Assignment No 1:

<u>Aim</u>:-Build Data Warehouse and Explore WEKA.

<u>Theory</u>:-

(i). Downloading and/or installation of WEKA data

mining toolkit Procedure:

1. Go to the Weka website, http://www.cs.waikato.ac.nz/ml/weka/, and download the software. On the left-hand side, click on the link that says download.

2. Select the appropriate link corresponding to the version of the software based on your operating system and whether or not you already have Java VM running on your machine.

3. The link will forward you to a site where you can download the software from a mirror site. Save the self-extracting executable to disk and then double click on it to install Weka. Answer yes or next to the questions during the installation.

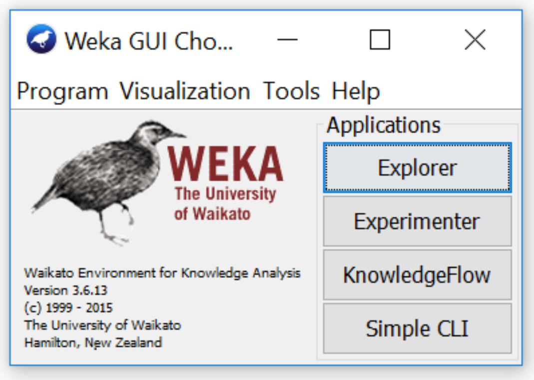
4. Click yes to accept the Java agreement if necessary. After you install the program Weka should appear on your start menu under Programs (if you are using Windows).

5. Running Weka from the start menu select Programs, then Weka. You will see the Weka GUI Chooser. Select Explorer. The Weka Explorer will then launch.

(ii). Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.

The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting point for launching Weka‘s main GUI applications and supporting tools. If one prefers a MDI (—multiple document interface‖) appearance, then this is provided by an alternative launcher called —Main‖ (class weka.gui.Main).

The GUI Chooser consists of four buttons—one for each of the four major Weka applications— and four menus.



The buttons can be used to start the following applications:

Explorer- An environment for exploring data with WEKA

a) Click on –explorer‖ button to bring up the explorer window.
b) Make sure the –preprocess‖ tab is highlighted.
c) Open a new file by clicking on –Open New file‖ and choosing a file with –.arff‖extension from the –Data‖ directory.
d) Attributes appear in the window below.
e) Click on the attributes to see the visualization on the right.
f) Click –visualize all‖ to see them all

Experimenter- An environment for performing experiments and conducting statistical tests between learning schemes.

a) Experimenter is for comparing results.
b) Under the –set up‖ tab click –New‖.
c) Click on –Add New‖ under –Data‖ frame. Choose a couple of arff format files from –Data‖ directory one at a time.
d) Click on –Add New‖ under –Algorithm‖ frame. Choose several algorithms, one at a time by clicking –OK‖ in the window and –Add New‖.
e) Under the –Run‖ tab click –Start‖.
f) Wait for WEKA to finish.
g) Under –Analyses‖ tab click on –Experiment‖ to see results.

Knowledge Flow- This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

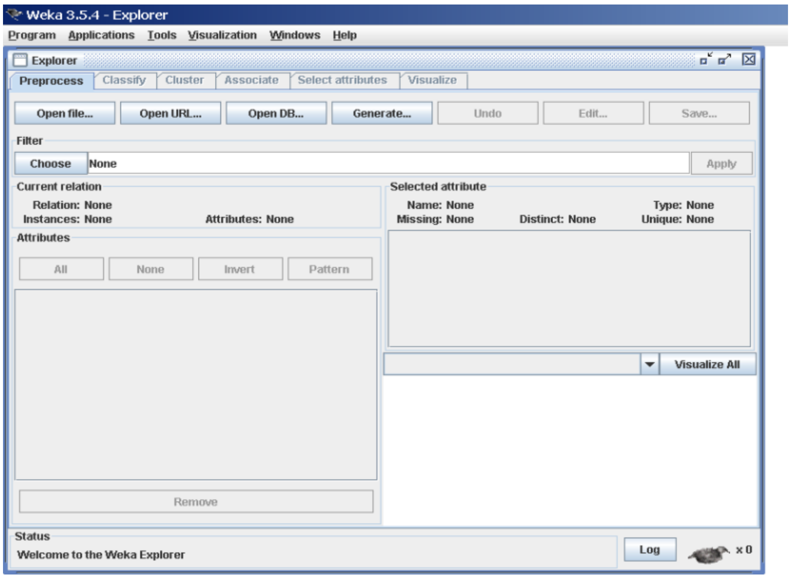
Simple CLI - Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

(iii). Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, classify panel, Cluster panel, Associate panel and Visualize panel)

When the Explorer is first started only the first tab is active; the others are greyed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.
The tabs are as follows:
1. Preprocess. Choose and modify the data being acted on.
2. Classify. Train and test learning schemes that classify or perform regression.
3. Cluster. Learn clusters for the data.
4. Associate. Learn association rules for the data.
5. Select attributes. Select the most relevant attributes in the data.
6. Visualize. View an interactive 2D plot of the data.

Once the tabs are active, clicking on them flicks between different screens, on which the respective actions can be performed. The bottom area of the window (including the status box, the log button, and the Weka bird) stays visible regardless of which section you are in.

1. Preprocessing



Loading Data:

The first four buttons at the top of the preprocess section enable you to load data into WEKA:

1. Open file.... Brings up a dialog box allowing you to browse for the datafile on the local file system.

2. Open URL ..... Asks for a Uniform Resource Locator address for where the data is stored.

3. Open DB..... Reads data from a database. (Note that to make this work you might

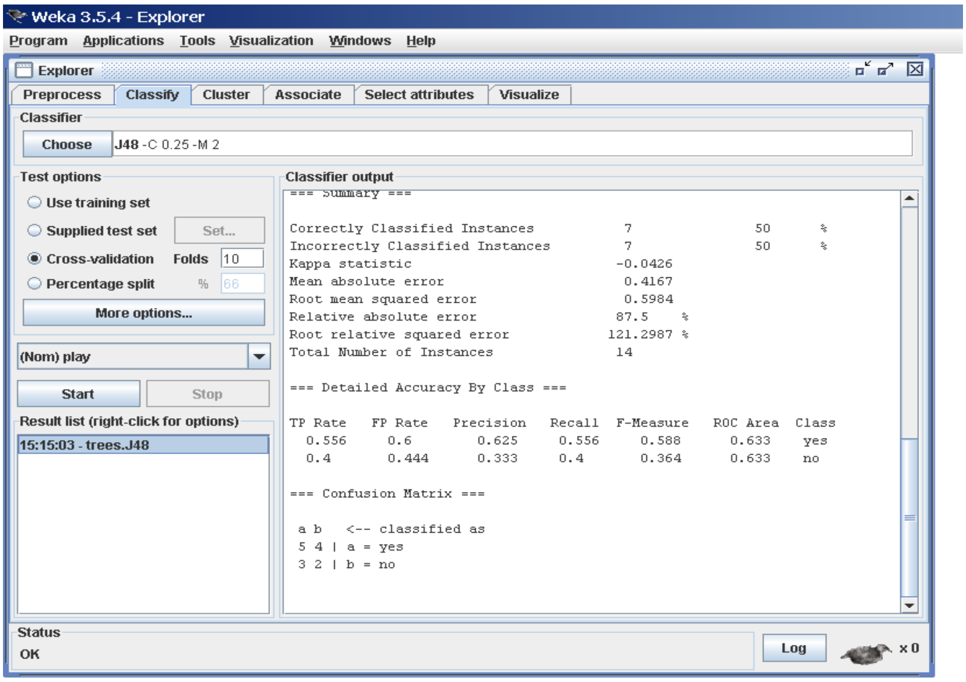
have to edit
the file in weka/experiment/DatabaseUtils.props.)

4. Generate. .. Enables you to generate artificial data from a variety of Data Generators.

Using the Open file.... button you can read files in a variety of formats:

WEKA's ARFF format, CSV format, C4.5 format, or serialized Instances format. ARFF files typically have a .arff extension, CSV files a .csv extension,C4.5 files a .data and .names extension, and serialized Instances objects a .bsi extension.

2. Classification:



Selecting a Classifier

At the top of the classify section is the Classifier box. This box has a text field that gives the name of the currently selected classifier, and its options. Clicking on the text box with the left mouse button brings up a Generic Object Editor dialog box, just the same as for filters, that you can use to configure the options of the current classifier. With a right click (or Alt+Shift+left click) you can once again copy the setup string to the clipboard or display the properties in a Generic Object Editor dialog box. The Choose button allows you to choose on4eof the classifiers that are available in WEKA.

Test Options:

The result of applying the chosen classifier will be tested according to the options that are set by clicking in the Test options box. There are four test modes:

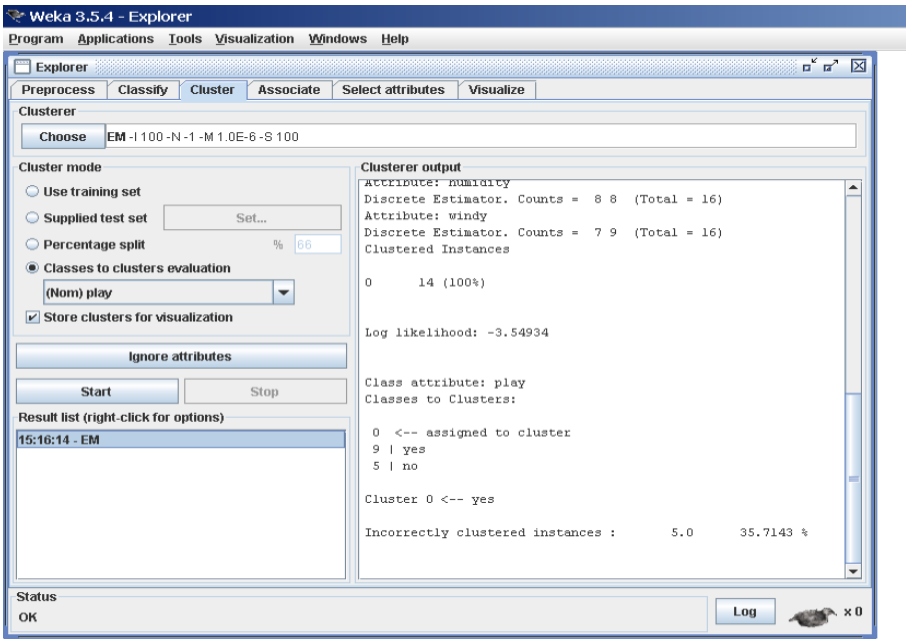
1. **Use training set:** The classifier is evaluated on how well it predicts the class of the instances it was trained on.

2. **Supplied test set:** The classifier is evaluated on how well it predicts the class of a set of instances loaded from a file. Clicking the Set... button brings up a dialog allowing you to choose the file to test on.

3. **Cross-validation:** The classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field.

4. **Percentage split:** The classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the % field.

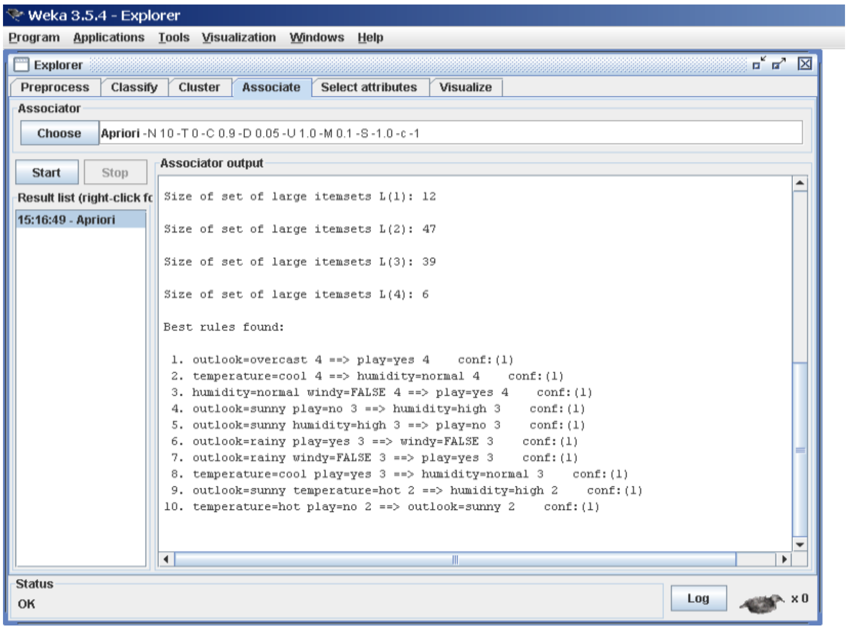
3. Clustering:



Cluster Modes:

The Cluster mode box is used to choose what to cluster and how to evaluate the results. The first three options are the same as for classification: Use training set, Supplied test set and Percentage split.

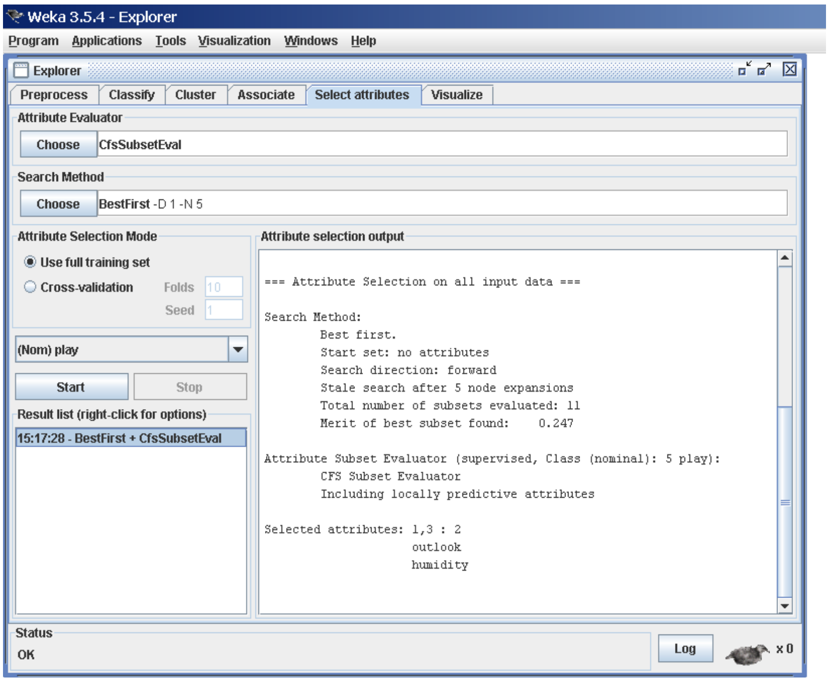
4. Associating:



Setting Up

This panel contains schemes for learning association rules, and the learners are chosen and configured in the same way as the clusters, filters, and classifiers in the other panels.

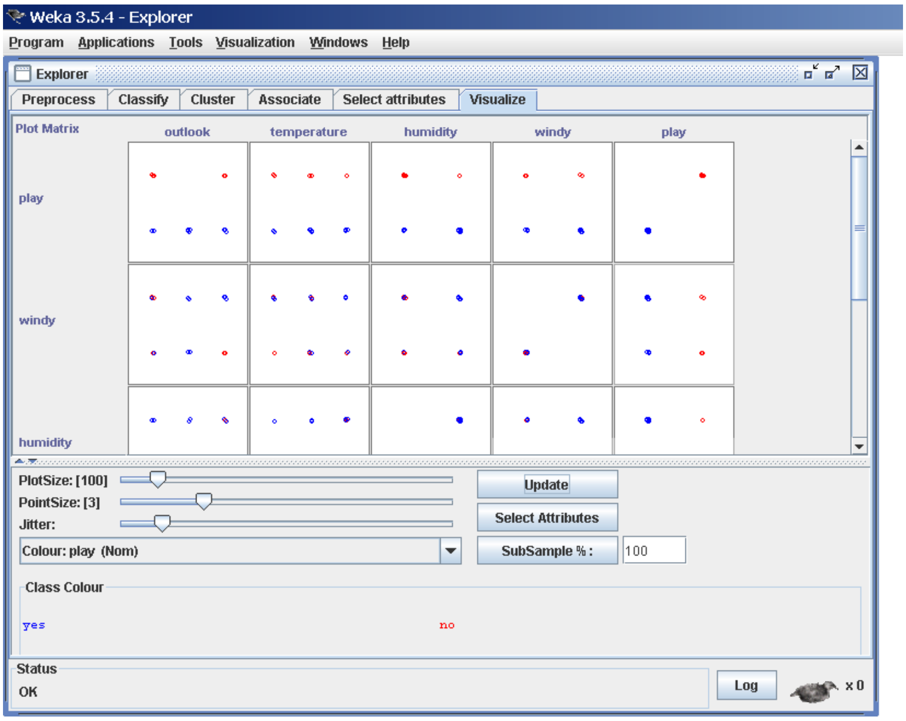
3. Selecting Attributes:



Searching and Evaluating

Attribute selection involves searching through all possible combinations of attributes in the data to find which subset of attributes works best for prediction. To do this, two objects must be set up: an attribute evaluator and a search method. The evaluator determines what method is used to assign a worth to each subset of attributes. The search method determines what style of search is performed.

3. Visualizing:



WEKA's visualization section allows you to visualize 2D plots of the current relation.

<u>Conclusion</u>:-

Hence we have successfully built Data Warehouse and explored WEKA.

Assignment No 2:

<u>Aim</u>:-Perform data pre-processing tasks and Demonstrate performing association rule mining on data sets.

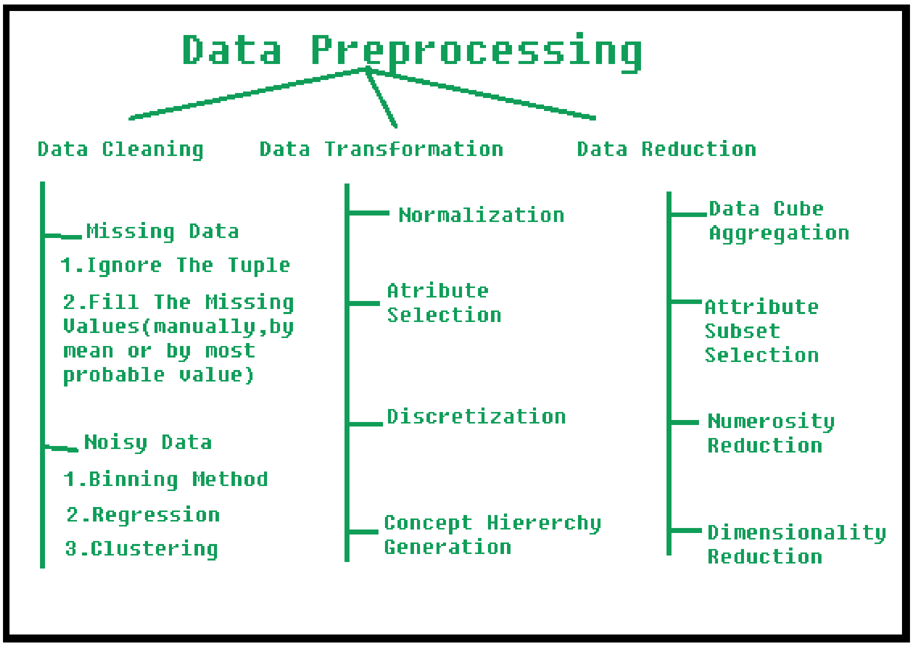
<u>Theory</u>:-

What Is Data Preprocessing?

Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning.

Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn't have a regular, uniform design.

Machines like to process nice and tidy information – they read data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

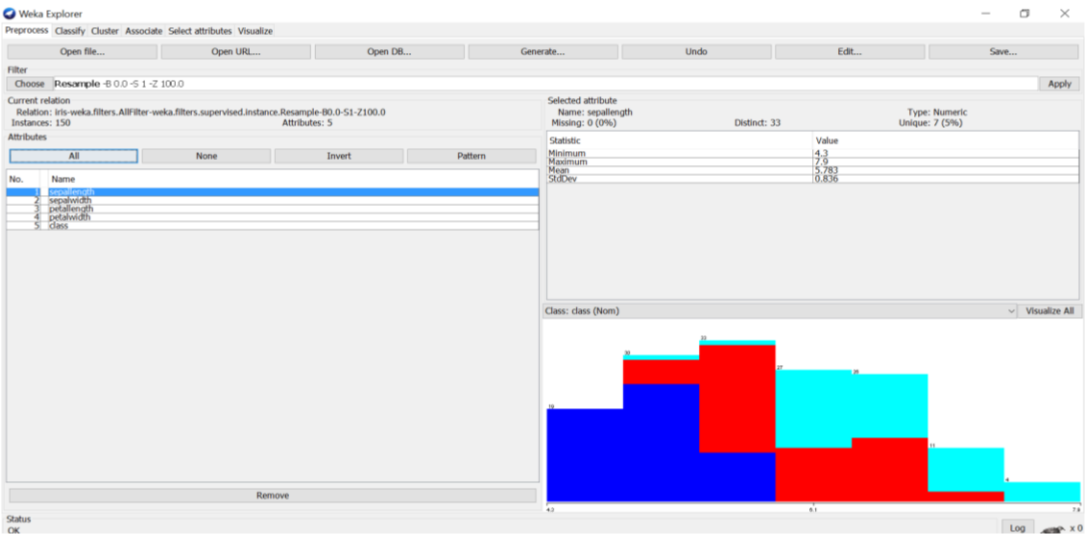


Association rule mining:

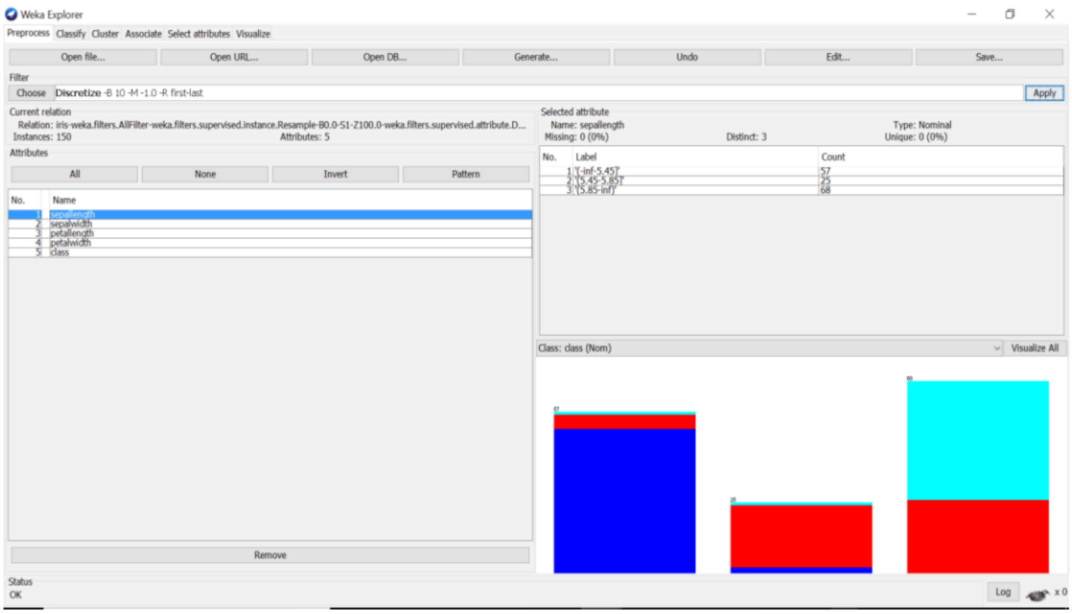
Association rule mining is a machine-learning method used to discover interesting relations between variables in large databases.It looks for patterns in the data and can be performed using various methods, such as the Apriori algorithm. It is intended to identify strong rules discovered using some measures of interestingness.

Procedure:-

For pre-processing the data after selecting the dataset (IRIS.arff).
Select Filter option & apply the resample filter & see the below results.



Select another filter option & apply the discretization filter, see the below results



Likewise, we can apply different filters for preprocessing the data & see the results in different dimensions.

Conclusion:- Hence we have performed data pre-processing tasks and demonstrated / performed association rule mining on data sets.

Assignment No 3:

Aim:-Demonstration of classification rule process on WEKA data-set using Naive Bayes algorithm.

Theory:-

Naïve Bayes Classifier Algorithm:

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes?

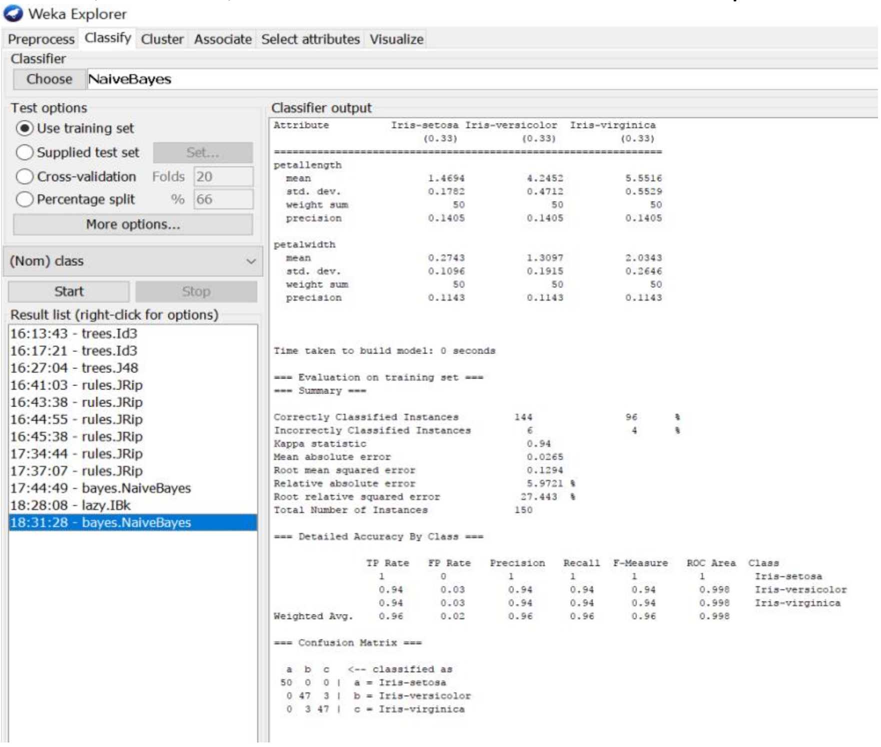
The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Procedure:-

1. Load the dataset (Iris-2D. arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under bayes section.
3. In which we selected Naïve-Bayes algorithm & click on start option with –use training set‖test option enabled.
4. Then we will get detailed accuracy by class consists of F-measure, TP rate, FP rate, Precision, Recall values& Confusion Matrix as represented below.



Conclusion:- Hence we have successfully demonstrated the classification rule process on WEKA data-set using Naive Bayes algorithm.

Assignment No 4:

Aim:-Implementation of OLAP operations.

Theory:-

OLAP Operations:

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations

- Roll-up (Drill-up)
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up (Drill-up):

Roll-up performs aggregation on a data cube in any of the following ways

- By climbing up a concept hierarchy for a dimension
- By dimension reduction
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

Drill-down:

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways

- By stepping down a concept hierarchy for a dimension
- By introducing a new dimension.
- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

Slice:

The slice operation selects one particular dimension from a given cube and provides a new sub-cube.

Dice:

Dice selects two or more dimensions from a given cube and provides a new sub-cube.

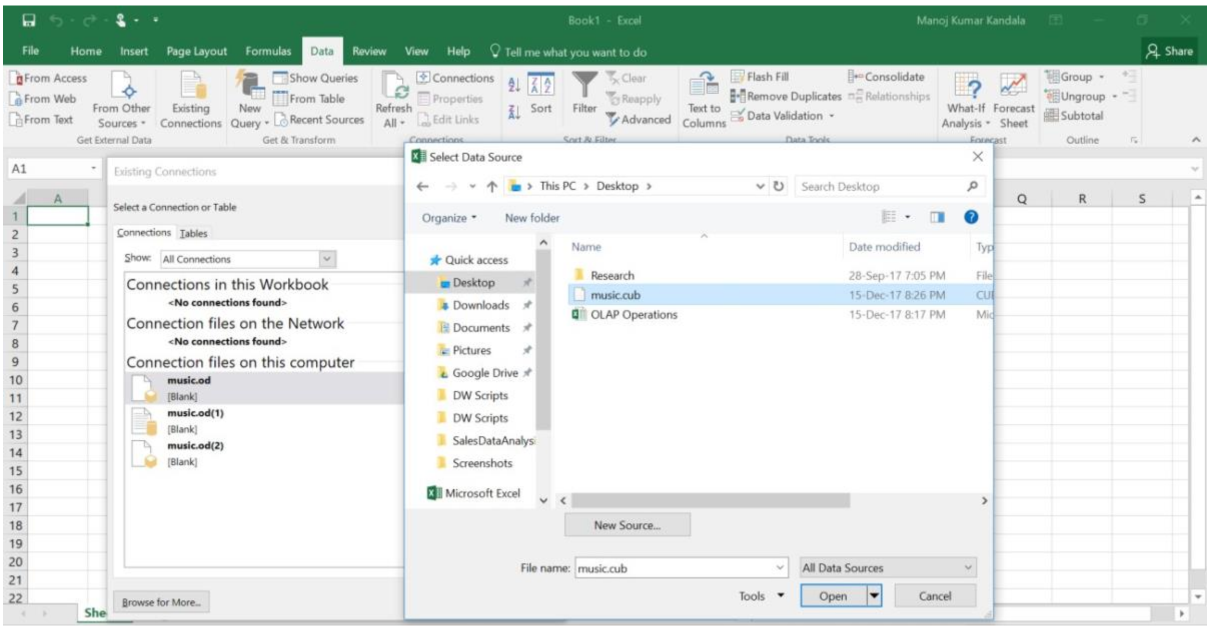
Pivot (rotate):

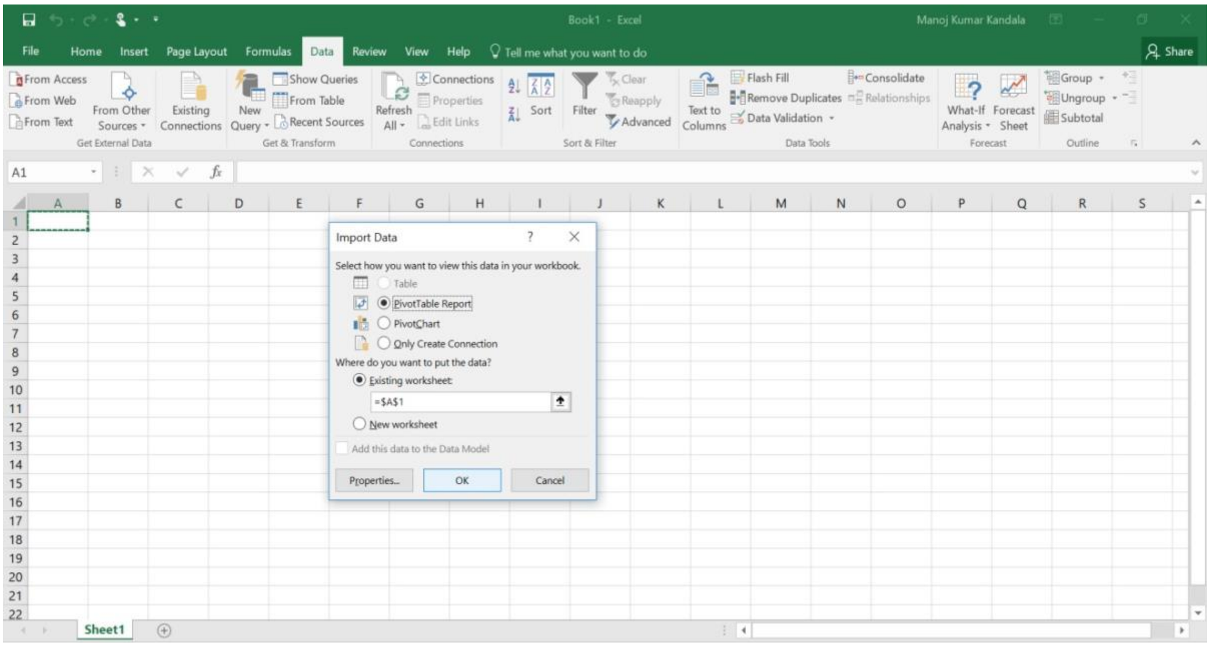
The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data.

Now, we are practically implementing all these OLAP Operations using **Microsoft Excel**.

Procedure for OLAP Operations:

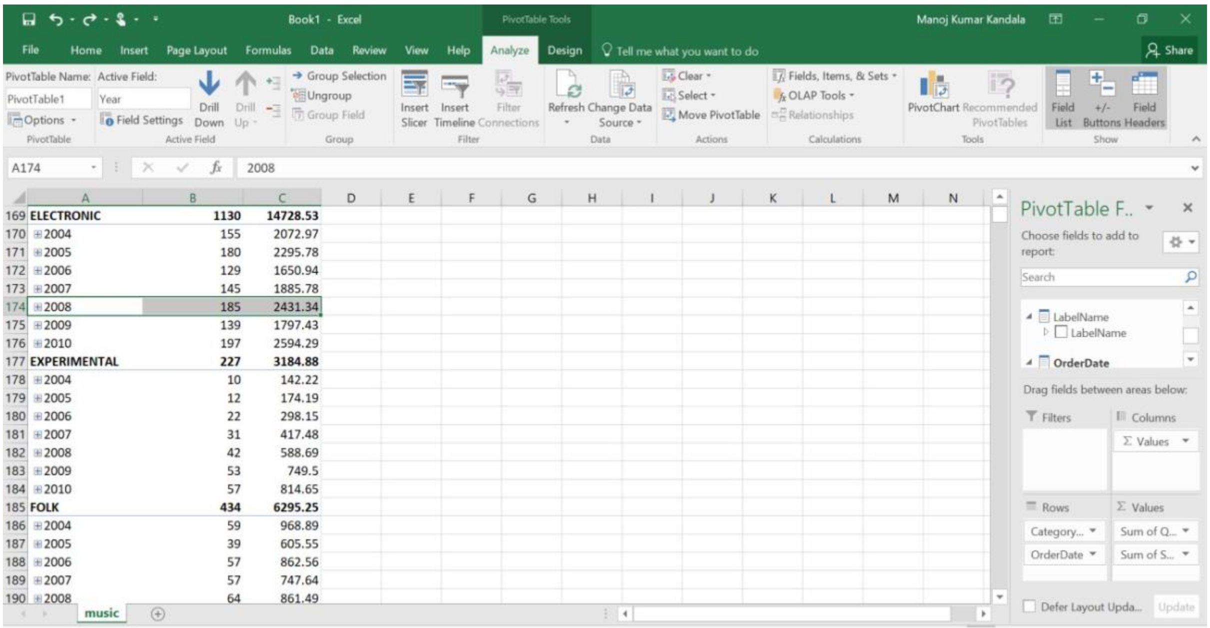
1.Open Microsoft Excel, go to **Data** tab in top & click on –**Existing Connections"**.
2. Existing Connections window will be opened, there **"Browse for more"** option should be clicked for importing **.cub extension file** for performing OLAP Operations. For sample, I took **music.cub** file.



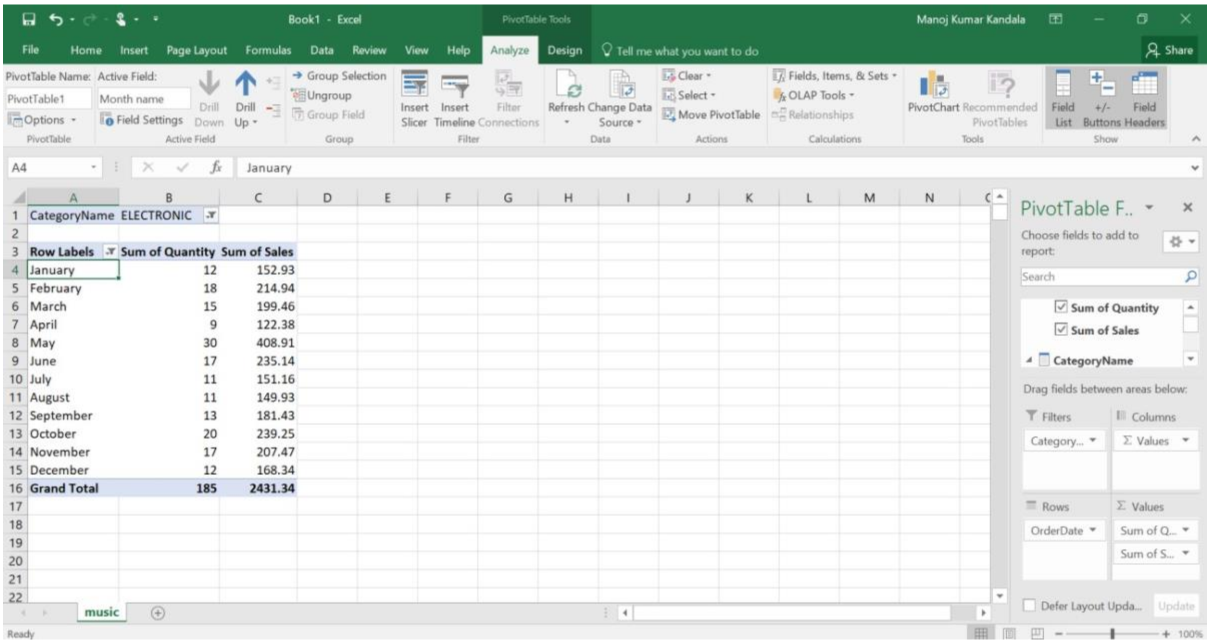


As shown in above window, select –**PivotTable Report"** and click **"OK".**

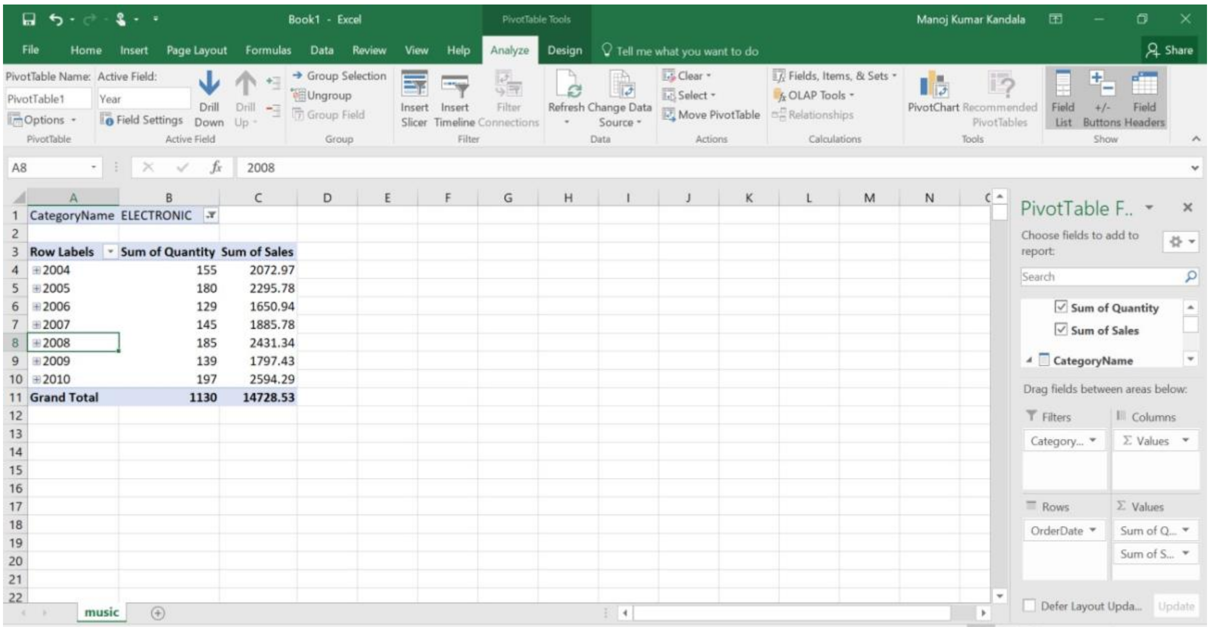
We got all the **music.cub** data for analysing different OLAP Operations. Firstly, we performed **drill-down operation** as shown below.



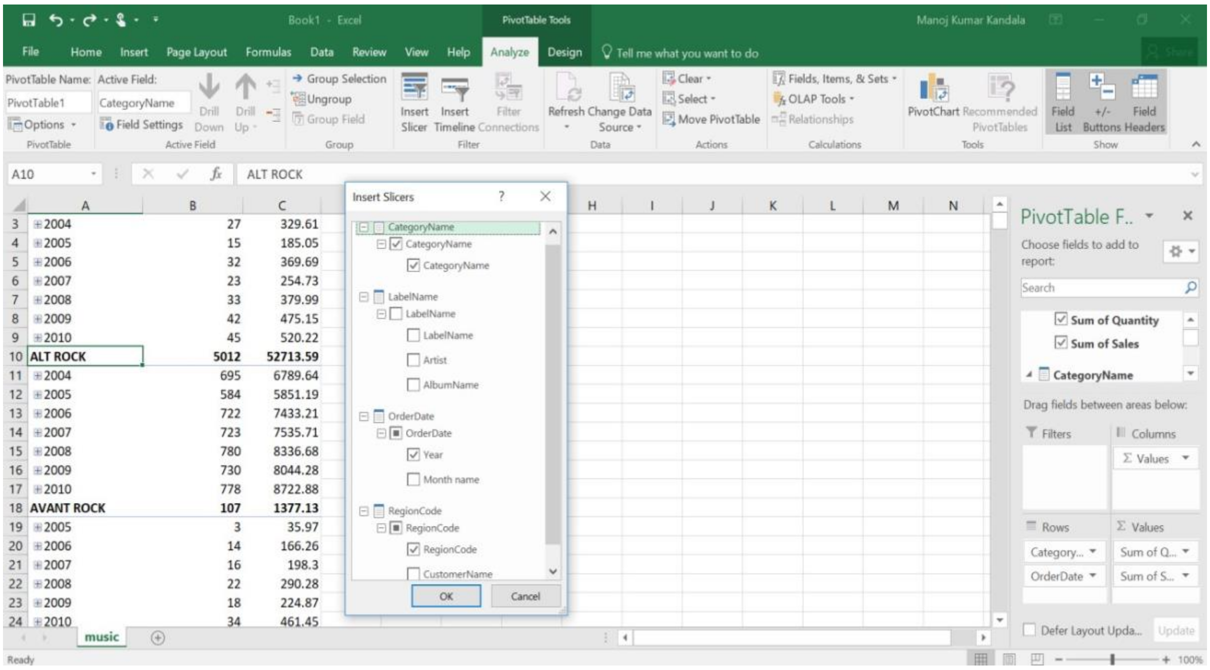
In the above window, we selected year **„2008‟** in **„Electronic‟** Category, then automatically the Drill-Down option is enabled on top navigation options. We will click on **„Drill-Down‟** option, then the below window will be displayed.



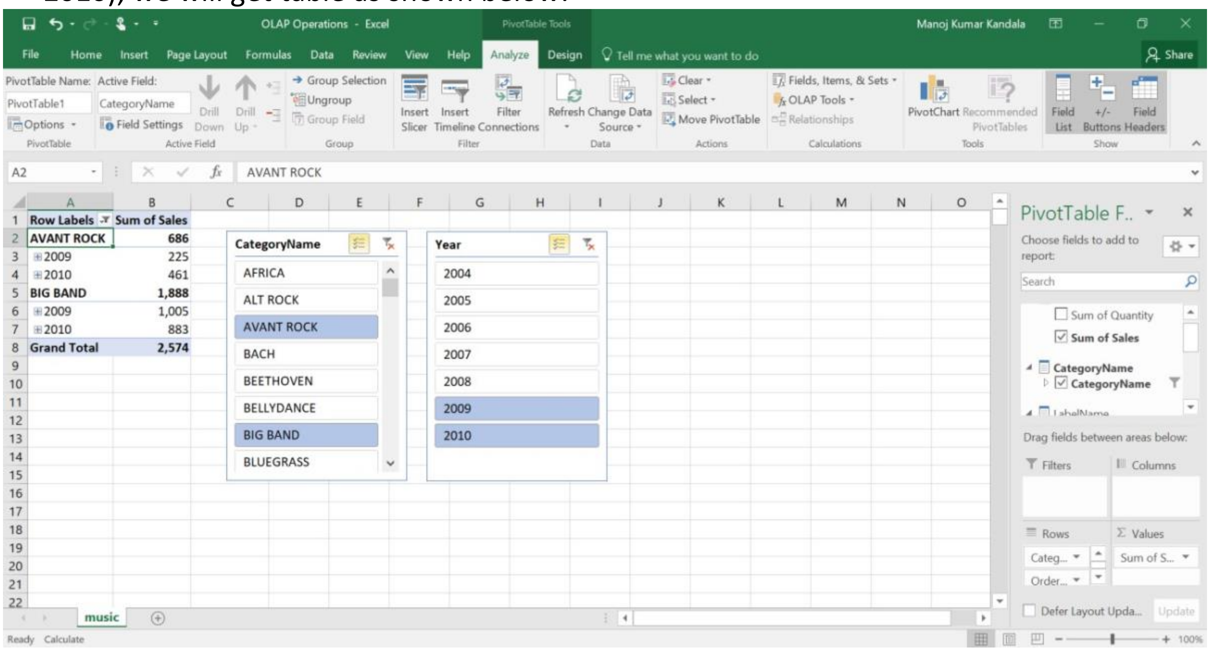
Now we are going to perform **roll-up (drill-up) operation**, in the above window I selected January month then automatically **Drill-up option** is enabled on top. We will click on **Drill-up** option, then the below window will be displayed.



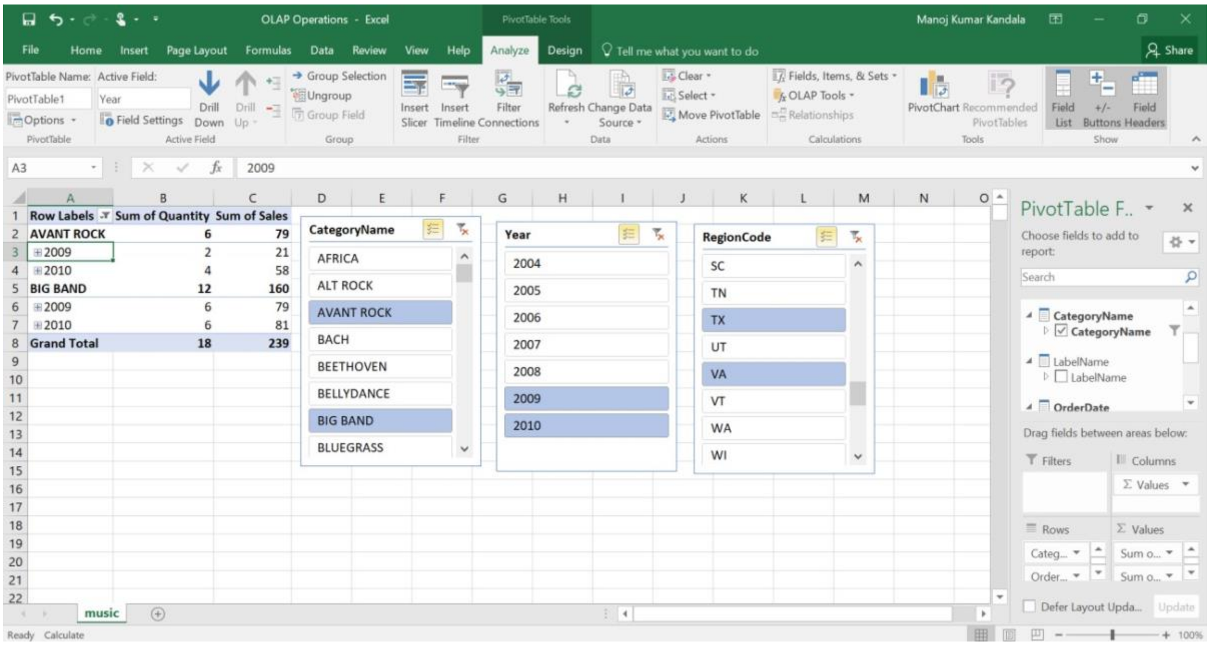
Next OLAP operation **Slicing** is performed by inserting slicer as shown in top navigation options.



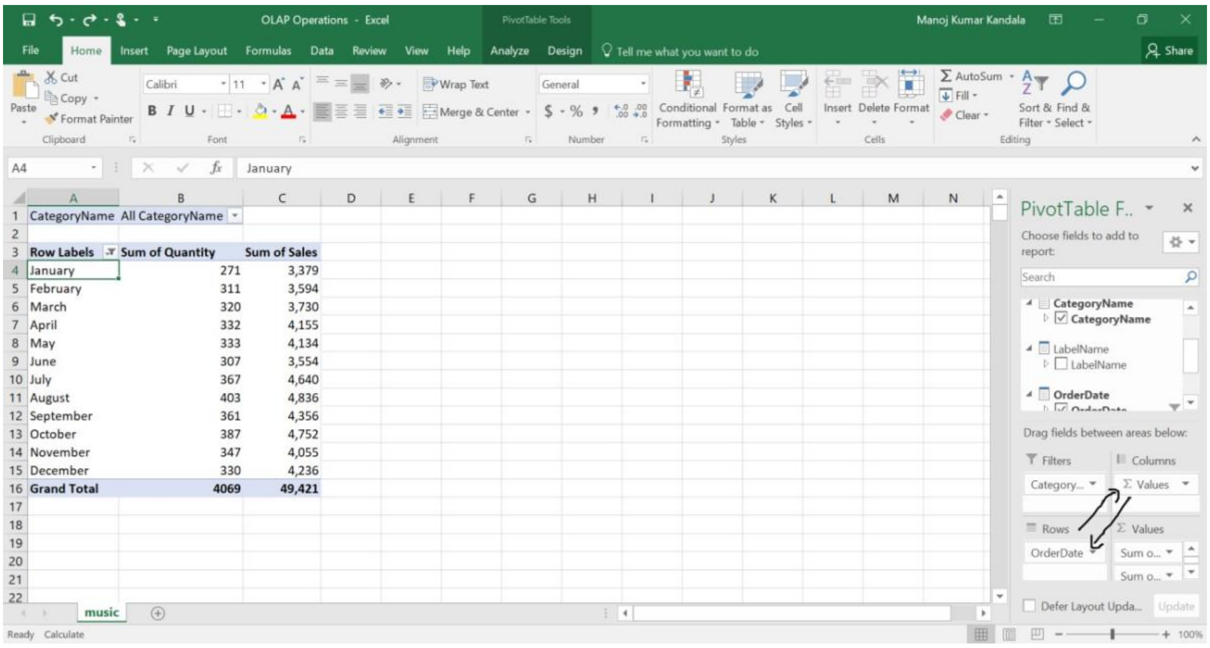
While inserting slicers for slicing operation, we select 2 Dimensions (for e.g. Category Name & Year) only with one Measure (for e.g. Sum of sales).After inserting a slice& adding a filter (Category Name: AVANT ROCK & BIG BAND; Year: 2009 & 2010), we will get table as shown below.



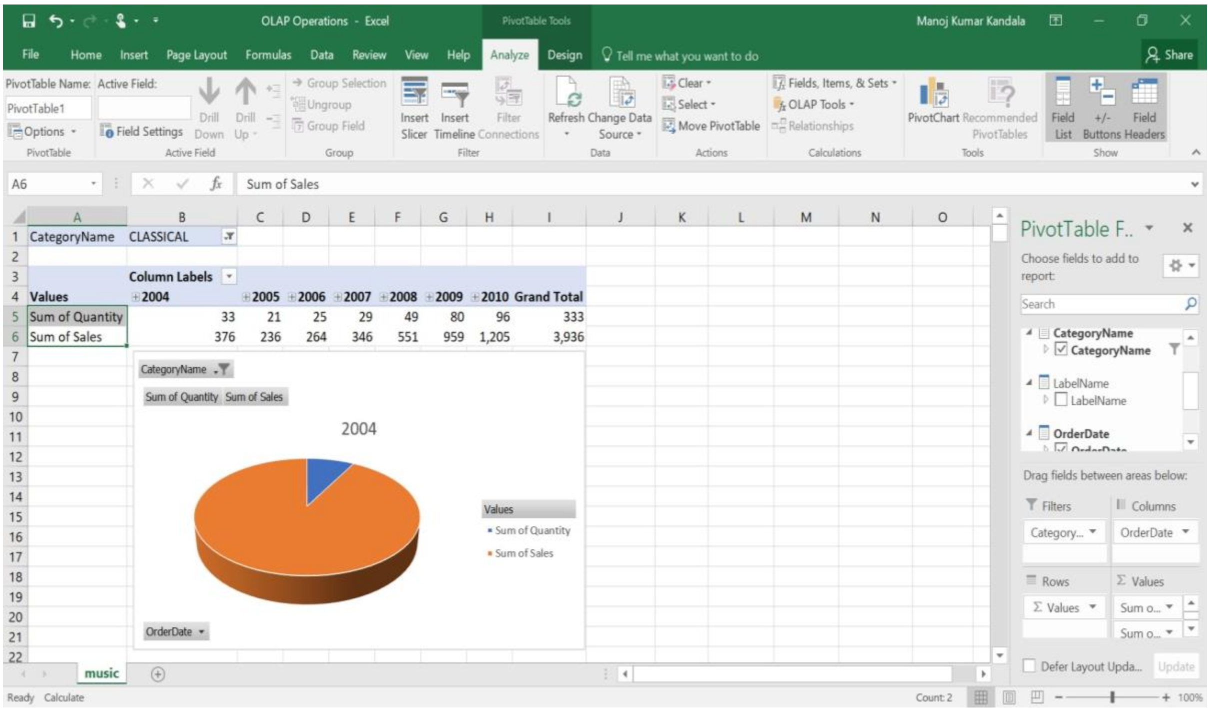
Dicing operation is similar to Slicing operation. Here we are selecting 3 dimensions (Category Name, Year, Region Code)& 2 Measures (Sum of Quantity, Sum of Sales) through „**insert slicer**" option. After that adding a filter for Category Name, Year & Region Code as shown below.



Finally, the **Pivot (rotate)** OLAP operation is performed by swapping rows (Order Date-Year) & columns (Values-Sum of Quantity & Sum of Sales) through right side bottom navigation bar as shown below.



After Swapping (rotating), we will get resultant as represented below with a pie-chart for Category-Classical& Year Wise data.



<u>Conclusion</u>:- Hence we have successfully implemented OLAP operations.

Assignment No 5:

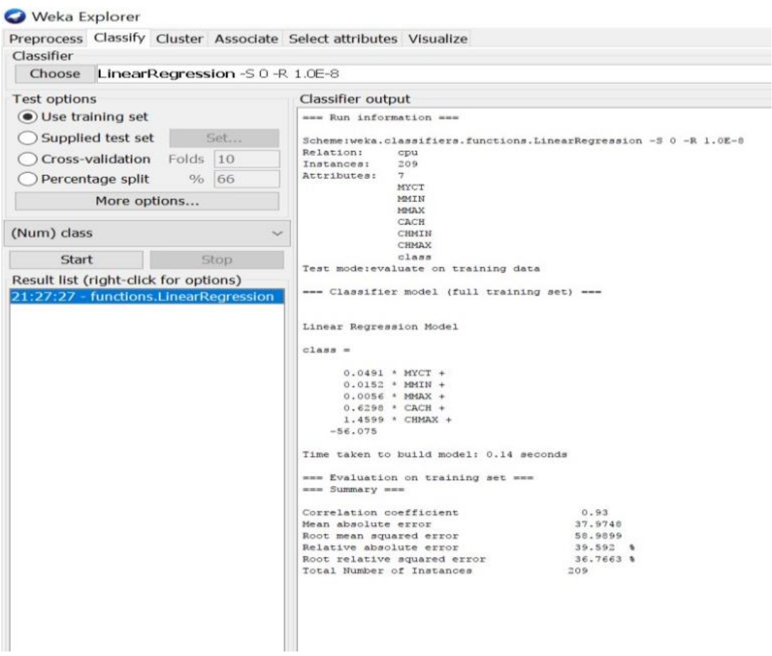
Aim:-Demonstrate performing Regression on data sets.

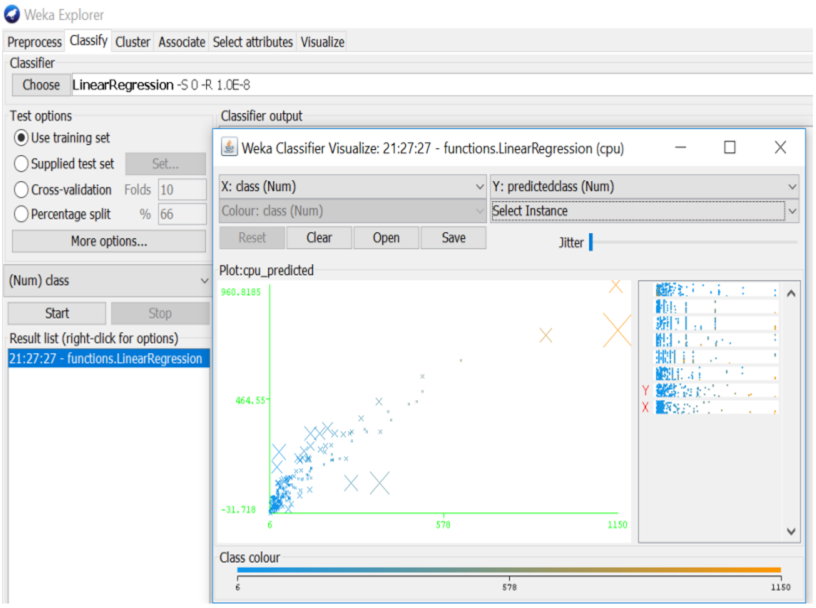
Theory:-

A. Load each dataset into Weka and build Linear Regression model. Study the clusters formed. Use Training set option. Interpret the regression model and derive patterns and conclusions from the regression results.

Procedure:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with use training set option.
4. Then we will get regression model & its result as shown below.
5. The patterns are visually mentioned below for regression model through visualize classifier errors option which is available in right click options.

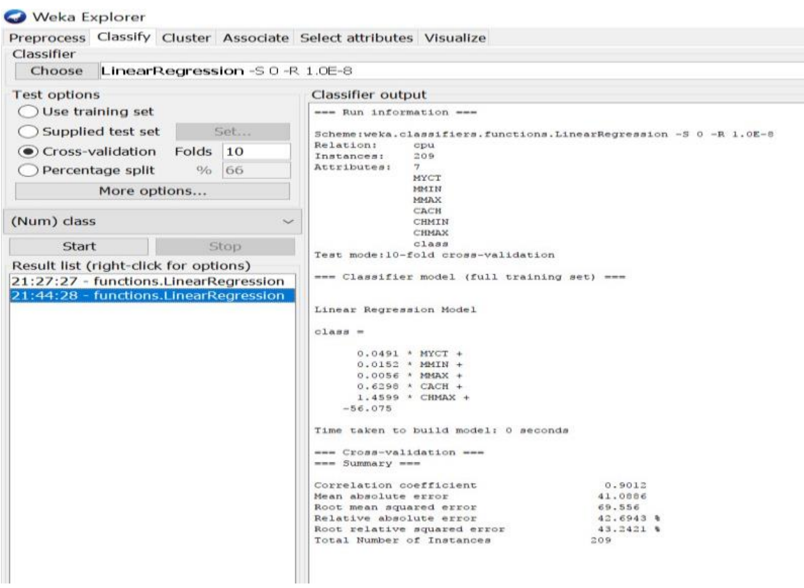




B. **Use options cross-validation and percentage split and repeat running the Linear Regression Model. Observe the results and derive meaningful results.**

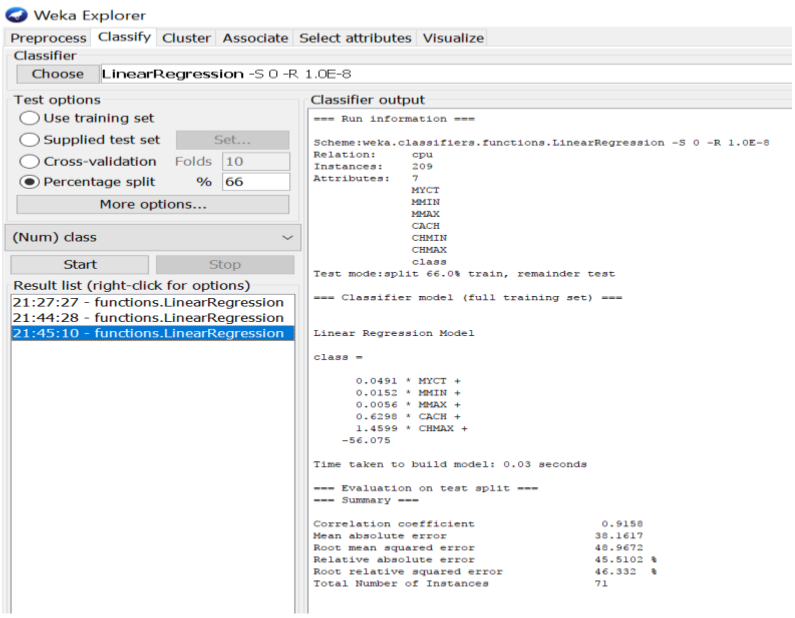
Procedure for cross-validation:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with cross validation option with 10 folds.
4. Then we will get regression model & its result as shown below.



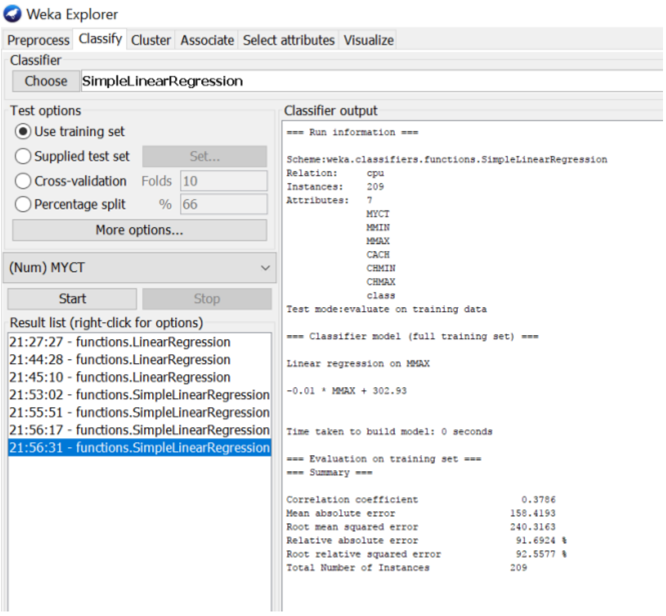
Procedure for percentage split:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Linear Regression algorithm & click on start option with percentage split option with 66% split.
4. Then we will get regression model & its result as shown below.



C. Explore Simple linear regression technique that only looks at one variable Procedure:

1. Load the dataset (Cpu.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different classification algorithms under functions section.
3. In which we selected Simple Linear Regression algorithm & click on start option with use training set option with one variable (MYCT).
4. Then we will get regression model & its result as shown below.



<u>Conclusion</u>:- Hence we have successfully performed Regression on data sets.

Assignment No 6:

Aim:-Demonstration of clustering rule process on data-set iris.arff using simple k-means.

Theory:-

Iris dataset :

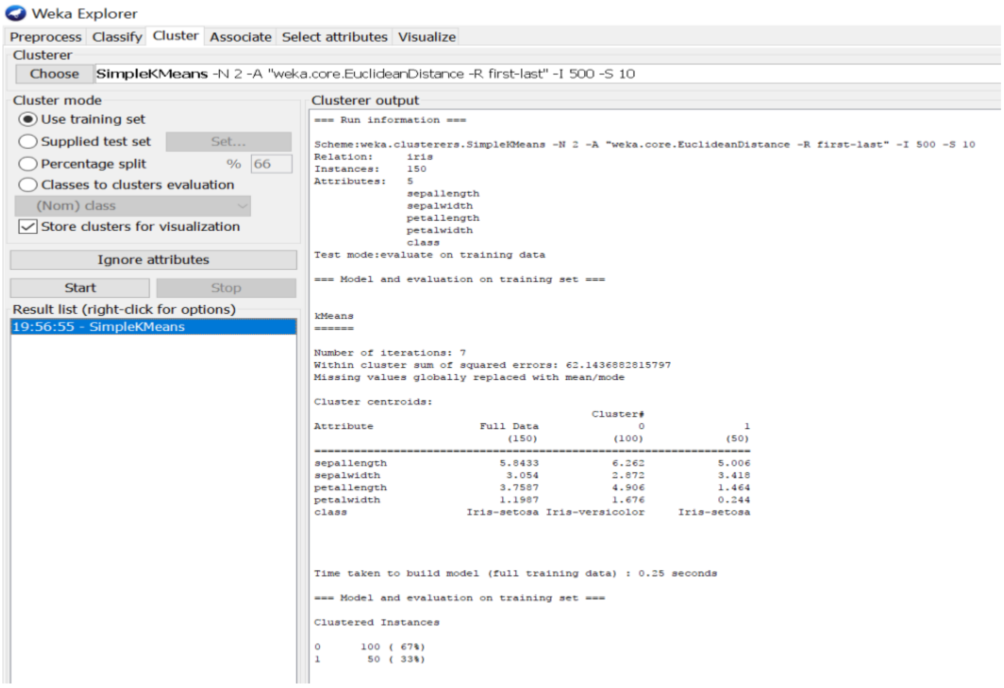
Iris dataset is the Hello World for the Data Science, so if you have started your career in Data Science and Machine Learning you will be practicing basic ML algorithms on this famous dataset. Iris dataset contains five columns such as Petal Length, Petal Width, Sepal Length, Sepal Width and Species Type.

Iris is a flowering plant, the researchers have measured various features of the different iris flowers and recorded digitally.

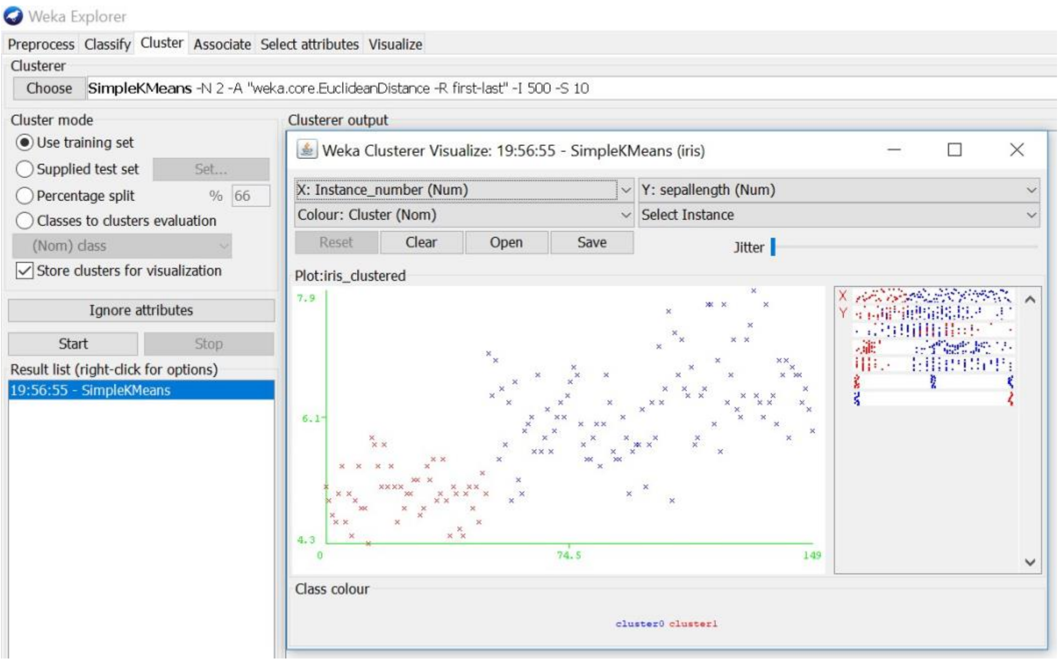
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa

Procedure:-

1. Load the dataset (Iris.arff) into weka tool
2. Go to classify option & in left-hand navigation bar we can see different clustering algorithms under lazy section.
3. In which we selected Simple K-Means algorithm & click on start option with –use training set‖ test option enabled.
4. Then we will get the sum of squared errors, centroids, No. of iterations & clustered instances as represented below.



5. If we right click on simple k means, we will get more options in which –Visualize cluster assignments‖ should be selected for getting cluster visualization as shown below.



<u>Conclusion</u>:-Hence we have demonstrated the clustering rule process on data-set iris.arff using simple k-means.

Assignment No 7:

Demonstration of any ETL tool.

Assignment No 8:

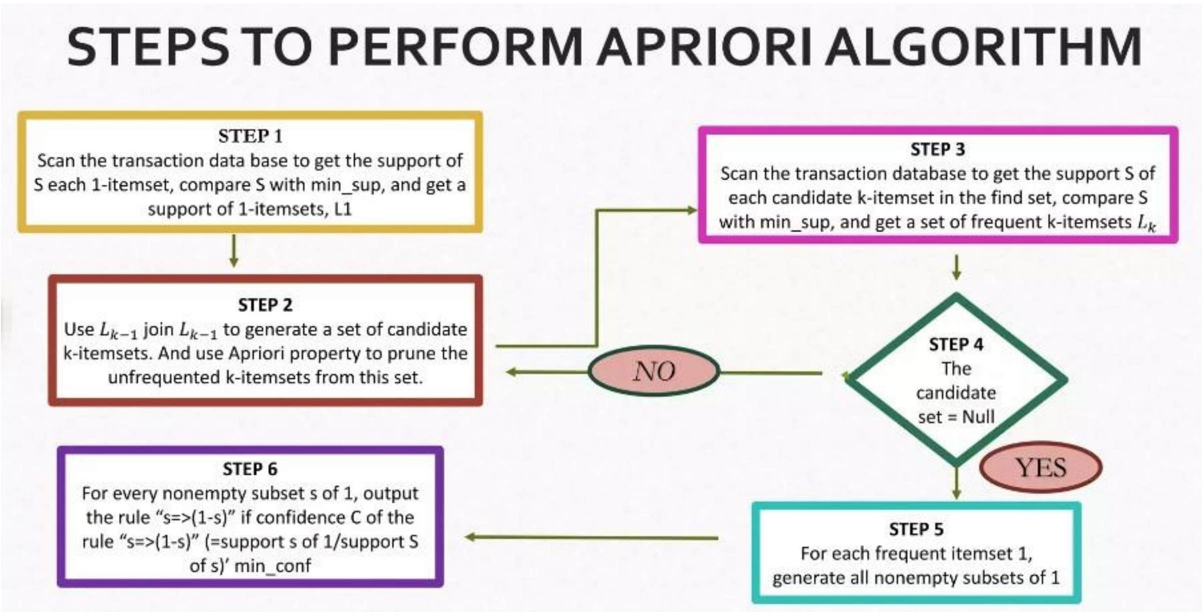
<u>Aim</u>:-Write a program of Apriori algorithm using any programming language.

<u>Theory</u>:-

Apriori algorithm:

Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.

To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called Apriori property which helps by reducing the search space.



Procedure:-

1. Load the dataset (Breast-Cancer.arff) into weka tool
2. Go to associate option & in left-hand navigation bar we can see different association algorithms.
3. In which we can select Aprori algorithm & click on select option.
4. Below we can see the rules generated with different support & confidence values for that selected dataset.

```
Relation:     breast-cancer
Instances:    286
Attributes:   10
              age
              menopause
              tumor-size
              inv-nodes
              node-caps
              deg-malig
              breast
              breast-quad
              irradiat
              Class
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.5 (143 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 10

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 6

Size of set of large itemsets L(3): 4

Size of set of large itemsets L(4): 1

Best rules found:

 1. inv-nodes=0-2 irradiat=no Class=no-recurrence-events 147 ==> node-caps=no 145    conf:(0.99)
 2. inv-nodes=0-2 irradiat=no 183 ==> node-caps=no 177    conf:(0.97)
 3. node-caps=no irradiat=no Class=no-recurrence-events 151 ==> inv-nodes=0-2 145    conf:(0.96)
 4. inv-nodes=0-2 Class=no-recurrence-events 167 ==> node-caps=no 160    conf:(0.96)
 5. inv-nodes=0-2 213 ==> node-caps=no 201    conf:(0.94)
 6. node-caps=no irradiat=no 188 ==> inv-nodes=0-2 177    conf:(0.94)
 7. node-caps=no Class=no-recurrence-events 171 ==> inv-nodes=0-2 160    conf:(0.94)
 8. irradiat=no Class=no-recurrence-events 164 ==> node-caps=no 151    conf:(0.92)
 9. inv-nodes=0-2 node-caps=no Class=no-recurrence-events 160 ==> irradiat=no 145    conf:(0.91)
10. node-caps=no 222 ==> inv-nodes=0-2 201    conf:(0.91)
```

Conclusion:- Hence we have implemented the Apriori algorithm.

Assignment No 9:

Case Study on Text Mining or any commercial application.