
**Steps :**

* 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





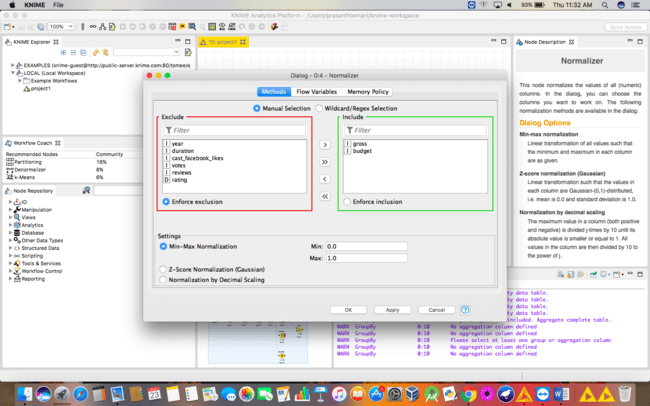
* **Normalizer :**

**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:

1. Gross
2. Budget
3. Min : 0.0
4. Max : 0.1

* Execute the normalizer and check for the values in the table where you will find the normalized values of the table.

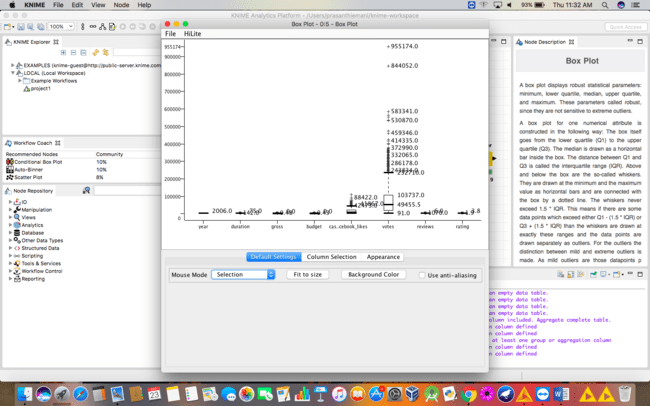


* **Boxplot :**

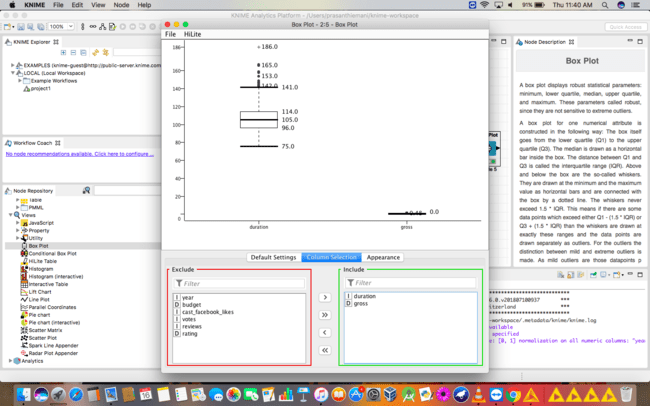
A box plot displays robust statistical parameters: minimum, lower quartile, median, upper quartile, and maximum. A box plot for one numerical attribute is constructed in the following way: The box itself goes from lower quartile (Q1) to upper quartile (Q3). Median is drawn as horizontal bar inside box. 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 :**

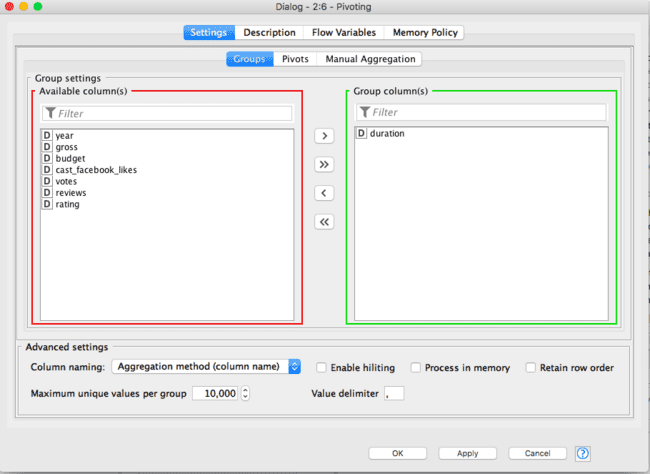
* Connect pivot with boxplot and have the connection between them.
* Configure pivoting with 3 different columns for same data type for :

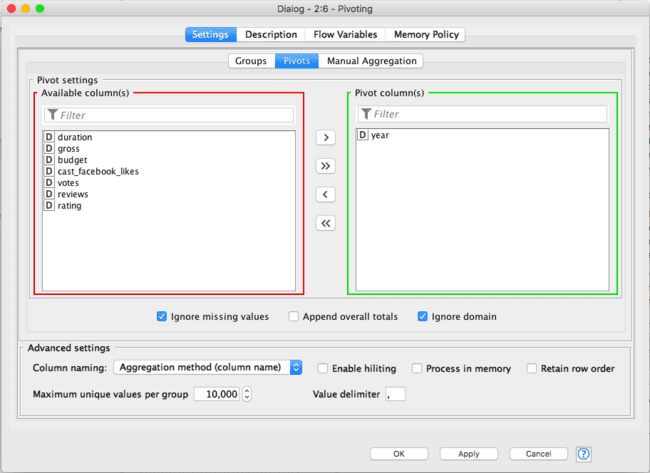
Groups – duration

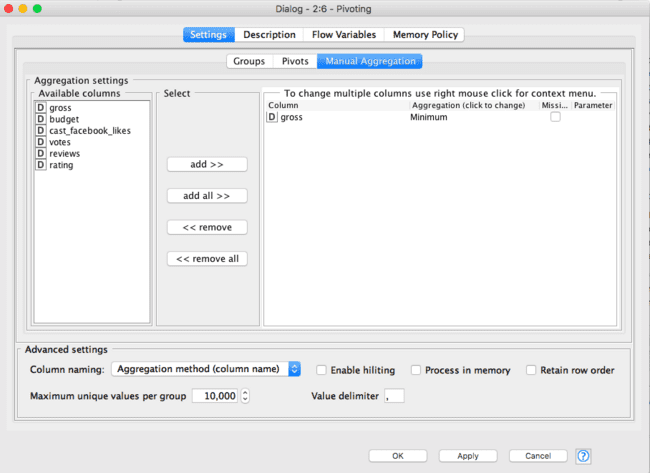
Pivots - year

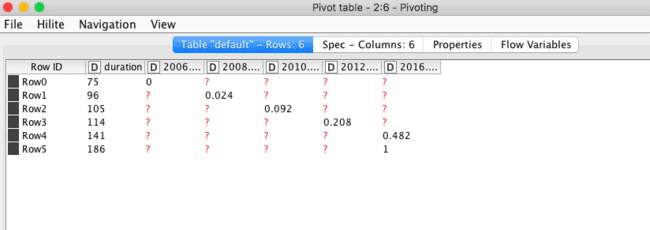
Manual Aggregation – gross

* Execute pivoting and checkout for the changes in the table.









* **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

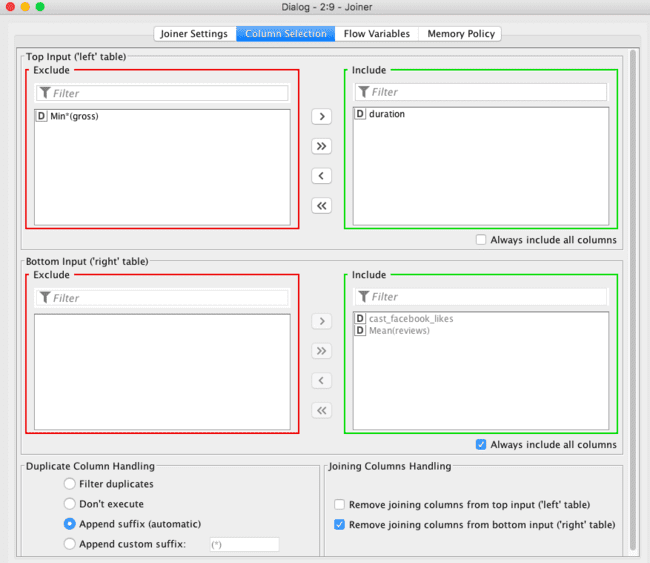
* Bottom Input(right table):

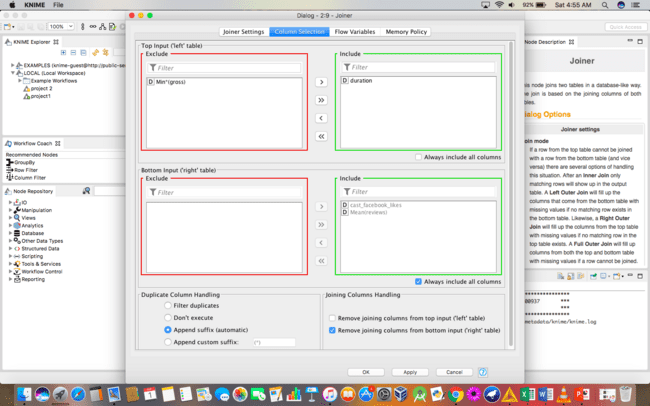
Include :

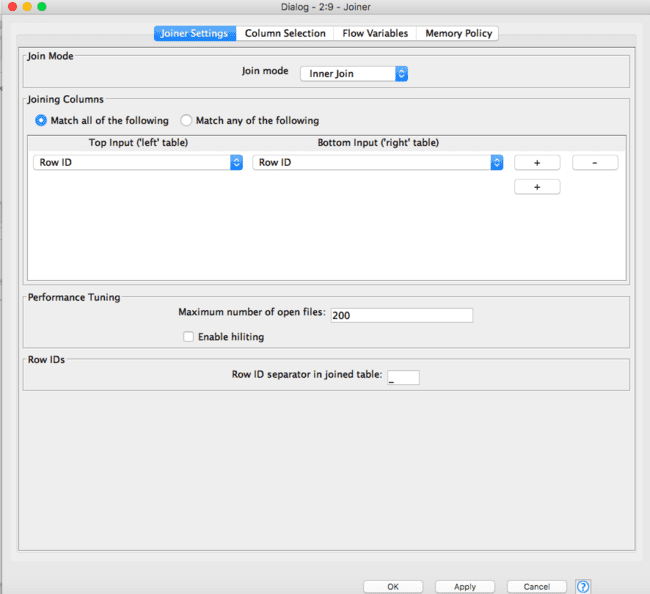
🡪cast\_facebook\_likes

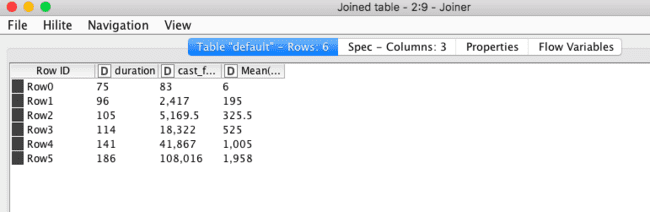
🡪 Mean

* Execute the joiner and check for the final table.







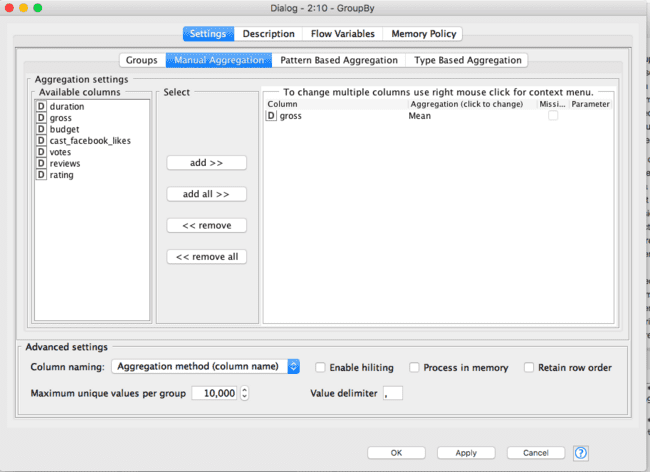


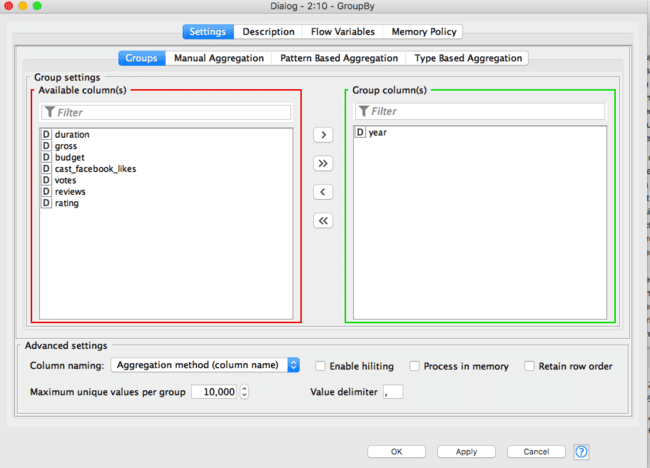
* **Group By :**

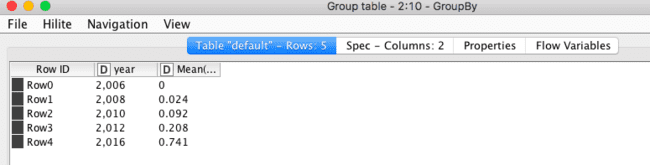
Groups the rows of a table by the unique values in the selected group columns. A row is created for each unique set of values of the selected group column. The remaining columns are aggregated based on the specified aggregation settings. The output table contains one row for each unique value combination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







**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)