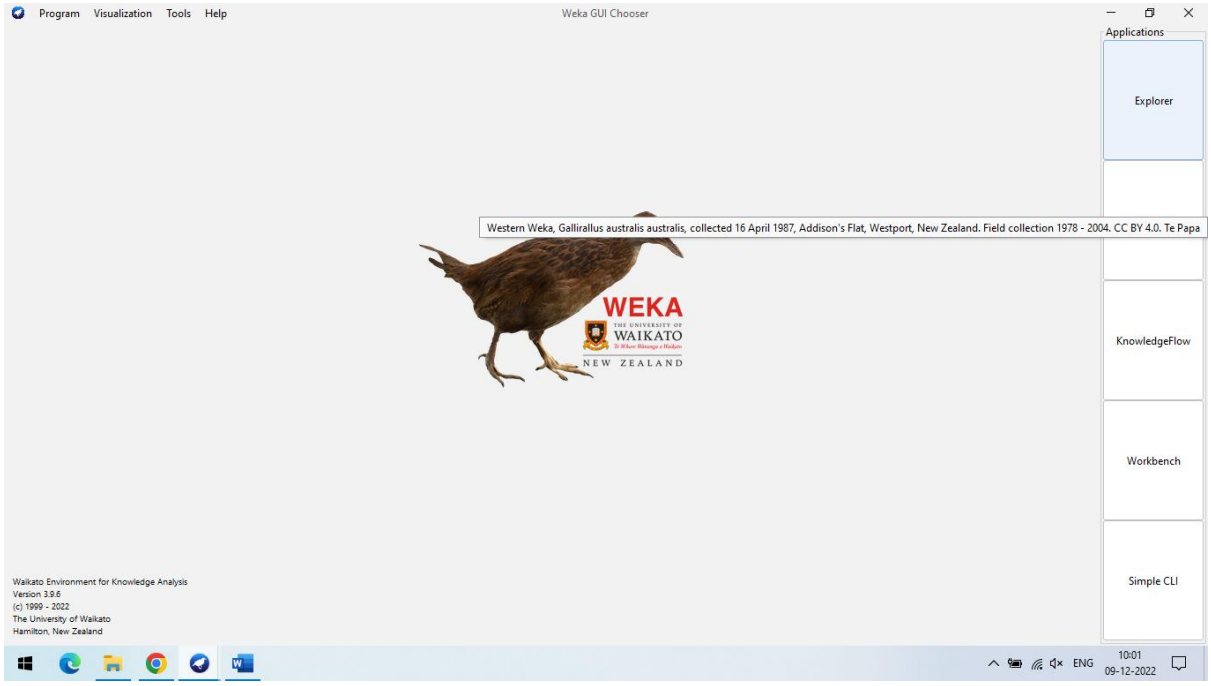
PRACTICAL-

AIM: Study about the tools Weka tool for Data Mining.

Theory:

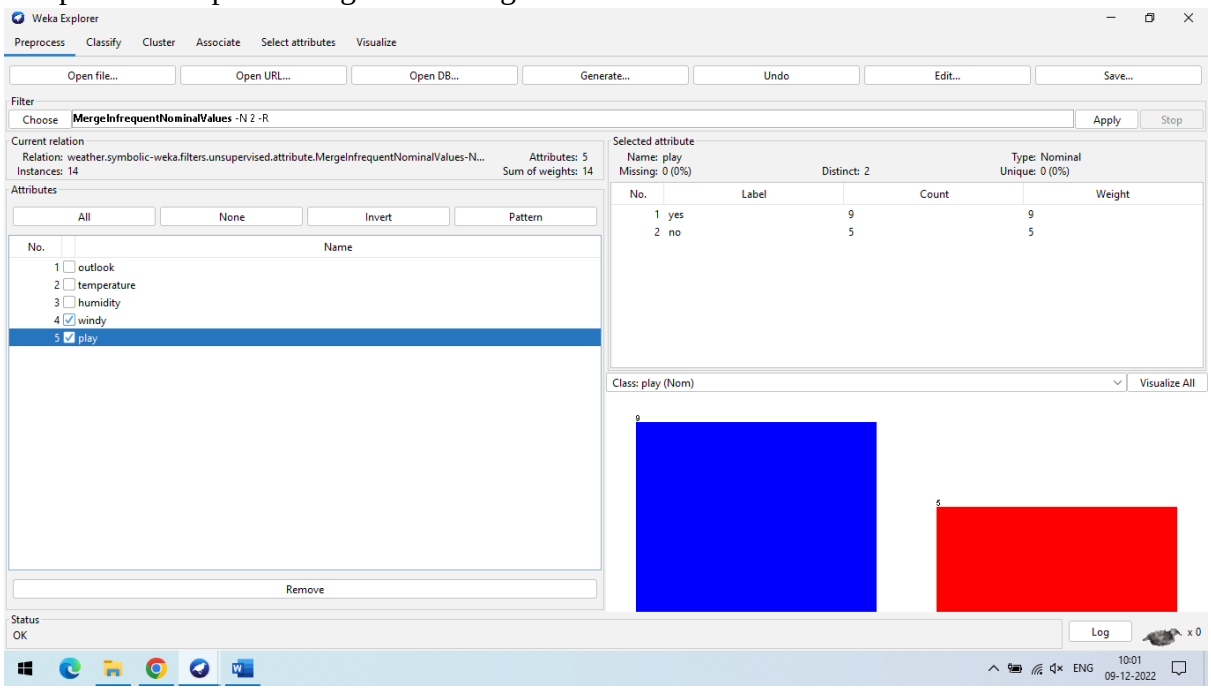
Weka Tool:

Weka is one of the very popular open source data mining tools developed at the University of Waikato in New Zealand in 1992. It is a Java based tool and can be used to implement various machine learning and data mining algorithms written in Java. The simplicity of using Weka has made it a landmark for machine learning and data mining implementation. Weka supports reading of files from several different databases and also allows importing the data from the internet, from web pages or from a remotely located SQL database server by entering the URL of resource. Among all the available data mining tools, Weka is the most commonly used of all due to its fast performance and support for major classification and clustering algorithm. Weka can be easily downloaded and deployed. Weka provides both, a GUI and CLI for performing data mining and does a good job of providing support for all the data mining tasks. Weka supports a variety of data formats like CSV (Commaseparated Value), ARFF and Binary. Weka focuses more on textual representation of the data rather than visualization although it does provide support to display some visualization but those are very generic. Also, Weka does not provide visual representation of results of processing in an effective and understanding manner like Rapid Miner. Weka performs accurately when the size of the data set is not large. If the size is large, then Weka does experience some performance issues. Weka provides support for filtering out data or attributes. Weka supports the following three graphical user interfaces.



1.The Explorer:

It is the most commonly used graphical user interface in Weka to implement data mining algorithms8. It supports exploratory data analysis to perform preprocessing, attribute selection, learning and visualization. This interface consists of different tabs to access various components for performing data mining. The different tabs are-



A) **Preprocessing** Using this tab, we can load input data files and perform preprocessing on this data using filters.

B) **Classify** This tab is used to implement different classification and regression algorithms. We can do this by selecting a particular classifier from this tab. For example, the K-NN or Naïve Bayesian algorithm can be implemented by using this tab.

C) **Associate** This tab is used to find out all association rules between different attributes of the data and which can be used for further mining. For example, Association rule mining, etc.

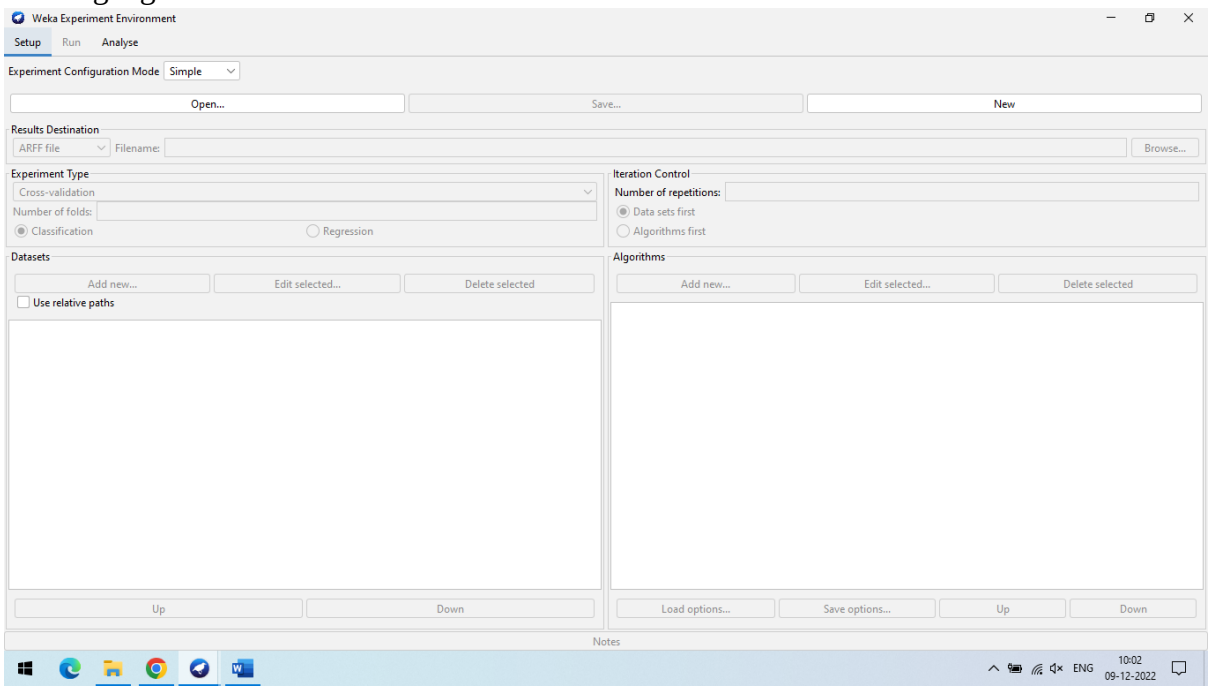
D) **Cluster** Using this tab, we can select a particular clustering algorithm to implement for our data set. Clustering algorithms like K-means can be implemented using this tab.

E) **Select attributes** This tab is used to select particular attributes from the data set useful for implementing the algorithm.

F) **Visualize** This tab is used to visualize the data whenever available or supported by a particular algorithm in the form of scatter plot matrix.

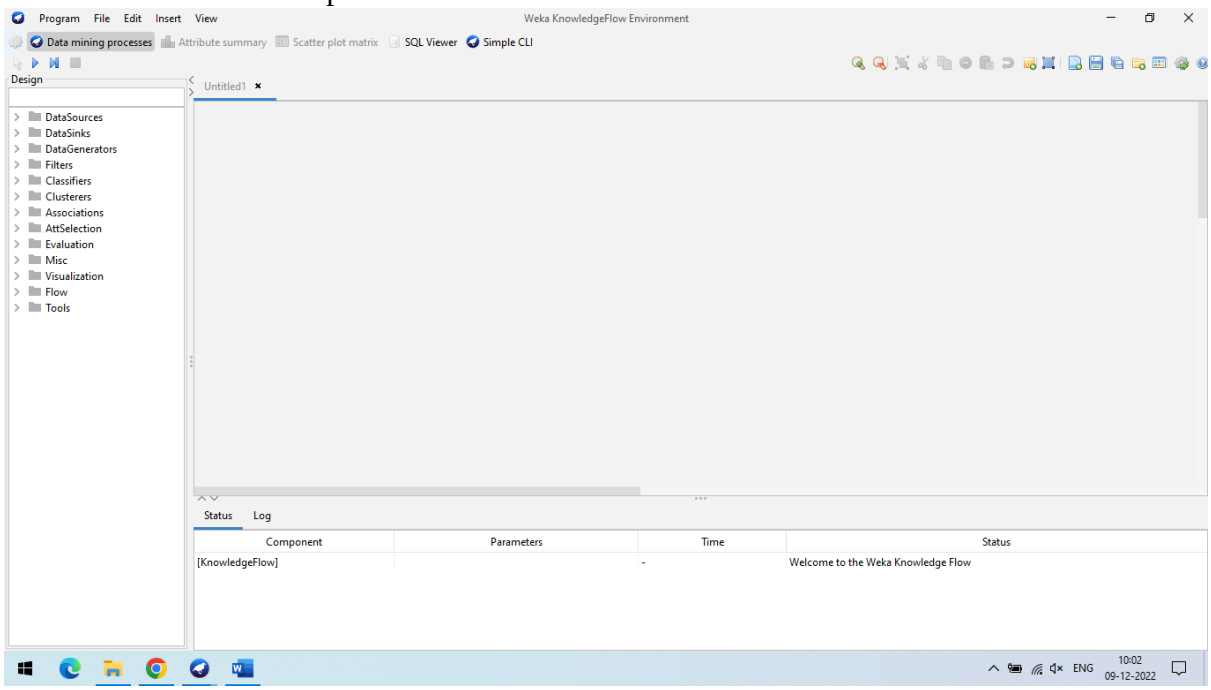
2. **The Experimenter**

This user interface provides experimental environment for testing and evaluating machine learning algorithms.



3. **The Knowledge**

Flow Knowledge flow is basically a component based interface similar to explorer. This interface is used for new process evaluations.



PRACTICAL-5

AIM: Perform Pre-processing on a dataset. Apply various Filters and discuss the effect of each filter applied.

a. Handle Missing Values

b. Handle Infrequent Nominal Values

c. Derive an attribute from the existing attribute

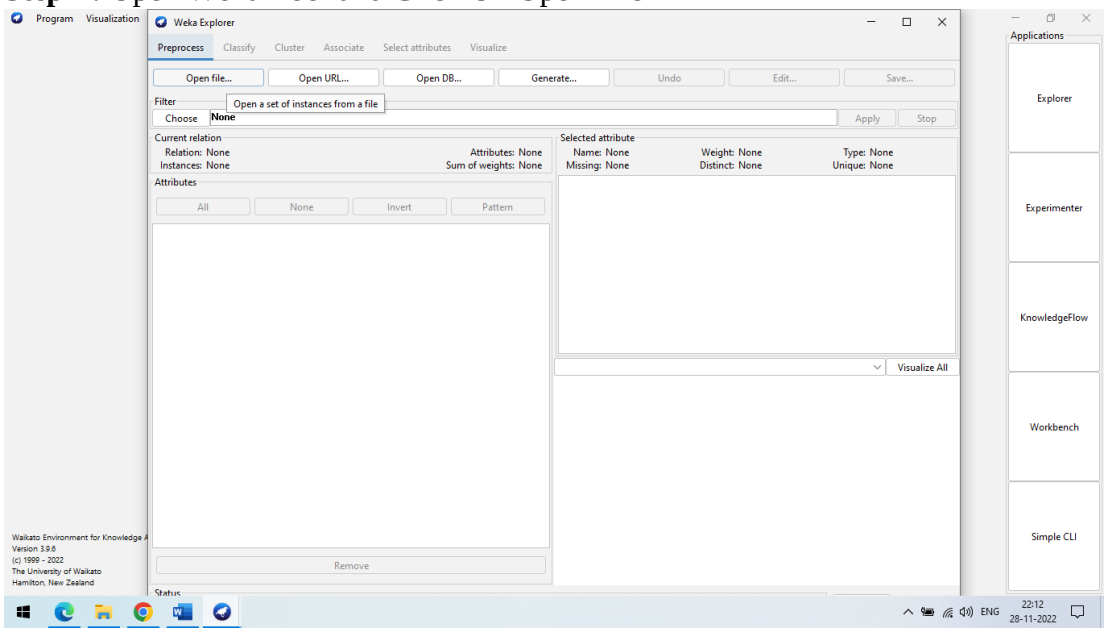
d. Sampling

e. Discretization

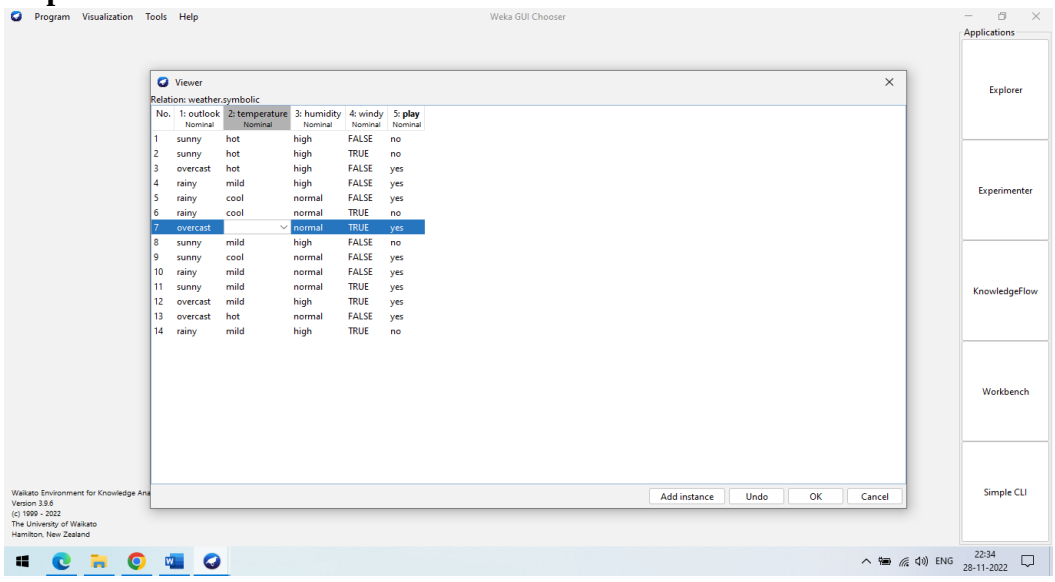
1)Use Weka Tool 2) Use XL Miner Tool.

a. Handle Missing Values

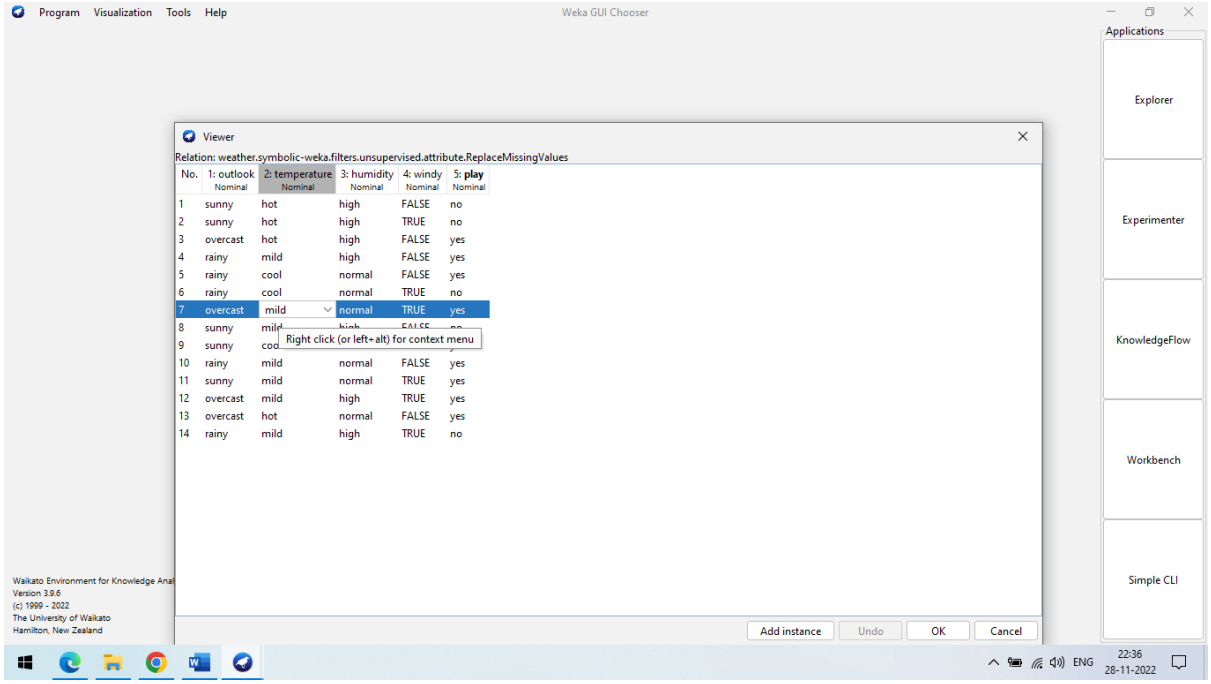
Step-1: Open Weka Tool and Click On Open File



Step-2: Select The Data Files weather.nominal.arff remove one value

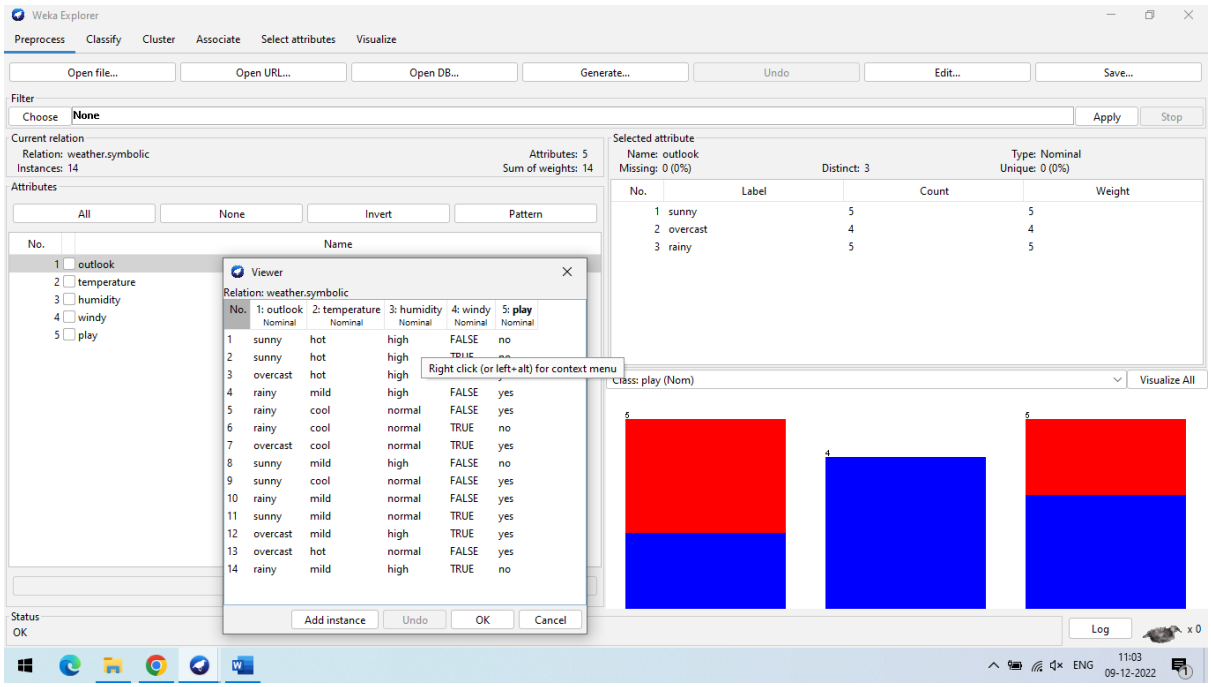


Step-3 : Choose→weka→filters→unsupervised→attribute→ReplaceMissingValue

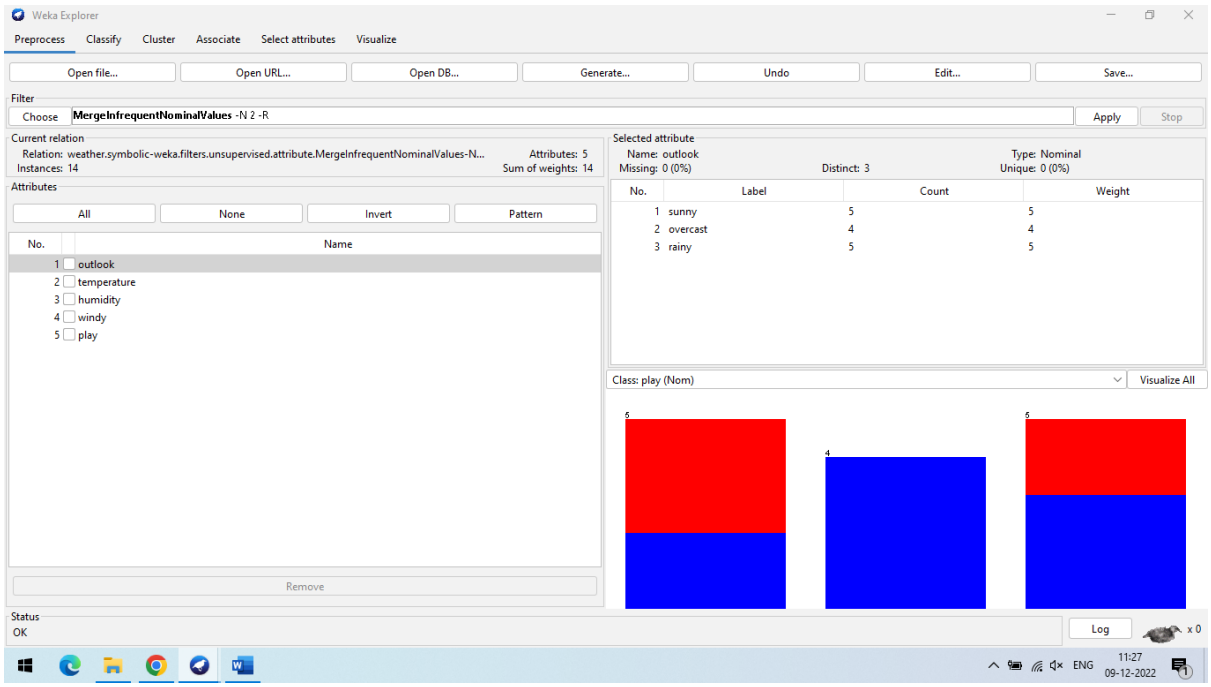


b. Handle Infrequent Nominal Values

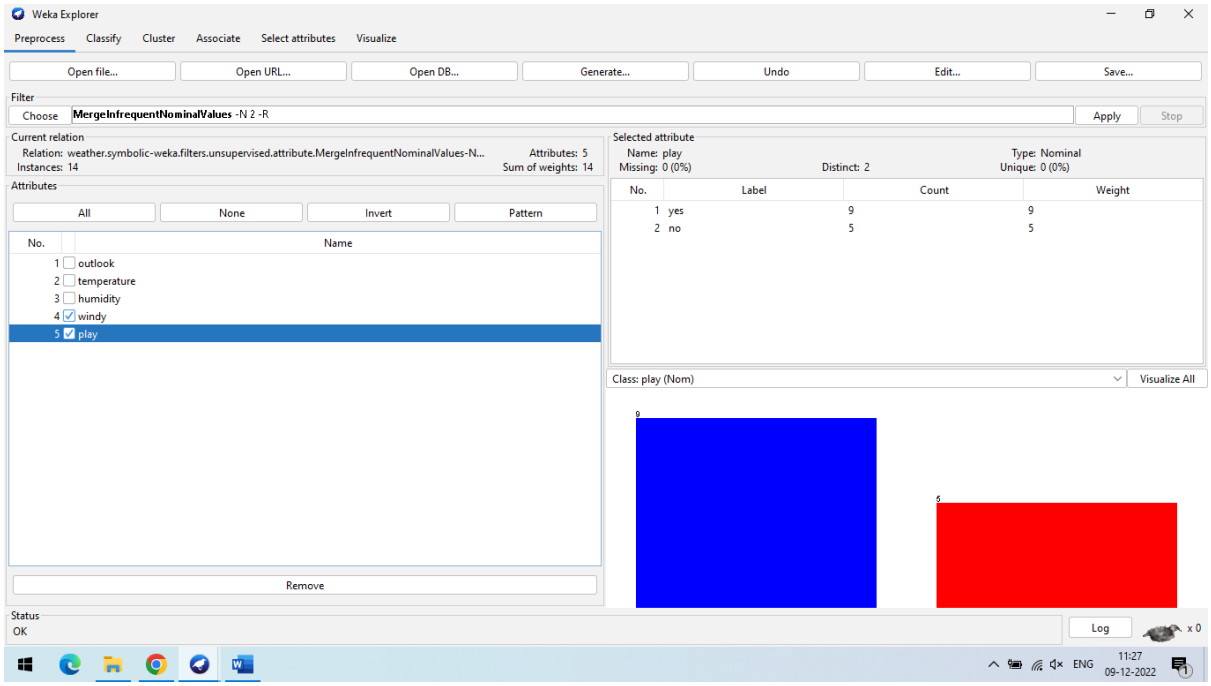
Step 1: First of all here we will use the weather.nominal.arff dataset.



Step-2 : After that we have to choose mergeInfrequentNominal for that we have to follow this path. weka→filters→unsupervised→attribute→mergeInfrequentNominalValues and then click on Apply button.

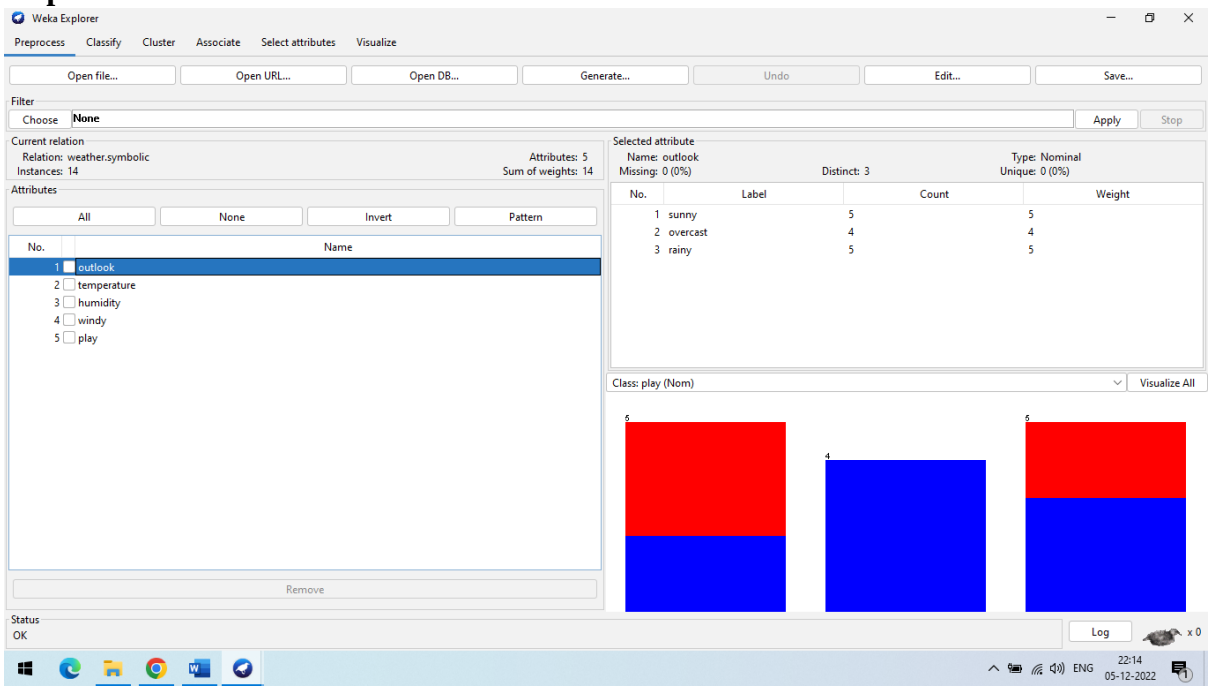


Step 3: After click on apply button we can get the output like this.



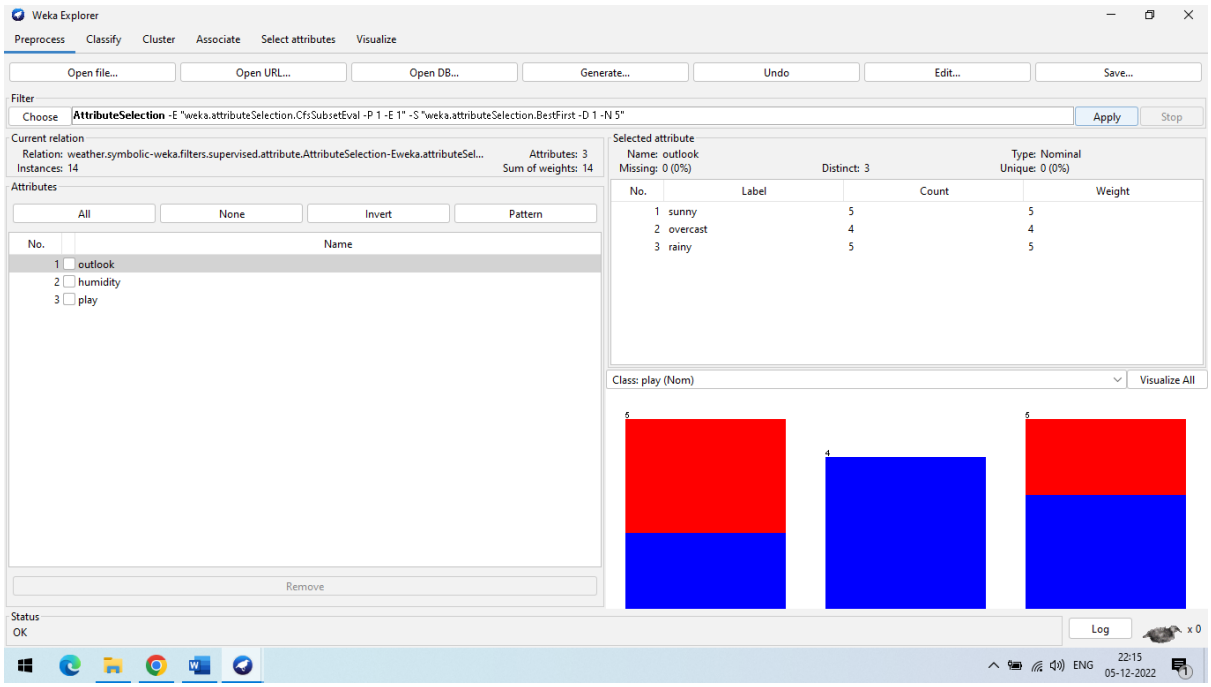
c. Derive an attribute from the existing attribute

Step-1 : we will use **weather-numeric.arff** database



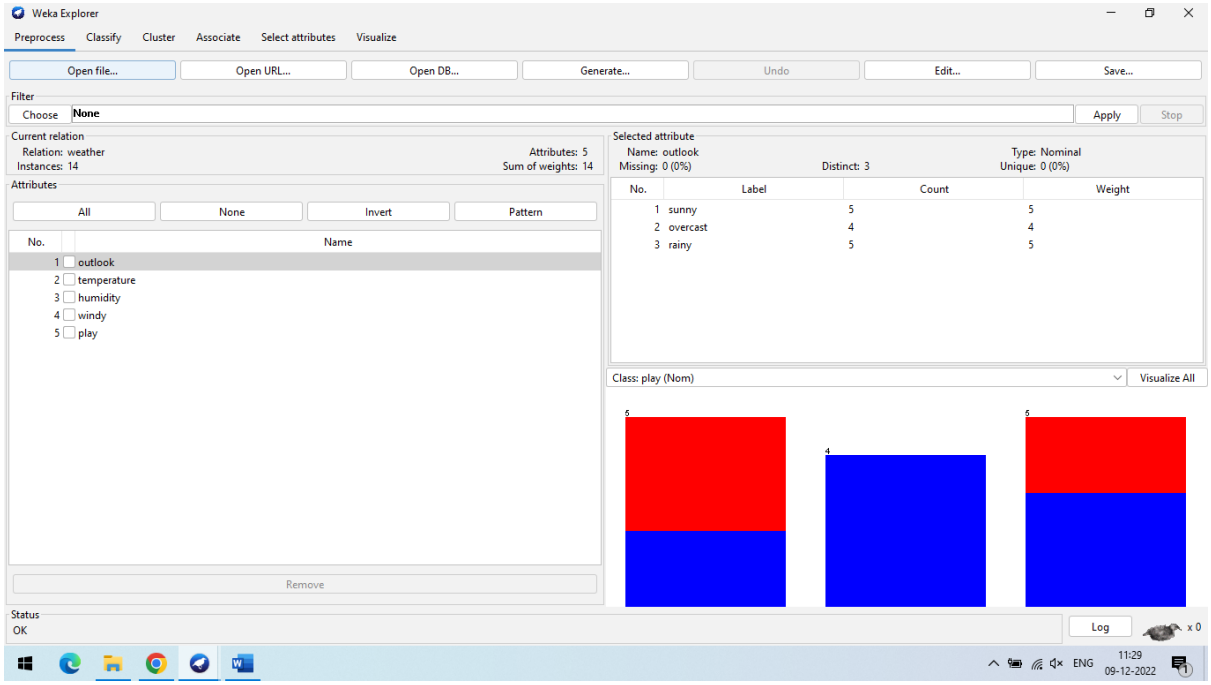
Step-2 : weka→filters→supervised→attribute→AttributeSelection

We notice that temperature and windy will be remove.

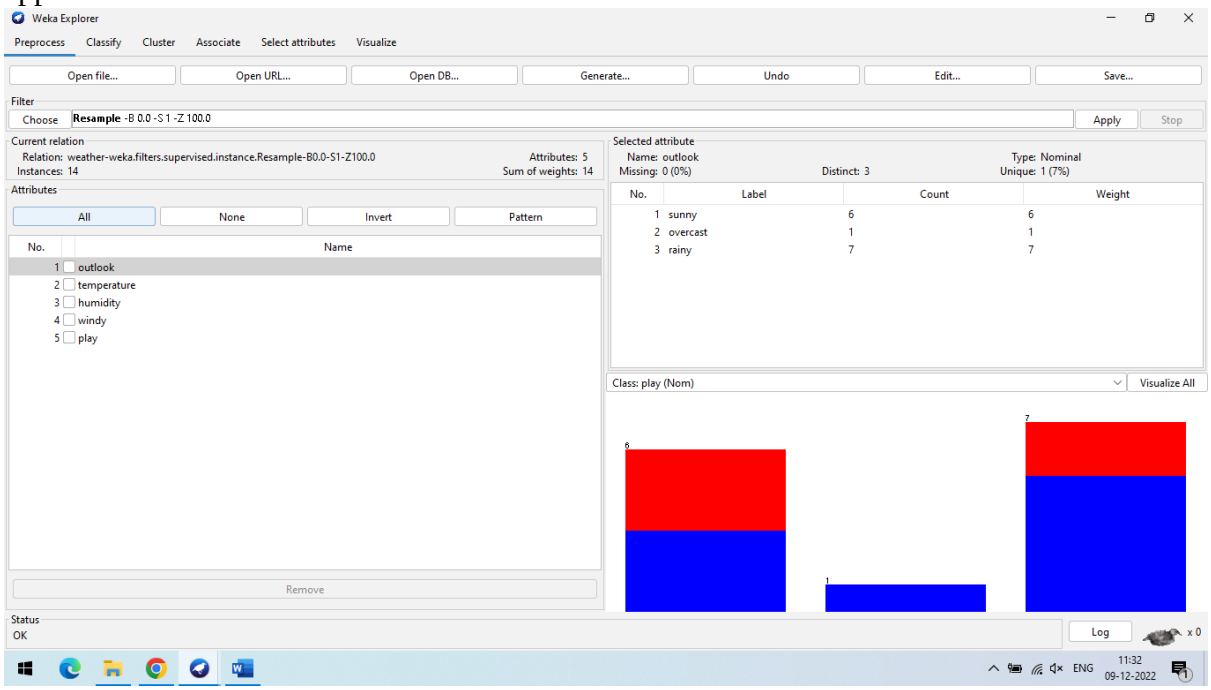


d. Sampling

Step 1: First of all we are using weather-numeric.arff dataset and select the humidity data.



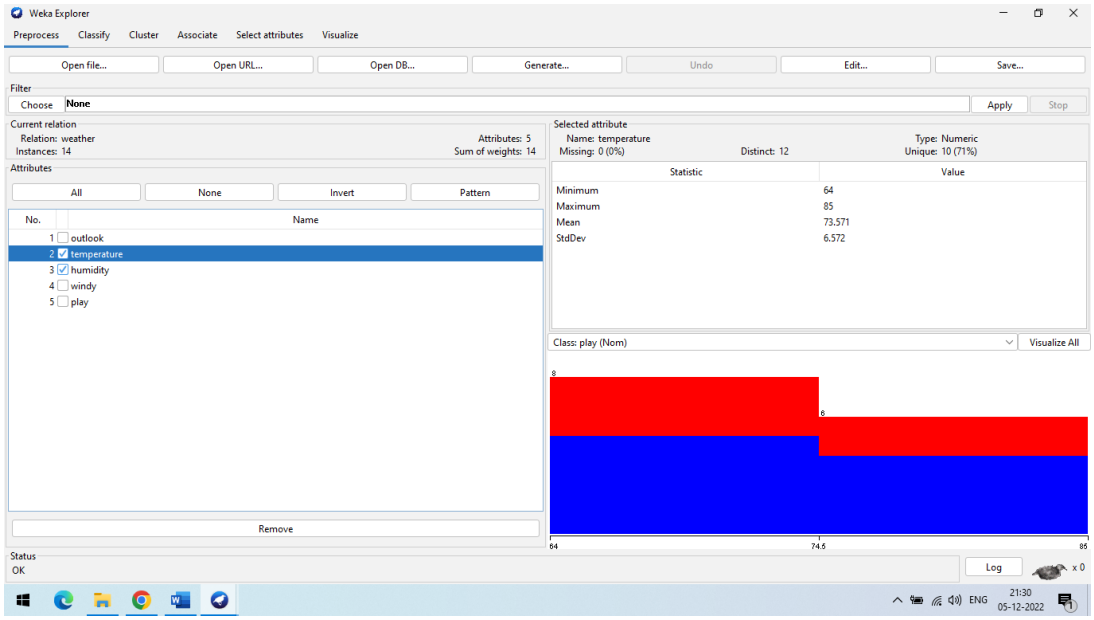
Step-2 : After that we have to follow this path for select the Resamble Choose→weka→filters→supervised→attribute→instance→Resample and then click on applu button.



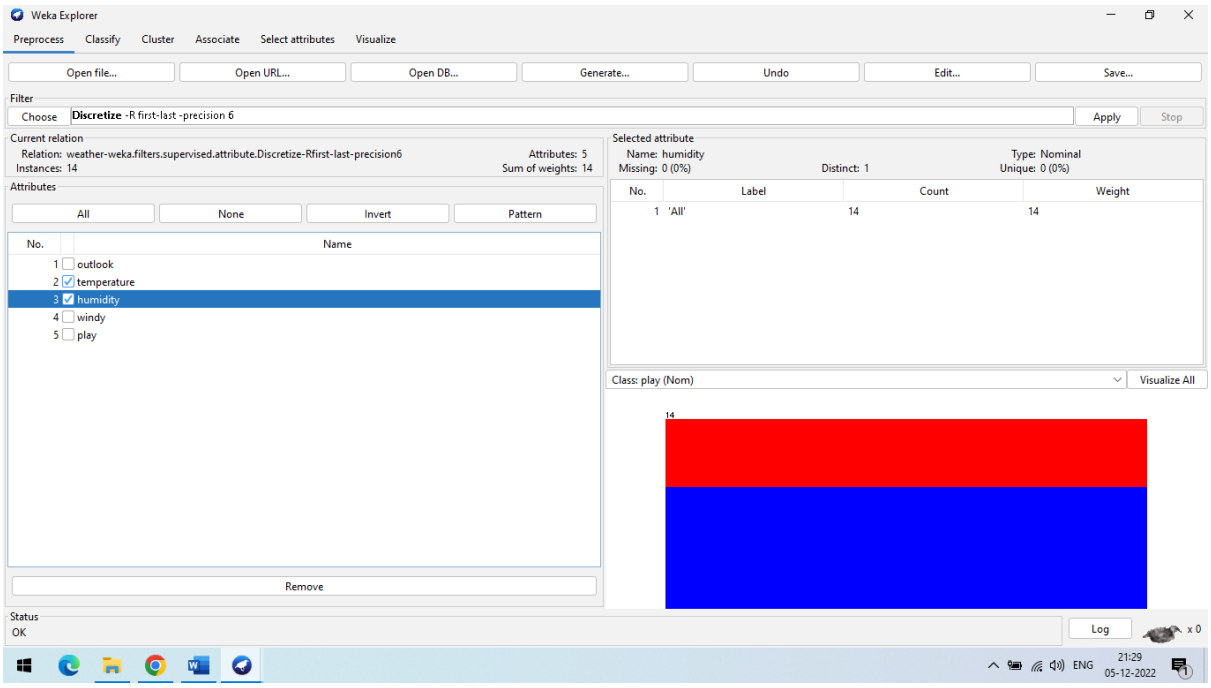
Note: Here you will notice that rainy,overcast and sunny will change.

e. Discretization

Step-1: we will use **weather-numeric.arff** database that contains two **numeric** attributes - **temperature** and **humidity**



Step-2 : Choose→weka→filters→supervised→attribute→Discretize

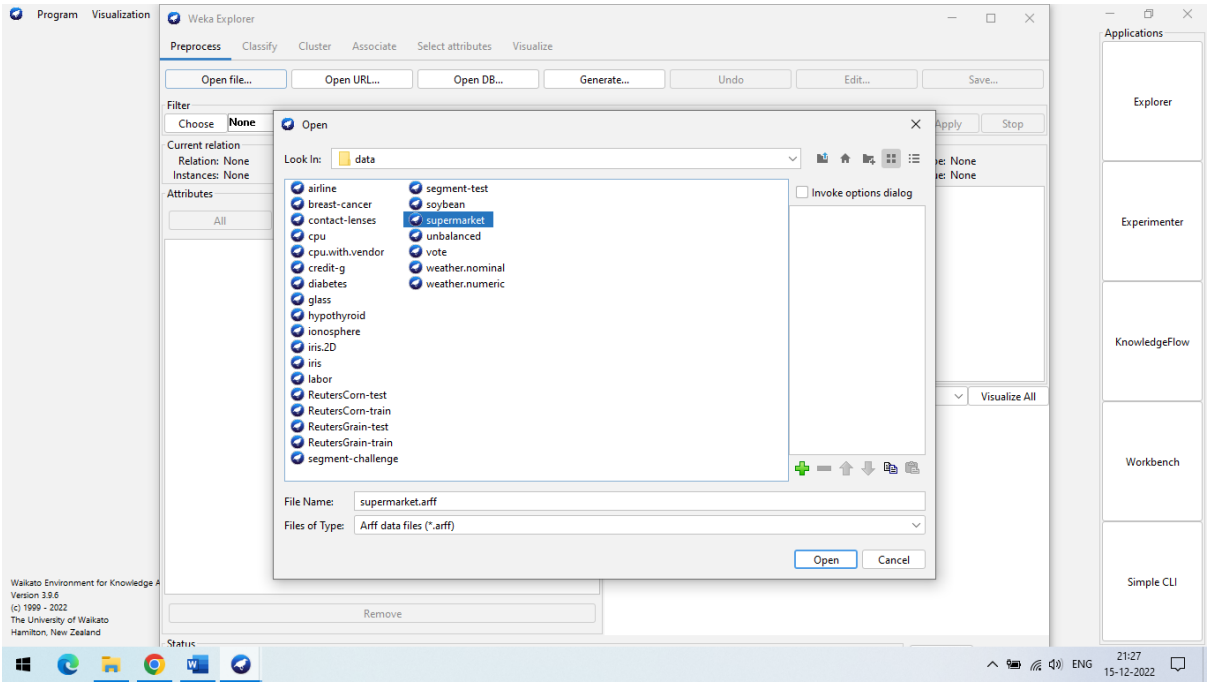


we will notice that these have changed from numeric to nominal types

PRACTICAL-8

AIM: Perform Association rule mining using WEKA tool.

Step 1: weka→explorer→open file→supermarket.arff
First of all we are using supermarket.arff dataset



Step 2: Go to **Associate->Choose->Apriori->right click**
Attribute configuration in Weka Explorer as per requirement

NAME

weka.associations.Apriori

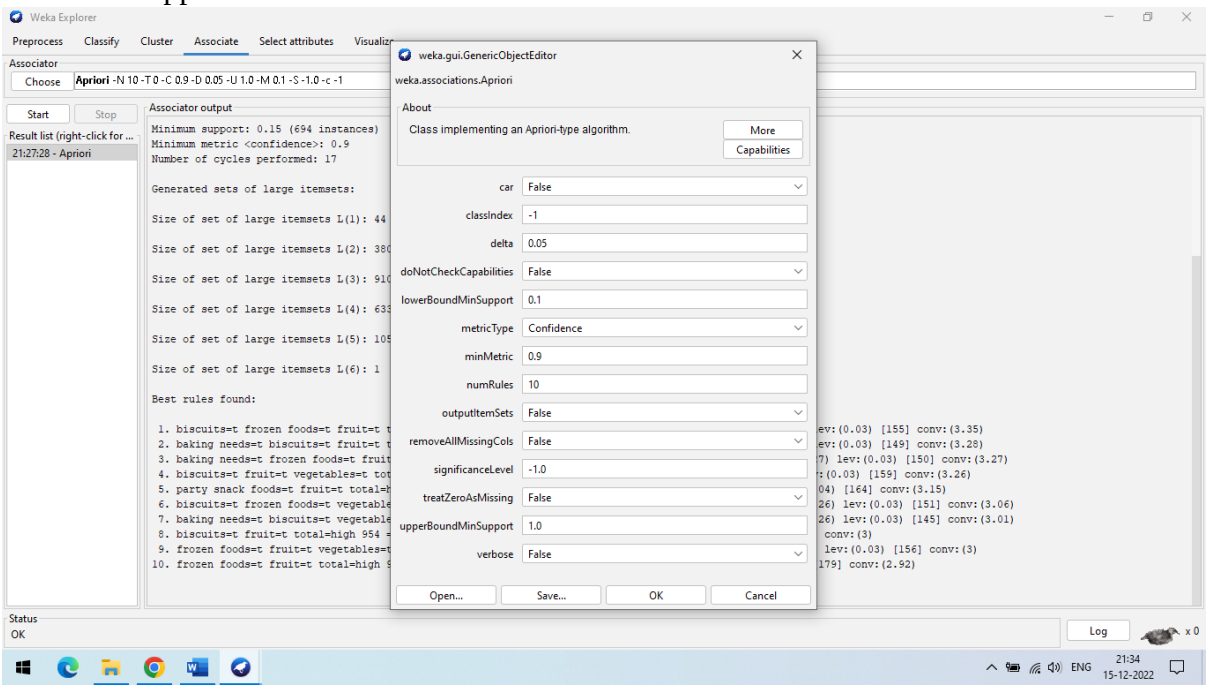
SYNOPSIS

Class implementing an Apriori-type algorithm. Iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence.
The algorithm has an option to mine class association rules. It is adapted as explained in the second reference.
For more information see:
R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules in Large Databases. In: 20th International Conference on Very Large Data Bases, 478-499, 1994.
Bing Liu, Wynne Hsu, Yiming Ma: Integrating Classification and Association Rule Mining. In: Fourth International Conference on Knowledge Discovery and Data Mining, 80-86, 1998.

OPTIONS

minMetric -- Minimum metric score. Consider only rules with scores higher than this value.
verbose -- If enabled the algorithm will be run in verbose mode.
numRules -- Number of rules to find.
lowerBoundMinSupport -- Lower bound for minimum support.
classIndex -- Index of the class attribute. If set to -1, the last attribute is taken as class attribute.
outputItemSets -- If enabled the itemsets are output as well.
car -- If enabled class association rules are mined instead of (general) association rules.

doNotCheckCapabilities -- If set, associator capabilities are not checked before associator is built (Use with caution to reduce runtime).
removeAllMissingCols -- Remove columns with all missing values.
Significancelevel -- Significance level. Significance test (confidence metric only).
Treatzeroasmissing -- If enabled, zero (that is, the first value of a nominal) is treated in the same way as a missing value.
Delta -- Iteratively decrease support by this factor. Reduces support until min support is reached or required number of rules has been generated.
Metrictype -- Set the type of metric by which to rank rules. Confidence is the proportion of the examples covered by the premise that are also covered by the consequence (Class association rules can only be mined using confidence). Lift is confidence divided by the proportion of all examples that are covered by the consequence. This is a measure of the importance of the association that is independent of support. Leverage is the proportion of additional examples covered by both the premise and consequence above those expected if the premise and consequence were independent of each other. The total number of examples that this represents is presented in brackets following the leverage. Conviction is another measure of departure from independence. Conviction is given by P(premise)P(!consequence) / P(premise, !consequence).
upperBoundMinSupport -- Upper bound for minimum support. Start iteratively decreasing minimum support from this value.



Step 3: Go to **Associate->Choose->Apriori->start**
The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules. Some key concepts for Apriori algorithm are :
• Frequent Itemsets: The sets of item which has minimum support (denoted by Li for ith-Itemset).
• Apriori Property: Any subset of frequent itemset must be frequent.
• Join Operation: To find Lk , a set of candidate kitemsets is generated by joining Lk-1 with itself.

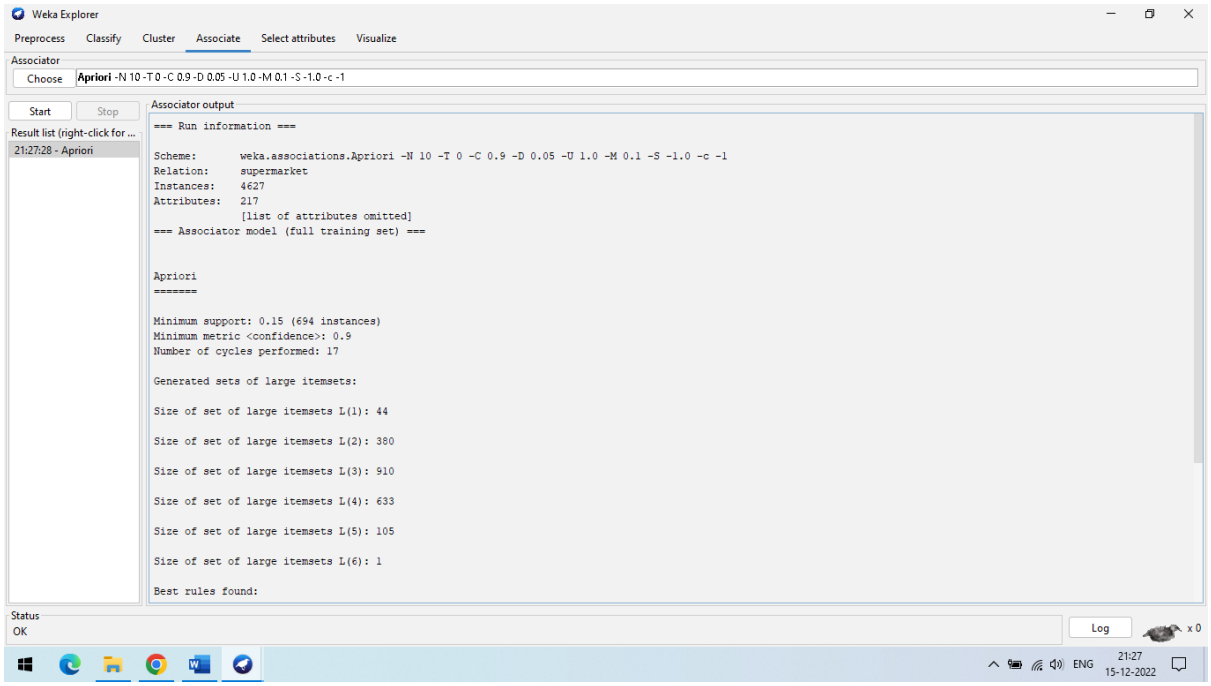
CAPABILITIES

Class -- Binary class, Missing class values, No class, Nominal class

Attributes -- Binary attributes, Empty nominal attributes, Missing values, Nominal attributes, Unary attributes

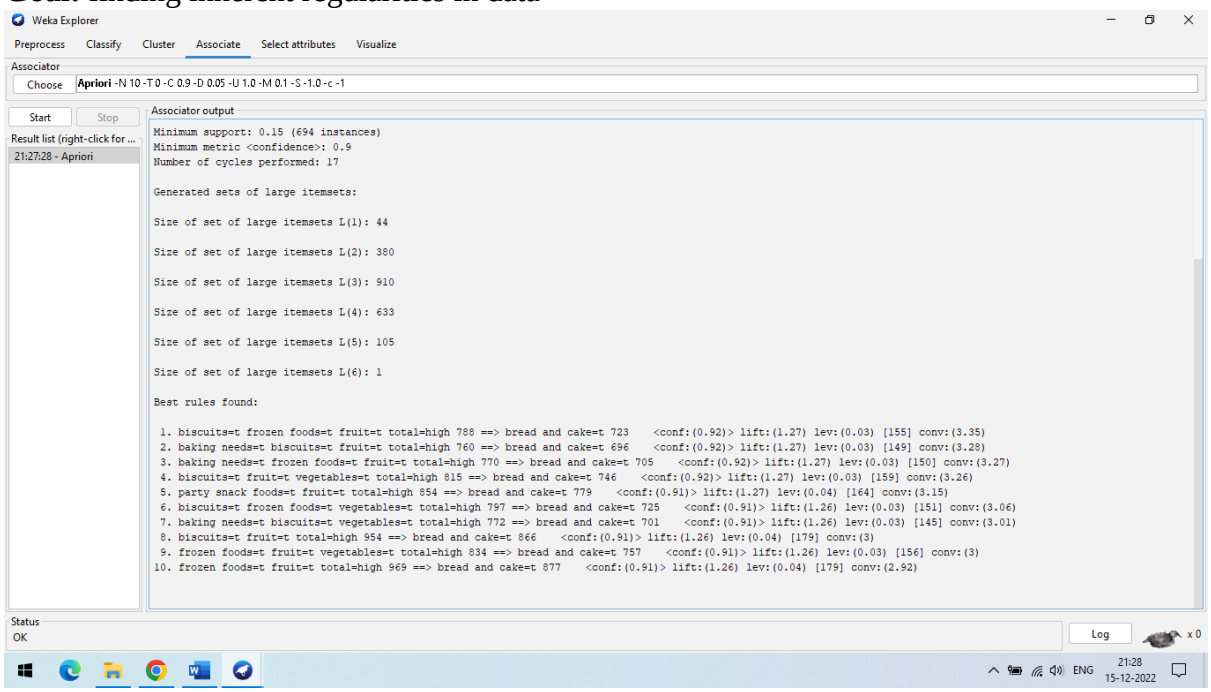
Additional

Minimum number of instances: 1



Apriori is the simplest algorithm which is used for mining of frequent patterns from the transaction database. The purpose of reducing the number of scans of database to extract frequent item set will be resolved in future due to our work is in progress for the same.

Goal: finding inherent regularities in data



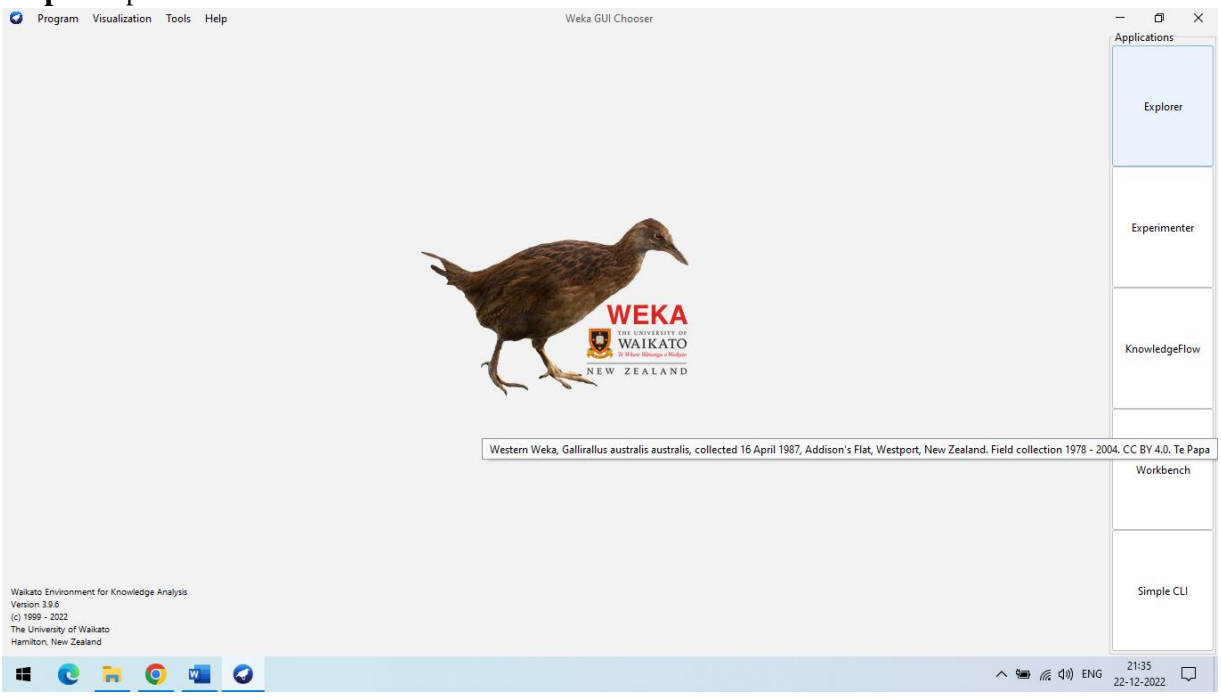
PRACTICAL-9

AIM: Perform classification with WEKA tool.
a. using Decision Tree Classifier
b. using Naïve Bayes Classifier
c. using Multilayer Perceptron.

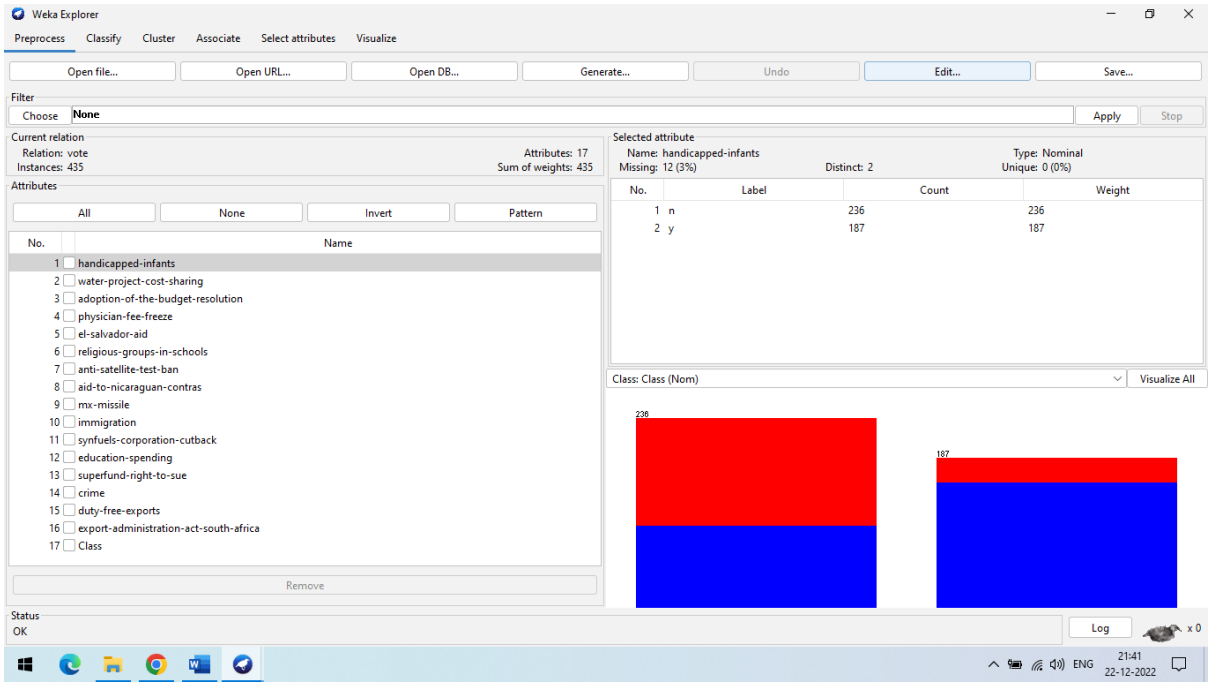
Steps:

(A) Using Decision Tree Classifier:- A decision tree is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from one class. Each non-leaf node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned. A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. Decision Tree Mining is a type of data mining technique that is used to build Classification Models. It builds classification models in the form of a tree-like structure, just like its name. This type of mining belongs to supervised class learning.

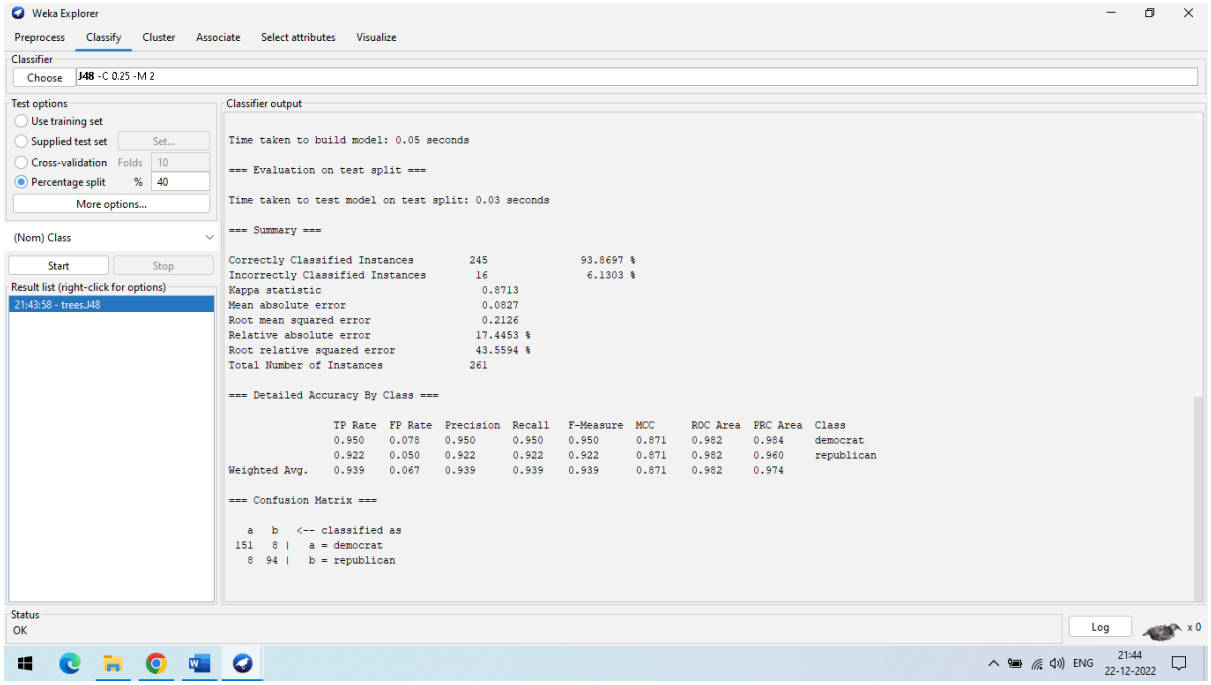
Step 1: Open Weka Tool



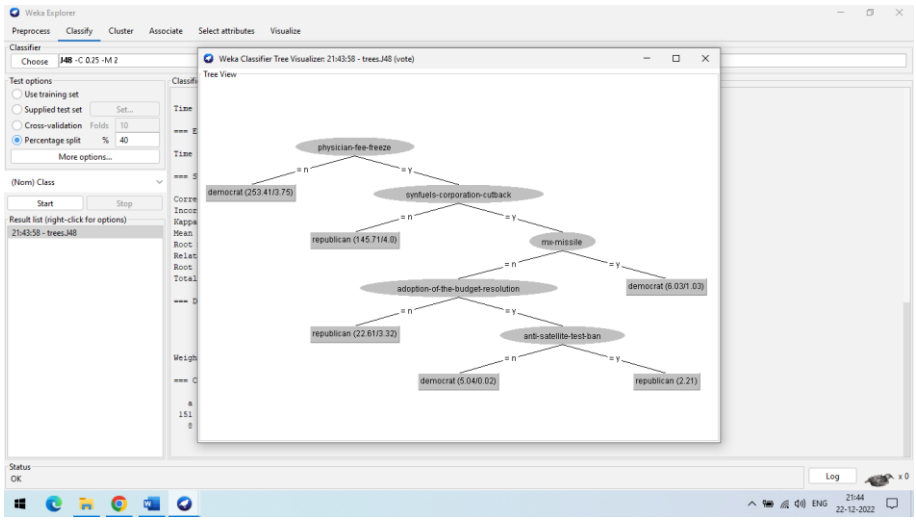
Step 2: weka→explorer→open file→ vote. Arff



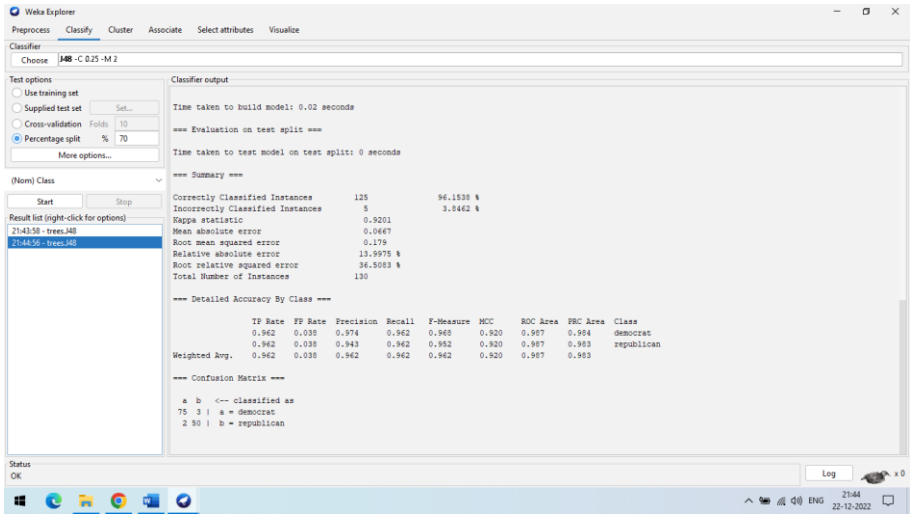
Step 3: Go to the classify >> choose filter j48 >> give percentage split(40%) >> click start



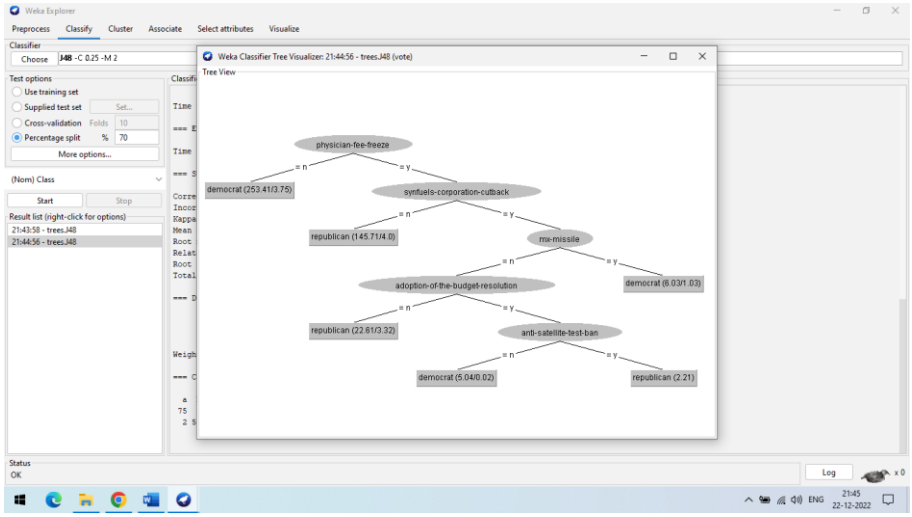
Step 4: right click on result list >> visualize tree



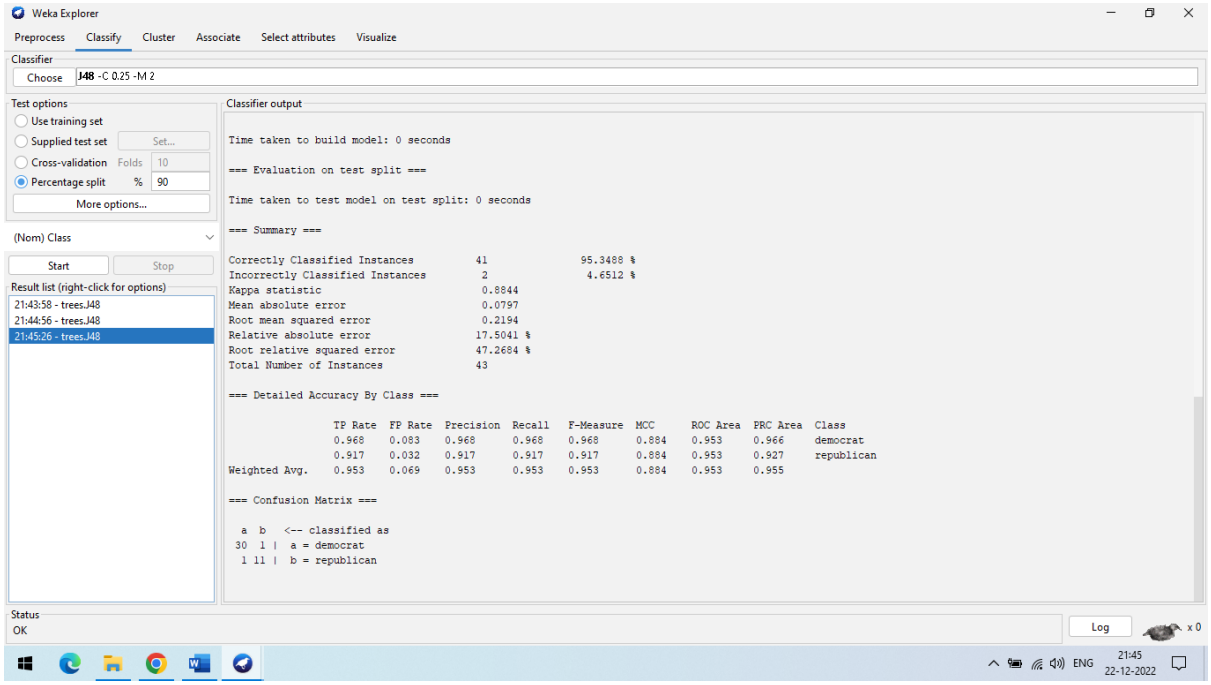
Step 5: Go to the classify >> choose filter j48 >> give percentage split(70%) >> click start



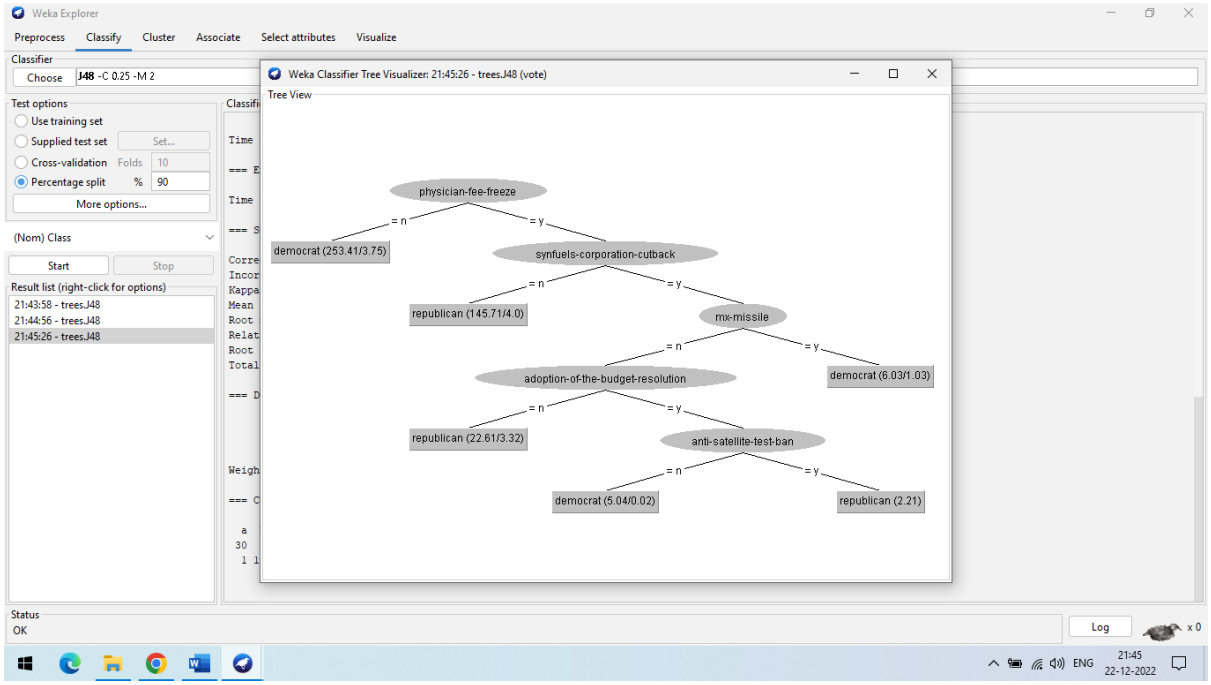
Step 6: right click on result list >> visualize tree



Step 7: Go to the classify >> choose filter j48 >> give percentage split(90%) >> click start



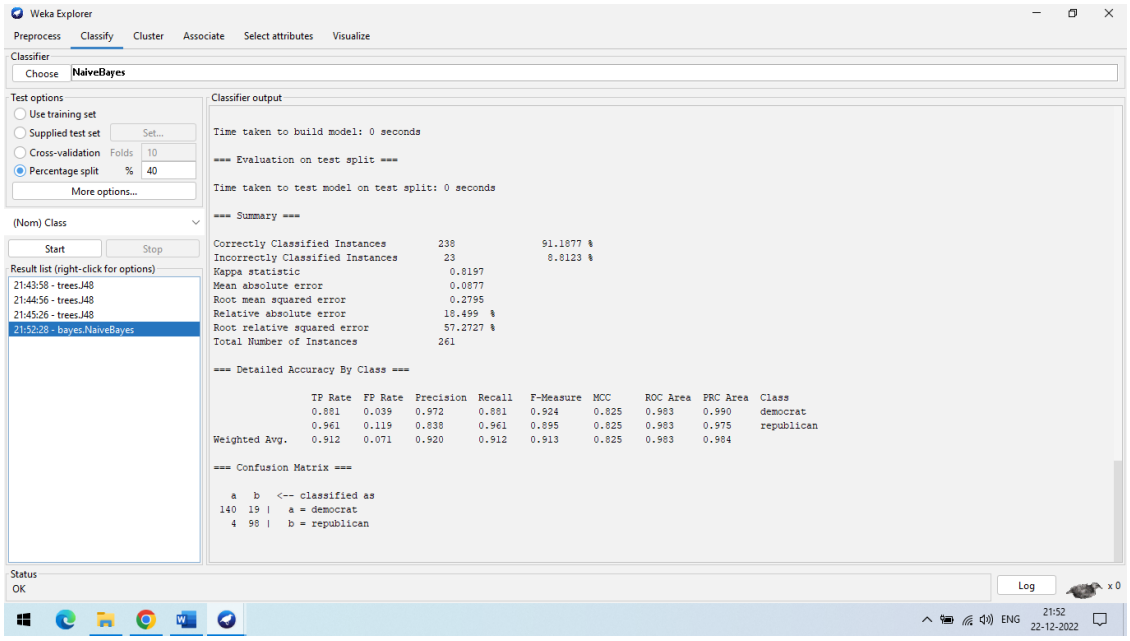
Step 8: right click on result list >> visualize tree



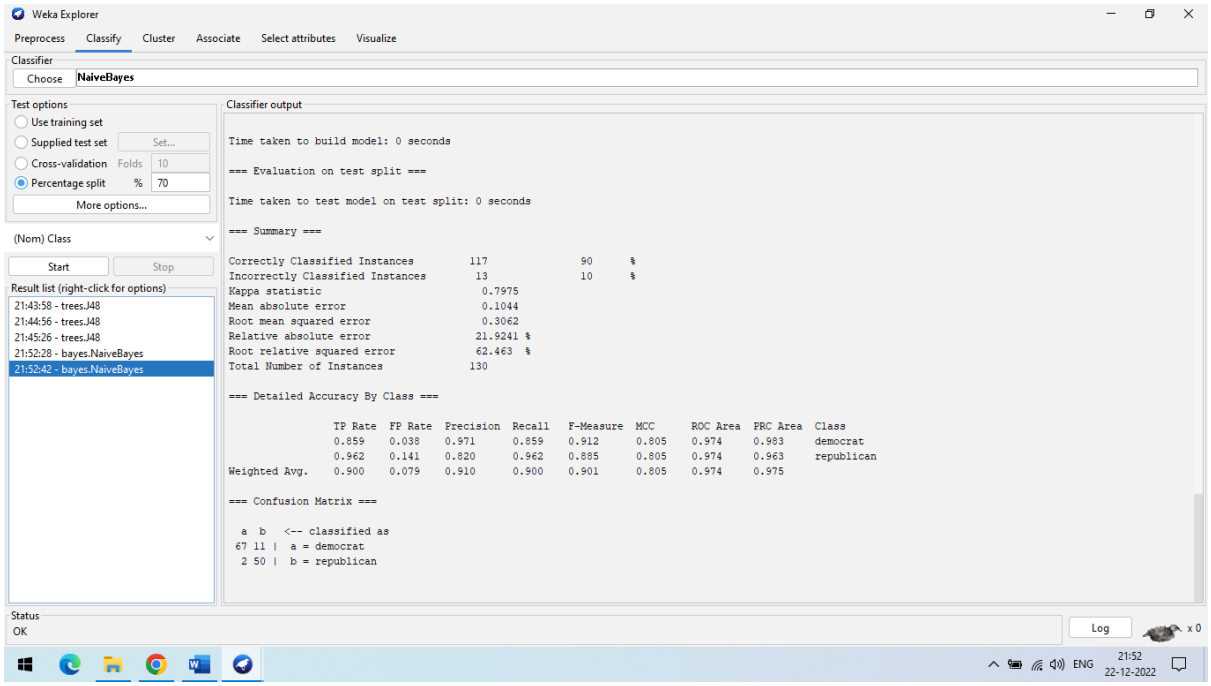
(B) Using Naïve Bayes Classifier: -

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 based on the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

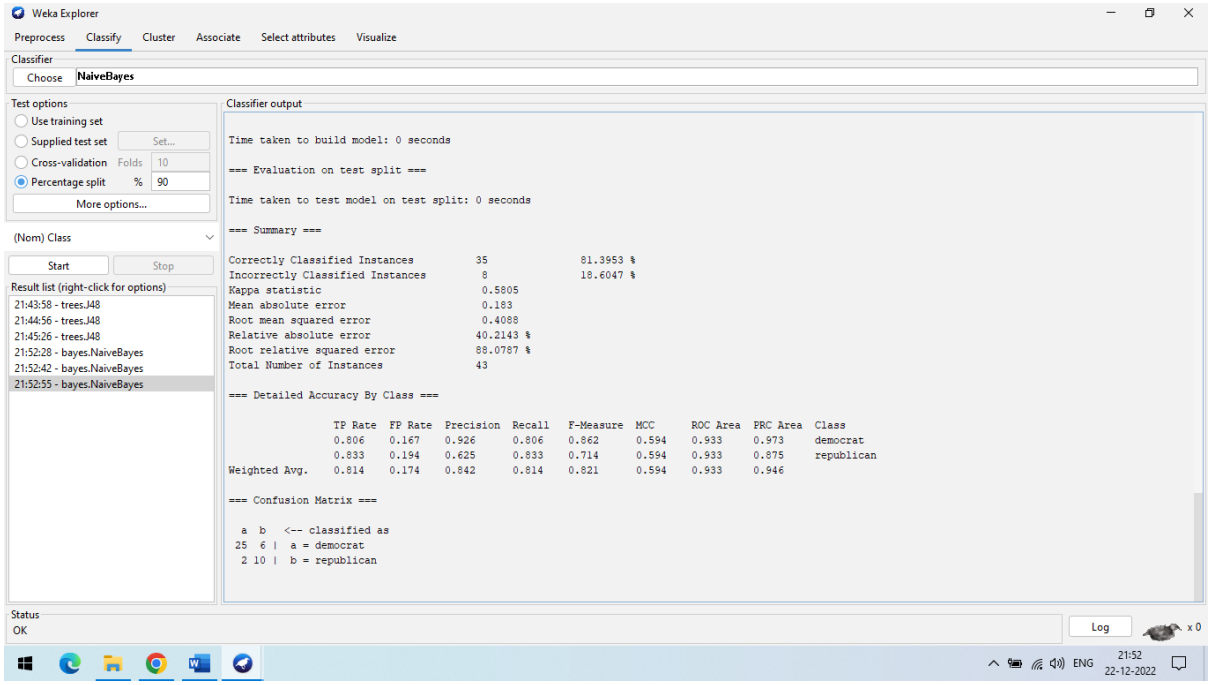
Step 1: Go to the classify >> choose filter NaiveBayes >> give percentage split(40%) >> click start.



Step 2: Go to the classify >> choose filter NaiveBayes >> give percentage split(70%) >> click start.



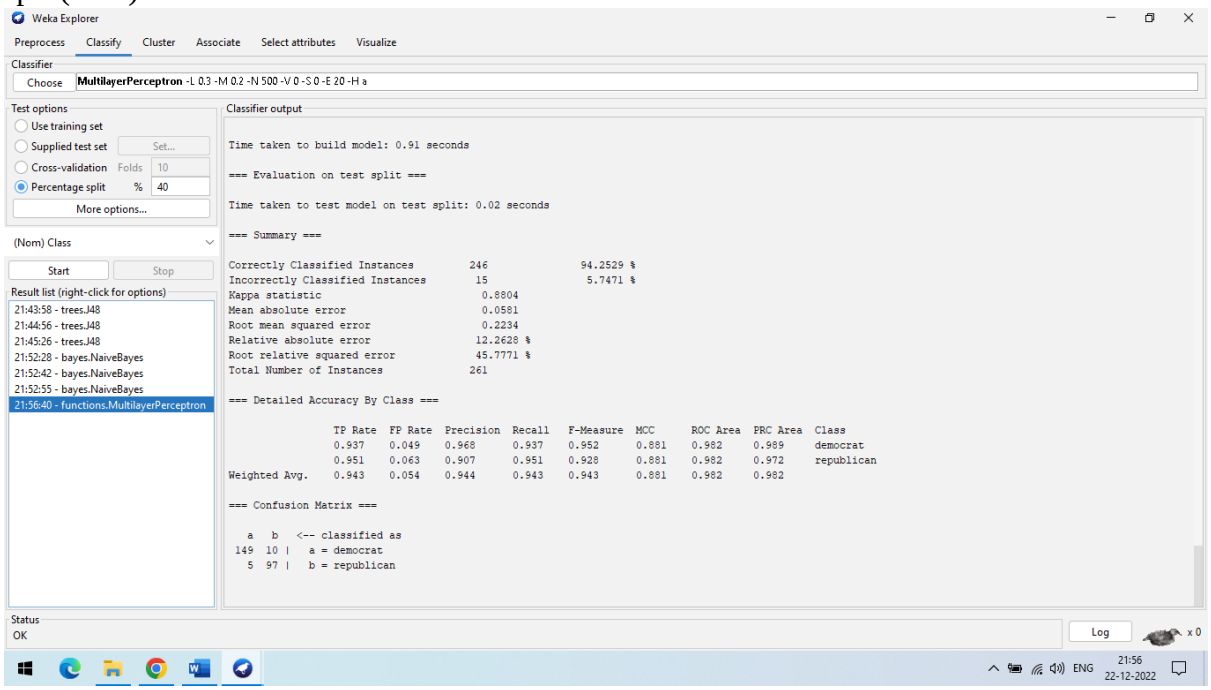
Step 3: Go to the classify >> choose filter NaiveBayes >> give percentage split(90%) >> click start.



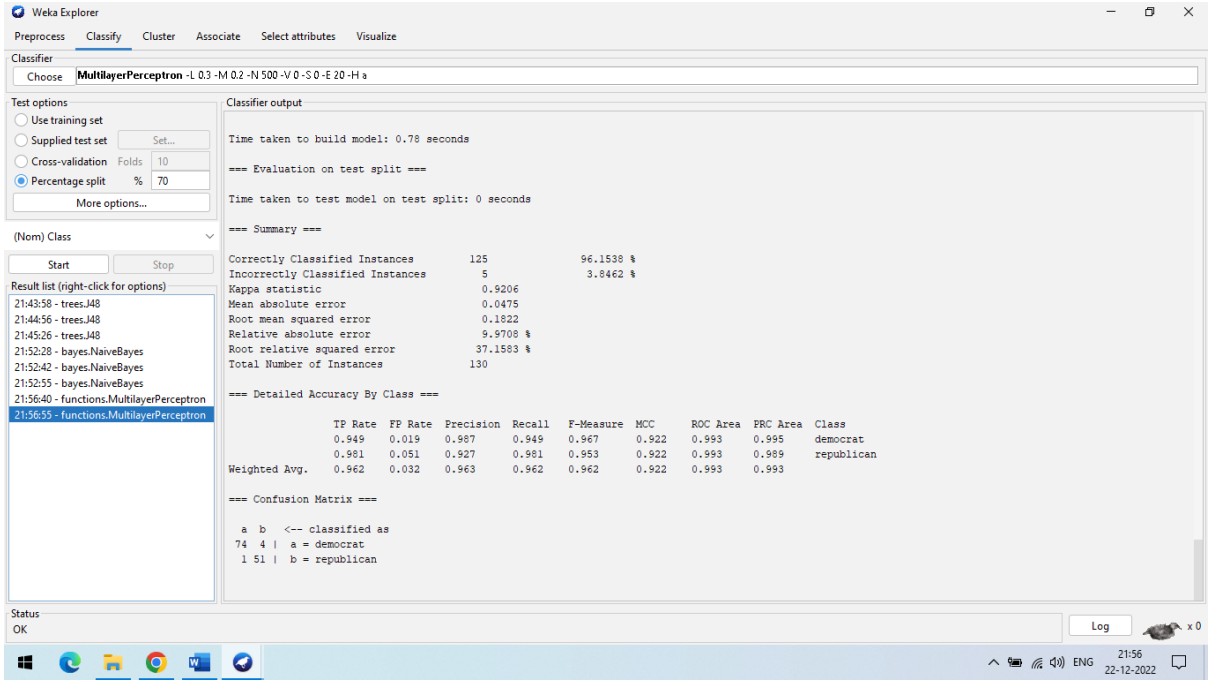
(c) Using Multilayer Perceptron:-

Multi-layer perception is also known as MLP. It is fully connected dense layers, which transform any input dimension to the desired dimension. A multi-layer perception is a neural network that has multiple layers. To create a neural network we combine neurons together so that the outputs of some neurons are inputs of other neurons. In the multi-layer perceptron diagram above, we can see that there are three inputs and thus three input nodes and the hidden layer has three nodes. The output layer gives two outputs, therefore there are two output nodes. The nodes in the input layer take input and forward it for further process, in the diagram above the nodes in the input layer forwards their output to each of the three nodes in the hidden layer, and in the same way, the hidden layer processes the information and passes it to the output layer. Every node in the multi-layer perception uses a sigmoid activation function. The sigmoid activation function takes real values as input and converts them to numbers between 0 and 1 using the sigmoid formula. Now that we are done with the theory part of multi-layer perception, let's go ahead and implement some code in python using the TensorFlow library.

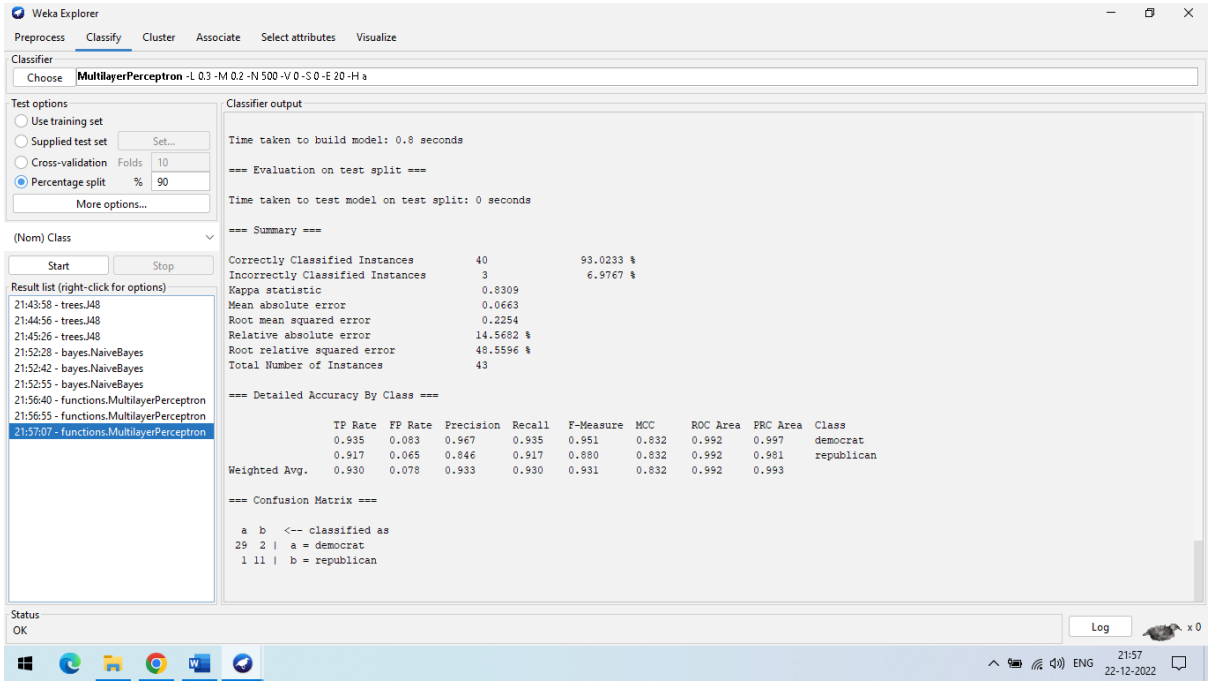
Step 1: Go to the classify >> choose filter MultilayerPerceptron >> give percentage split(40%) >> click start.



Step 2: Go to the classify >> choose filter MultilayerPerceptron >> give percentage split(70%) >> click start.



Step 3: Go to the classify >> choose filter MultilayerPerceptron >> give percentage split(90%) >> click start.



❖ Accuracy Table For Classification:

Split Percentage (%)	Accuracy (%)			Total instances					
	J48	Naïve Bayes	Multilayer Perceptron	J48		Naïve Bayes		Multilayer Perceptron	
				Correctly identified	Incorrectly identified	Correctly identified	Incorrectly identified	Correctly identified	Incorrectly identified
70	96.1538	90	96.1538	125	5	117	13	125	5
40	93.8697	91.1877	94.2529	245	16	238	23	246	15
90	95.3488	81.3953	93.0233	41	2	35	8	40	3

PRACTICAL-10

AIM: Perform Clustering using WEKA tool.

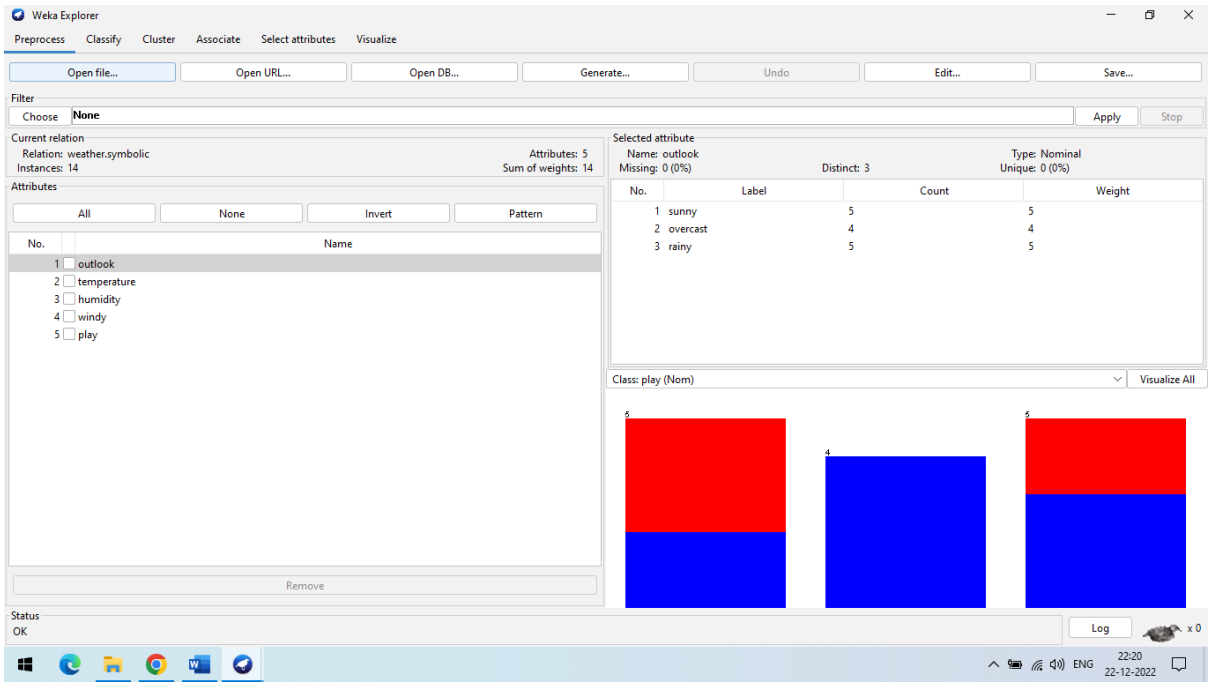
Steps:

A clustering algorithm finds groups of similar instances in the entire dataset.Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. WEKA supports several clustering algorithms such as EM, FilteredClusterer, HierarchicalClusterer, SimpleKMeans and so on.

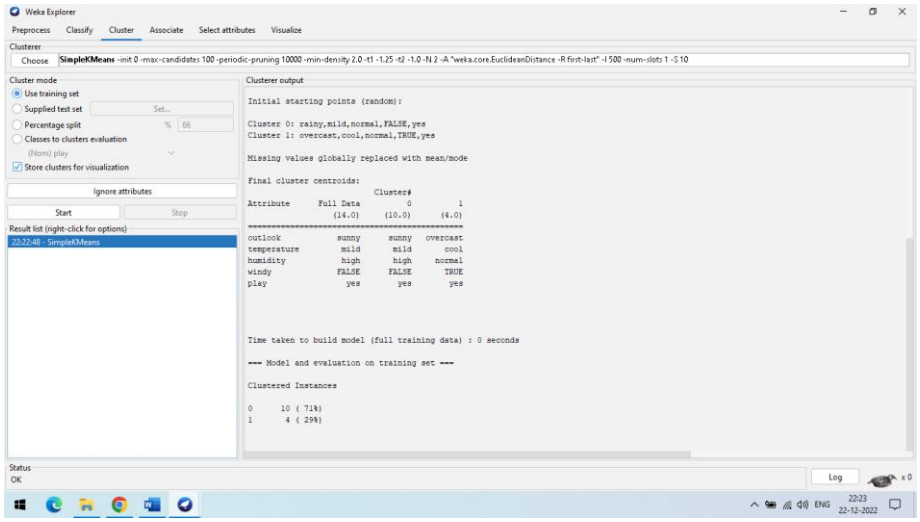
1.Using Simplekmeans

The WEKA SimpleKMeans algorithm uses Euclidean distance measure to compute distances between instances and clusters.

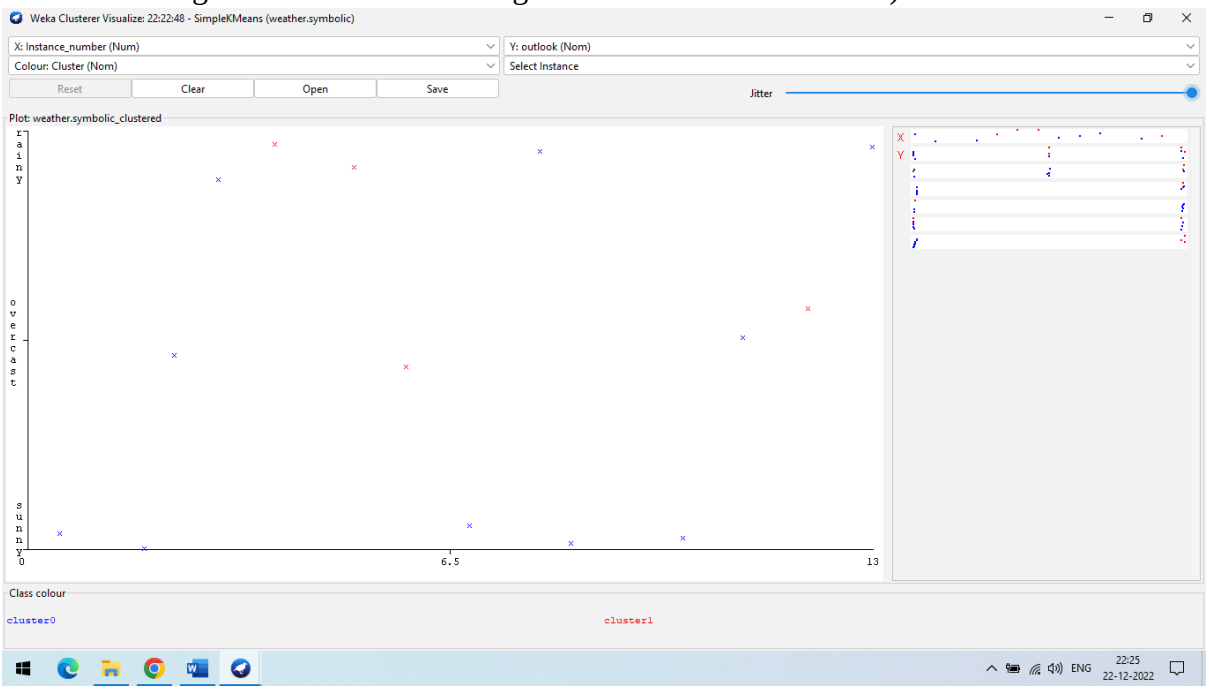
Step 1: open the WEKA tool and select the nominal data in the dataset



Step 2: To perform clustering, select the "Cluster" tab in the Explorer and click on the "Choose" button and choose a simpleKMeans algorithm.and then click start



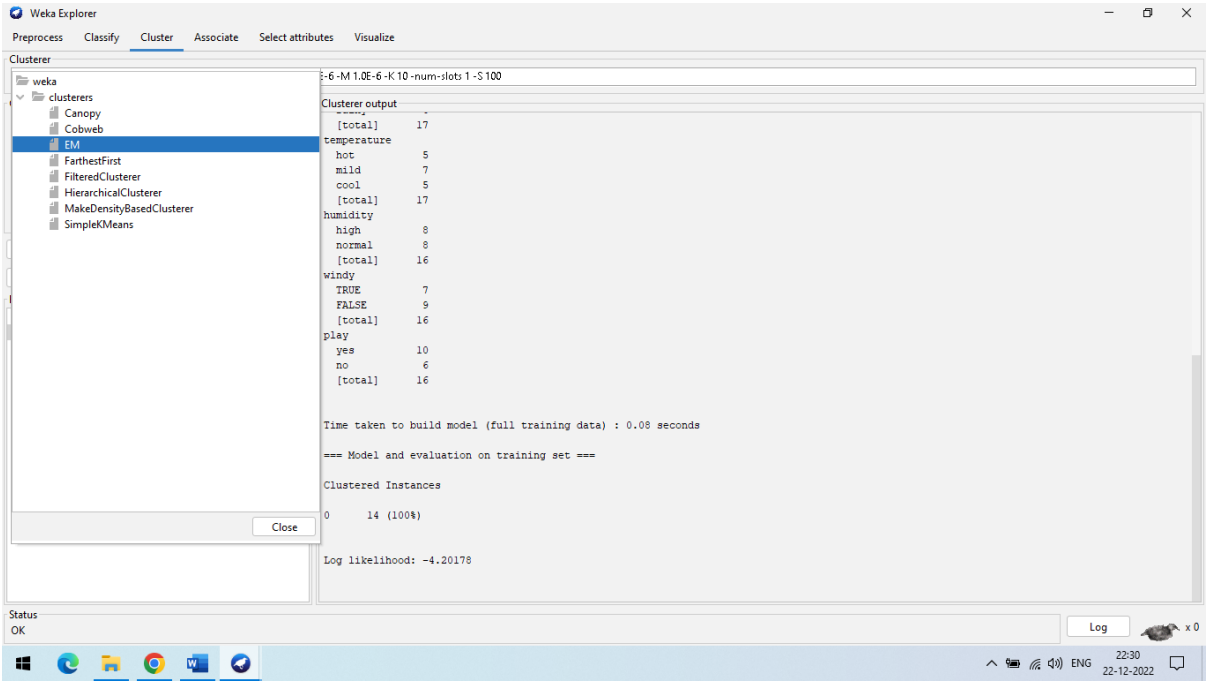
Step 3: Right click on simpleKMeans -> visualize cluster assignment(Another way to grasp the characteristics of each cluster is to visualize them. To do so, right-click the result set on the result. Selecting to visualize cluster assignments from the list column)



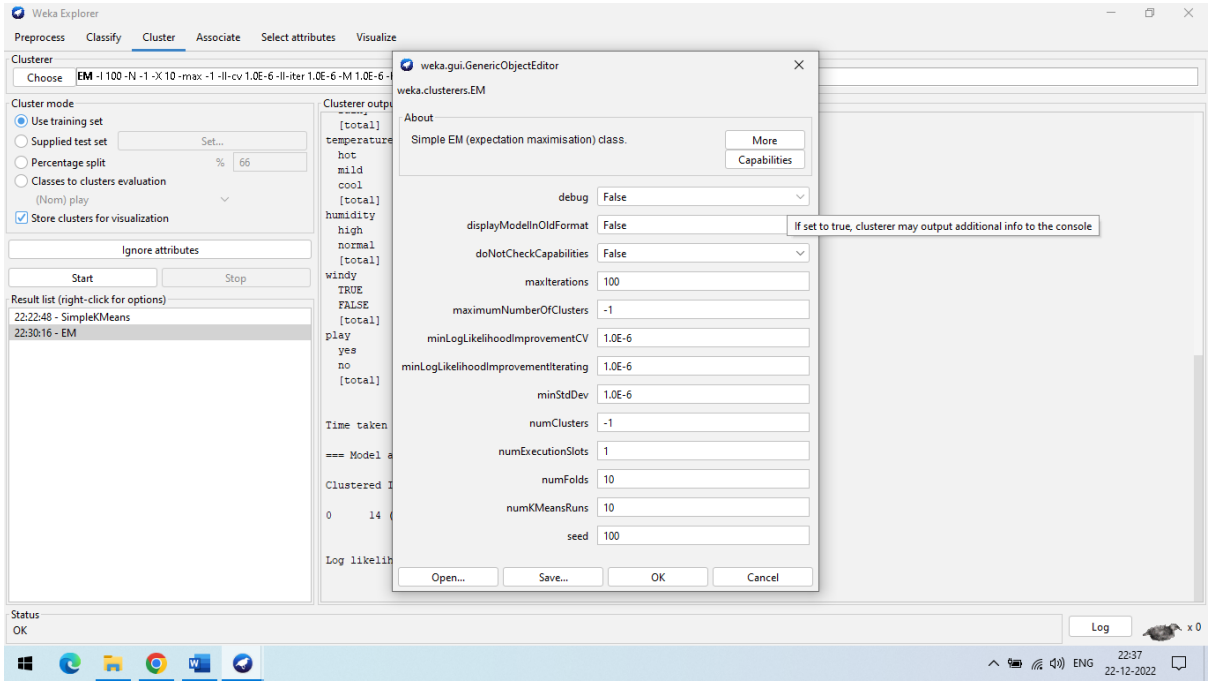
2. Using EM Clustering

In Weka EM selects the number of clusters automatically by maximizing the logarithm of the likelihood of future data, estimated using cross-validation. Beginning with one cluster, it continues to add clusters until the estimated log-likelihood decreases.

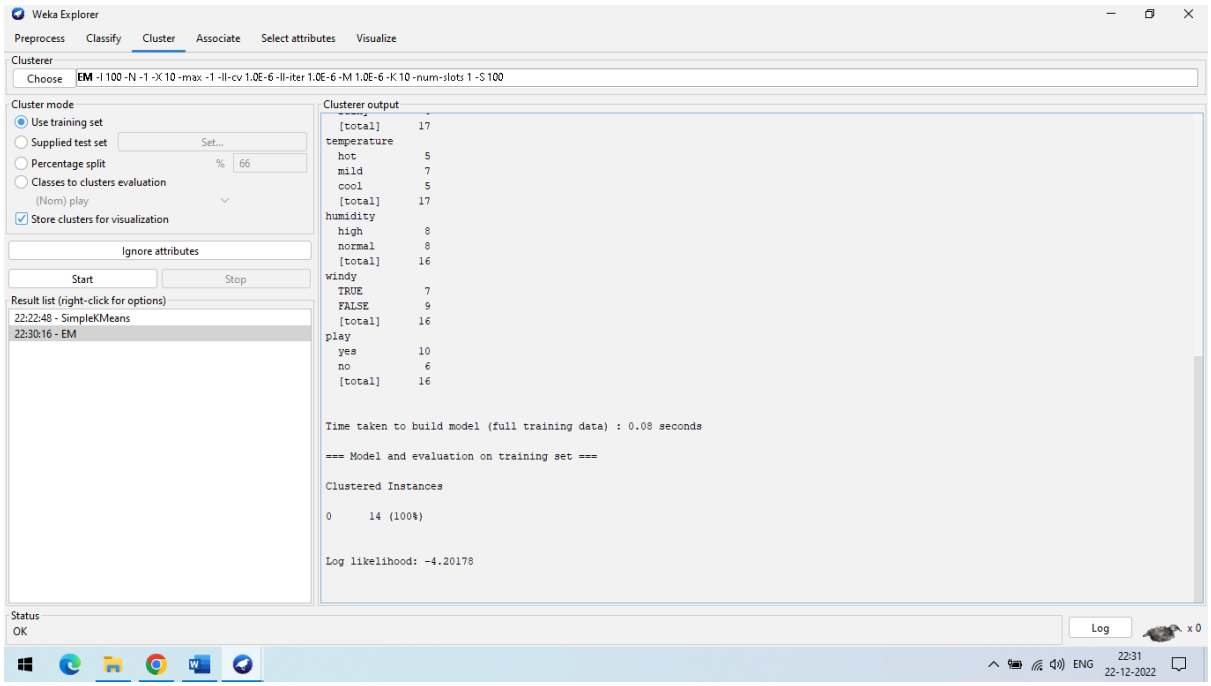
Step 1: weka→explorer→open file→weather.nominal.arff->cluster-> EM algorithm
(To perform clustering, select the "Cluster" tab in the Explorer and click on the "Choose" button and choose a EM algorithm.)



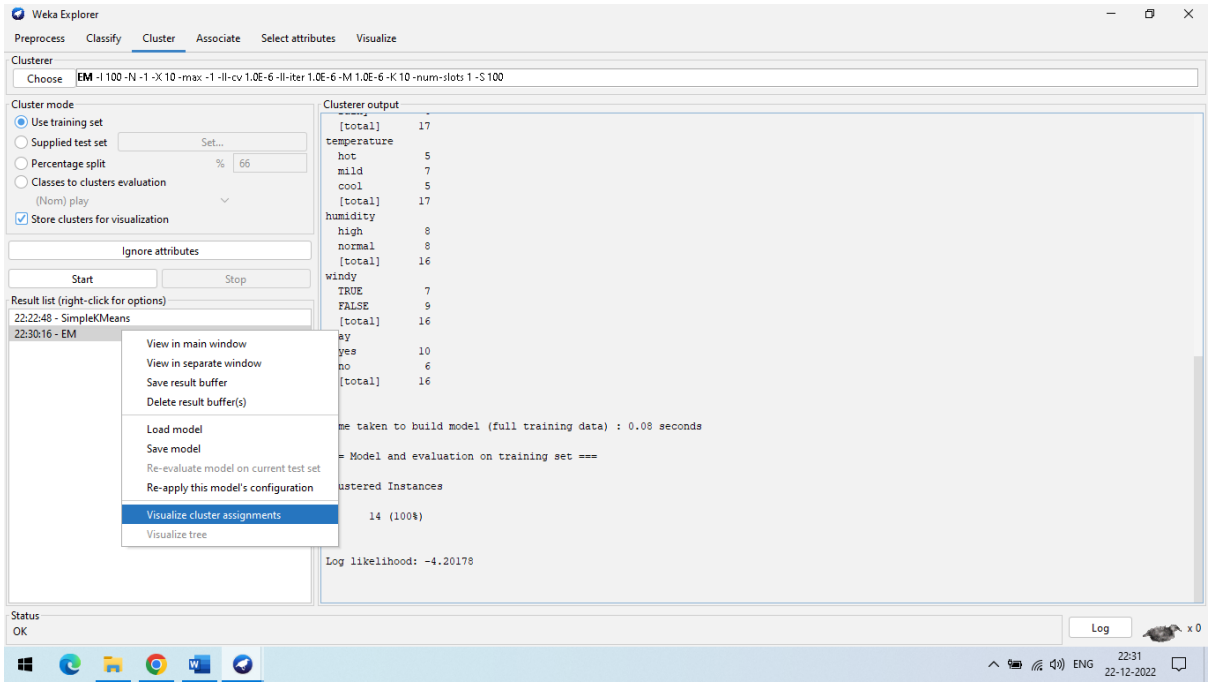
Step 2: Click Cluster to apply the clustering algorithm to our loaded data click on the choose button and change the value if you want**.**



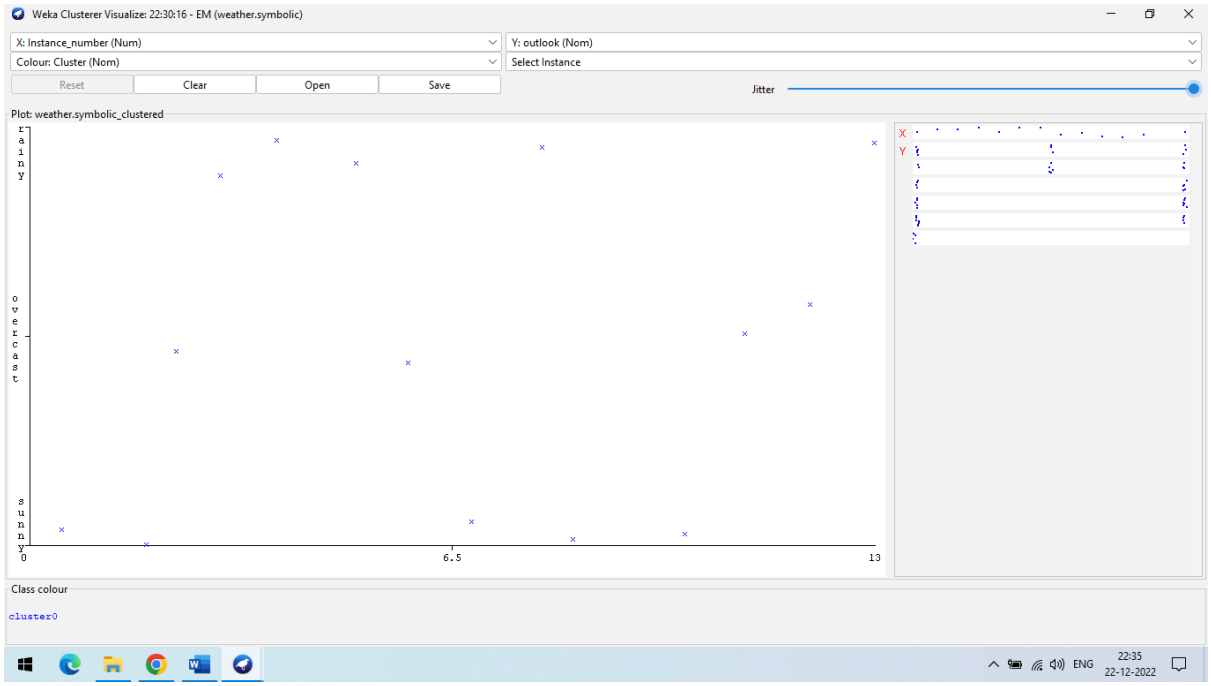
Step 3: After that, Select EM to Start the Clustering.



Step 4 Right-click on EM in result section(Another way to grasp the characteristics of each cluster is to visualise them. To do so, right-click the result set on the result. Selecting to visualise cluster assignments from the list column.)



Step 5 Click on visualize cluster assignment .(Another way to grasp the characteristics of each cluster is to visualise them. To do so, right-click the result set on the result. Selecting to visualise cluster assignments from the list column.)



PRACTICAL-6

AIM: Perform different binning techniques to smooth out the noise in the dataset. Make sure that the user should have the choice to apply all the possible techniques. Show the output of different bins. Use histogram to partition the dataset into groups.

Theory:

Binning:

Why is Binning Used?

Binning or discretization is used to transform a continuous or numerical variable into a categorical feature. Binning of continuous variables introduces non-linearity and tends to improve the performance of the model. It can also be used to identify missing values or outliers.

What is the Purpose of Binning Data?

Binning, also called discretization, is a technique for reducing continuous and discrete data cardinality. Binning groups related values together in bins to reduce the number of distinct values.

Example of Binning

Histograms are an example of data binning used to observe underlying distributions. They typically occur in one-dimensional space and equal intervals for ease of visualization.

Approaches to perform smoothing:

1. **Smoothing by bin means:**

➢ In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

2. Smoothing by bin median:

➢ In this method each bin value is replaced by its bin median value.

3. Smoothing by bin boundary:

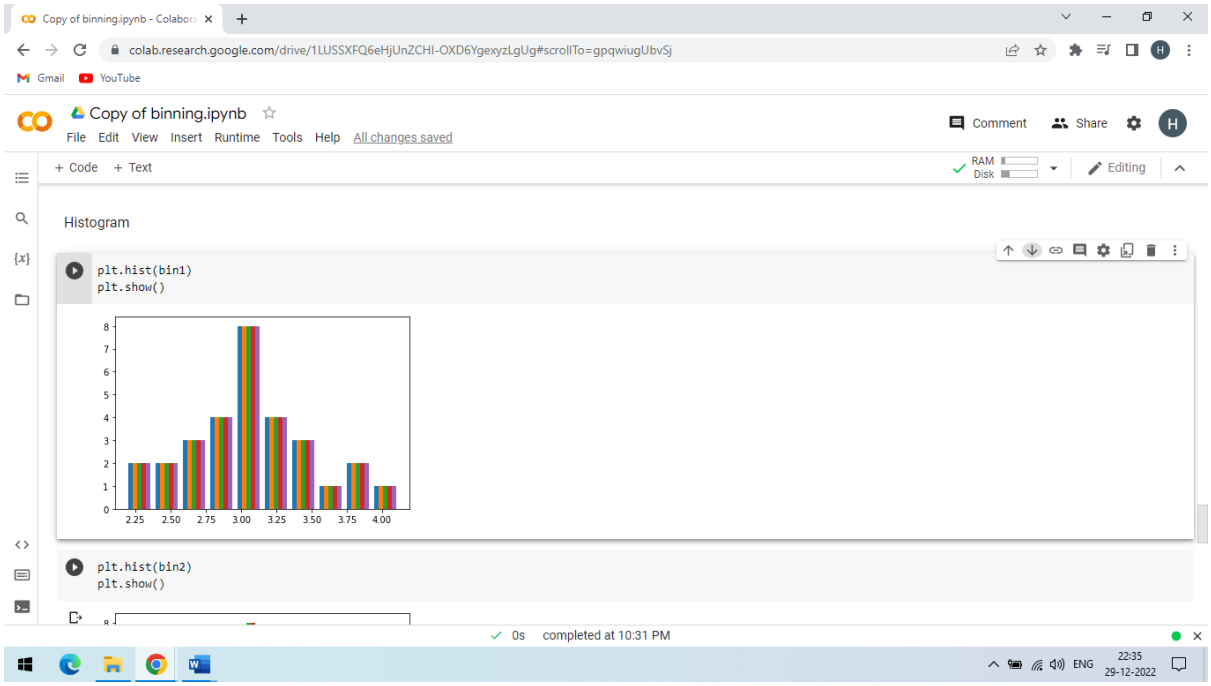
➢ In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

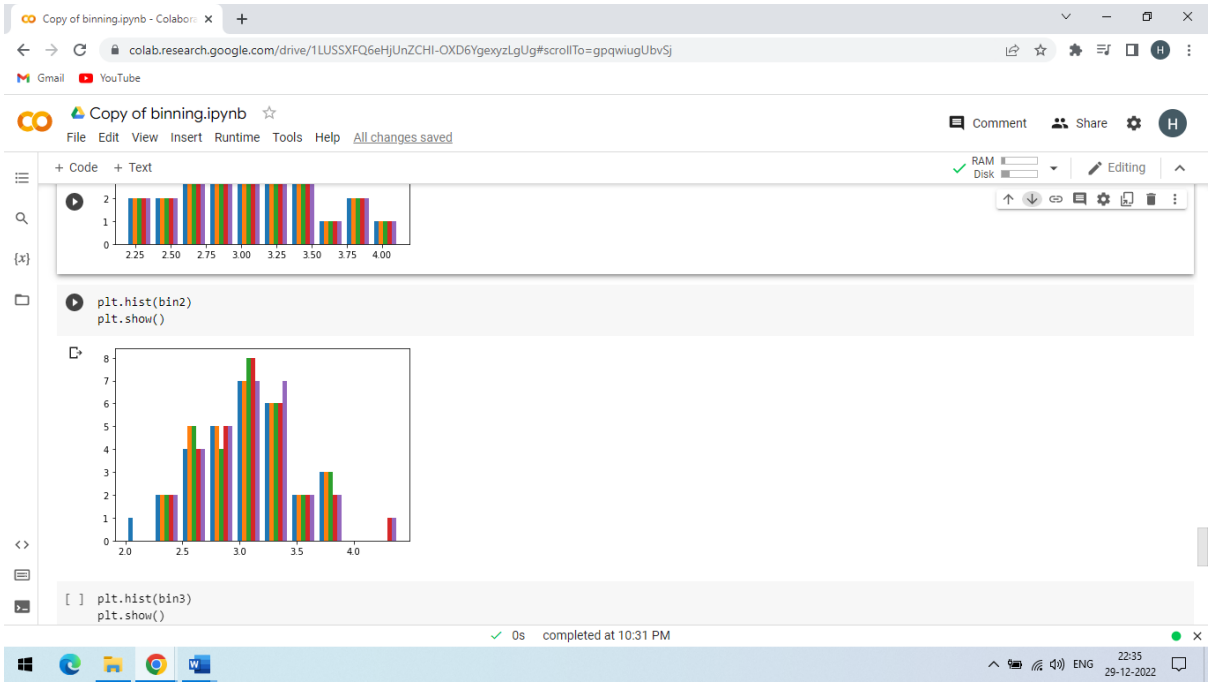
Code:

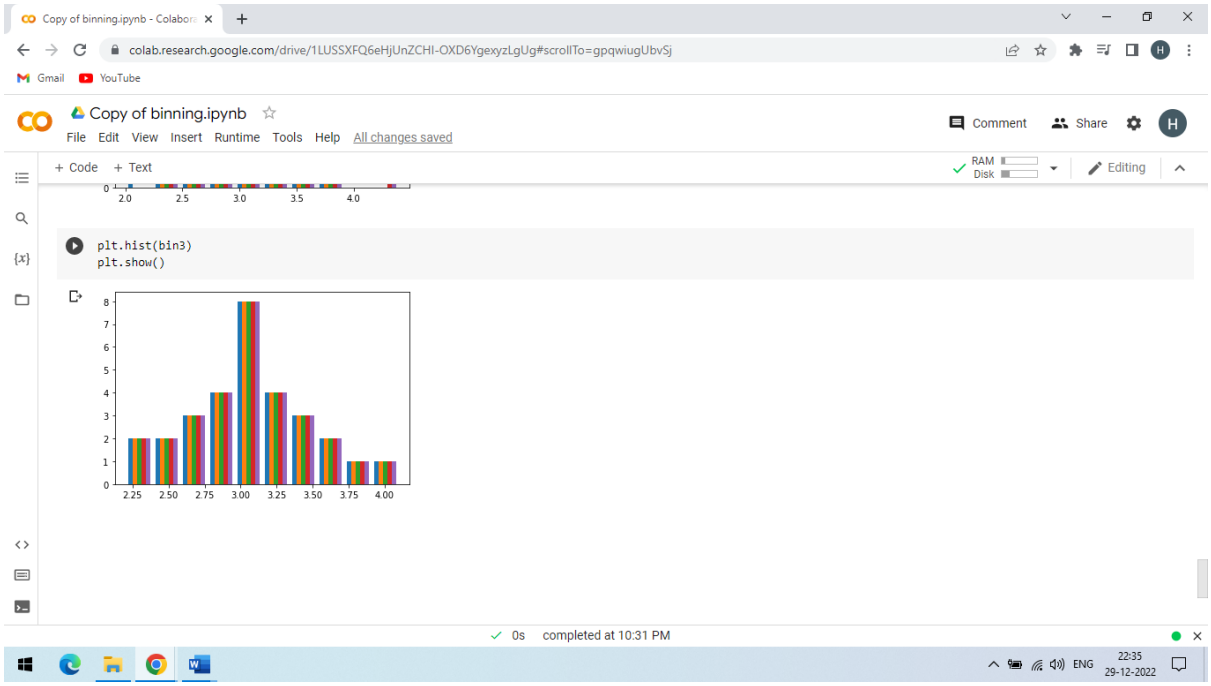
```
import numpy as np
import math
from sklearn.datasets import load_iris
from sklearn import datasets, linear_model, metrics
import matplotlib.pyplot as plt
# load iris data set
dataset = load_iris()
print(dataset)
a = dataset.data
b = np.zeros(150)
print(dataset)
# take 1st column among 4 column of data set
for i in range (150):
  b[i]=a[i,1]
#sort the array
b=np.sort(b)
```

```
print(b.size)
# create bins
bin1=np.zeros((30,5))
bin2=np.zeros((30,5))
bin3=np.zeros((30,5))
print(bin1.size)
# Bin mean
for i in range (0,150,5):
  k=int(i/5)
  mean=(b[i] + b[i+1] + b[i+2] + b[i+3] + b[i+4])/5
  for j in range(5):
    bin1[k,j]=mean
print("Bin Mean: \n",bin1)
# Bin boundaries
for i in range (0,150,5):
  k=int(i/5)
  for j in range (5):
    if (b[i+j]-b[i]) < (b[i+4]-b[i+j]):
      bin2[k,j]=b[i]
    else:
      bin2[k,j]=b[i+4]
print("\n")
print("Bin Boundaries: \n",bin2)
# Bin median
for i in range (0,150,5):
  k=int(i/5)
  for j in range (5):
   bin3[k,j]=b[i+2]
print("Bin Median: \n",bin3)
```

Output:


Copy of binning.ipynb - Colabora
colab.research.google.com/drive/1LUSSXFQ6eHjUnZCHI-OXD6YgexyzLgUg#scrollTo=gpqwiugUbvSj
Gmail
YouTube
Copy of binning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Histogram
plt.hist(bin1)
plt.show()
plt.hist(bin2)
plt.show()
0s completed at 10:31 PM
ENG
22:35
29-12-2022


Copy of binning.ipynb - Colabora
colab.research.google.com/drive/1LUSSXFQ6eHjUnZCHI-OXD6YgexyzLgUg#scrollTo=gpqwiugUbvSj
Gmail
YouTube
Copy of binning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
plt.hist(bin2)
plt.show()
[ ] plt.hist(bin3)
plt.show()
0s completed at 10:31 PM
ENG
22:35
29-12-2022


Copy of binning.ipynb - Colabora
colab.research.google.com/drive/1LUSSXFQ6eHjUnZCHI-OXD6YgexyzLgUg#scrollTo=gpqwiugUbvSj
Gmail
YouTube
Copy of binning.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
plt.hist(bin3)
plt.show()
0s completed at 10:31 PM
ENG
22:35
29-12-2022

PRACTICAL 7

AIM:- Perform regression on the data set using R programming.

Solution:

Theory:

Regression :-

- Regression refers to a type of supervised machine learning technique that is used to predict any continuous-valued attribute.
- Regression involves the technique of fitting a straight line or a curve on numerous data points. It happens in such a way that the distance between the data points and cure comes out to be the lowest.

Regression Analysis in R:-

- Regression analysis is a group of statistical processes used in R programming and statistics to determine the relationship between dataset variables. Generally, regression analysis is used to determine the relationship between the dependent and independent variables of the dataset.
- Regression analysis helps to understand how dependent variables change when one of the independent variables is changes and other independent variables are kept constant. This helps in building a regression model and further, helps in forecasting the values with respect to a change in one of the independent variables.
- On the basis of types of dependent variables, a number of independent variables, and the shape of the regression line, there are 4 types of regression analysis techniques i.e., Linear Regression, Logistic Regression, Multinomial Logistic Regression and Ordinal Logistic Regression.

Code :

```
data <- read.csv('D:/Hemil.csv')
View(data)
nrow(data)
ncol(data)
X <- data$weight
Y <- data$height
relation <- lm(Y~X)
plot(X,Y)
plot(X,Y,col = "blue",main = "X & Y Regression",cex = 1,pch = 16,xlab = "X",ylab = "Y")
abline(relation)
summary(relation)
```

OUTPUT:

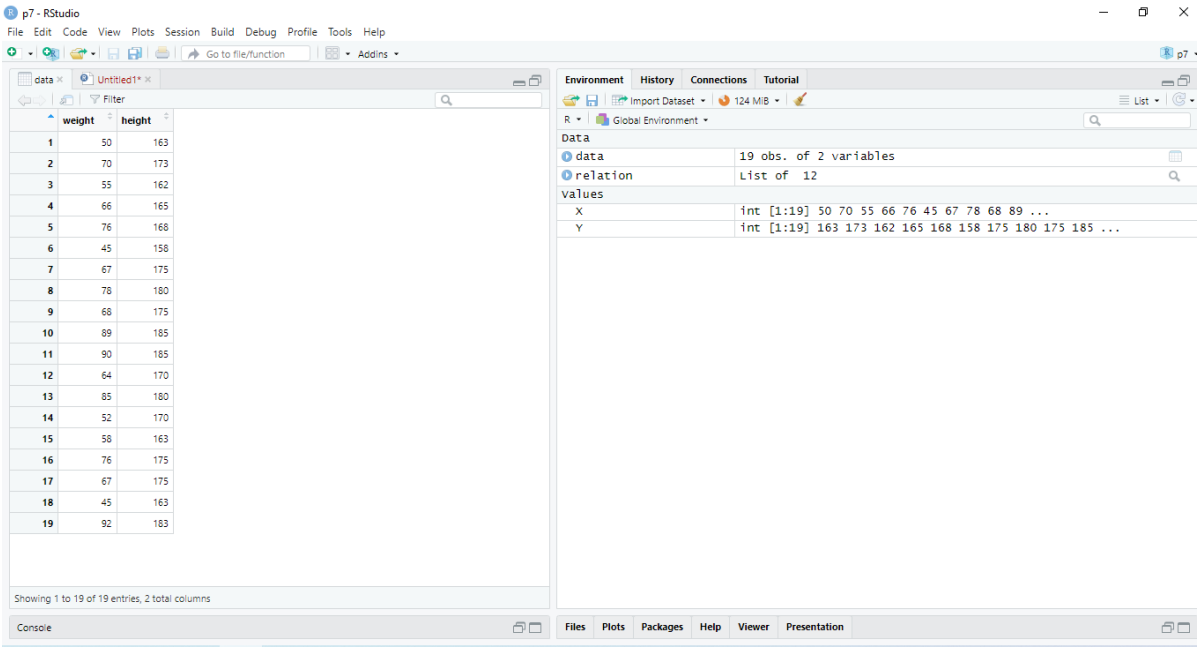


Figure 1 Display the relationship between dataset variables

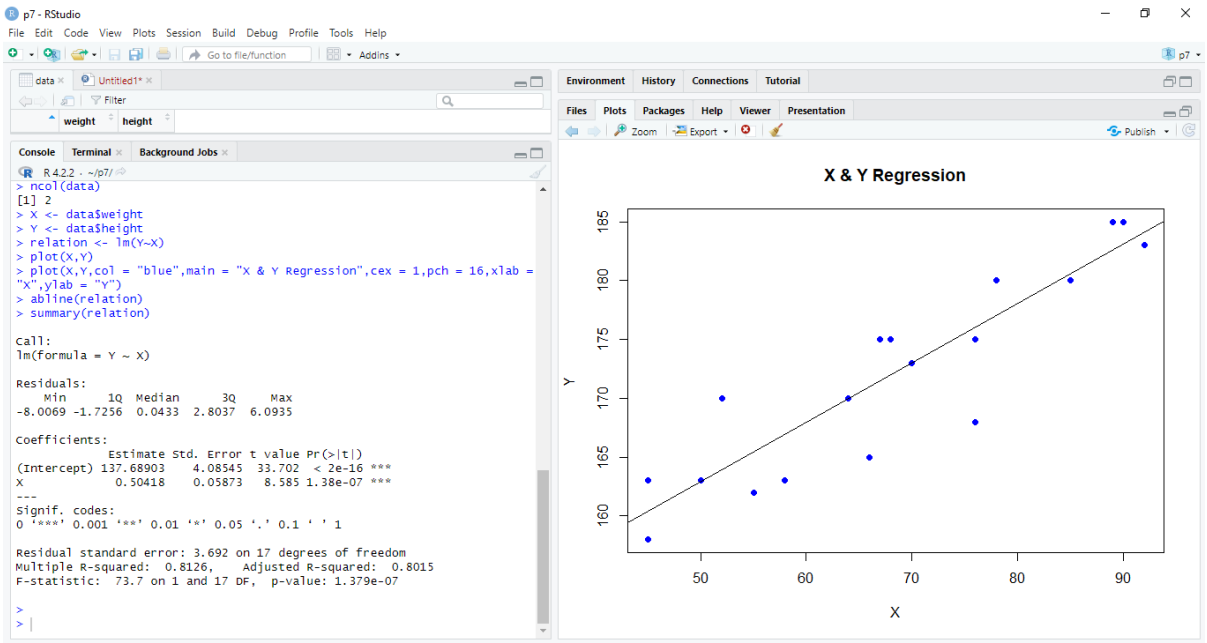


Figure 2 X&Y Regression Plots Diagram

PRACTICAL-1

AIM:- Design and Create cube by identifying measures and dimensions for Star Schema, Snowflake schema and fact Constellation Schema.

Solution:

Software Used: Analysis services- SQL Server-2008

Theory:

1) Creating a New Analysis Services Project

To create a new Analysis Services project, use the New Project dialog box in BIDSTo create a new Analysis Services project, follow these steps:

1. Select Microsoft SQL Server 2008 ⇒SQL Server Business Intelligence. Development Studio from the Programs menu to launch Business Intelligence Development Studio.
2. Select File ⇒ New Project. ⇒
3. In the New Project dialog box, select the Business Intelligence Projects project type.
4. Select the Analysis Services Project template.
5. Name the new project AdventureWorksCube1 and select a convenient location to save it.
6. Click OK to create the new project.

(1) **Star Schema:**

- The star schema architecture is the simplest data warehouse schema.
- It is called a star schema because the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. Usually the fact tables in a star schema are in third normal form(3NF) whereas dimensional tables are de-normalized.
- Despite the fact that the star schema is the simplest architecture, it is most commonly used nowadays and is recommended by Oracle.

Fact Tables:-

- A fact table typically has two types of columns: foreign keys to dimension tables and measures those that contain numeric facts.
- A fact table can contain fact's data on detail or aggregated level.

Dimension Tables:-

- A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension hasn't got a hierarchies and levels it is called flat dimension or list.
- The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value.
- They are normally descriptive, textual values. Dimension tables are generally small in size then fact table.

The main characteristics of star schema:

- Simple structure -> easy to understand schema
- Great query effectives -> small number of tables to join
- Relatively long time of loading data into dimension tables -> de-normalization, redundancy data caused that size of the table could be large.
- The most commonly used in the data warehouse implementations -> widely supported by a large number of business intelligence tools.

(1) Snowflake Schema:

- A snowflake schema is a logical arrangement of tables in a multidimensional database such that the entity relationship diagram resembles a snowflake shape.
- The snowflake schema is represented by centralized fact tables which are connected to multiple dimensions.
- Star and snowflake schemas are most commonly found in dimensional data warehouses and data marts where speed of data retrieval is more important than the efficiency of data manipulations. As such, the tables in these schemas are not normalized much, and are frequently designed at a level of normalization short of third normal form.
- Normalization splits up data to avoid redundancy (duplication) by moving commonly repeating groups of data into new tables. Normalization therefore tends to increase the number of tables that need to be joined in order to perform a given query, but reduces the space required to hold the data and the number of places where it needs to be updated if the data changes

(2) Fact constellation Schema:

- Fact constellation is a measure of online analytical processing, which is a collection of multiple fact tables sharing dimension tables, viewed as a collection of stars.
- This is an improvement over Star schema.
- For each star schema it is possible to construct fact constellation schema(for example by splitting the original star schema into more star schemes each of them describes facts on another level of dimension hierarchies).
- The fact constellation architecture contains multiple fact tables that share many dimension tables.
- Moreover, dimension tables are still large.**tep 1:** shows the Solution Explorer window of the new project, ready to be populated with objects and Defining a data source.

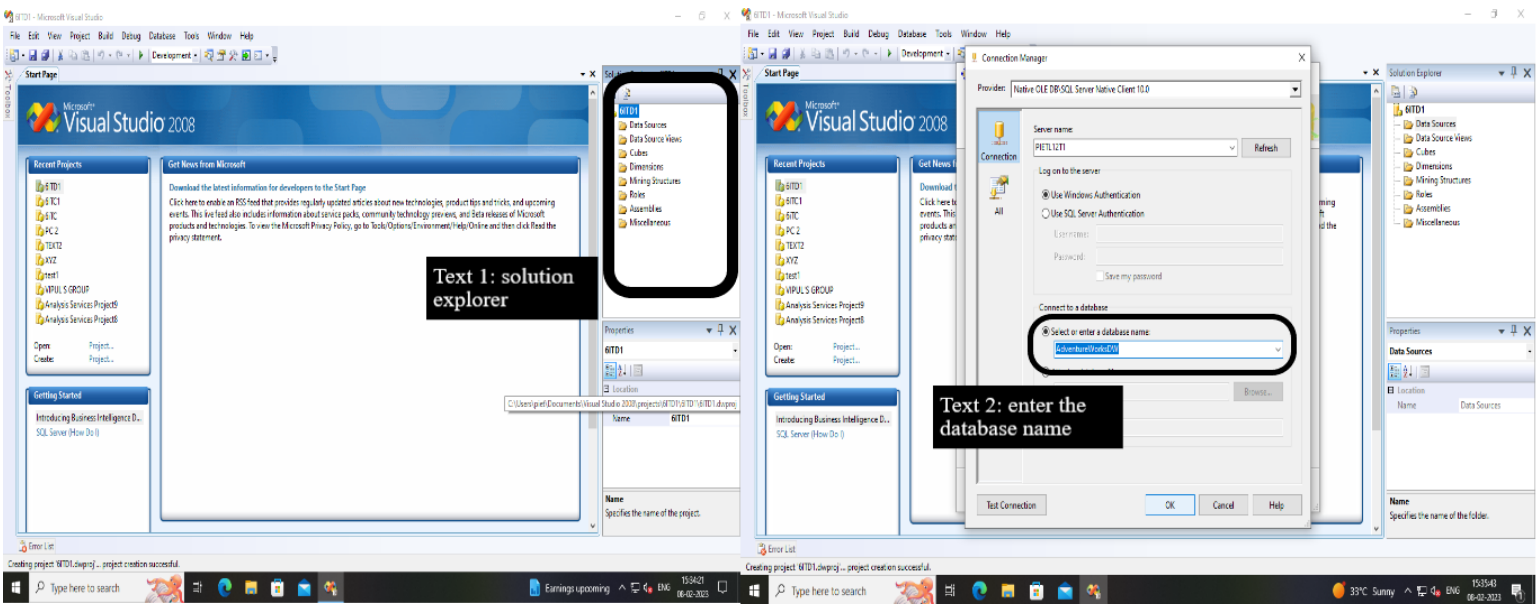


Figure 1 New Analysis Services project& Setting up a connection

step 2: create data source view Wizard select the tables and show the analysis service.

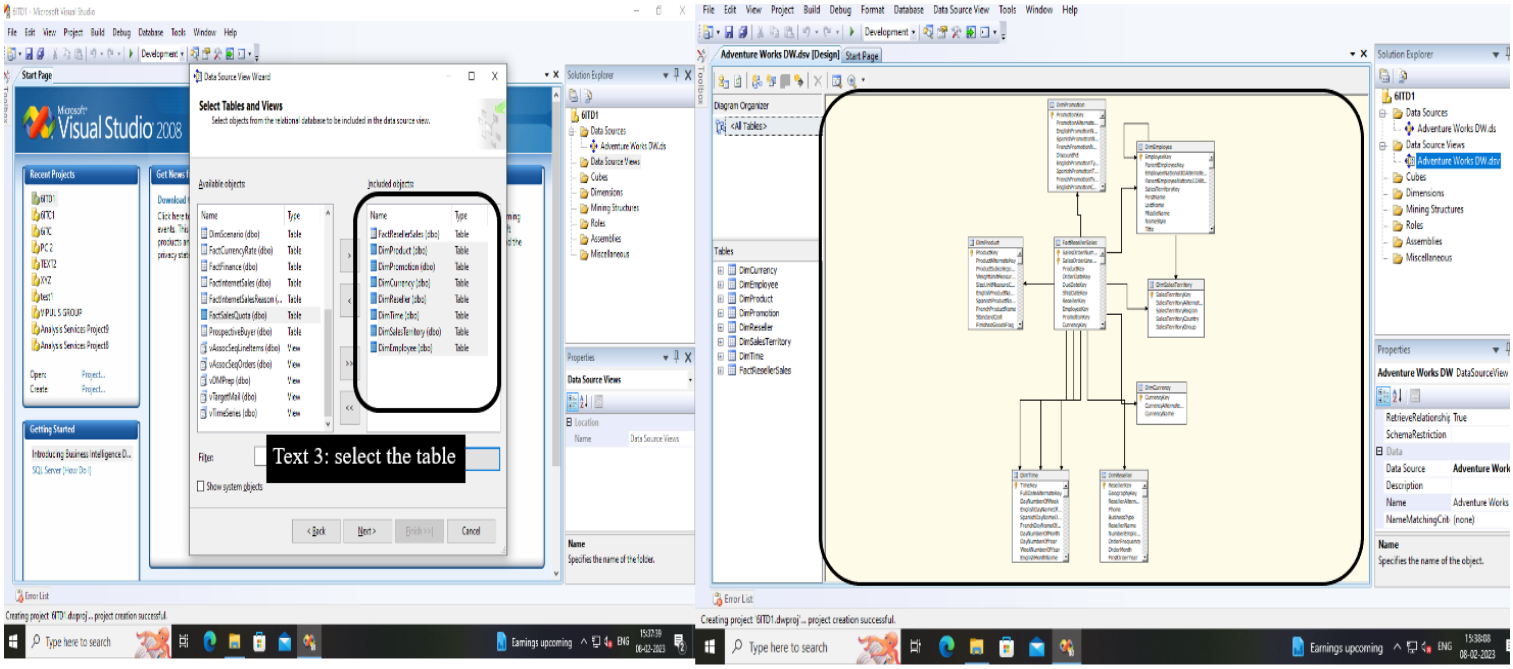


Figure 2 Selecting tables for the data source view & The Sales data source view

step 3: Deploying and Processing a Cube.

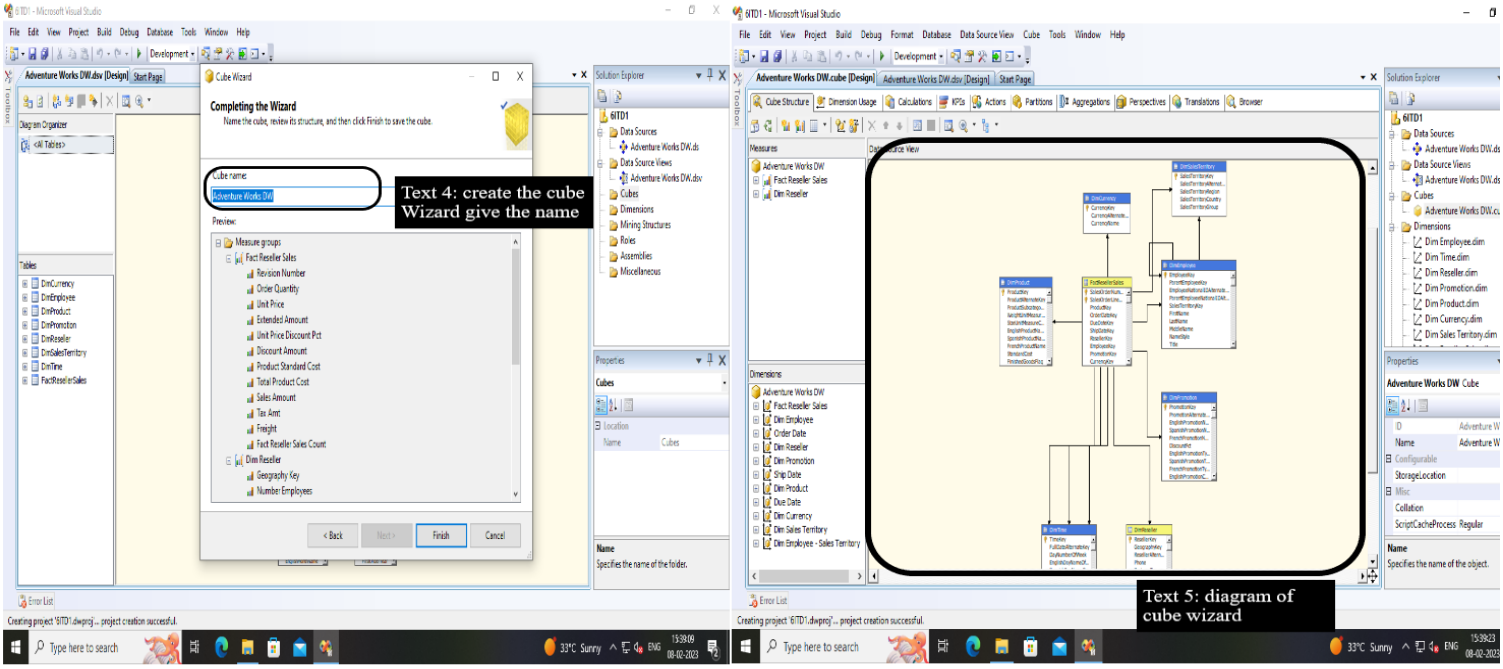


Figure 3 cube wizard diagram

step 4: Create Star Schema, Snowflake schema and fact Constellation Schema.

A) star schema

B) Snowflake schema

C) Fate collection schema

Figure 4 Diagram for schema

PRACTICAL-2

AIM: - Make an OLAP cube and perform Roll Up and Drill Down operations on it. Show the Apex and Base cuboid for the same. Draw Star-net query model for the cube.

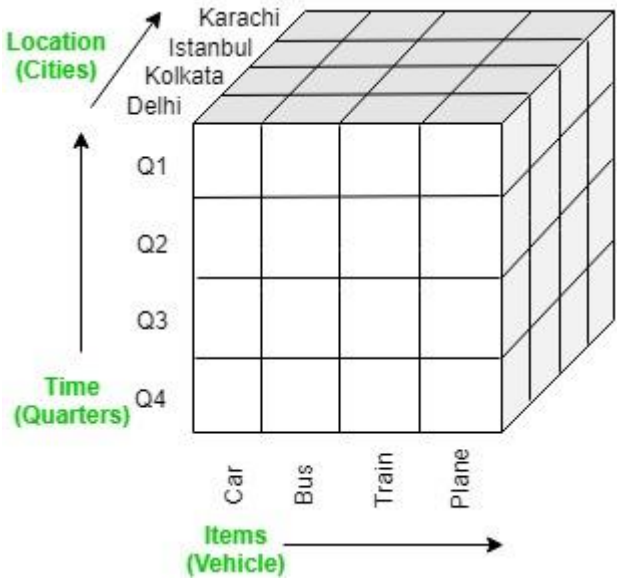
Theory:

Roll-Up:-

The roll-up operation (also known as drill-up or aggregation operation) performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction. Roll-up is like zooming-out on the data cubes. Figure shows the result of roll-up operations performed on the dimension location. The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country. The roll-up operation groups the information by levels of temperature.

Unlike the CUBE subclause, ROLLUP does not yield all possible grouping sets based on the specified columns. It just makes a subset of those. The ROLLUP presupposes a hierarchy between the input columns and yields all grouping sets that make sense only if the hierarchy is considered.

The following diagram illustrates how roll-up works

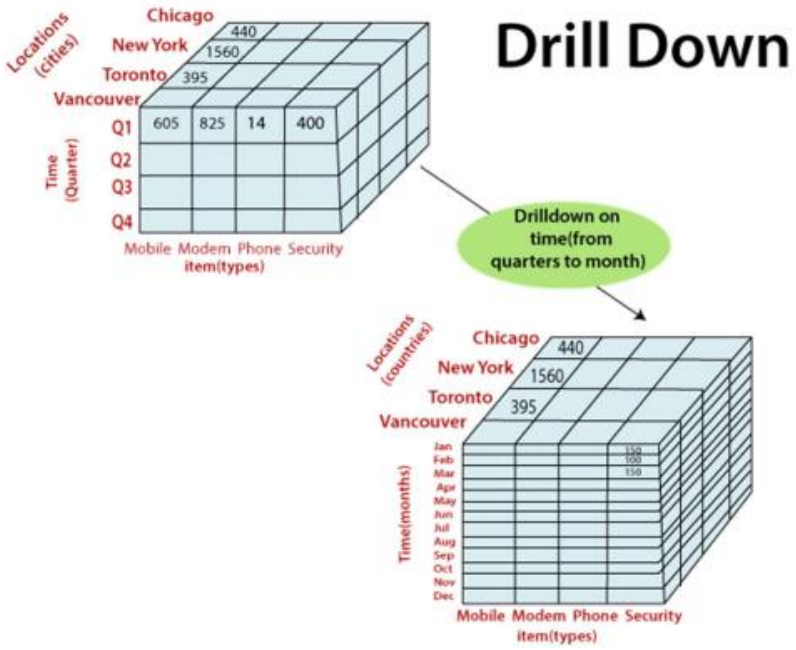


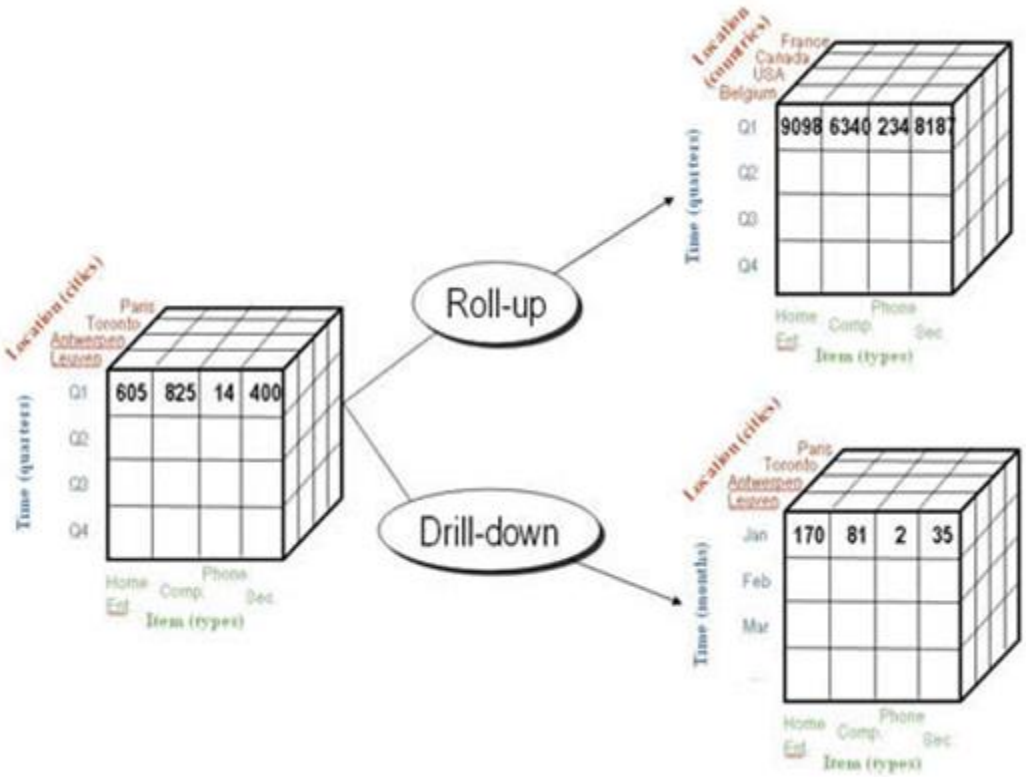
Drill-Down

The drill-down operation (also called roll-down) is the reverse operation of roll-up. Drill-down is like zooming-in on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either stepping down a concept hierarchy for a dimension or adding additional dimensions. Kirti Patel(200303108034) Data Mining & Business Intelligence (203105454) Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more

detailed level of the month. Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

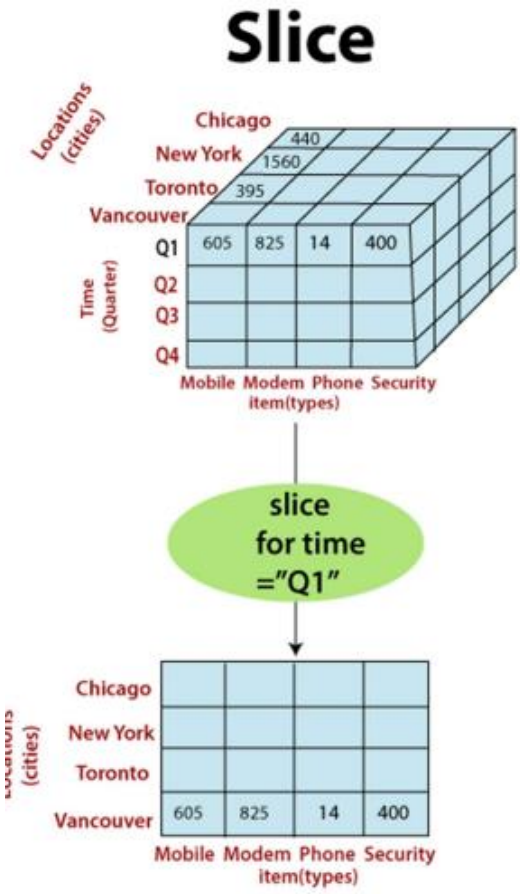
Example Drill-down adds more details to the given data The following diagram illustrates how Drill-down works.





Slice

A slice is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a threedimensional cube resulting in a two-dimensional site. So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube. For example, if we make the selection, temperature=cool we will obtain the following cube: The following diagram illustrates how Slice works.



Here Slice is functioning for the dimensions "time" using the criterion time = "Q1". It will form a new sub-cubes by selecting one or more dimensions.

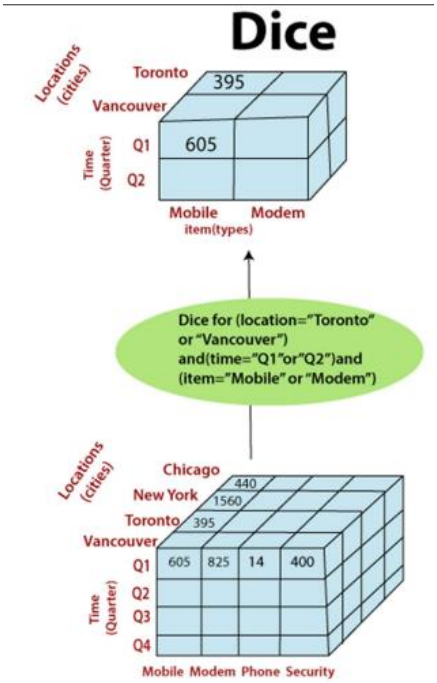
Dice

The dice operation describes a subcube by operating a selection on two or more dimension.

For example, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following subcube (still two-dimensional)

Temperature	cool	hot
Day 3	0	1
Day 4	0	0

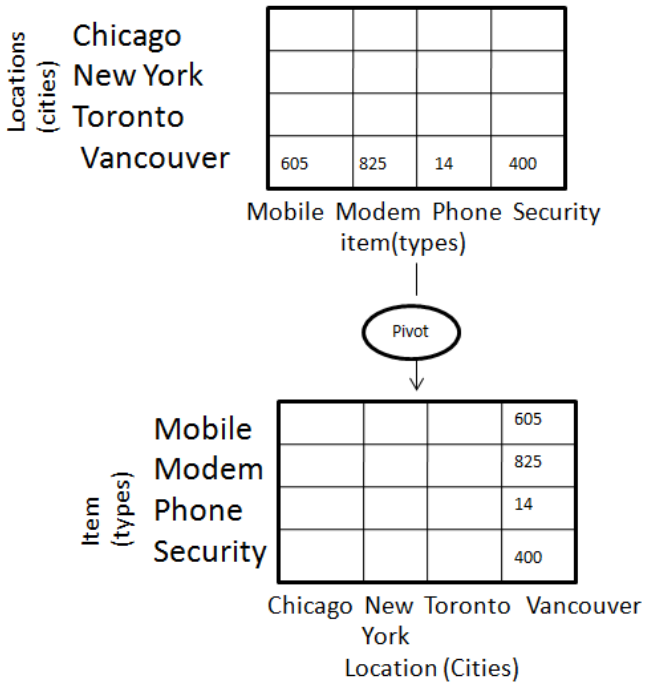
Consider the following diagram, which shows the dice operations.



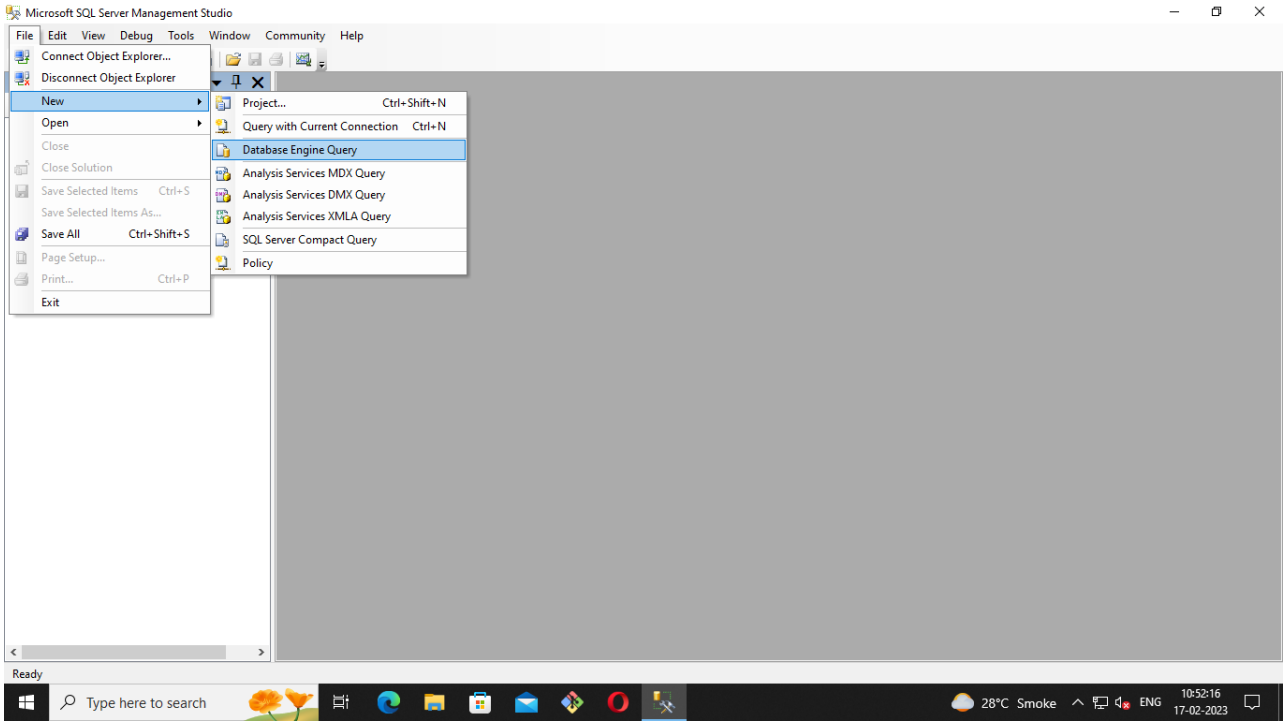
The dice operation on the cubes based on the following selection criteria involves three dimensions. 1. (location = "Toronto" or "Vancouver") 2. (time = "Q1" or "Q2") 3. (item =" Mobile" or "Modem")

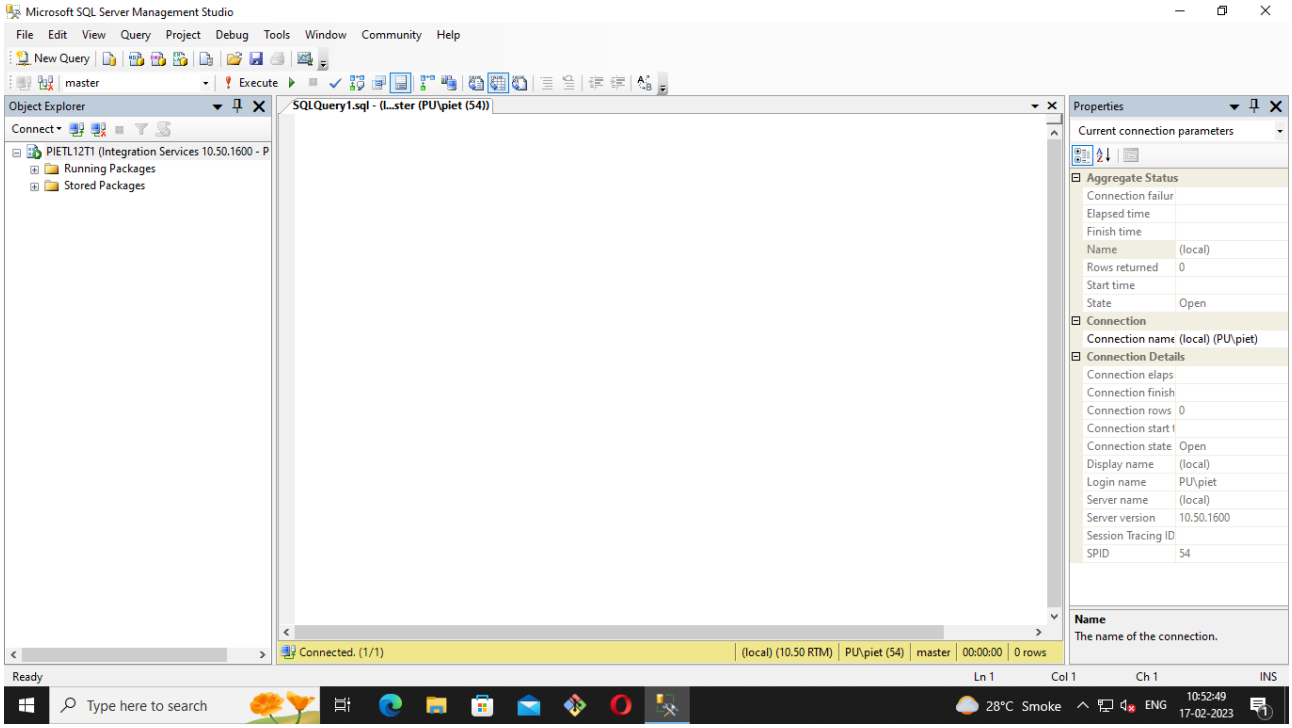
Pivot

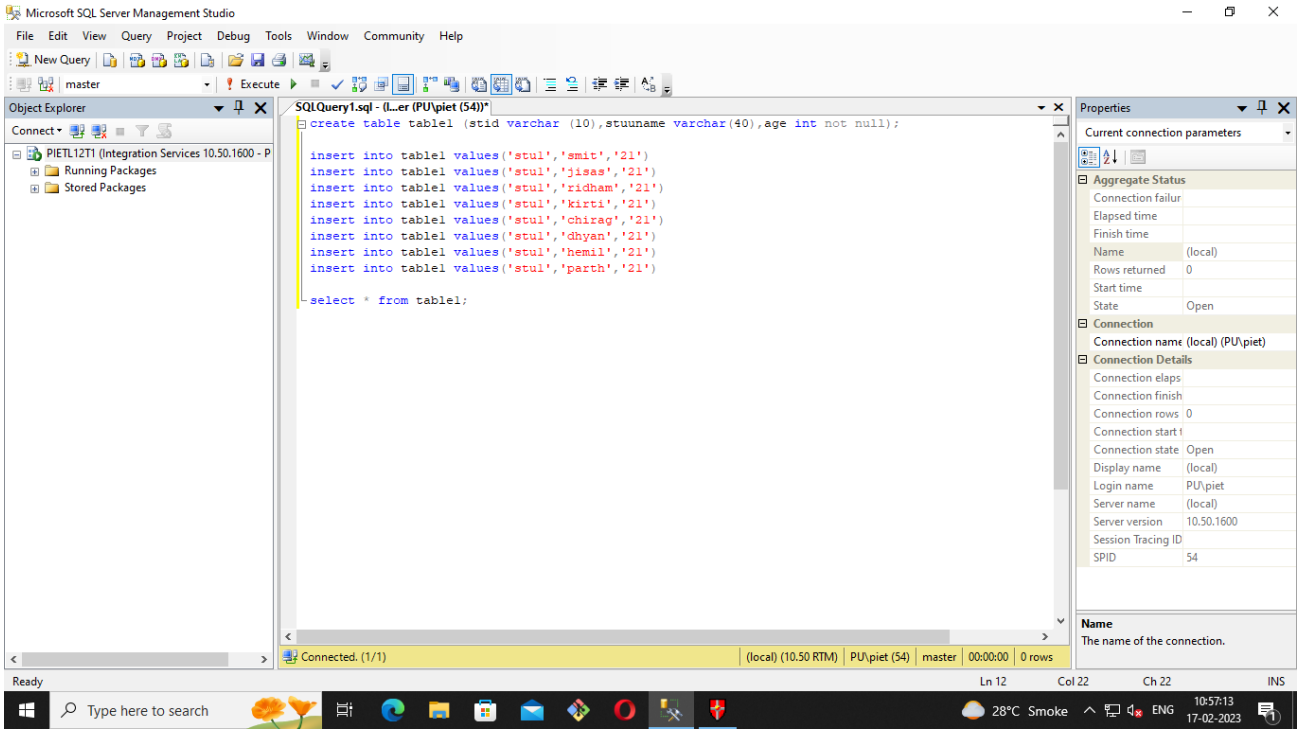
The pivot operation is also called a rotation. Pivot is a visualization operations which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.

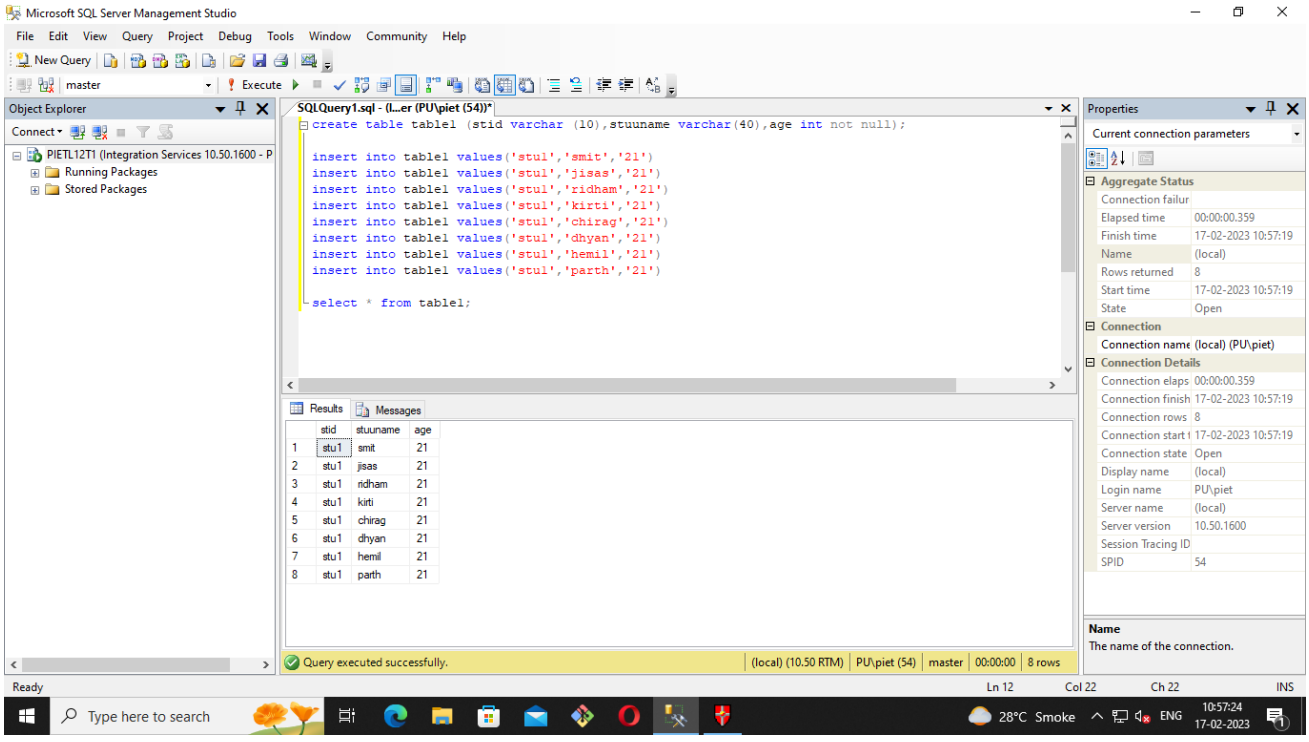


Query Examples of CUBE and ROLL UP:-






Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Community Help
New Query
master
Execute
Object Explorer
Connect
PIETL12T1 (Integration Services 10.50.1600 - P
Running Packages
Stored Packages
SQLQuery1.sql - (l...er (PU\piet (54))*
create table table1 (stid varchar (10),stuuname varchar(40),age int not null);
insert into table1 values('stu1','smit','21')
insert into table1 values('stu1','jisas','21')
insert into table1 values('stu1','ridham','21')
insert into table1 values('stu1','kirti','21')
insert into table1 values('stu1','chirag','21')
insert into table1 values('stu1','dhyan','21')
insert into table1 values('stu1','hemil','21')
insert into table1 values('stu1','parth','21')
select * from table1;
Connected. (1/1)
(local) (10.50 RTM)
PU\piet (54)
master
00:00:00
0 rows
Properties
Current connection parameters
Aggregate Status
Connection failur
Elapsed time
Finish time
Name (local)
Rows returned 0
Start time
State Open
Connection
Connection name (local) (PU\piet)
Connection Details
Connection elaps
Connection finish
Connection rows 0
Connection start t
Connection state Open
Display name (local)
Login name PU\piet
Server name (local)
Server version 10.50.1600
Session Tracing ID
SPID 54
Name
The name of the connection.
Ready
Ln 12
Col 22
Ch 22
INS
Type here to search
28°C Smoke
ENG
10:57:13
17-02-2023


Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Community Help
New Query
master
Execute
Object Explorer
Connect
PIETL12T1 (Integration Services 10.50.1600 - P
Running Packages
Stored Packages
SQLQuery1.sql - (l...er (PU\piet (54))*
create table table1 (stid varchar (10),stuuname varchar(40),age int not null);
insert into table1 values('stu1','smit','21')
insert into table1 values('stu1','jisas','21')
insert into table1 values('stu1','ridham','21')
insert into table1 values('stu1','kirti','21')
insert into table1 values('stu1','chirag','21')
insert into table1 values('stu1','dhyan','21')
insert into table1 values('stu1','hemil','21')
insert into table1 values('stu1','parth','21')
select * from table1;
Results
Messages
stid stuuname age
1 stu1 smit 21
2 stu1 jisas 21
3 stu1 ridham 21
4 stu1 kirti 21
5 stu1 chirag 21
6 stu1 dhyan 21
7 stu1 hemil 21
8 stu1 parth 21
Query executed successfully.
(local) (10.50 RTM)
PU\piet (54)
master
00:00:00
8 rows
Properties
Current connection parameters
Aggregate Status
Connection failur
Elapsed time 00:00:00.359
Finish time 17-02-2023 10:57:19
Name (local)
Rows returned 8
Start time 17-02-2023 10:57:19
State Open
Connection
Connection name (local) (PU\piet)
Connection Details
Connection elaps 00:00:00.359
Connection finish 17-02-2023 10:57:19
Connection rows 8
Connection start t 17-02-2023 10:57:19
Connection state Open
Display name (local)
Login name PU\piet
Server name (local)
Server version 10.50.1600
Session Tracing ID
SPID 54
Name
The name of the connection.
Ready
Ln 12
Col 22
Ch 22
INS
Type here to search
28°C Smoke
ENG
10:57:24
17-02-2023

PRACTICAL-3

AIM: Create calculated member using arithmetic operators and member property of dimension member.

Theory:

Calculated members are members of a dimension or a measure group that are defined based on a combination of cube data, arithmetic operators, numbers, and functions.

For example, you can create a calculated member that calculates the sum of two physical measures in the cube. Calculated member definitions are stored in cubes, but their values are calculated at query time.

It is Experimental Cube Data Source View shown in figure.

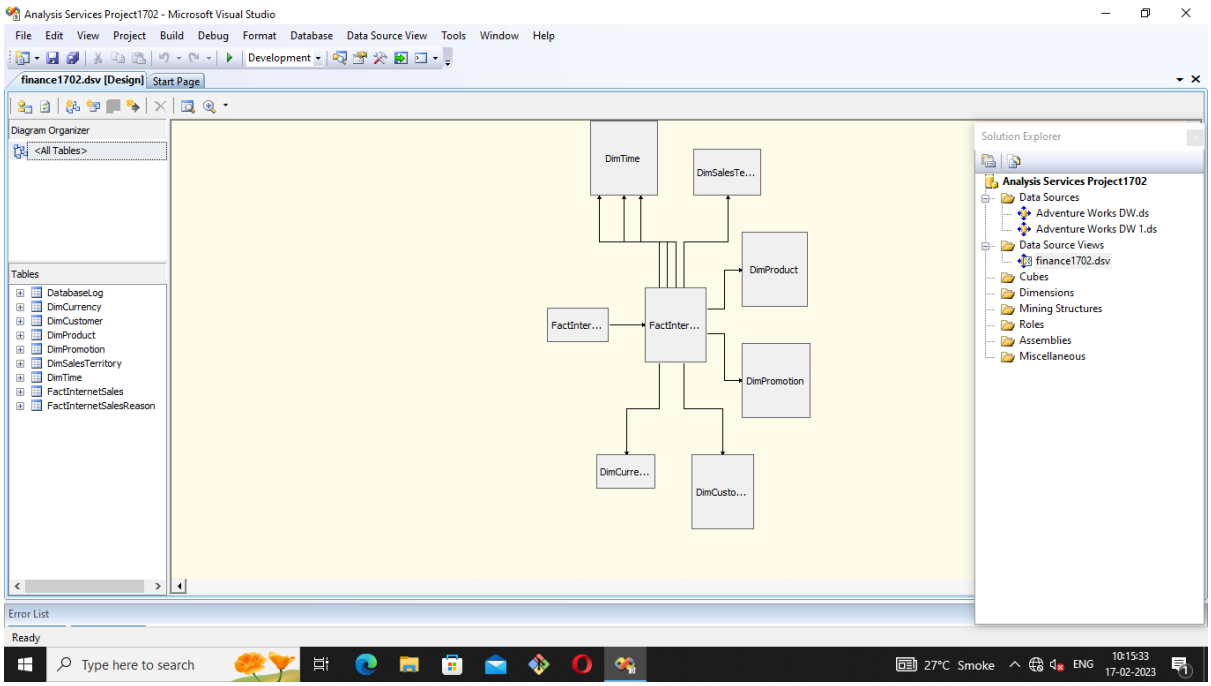


Figure-1 Cube Data Source View

Follow the Steps to create new calculate measure

1. Open Cube Designer for the Analysis Services Tutorial cube, and then click the Calculations tab.

2. On the toolbar of the Calculations tab, click New Calculated Member. A new form appears in the Calculation Expressions pane within which you define the properties of this new calculated member. The new member also appears in the Script Organizer pane.

3. In the Name box, change the name of the calculated measure to [Total Sales Amount].

4. On the Metadata tab in the Calculation Tools pane of the Calculations tab, expand Measures and then expand Internet Sales to view the metadata for the Internet Sales measure group.

5. Drag Internet Sales-Sales Amount from the Metadata tab in the Calculation Tools pane into the Expression box in the Calculation Expressions pane.

6. In the Expression box, type a plus sign (+)after [Measures]. [Fact Internet Sales Count-Amount].

7. In the Format string list, select "Currency".

8. In the Non-empty behavior list, select the check boxes for Fact Internet Sales Count-Tax Amount Fact Internet Sales, and then click OK.

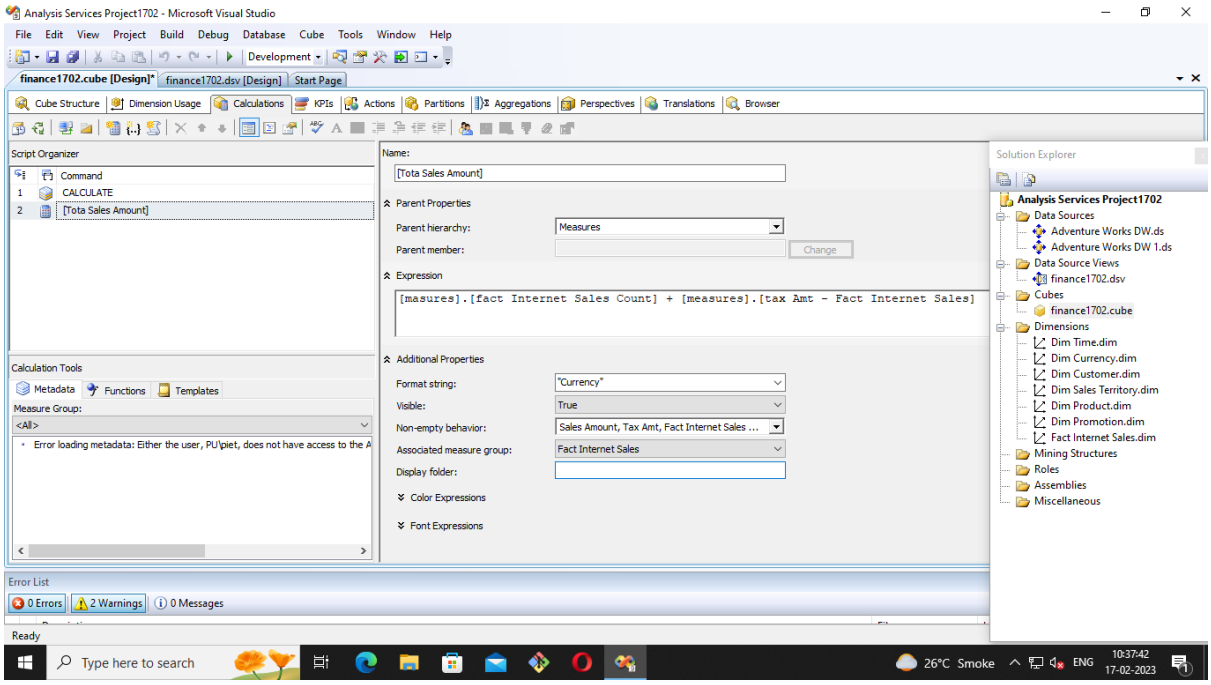


Figure-2 Defining Calculated Members

9. On the toolbar of the Calculations tab, click Script View, and then review the calculation script in the Calculation Expressions pane.

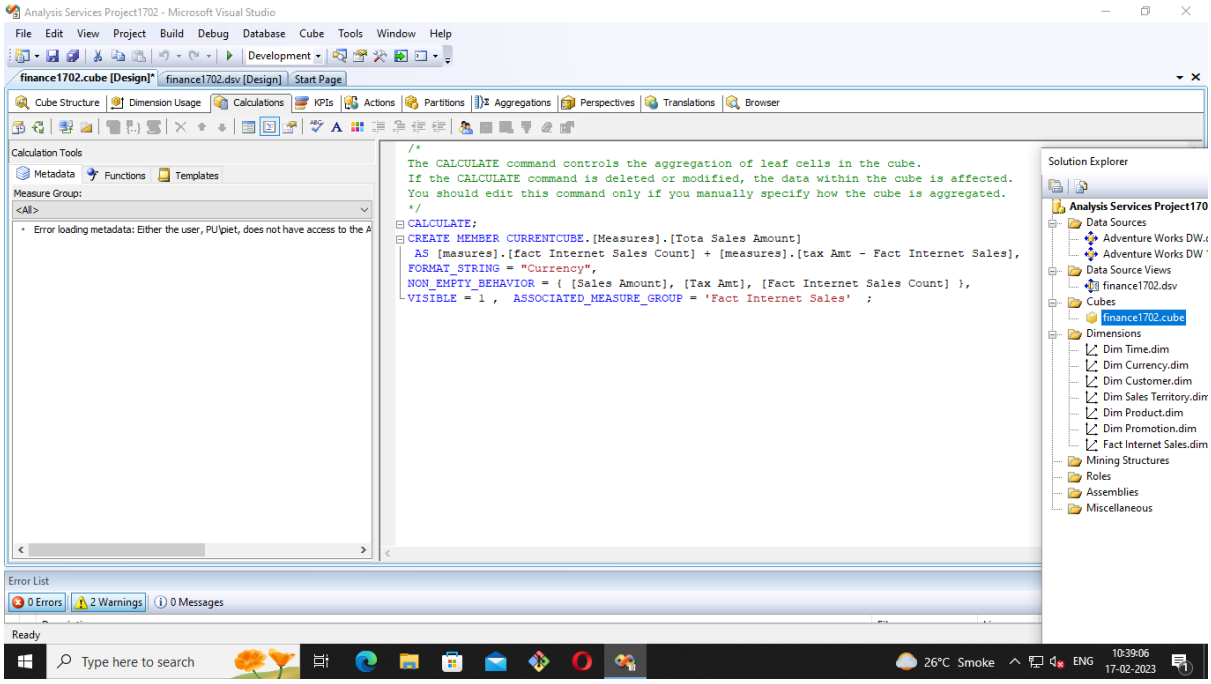


Figure-3 shows the calculation scripts

Browsing the New Calculated Members

1 .Drag and drop the Calculated Measure.

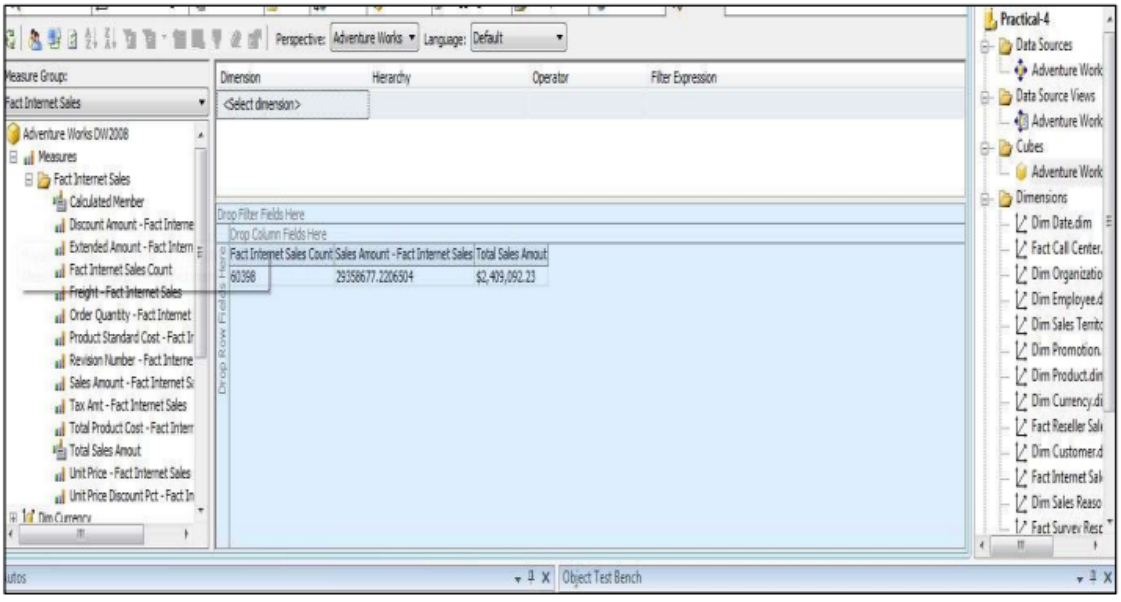


Figure-4 Browsing the New Calculated Members

PRACTICAL-4

AIM: Design and Create cube by identifying measures and dimensions for Design storage using storage mode MOLAP, ROLAP and HOLAP

Theory:

Microsoft SQL Server Analysis Services provides several standard storage configurations for storage modes and caching options.

Storage modes are ways to physically store the data in cubes and dimensions.

Regardless of the storage mode, the Analysis Services engines accepts Multidimensional Expressions (MDX) and return cell sets. Multidimensional Expressions (MDX) is a query language for OLAP databases. Much like SQL, it is a query language for relational databases.

There are three storage mode available in SQL Server 2008.

- MOLAP

- ROLAP

-HOLAP

User can select the storage mode from the settings.

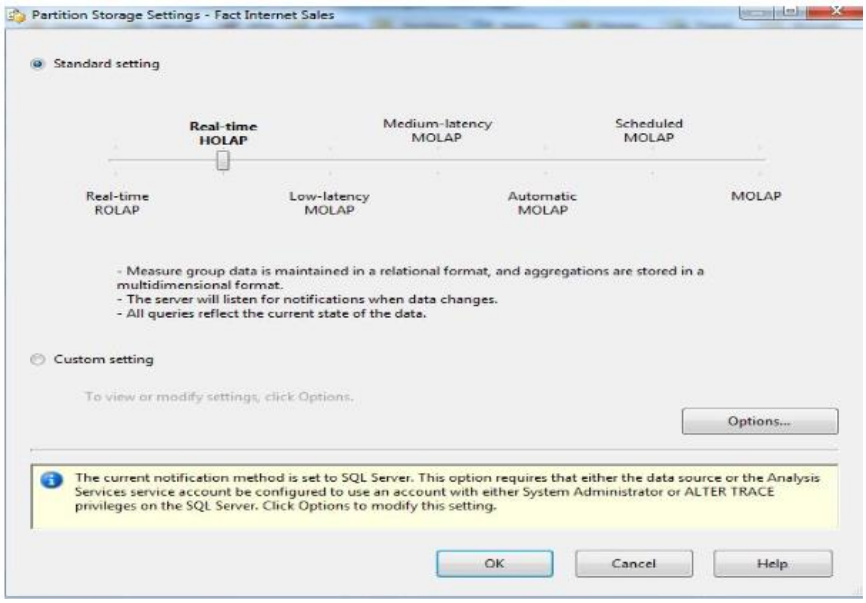


Figure-1 Storage Mode

Each setting explain as under:

Multidimensional OLAP (MOLAP):

MOLAP storage mode uses array-based multidimensional storage engine for multidimensional views of data. With Multidimensional data stores, the storage utilization is low if the data set is sparse. Many MOLAP servers use two levels of data storage representation to handle dense and sparse data sets.

Scheduled MOLAP:

Scheduled MOLAP used for a data source when only daily updates are required. Queries are always against data in the MOLAP cache, which is not discarded until a new cache is built and its objects are processed.

Automatic MOLAP:

Automatic MOLAP used for a data source when query performance is of key importance. It is automatically processes MOLAP objects whenever required after the latency interval. Queries do not return the most recent data while the new cache is being built and processed.

Medium- Latency MOLAP:

Medium-Latency MOLAP used for a data source with frequent (or less frequent) updates when query performance is more important than always providing the most current data. It is automatically processes MOLAP objects whenever required after the latency interval. Performance is slower while the MOLAP objects are being reprocessed.

Low-Latency MOLAP:

Low-Latency MOLAP used for a data source with frequent updates when query performance is somewhat more important than always providing the most current data. It is automatically processes MOLAP objects whenever required after the latency interval. Performance is slower while the MOLAP objects are being reprocessed.

Hybrid Online Analytical Processing (HOLAP):

Hybrid Online Analytical Processing used for a data source with frequent and continuous updates (but not so frequent as to require real-time ROLAP) and users always require the latest data. This method normally provides better overall performance than ROLAP storage. Users can get MOLAP performance from this setting if the data source stays silent long enough.

Real Time ROLAP (ROLAP):

Real time ROLAP used for a data source with very frequent and continuous updates when the

very latest data is always required by users. Depending on the types of queries generated by client applications, this method is liable to give the slowest response times.

Experimental Cube Data Source View:

Experimental Cube Data Source use the dimension like Fact Currency Rate, Fact Internet Sales, Fact Internet Sales Reason, and Fact Resellers Sales.

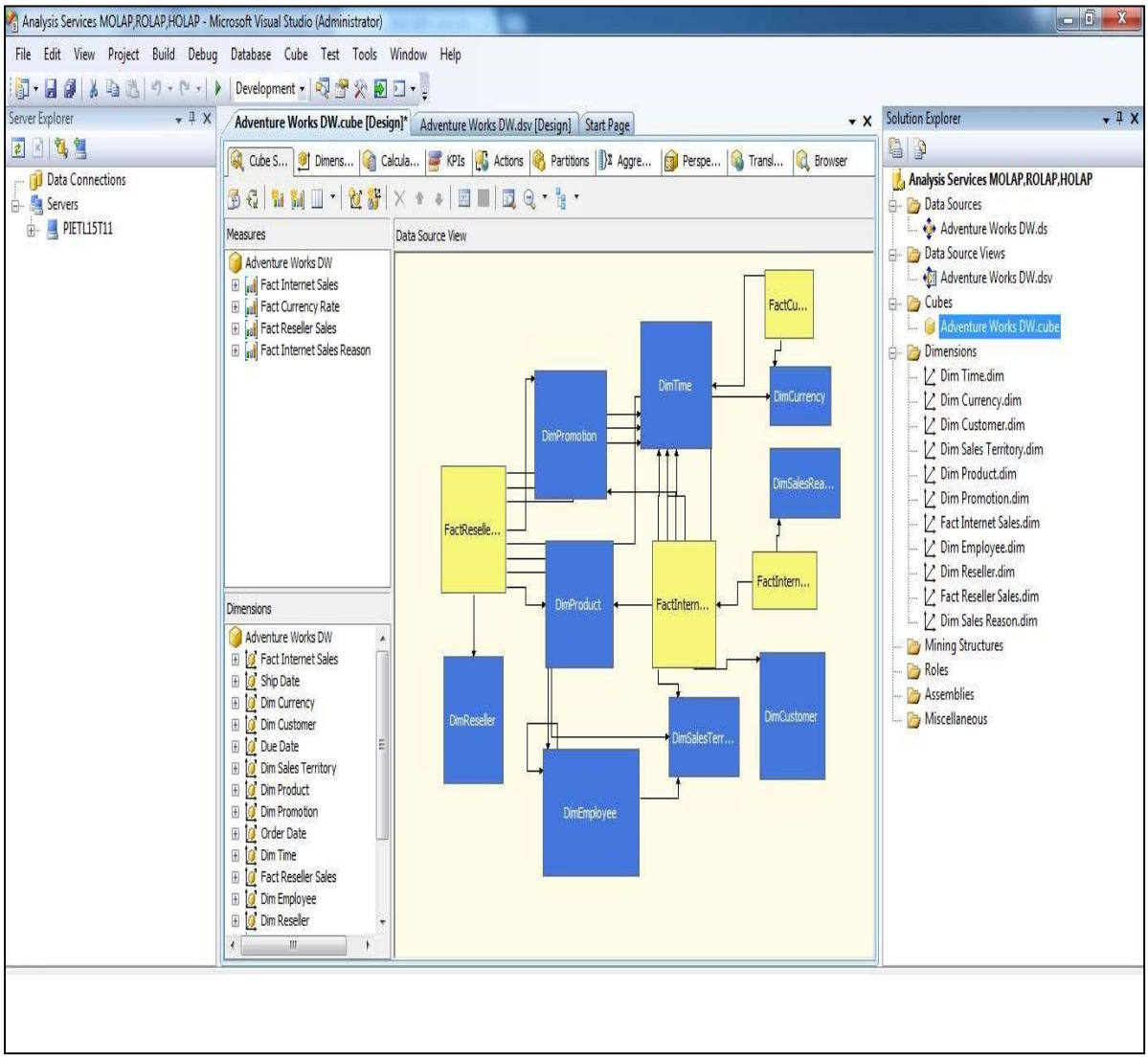


Figure-2 Cube Data Source View

Type of Storage Settings available:

1. Standard Setting

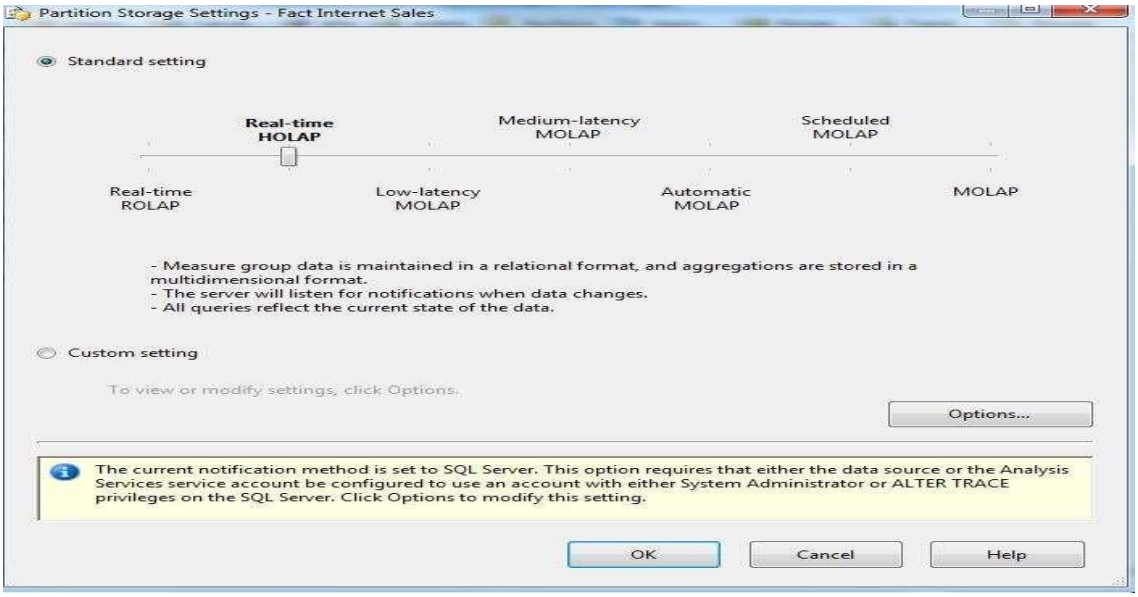


Figure-3 Standard Storage settings Options

2. Custom Settings

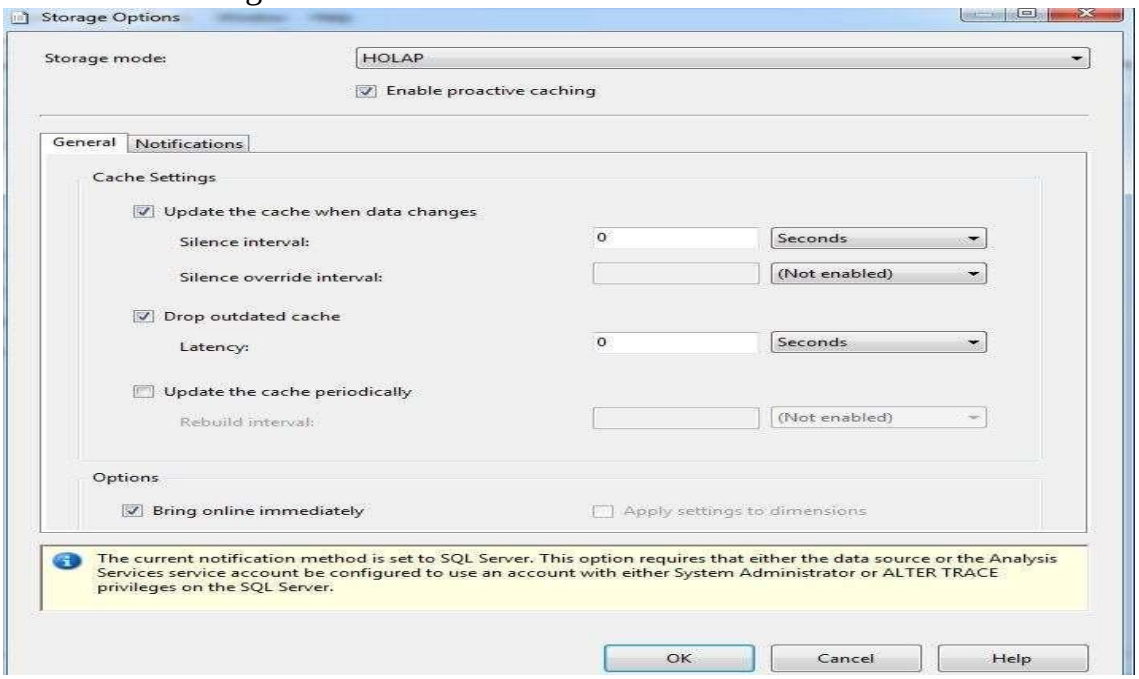


Figure-4 Custom Setting Storage Options

Implantation of different Storage Mode on the Above Experimental Cube Data Source.

1. MOLAP (Multidimensional OLAP)

This is the default and most frequently used storage mode. MOLAP Stores aggregations and leaf level data in binary files separate from the relational data warehouse. MOLAP makes a copy of the fact and dimension data. The MOLAP storage mode is very efficient and provides the fastest query performance.

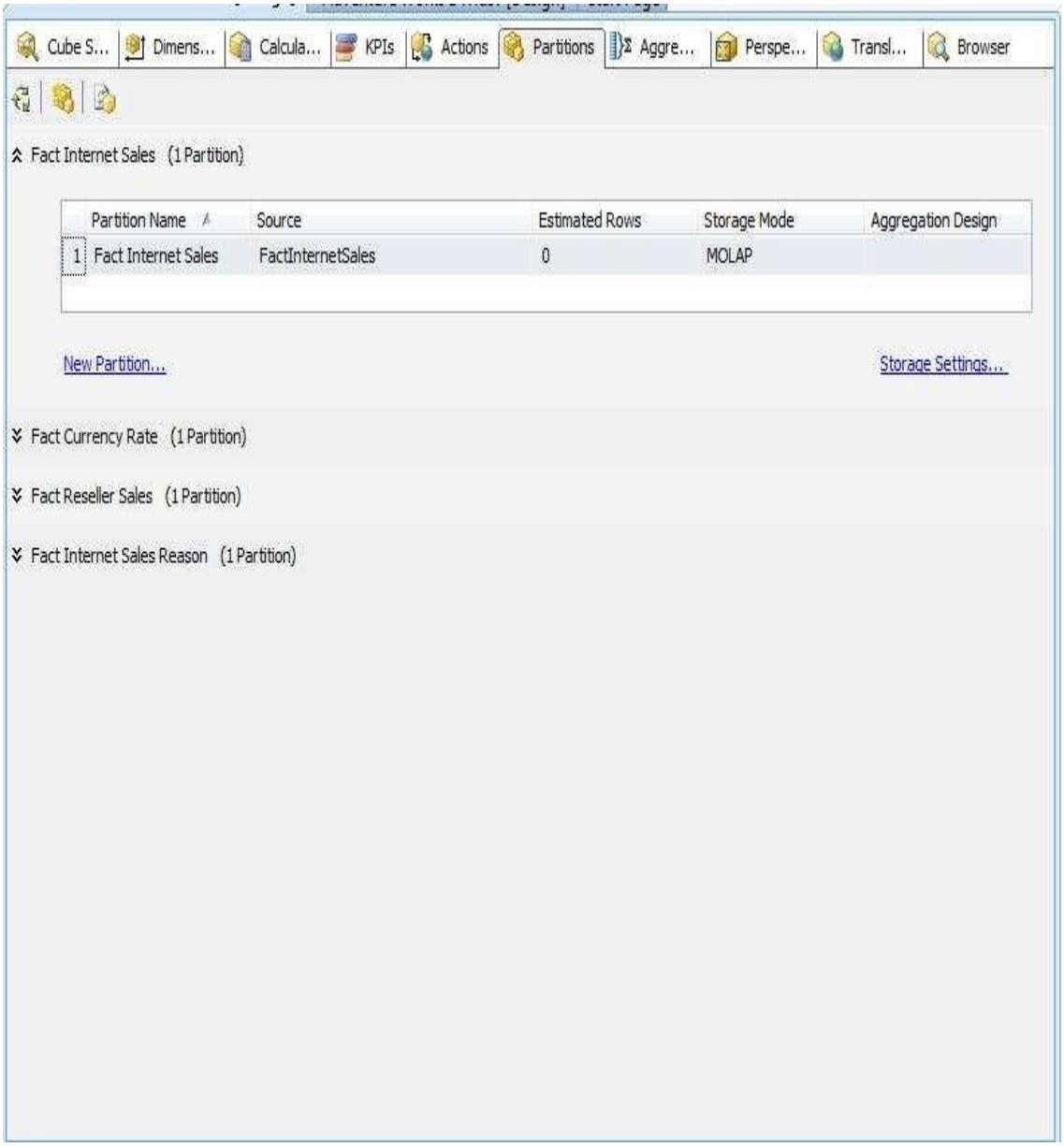


Figure-5 Select Storage Mode-MOLAP

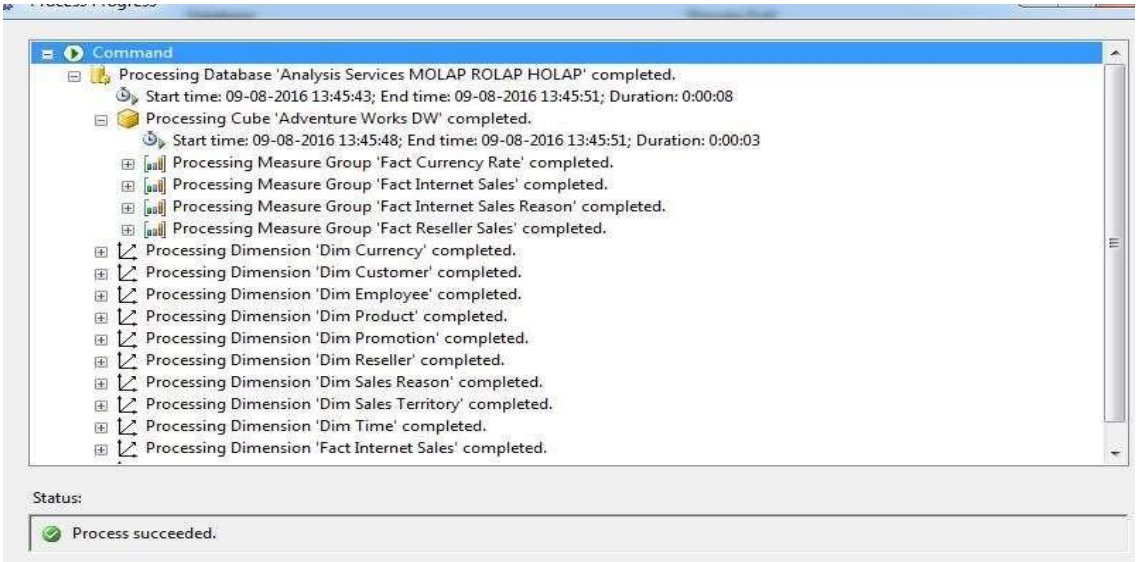


Figure-6 Show Processing Time of MOLAP

2. ROLAP (Relational OLAP)

Relational OLAP stores aggregations in indexed views in the relational database. ROLAP leaves the fact and dimensions data in relational tables.

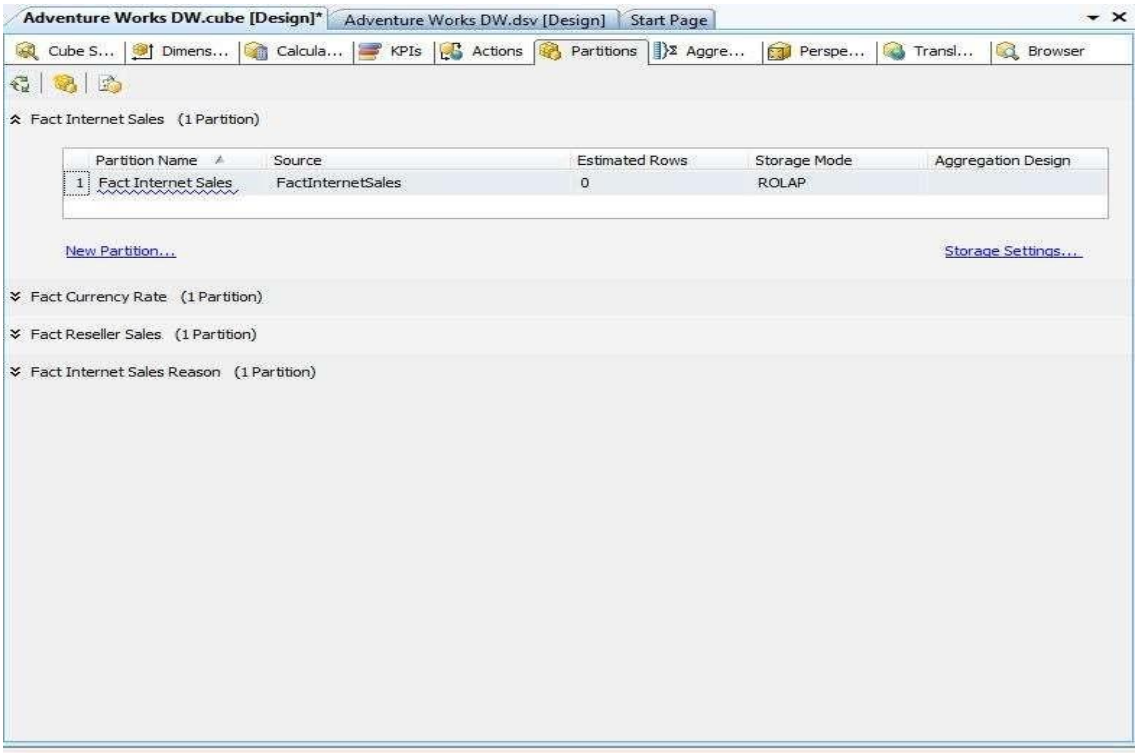


Figure-7 Select Storage Mode-ROLAP

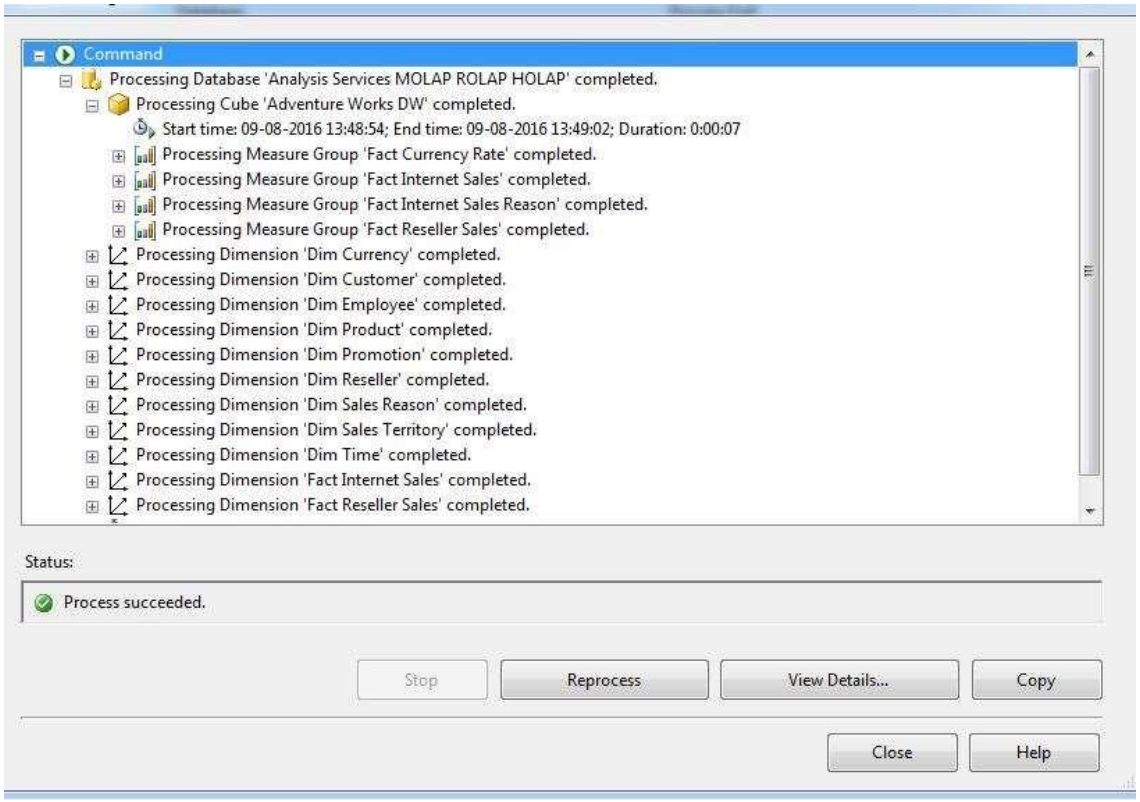


Figure-8 Show Processing Time of ROLAP

HOLAP (Hybrid online analytical processing)

Hybrid OLAP stores aggregations in binary files and leaves fact and dimension data in the relational database.

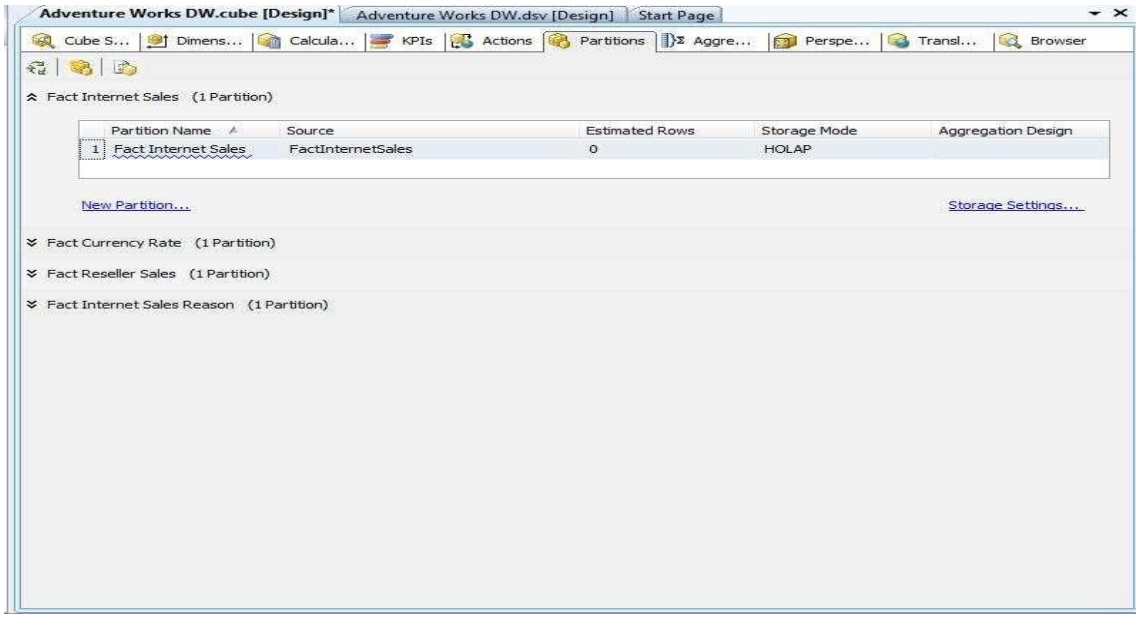


Figure-9 Select Storage Mode-HOLAP

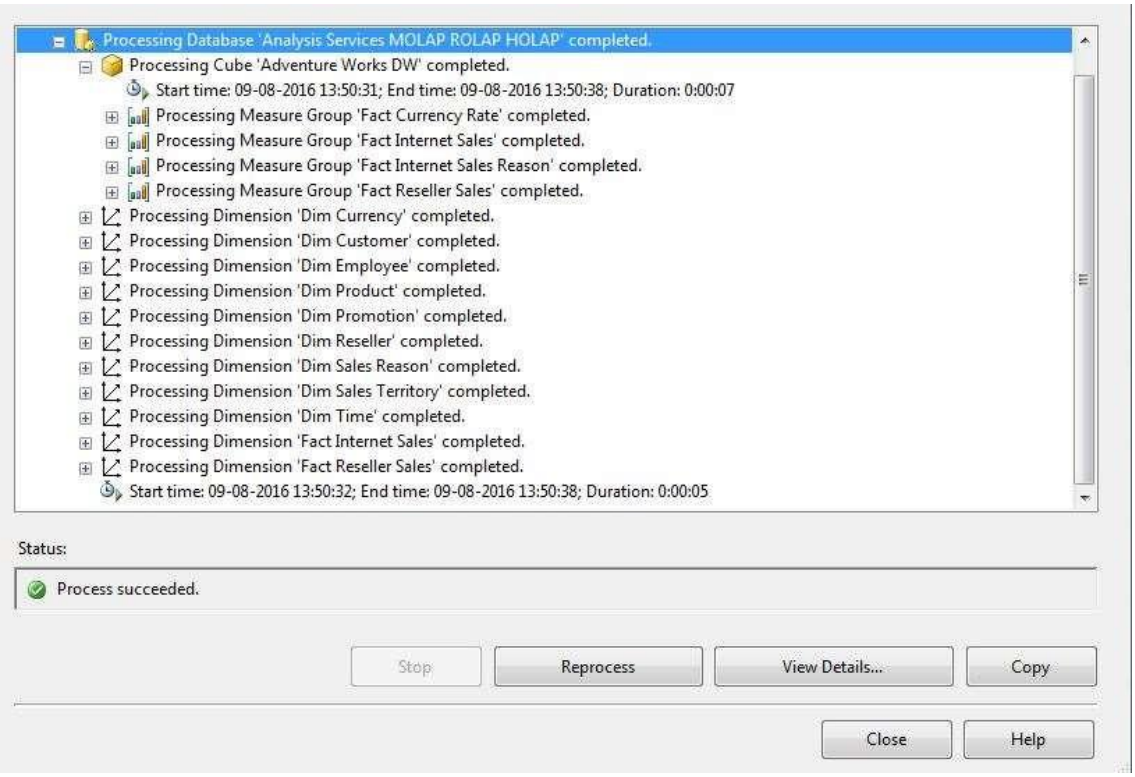


Figure-10 Show Processing Time of HOLAP

Storing Cubes and Dimensions

Dimensions can be stored as:

-MOLAP

-ROLAP

Cubes (partitions) can stored as:

-MOLAP

-ROLAP

-HOLAP

Comparison:

MOLAP	ROLAP	HOLAP
Fastest(almost always)	Slowest(almost always)	As fast as MOLAP for aggregations
More Total disk storage	Supports some real-time(with some caveats)	As slow as ROLAP for leaf level data
Snapshot in time		