VISHWAKARMA GOVERNMENT ENGINEERING COLLEGE, CHANDKHEDA

Subject: Data Mining & Business Intelligence (2170715)

INDEX

Faculty Coordinator Prof. J.J.Jadav
Computer Engineering Department

Practical List
ODD-2020

No.	Name of Experiment	CO mapping
1.	Identify how data mining is an interdisciplinary field by example.	**CO1**
2.	**Write programs to perform the following tasks (any language).** **1.1 Noisy data handling** 1) Equal Width Binning 2) Equal Frequency /Depth Binning **1.2 Normalization Techniques** 1) Min max normalization 2) Z score normalization 3) Decimal scaling 4)Five Number Summary	**CO2**
3.	To perform hand on experiments of data preprocessing with sample data on Orange tool.	**CO2**
4.	Implement Apriori algorithm of association rule data mining technique in any Programming language.	**CO3**
5.	To perform hand on experiments with sample data sets on XLMiner.	**CO3**
6.	To perform hand on experiments with sample data sets on Weka.	**CO3**
7.	Design and Create cube by identifying measures and dimensions for Star Schema, Snowflake schema by SQL Server Analysis Service.	**CO4**
8.	Design and Create cube by identifying measures and dimensions for Design storage for cube using storage mode MOLAP, ROLAP and HOALP	**CO4**
9.	Design and create data mining models using Analysis Services of SQL Server.	**CO4**
10.	Refer any two research papers of Advance Mining topic and write down summary of it.	**CO5**

Practical-1

Aim: Identify how data mining is an interdisciplinary field by example.

"Data mining is the process of uncovering patterns and finding anomalies and relationships in large datasets that can be used to make predictions about future trends. The main purpose of data mining is extracting valuable information from available data."
"Data mining is considered an interdisciplinary field that joins the techniques of computer science and statistics. Note that the term "data mining" is a misnomer. It is primarily concerned with discovering patterns and anomalies within datasets, but it is not related to the extraction of the data itself."
"Data mining is an interdisciplinary field bringing together techniques from machine learning, pattern recognition, statistics, databases and visualization to address the issue of information extraction from large data bases."

Here are some important areas where data mining is widely used:

- Future Healthcare
- Market Basket Analysis
- Education
- Manufacturing Engineering
- CRM (Customer Relationship Management)
- Fraud Detection
- Intrusion Detection
- Lie Detection
- Customer Segmentation
- Financial Banking
- Corporate Surveillance
- Research Analysis
- Criminal Investigation
- Bio Informatics

Future Healthcare:
Data mining holds great potential to improve health systems. It uses data and analytics to identify best practices that improve care and reduce costs. Researchers use data mining approaches like multi-dimensional databases, machine learning, soft computing, data visualization and statistics. Mining can be used to predict the volume of patients in every category. Processes are developed that make sure that the patients receive appropriate care at the right place and at the right time. Data mining can also help healthcare insurers to detect fraud and abuse.

Market Basket Analysis:
Market basket analysis is a modelling technique based upon a theory that if you buy a certain group of items you are more likely to buy another group of items. This technique may allow the retailer to understand the purchase behaviour of a buyer. This information may help the retailer to know the buyer's needs and change the store's layout accordingly. Using differential analysis comparison of results between different stores, between customers in different demographic groups can be done.

Education:
There is a new emerging field, called Educational Data Mining, concerns with developing methods that discover knowledge from data originating from educational Environments. The goals of EDM are identified as predicting students' future learning behaviour, studying the effects of educational support, and advancing scientific knowledge about learning. Data mining can be used by an institution to take accurate decisions and also to predict the results of the student. With the results the institution can focus on what to teach and how to teach. Learning pattern of the students can be captured and used to develop techniques to teach them.

Manufacturing Engineering:
Knowledge is the best asset a manufacturing enterprise would possess. Data mining tools can be very useful to discover patterns in complex manufacturing process. Data mining can be used in system-level designing to extract the relationships between product architecture, product portfolio, and customer needs data. It can also be used to predict the product development span time, cost, and dependencies among other tasks.

CRM:
Customer Relationship Management is all about acquiring and retaining customers, also improving customers' loyalty and implementing customer focused strategies. To maintain a proper relationship with a customer a business need to collect data and analyse the information. This is where data mining plays its part. With data mining technologies the collected data can be used for analysis. Instead of being confused where to focus to retain customer, the seekers for the solution get filtered results.

Fraud Detection:
Billions of dollars have been lost to the action of frauds. Traditional methods of fraud detection are time consuming and complex. Data mining aids in providing meaningful patterns and turning data into information. Any information that is valid and useful is knowledge. A perfect fraud detection system should protect information of all the users. A supervised method includes collection of sample records. These records are classified fraudulent or non-fraudulent. A model is built using this data and the algorithm is made to identify whether the record is fraudulent or not.

Intrusion Detection:
Any action that will compromise the integrity and confidentiality of a resource is an intrusion. The defensive measures to avoid an intrusion includes user authentication, avoid programming errors, and information protection. Data mining can help improve intrusion detection by adding a level of focus to anomaly detection. It helps an analyst to distinguish an activity from common everyday network activity. Data mining also helps extract data which is more relevant to the problem.

Lie Detection:
Apprehending a criminal is easy whereas bringing out the truth from him is difficult. Law enforcement can use mining techniques to investigate crimes, monitor communication of suspected terrorists. This filed includes text mining also. This process seeks to find meaningful patterns in data which is usually unstructured text. The data sample collected from previous investigations are compared and a model for lie detection is created. With this model processes can be created according to the necessity.

Customer Segmentation:
Traditional market research may help us to segment customers but data mining goes in deep and increases market effectiveness. Data mining aids in aligning the customers into a distinct segment and can tailor the needs according to the customers. Market is always about retaining the customers. Data mining allows to find a segment of customers based on vulnerability and the business could offer them with special offers and enhance satisfaction.

Financial Banking:
With computerised banking everywhere huge amount of data is supposed to be generated with new transactions. Data mining can contribute to solving business problems in banking and finance by finding patterns, causalities, and correlations in business information and market prices that are not immediately apparent to managers because the volume data is too large or is generated too quickly to screen by experts. The managers may find these information for better segmenting, targeting, acquiring, retaining and maintaining a profitable customer.

Corporate Surveillance:
Corporate surveillance is the monitoring of a person or group's behaviour by a corporation. The data collected is most often used for marketing purposes or sold to other corporations, but is also regularly shared with government agencies. It can be used by the business to tailor their products desirable by their customers. The data can be used for direct marketing purposes, such as the targeted advertisements on Google and Yahoo, where ads are targeted to the user of the search engine by analyzing their search history and emails.

Research Analysis
History shows that we have witnessed revolutionary changes in research. Data mining is helpful in data cleaning, data pre-processing and integration of databases. The researchers can find any similar data from the database that might bring any change in the research. Identification of any co-occurring sequences and the correlation between any activities can be known. Data visualisation and visual data mining provide us with a clear view of the data.

Criminal Investigation

Criminology is a process that aims to identify crime characteristics. Actually crime analysis includes exploring and detecting crimes and their relationships with criminals. The high volume of crime datasets and also the complexity of relationships between these kinds of data have made criminology an appropriate field for applying data mining techniques. Text based crime reports can be converted into word processing files. These information can be used to perform crime matching process.

Bio Informatics

Data Mining approaches seem ideally suited for Bioinformatics, since it is data-rich. Mining biological data helps to extract useful knowledge from massive datasets gathered in biology, and in other related life sciences areas such as medicine and neuroscience. Applications of data mining to bioinformatics include gene finding, protein function inference, disease diagnosis, disease prognosis, disease treatment optimization, protein and gene interaction network reconstruction, data cleansing, and protein sub-cellular location prediction.

Practical-2

Aim: Write a program to perform the following tasks.

1.1: Noisy data handling

1) Equal Width Binding:

```
def equiwidth(arr1, m):
    a = len(arr1)
    w = int((max(arr1) - min(arr1)) / m)
    min1 = min(arr1)
    arr = []
    for i in range(0, m + 1):
        arr = arr + [min1 + w * i]
    arri=[]

    for i in range(0, m):
        temp = []
        for j in arr1:
            if j > arr[i] and j < arr[i+1]:
                temp += [j]
        arri += [temp]
    print(arri)

if __name__ == '__main__':
    data = [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]
    m = 3

    print("\n\nequal width binning")
    equiwidth(data, 3)
```

Output:

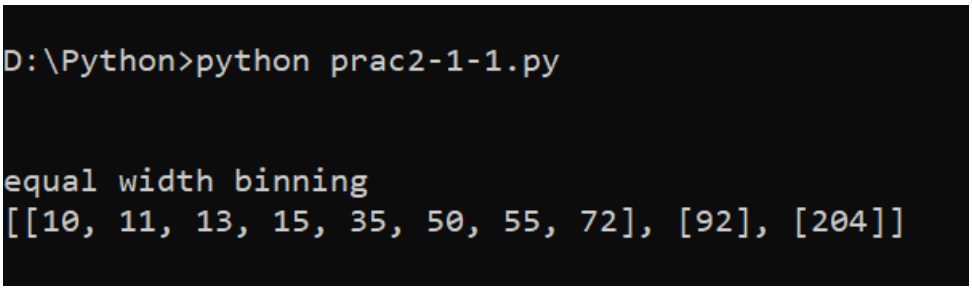


Fig-1.1

2) Equal Frequency Depth Binding

```
def equifreq(arr1, m):

    a = len(arr1)
    n = int(a / m)
    for i in range(0, m):
        arr = []
        for j in range(i * n, (i + 1) * n):
            if j >= a:
                break
            arr = arr + [arr1[j]]
        print(arr)

if __name__ == '__main__':
    data = [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]
    m = 3

    print("equal frequency binning")
    equifreq(data, m)
```

Output:

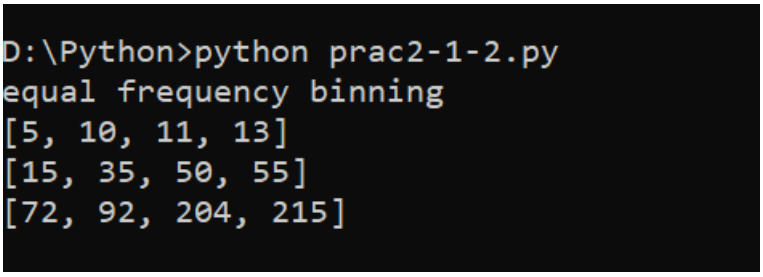


Fig 1.2

1.2: Normalization Techniques

1) Min Max Noramalization:

```
import numpy as np
if __name__ == '__main__':
    x = np.array([15, 2, 4, 12, 1 , 3, 5, 6, 8, 9, 10])
    ans = []
    for a in x:
        ans.append((a - x.min()) / (x.max() - x.min()))
    print("Input  Array:\n",x)
    print("\n\nNormalized Array:\n",ans)
```

2) Z score normalization:

```
if __name__ == '__main__':
    x = list(map(int, input("Enter Values:\n").split()))
    y = sum(x)/len(x)
    z = 0
    for a in x:
        z += (a-y)**2
    z = (z/len(x))**(1/2)
    ans = []
```

```
    for a in x:
        ans.append((a-y)/z)
    print("Input Array:\n",x)
    print("\nZ Score Normalized array\n",ans)
```

Output:

```
D:\Python>python prac2-2-2.py
Enter Values:
2 4 3 8 5 6 11
Input Array:
 [2, 4, 3, 8, 5, 6, 11]

Z Score Normalized array
 [-1.243796487762486, -0.5472704546154938, -0.89553347118899, 0.8457816116784908, -0.19900743804199772, 0.14925557853149843, 1.8905706613989792]
```

Fig 2.2

3) Decimal Scaling:

```
if __name__ == '__main__':
    x = input("Enter Input Array\n:").split( )
    max = "0"
    for a in x:
        if max < a:
            max = a
            b = len(max)

    ans = []
    for a in x:
        ans.append(int(a)/(10**b))
    print("Desimal Normalized array",ans)
```

Output:

```
D:\Python>python prac2-2-3.py
Enter Input Array
:2 4 3 8 5 6 11
Desimal Normalized array [0.2, 0.4, 0.3, 0.8, 0.5, 0.6, 1.1]
```

Fig 2.3

4) Five Number Summary:

```
def find_median(list_m):
    no_of_elements = len(list_m)
    if no_of_elements % 2 == 0:
        median_1 = list_m[(no_of_elements // 2)-1]
        median_2 = list_m[(no_of_elements // 2 + 1)-1]
        median_lis = (median_1 + median_2) / 2
        position = ((no_of_elements + 1) / 2)
    else:
        median_lis = list_m[((no_of_elements + 1) // 2)-1]
        position = ((no_of_elements + 1) / 2)
    return int(median_lis), position


if __name__ == '__main__':
    print("Enter the data: ")
    lis = list(map(int, input().split()))
    lis = sorted(lis)

    median1, pos = find_median(lis)

    count_a = 0
    count_b = 0

    for i in range(len(lis)):
        if i < pos-1:
            count_a += 1
        elif i > pos-1:
            count_b += 1
```

```
quad1 = int((count_a+1)/2)
quad3 = int(pos) + int((count_b + 1) / 2)
iqr = quad3-quad1

print("Minimum : {}".format(min(lis)))
print("Q1 : {}".format(lis[quad1-1]))
print("Median : {}".format(median1))
print("Q3 : {}".format(lis[quad3-1]))
print("Maximum : {}".format(max(lis)))
print("IQR : {}".format(lis[quad3-1]-lis[quad1-1]))
```

Output:

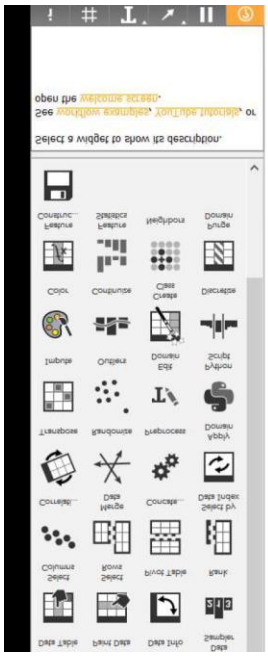
```
D:\Python>python prac2-2-4.py
Enter the data:
2 4 3 8 5 6 11
Minimum : 2
Q1 : 3
Median : 5
Q3 : 8
Maximum : 11
IQR : 5
```

Fig 2.4

Practicals-3

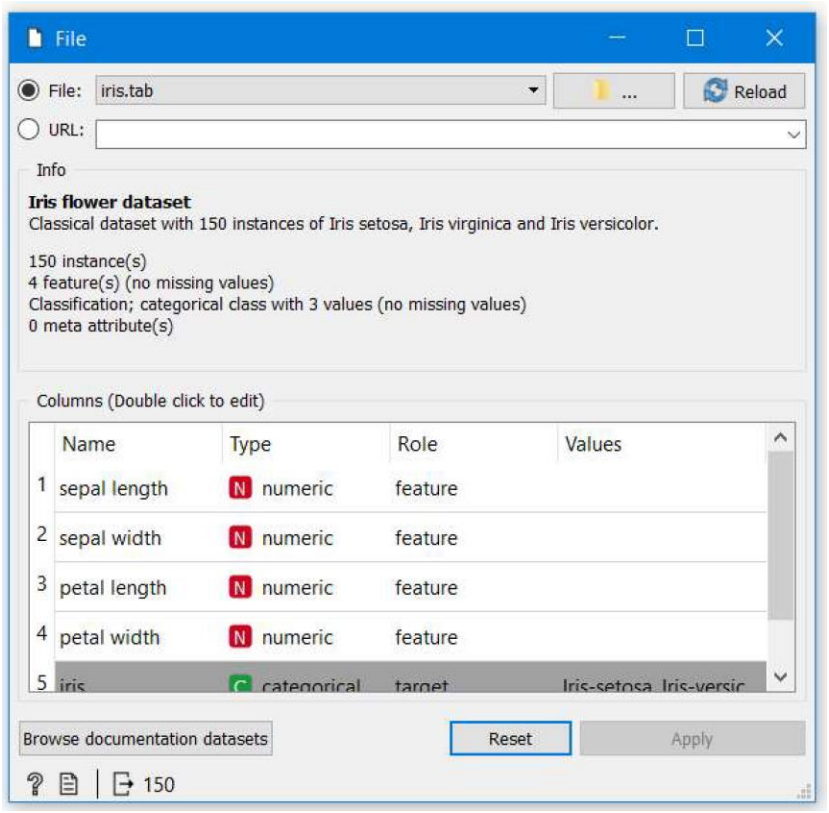
Aim: To perform hand-on experiments of data preprocessing with sample data on the Orange tool.

1. Open Orange tool. It will look like Screenshot 1.



Scrreenshot-1

2. From that, choose “File” from the pane on the left side. Refer to Screenshot 2.



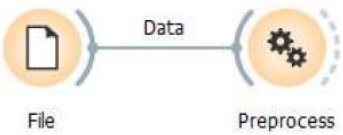
Screenshot-2

3. Here, we will choose a default data set known as “iris.tab” as shown above and we will perform the preprocessing on it. Close the dialog box. Now, the diagram will have a “File” point looking like shown in Screenshot 3.

Screenshot-3

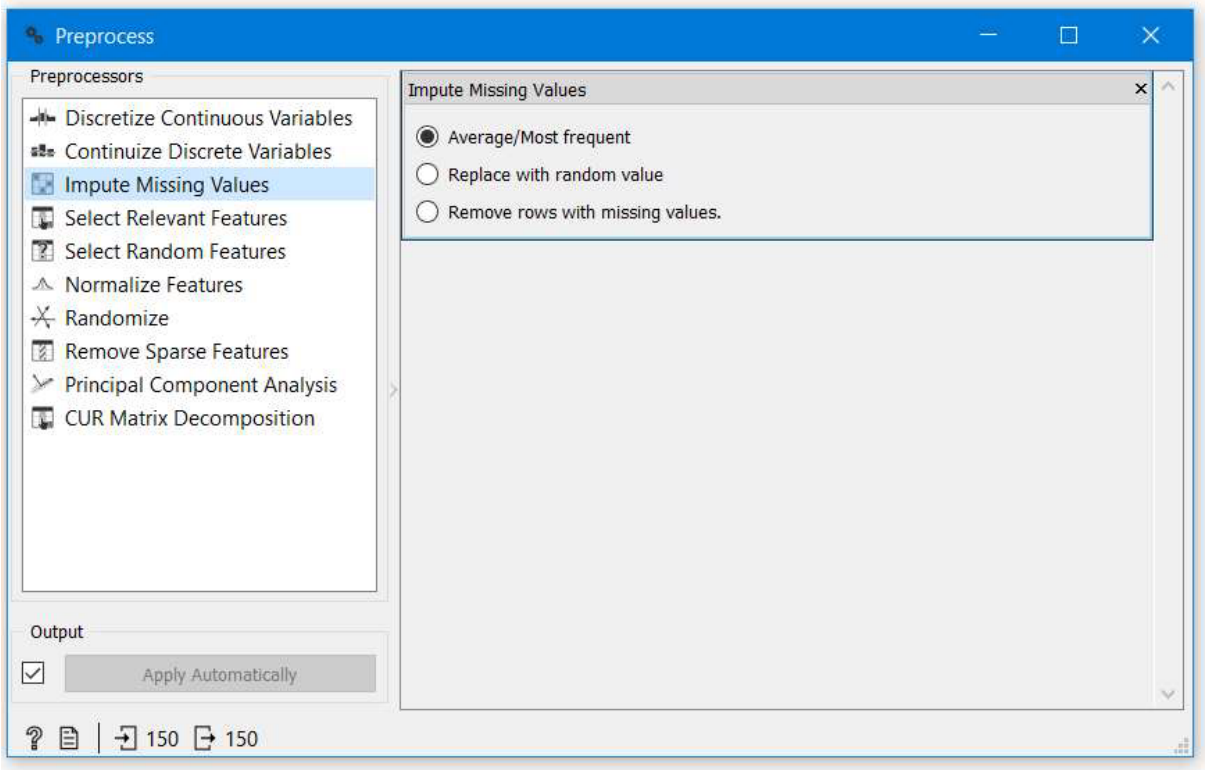
4. Now select the “Preprocess” option from the left side pane and then the diagram will look like this (Screenshot 4a). Then drag a point from the left side and extend it till the Preprocess point. Refer to Screenshots 4a and 4b.

Screenshot-4.1



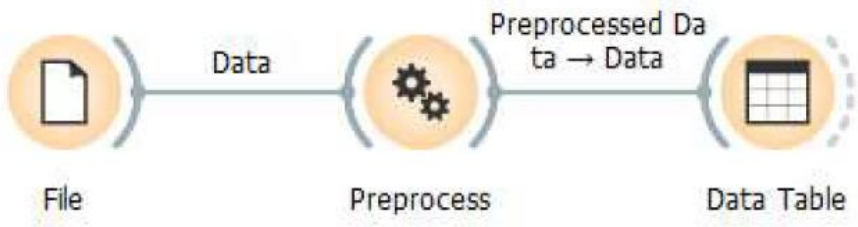
Screenshot-4.2

5. Then select the Preprocess point as shown above and then we have to select as to how to preprocess our data. We will select the “Impute Missing Values” tab and then select “Average/Most Frequent” from the available options shown in Screenshot 5.



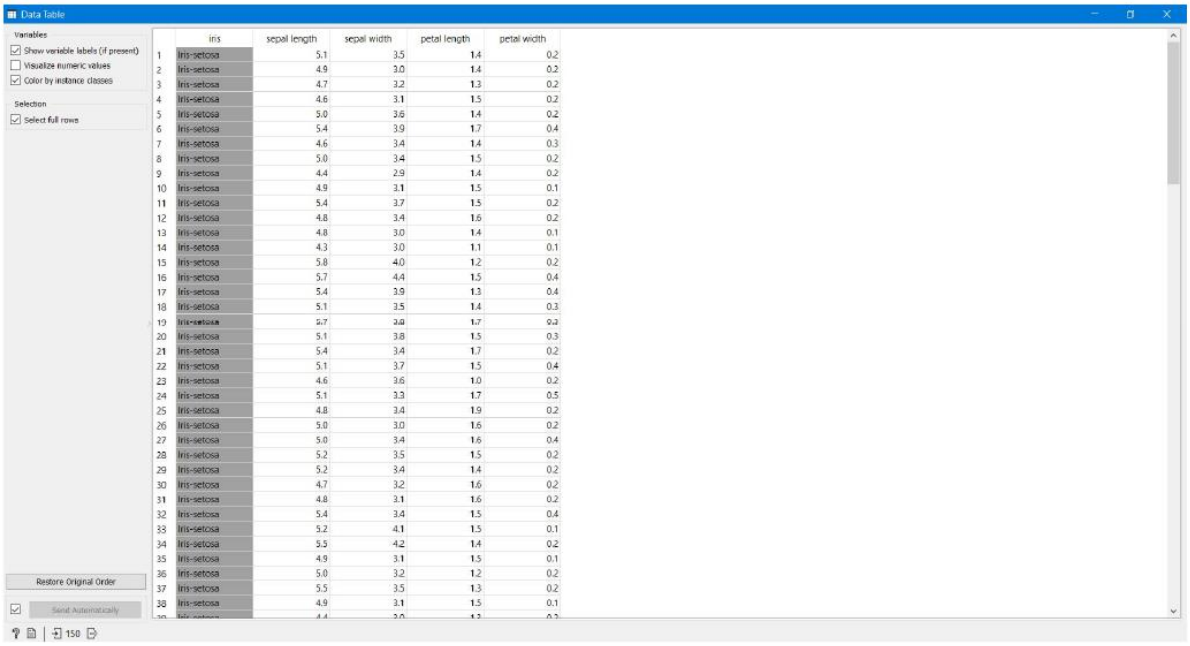
Screenshot-5

6. Then select “Data Table” from the left side pane and then the diagram will look like as shown in Screenshot 6. Then just like the step no. 4 drag from “Preprocess” to “Data Table”.



Screenshot-6

7. The “Data Table” will look like as shown in Screenshot 7.

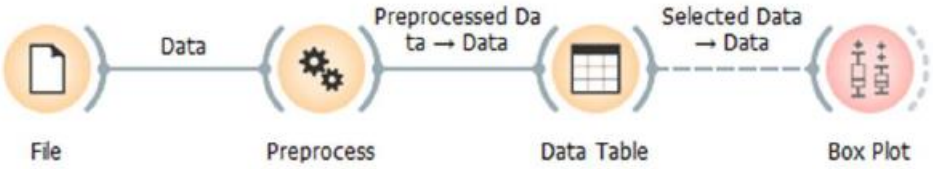


Screenshot-7

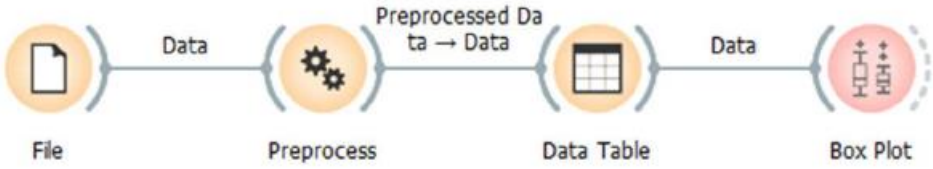
8. Then select “Visualize” instead of “Data” from the left side panel and the screen will look like as shown in Screenshot 8.

Screenshot-8

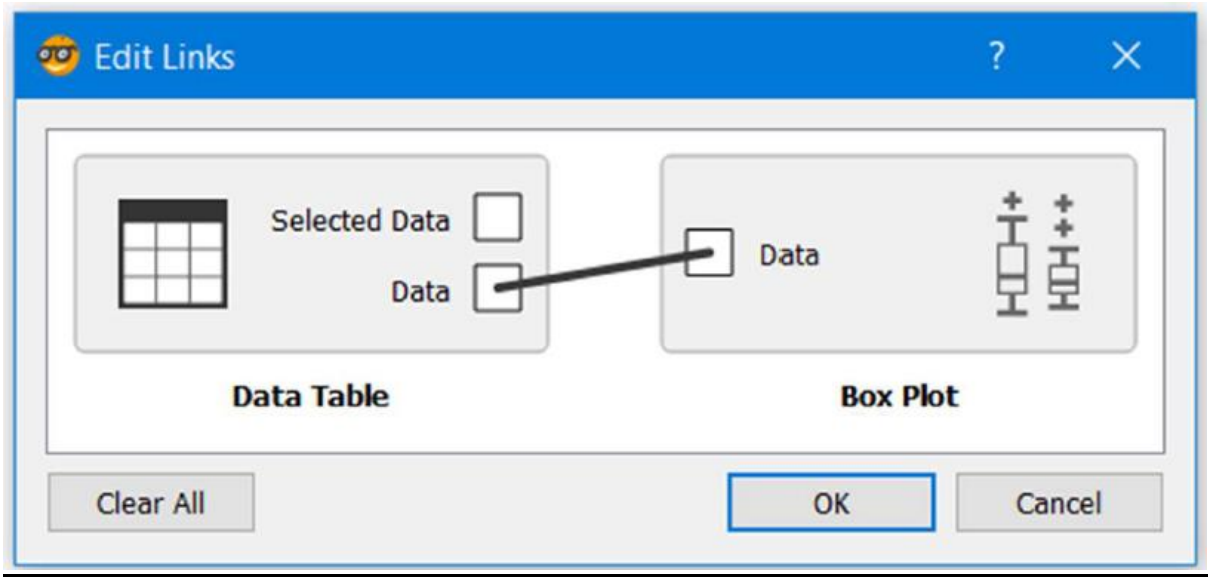
9. Select “Box Plot” from the options on the left side pane and drag the point from “Data Table” to the “Box point” and the diagram will look like as shown in Screenshot 9a. In this, click on the line formed in the middle and select the correspondence “Data to Data” looking in Screenshot 9b. Then finally, the diagram will look as shown in Screenshot 9c.



Screenshot-9.1

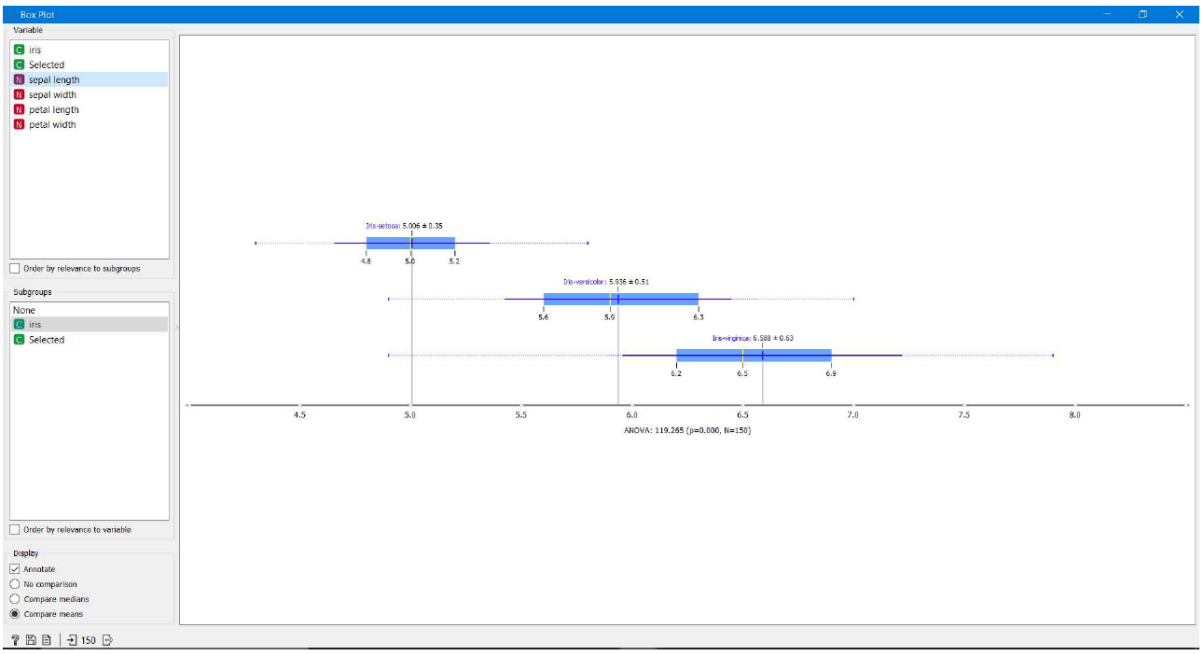


Screenshot-9.2



Screenshot-9.3

10. Then click on the “Box Plot” and the screen will look like as shown in Screenshot 10.



Screenshot-10

Practicals-4

Aim: Implement Apriori algorithm of association rule data mining technique in any Programming language.

In [1]:
```
import sys
!{sys.executable} -m pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.6/dist-packages (3.0.0)
Requirement already satisfied: py4j==0.10.9 in /usr/local/lib/python3.6/dist-packages (from pyspark) (0.10.9)
```

In [2]:
```
import sys
!{sys.executable} -m pip install spark
```

```
Requirement already satisfied: spark in /usr/local/lib/python3.6/dist-packages (0.2.1)
```

In [3]:
```
from pyspark import SparkContext
sc = SparkContext("local", "Apriori")
```

In [4]:
```
file = sc.textFile("transaction.txt")
print(file.collect())
```

```
['A1,A2,A5,', 'A2,A4,,', 'A2,A3,,', 'A1,A2,A4,', 'A1,A3,,', 'A2,A3,,', 'A1,A3,,', 'A1,A2,A3,A5', 'A1,A2,A3,']
```

In [5]:
```
lblitems = file.map(lambda line: line.split(','))
print(lblitems.collect())
```

```
[['A1', 'A2', 'A5', ''], ['A2', 'A4', '', ''], ['A2', 'A3', '', ''], ['A1', 'A2', 'A4', ''], ['A1', 'A3', '', ''], ['
A2', 'A3', '', ''], ['A1', 'A3', '', ''], ['A1', 'A2', 'A3', 'A5'], ['A1', 'A2', 'A3', '']]
```

In [6]:
```
wlitems = file.flatMap(lambda line:line.split(','))
print(wlitems.collect())
```

```
['A1', 'A2', 'A5', '', 'A2', 'A4', '', '', 'A2', 'A3', '', '', 'A1', 'A2', 'A4', '', 'A1', 'A3', '', '', 'A2', 'A3',
'', '', 'A1', 'A3', '', '', 'A1', 'A2', 'A3', 'A5', 'A1', 'A2', 'A3', '']
```

In [7]:
```
uniqueItems = wlitems.distinct()
supportRdd = wlitems.map(lambda item: (item , 1))
def sumOperator(x,y):
    return x+y
supportRdd = supportRdd.reduceByKey(sumOperator)
supports = supportRdd.map(lambda item: item[1])
```

In [8]:
```
minSupport = supports.min()
minSupport = 2 if minSupport == 1 else minSupport
supportRdd = supportRdd.filter(lambda item: item[1] >= minSupport )
baseRdd = supportRdd.map(lambda item: ([item[0]] , item[1]))
print("1 itemset")
print(supportRdd.collect())
supportRdd = supportRdd.map(lambda item: item[0])
supportRddCart = supportRdd
```

```
1 itemset
[('A1', 6), ('A2', 7), ('A5', 2), ('', 13), ('A4', 2), ('A3', 6)]
```

In [9]:
```
def removeReplica(record):
    if isinstance(record[0], tuple):
        x1 = record[0]
        x2 = record[1]
    else:
        x1 = [record[0]]
        x2 = record[1]
    if any(x == x2 for x in x1) == False:
        a = list(x1)
        a.append(x2)
        a.sort()
        result = tuple(a)
        return result
    else:
        return x1
```

In [10]:
```
c = 2
while supportRdd.isEmpty() == False:
    combined = supportRdd.cartesian(uniqueItems)
    combined = combined.map(lambda item: removeReplica(item))
    combined = combined.filter(lambda item: len(item) == c)
    combined = combined.distinct()
    combined_2 = combined.cartesian(lblitems)
    combined_2 = combined_2.filter(lambda item: all(x in item[1] for x in item[0]))
    combined_2 = combined_2.map(lambda item: item[0])
    combined_2 = combined_2.map(lambda item: (item , 1))
    combined_2 = combined_2.reduceByKey(sumOperator)
    combined_2 = combined_2.filter(lambda item: item[1] >= minSupport)
    baseRdd = baseRdd.union(combined_2)
    combined_2 = combined_2.map(lambda item: item[0])
    supportRdd = combined_2
    print("{} itemset".format(c))
    print(supportRdd.collect())
    print()
    c = c + 1
```

```
2 itemset
[('A1', 'A2'), ('A1', 'A5'), ('', 'A1'), ('A2', 'A5'), ('', 'A2'), ('A1', 'A3'), ('A2', 'A4'), ('A2', 'A3'), ('', 'A4
'), ('', 'A3')]

3 itemset
[('A1', 'A2', 'A5'), ('', 'A1', 'A2'), ('A1', 'A2', 'A3'), ('', 'A2', 'A4'), ('', 'A2', 'A3'), ('', 'A1', 'A3')]

4 itemset
[]
```

In [11]:
```
class Filter():
    def __init__(self):
        self.stages = 1
    def filterForConf(self, item , total):
        if len(item[0][0]) > len(item[1][0]):
            if self.checkItemSets(item[0][0] , item[1][0]) == False:
                pass
            else:
                return item
        else:
            pass
        self.stages = self.stages + 1
    def checkItemSets(self, item_1 , item_2):
        if len(item_1) > len(item_2):
            return all(any(k == l for k in item_1 ) for l in item_2)
        else:
            return all(any(k == l for k in item_2 ) for l in item_1)
    def calculateConfidence(self, item):
        parent = set(item[0][0])
        if isinstance(item[1][0] , str):
            child  = set([item[1][0]])
        else:
            child  = set(item[1][0])
        parentSupport = item[0][1]
        childSupport = item[1][1]
        support = (parentSupport / childSupport)*100
        return list([ list(child) ,  list(parent.difference(child)) , support ])
calcuItems = baseRdd.cartesian(baseRdd)
ff = Filter()
total = calcuItems.count()
baseRddConfidence = calcuItems.filter(lambda item: ff.filterForConf(item , total))
baseRddConfidence = baseRddConfidence.map(lambda item: ff.calculateConfidence(item))
print(baseRddConfidence.collect())
```

```
[[['A1'], ['A2'], 66.66666666666666], [['A2'], ['A1'], 57.14285714285714], [['A1'], ['A5'], 33.33333333333333], [['A1
'], [''], 83.33333333333334], [['A5'], ['A1'], 100.0], [[''], ['A1'], 38.46153846153847], [['A1'], ['A3'], 66.6666666
6666666], [['A2'], ['A5'], 28.571428571428571], [['A5'], ['A2'], 100.0], [['A2'], [''], 85.71428571428571], [['A2'],
['A4'], 28.57142857142857], [[''], ['A2'], 46.15384615384615], [['A3'], ['A1'], 66.66666666666666], [['A4'], ['A2'],
100.0], [['A2'], ['A3'], 57.14285714285714], [['A3'], ['A2'], 66.66666666666666], [[''], ['A4'], 15.384615384615385],
[['A4'], [''], 100.0], [[''], ['A3'], 38.46153846153847], [['A3'], [''], 83.33333333333334], [['A1'], ['A2', 'A5'], 3
3.33333333333333], [['A2'], ['A5', 'A1'], 28.57142857142857], [['A5'], ['A2', 'A1'], 100.0], [['A1'], ['', 'A2'], 50.
0], [['A1'], ['A3', 'A2'], 33.33333333333333], [['A2'], ['', 'A1'], 42.857142857142854], [['A2'], ['A3', 'A1'], 28.57
142857142857], [[''], ['A2', 'A1'], 23.076923076923077], [['A3'], ['A2', 'A1'], 33.33333333333333], [['A1'], ['', 'A3
'], 50.0], [['A2'], ['', 'A4'], 28.57142857142857], [['A2'], ['', 'A3'], 42.857142857142854], [[''], ['A4', 'A2'], 1
5.384615384615385], [['A4'], ['', 'A2'], 100.0], [[''], ['A3', 'A2'], 23.076923076923077], [['A3'], ['', 'A2'], 50.
0], [[''], ['A3', 'A1'], 23.076923076923077], [['A3'], ['', 'A1'], 50.0], [['A2', 'A1'], ['A5'], 50.0], [['A5', 'A1
'], ['A2'], 100.0], [['A2', 'A5'], ['A1'], 100.0], [['A2', 'A1'], [''], 75.0], [['A2', 'A1'], ['A3'], 50.0], [['', 'A
1'], ['A2'], 60.0], [['', 'A2'], ['A1'], 50.0], [['A3', 'A1'], ['A2'], 50.0], [['A3', 'A2'], ['A1'], 50.0], [['', 'A1
'], ['A3'], 60.0], [['', 'A2'], ['A4'], 33.33333333333333], [['A4', 'A2'], [''], 100.0], [['', 'A2'], ['A3'], 50.0],
[['A3', 'A1'], [''], 75.0], [['', 'A4'], ['A2'], 100.0], [['A3', 'A2'], [''], 75.0], [['', 'A3'], ['A2'], 60.0],
[['', 'A3'], ['A1'], 60.0]]
```

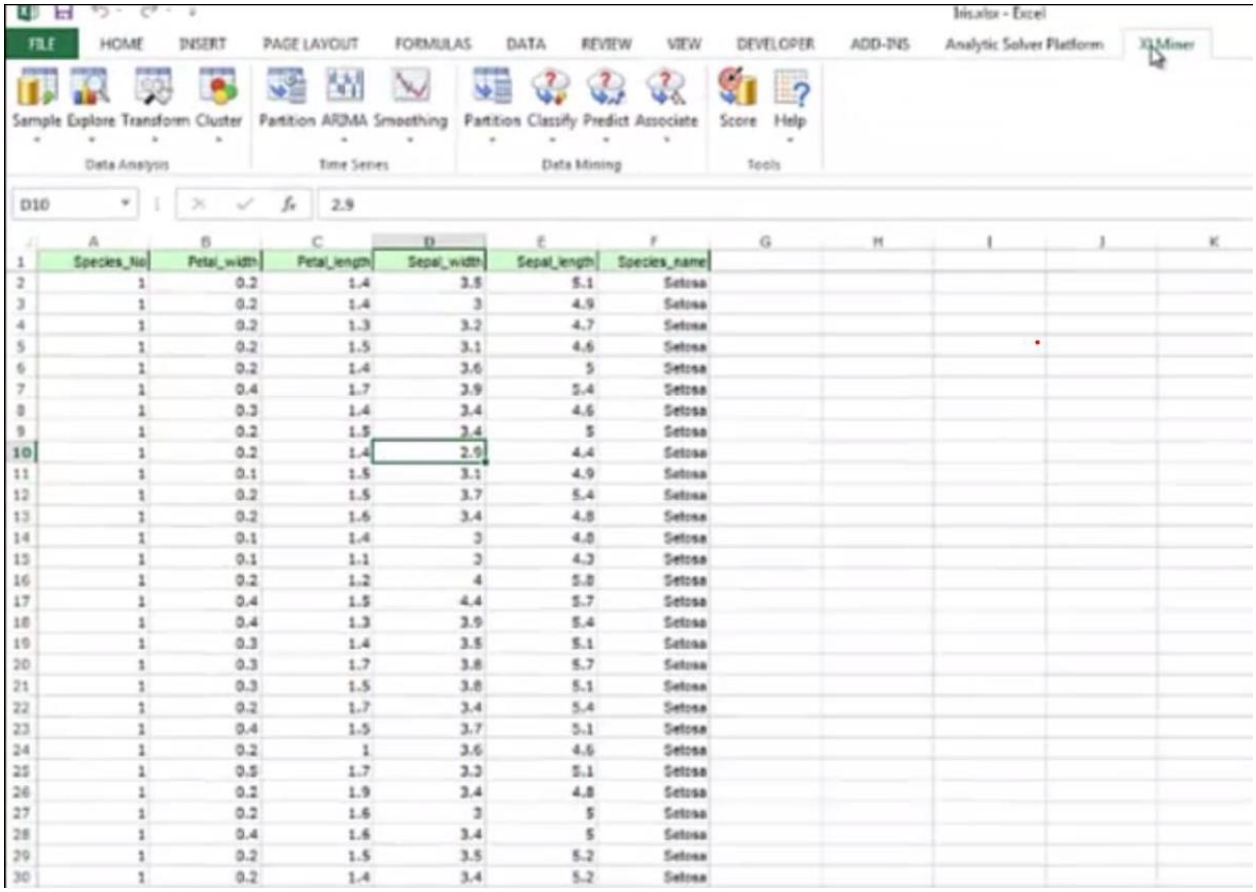
Practicals-5

Aim: To perform hand on experiments with sample data sets on XLMiner.

Given below functions XLMiner provides

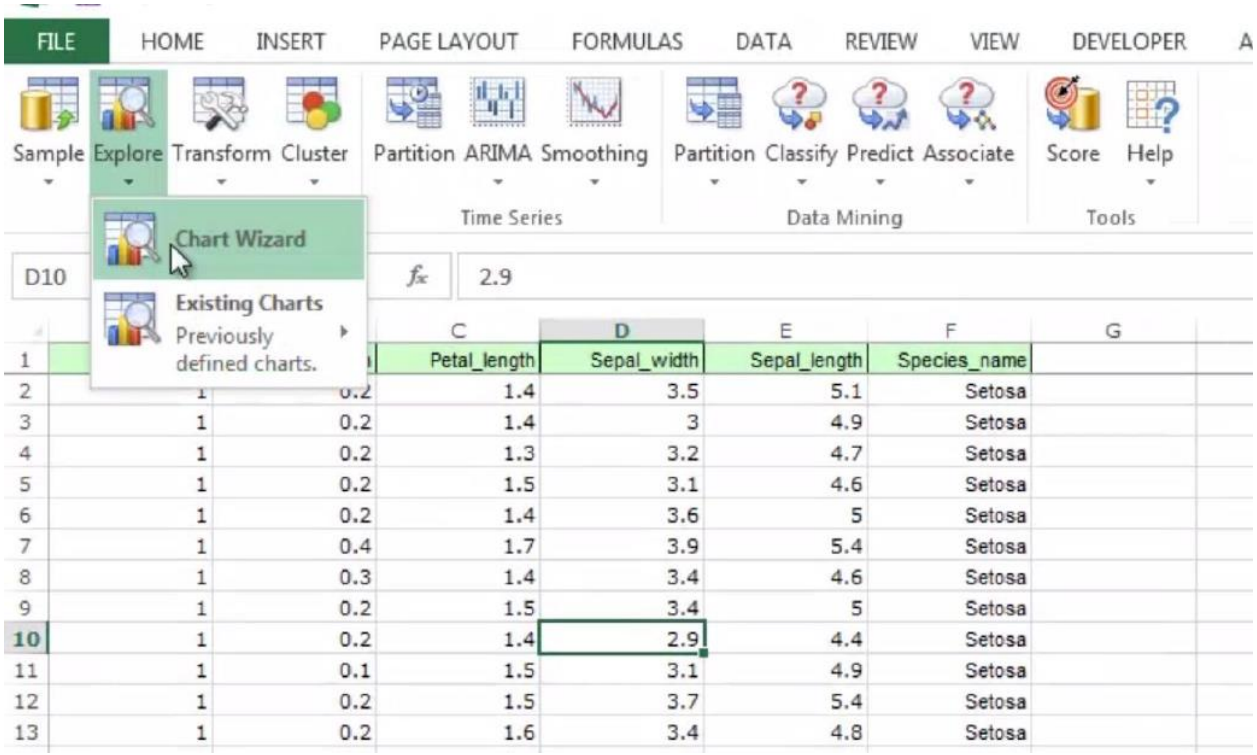
- Example Data Sets
- Data Analysis
- Time Series
- Data Mining
- Applying Your Model
- Resources

Open an excel file (I'm using the iris dataset).



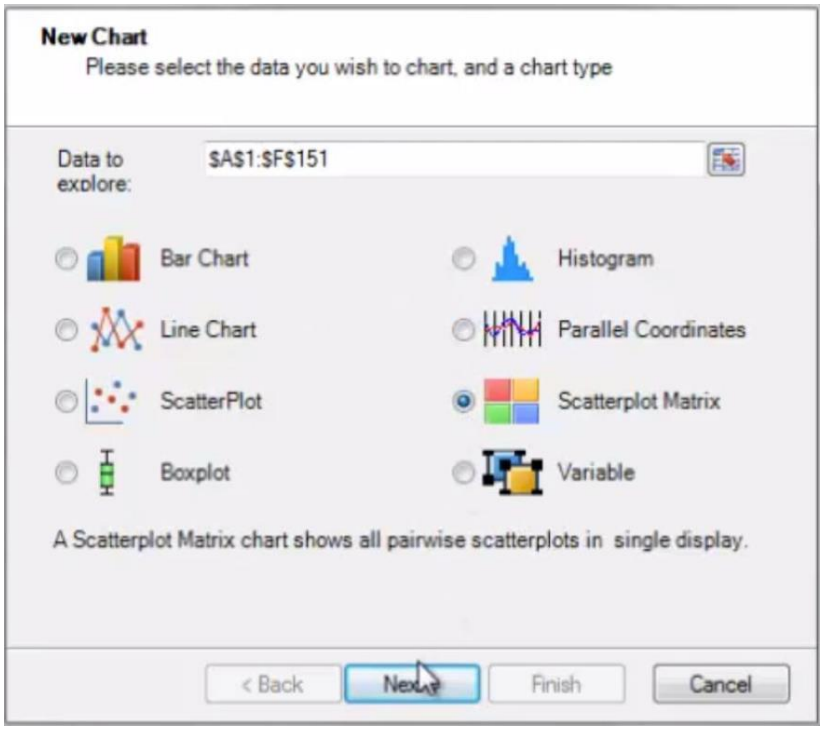
	A	B	C	D	E	F
1	Species_No	Petal_width	Petal_length	Sepal_width	Sepal_length	Species_name
2	1	0.2	1.4	3.5	5.1	Setosa
3	1	0.2	1.4	3	4.9	Setosa
4	1	0.2	1.3	3.2	4.7	Setosa
5	1	0.2	1.5	3.1	4.6	Setosa
6	1	0.2	1.4	3.6	5	Setosa
7	1	0.4	1.7	3.9	5.4	Setosa
8	1	0.3	1.4	3.4	4.6	Setosa
9	1	0.2	1.5	3.4	5	Setosa
10	1	0.2	1.4	2.9	4.4	Setosa
11	1	0.1	1.5	3.1	4.9	Setosa
12	1	0.2	1.5	3.7	5.4	Setosa
13	1	0.2	1.6	3.4	4.8	Setosa
14	1	0.1	1.4	3	4.8	Setosa
15	1	0.1	1.1	3	4.3	Setosa
16	1	0.2	1.2	4	5.8	Setosa
17	1	0.4	1.5	4.4	5.7	Setosa
18	1	0.4	1.3	3.9	5.4	Setosa
19	1	0.3	1.4	3.5	5.1	Setosa
20	1	0.3	1.7	3.8	5.7	Setosa
21	1	0.3	1.5	3.8	5.1	Setosa
22	1	0.2	1.7	3.4	5.4	Setosa
23	1	0.4	1.5	3.7	5.1	Setosa
24	1	0.2	1	3.6	4.6	Setosa
25	1	0.5	1.7	3.3	5.1	Setosa
26	1	0.2	1.9	3.4	4.8	Setosa
27	1	0.2	1.6	3	5	Setosa
28	1	0.4	1.6	3.4	5	Setosa
29	1	0.2	1.5	3.5	5.2	Setosa
30	1	0.2	1.4	3.4	5.2	Setosa

Click on XLMiner add-on and select chart wizard.

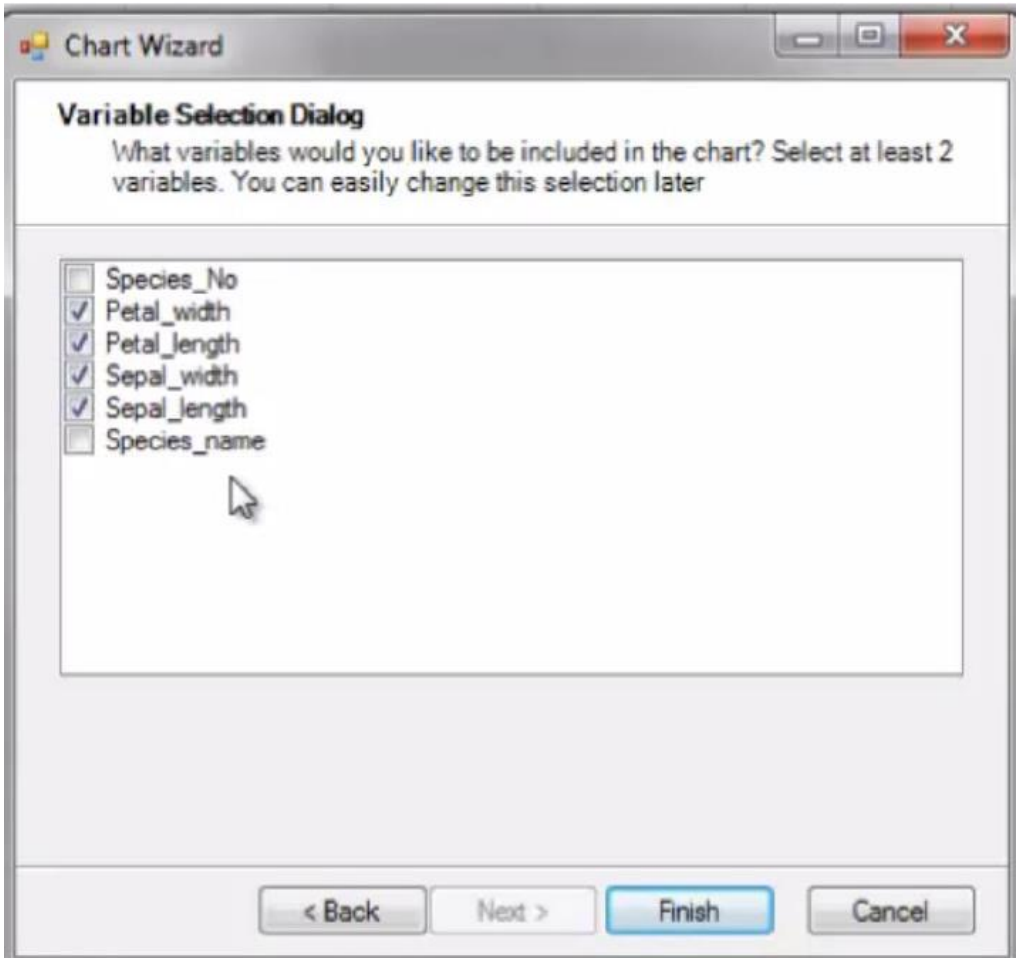


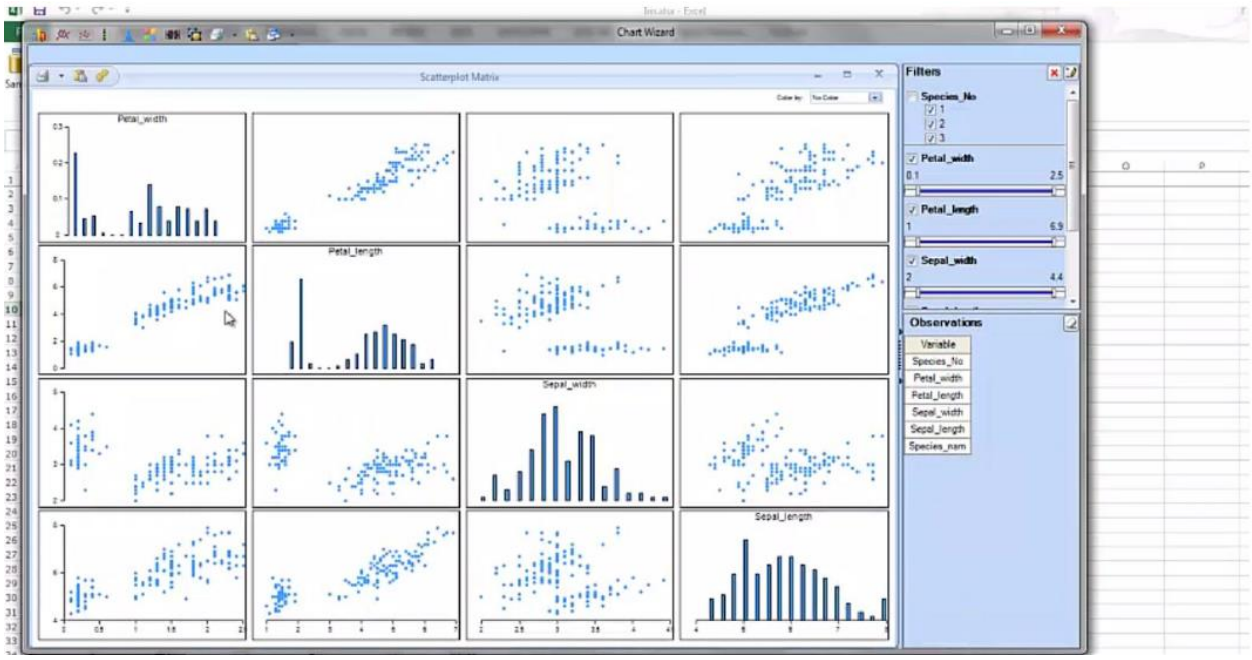
	C	D	E	F	G
1	Petal_length	Sepal_width	Sepal_length	Species_name	
2	1.4	3.5	5.1	Setosa	
3	1.4	3	4.9	Setosa	
4	1.3	3.2	4.7	Setosa	
5	1.5	3.1	4.6	Setosa	
6	1.4	3.6	5	Setosa	
7	1.7	3.9	5.4	Setosa	
8	1.4	3.4	4.6	Setosa	
9	1.5	3.4	5	Setosa	
10	1.4	2.9	4.4	Setosa	
11	1.5	3.1	4.9	Setosa	
12	1.5	3.7	5.4	Setosa	
13	1.6	3.4	4.8	Setosa	

Visulizing scatter plots select it and click on the next.



Select independent variables to explore the relation between them.





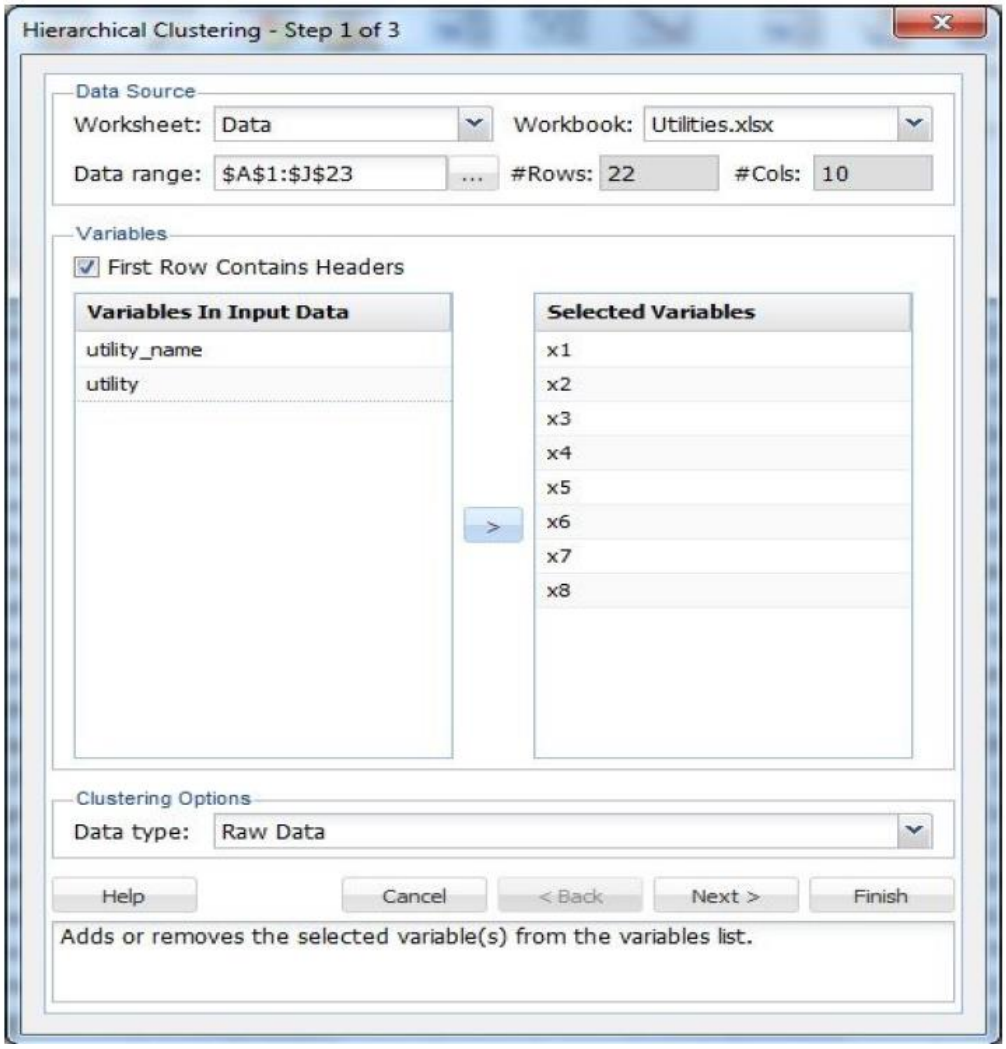
Selecting color by the name of the species.

Here we can clearly distinguish between various species and find their patterns and relationships among them.

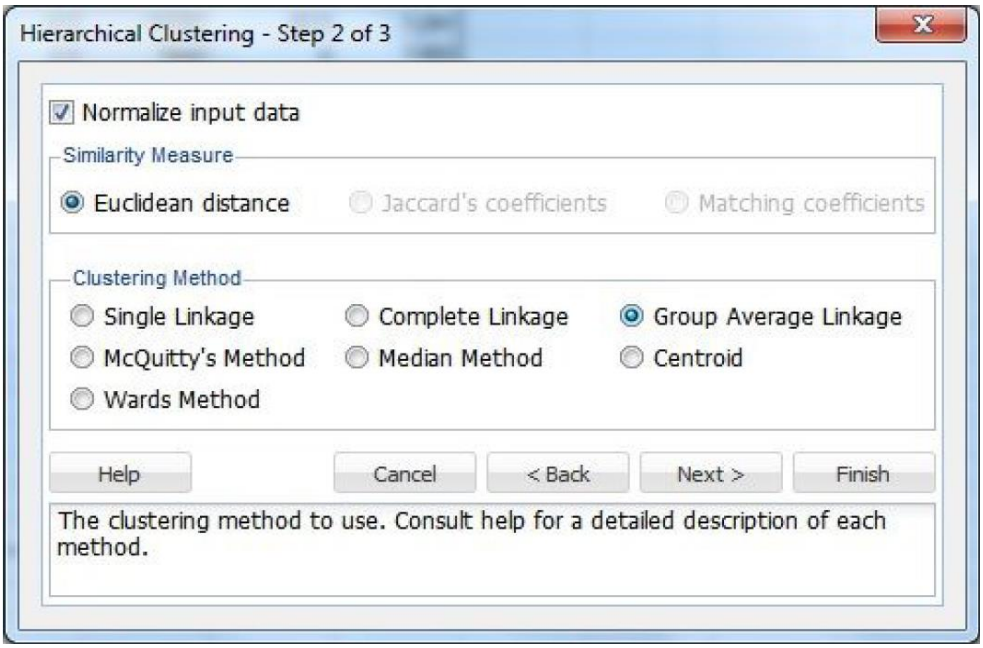
Now lest create a cluster analysis model using the US utility dataset.

utility_name	utility	x1	x2	x3	x4	x5	x6	x7	x8
Arizona	1	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	2	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	3	1.43	15.4	113	53	3.4	9212	0	1.058
Common	4	1.02	11.2	168	56	0.3	6423	34.3	0.7
Consolid	5	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	6	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	7	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	8	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	9	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	10	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	11	0.75	7.5	173	51.5	6.5	17441	0	0.768
NewEngla	12	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	13	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	14	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	15	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	16	1.16	9.9	252	56	9.2	15991	0	0.62
SanDiego	17	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	18	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	19	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsi	20	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	21	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	22	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

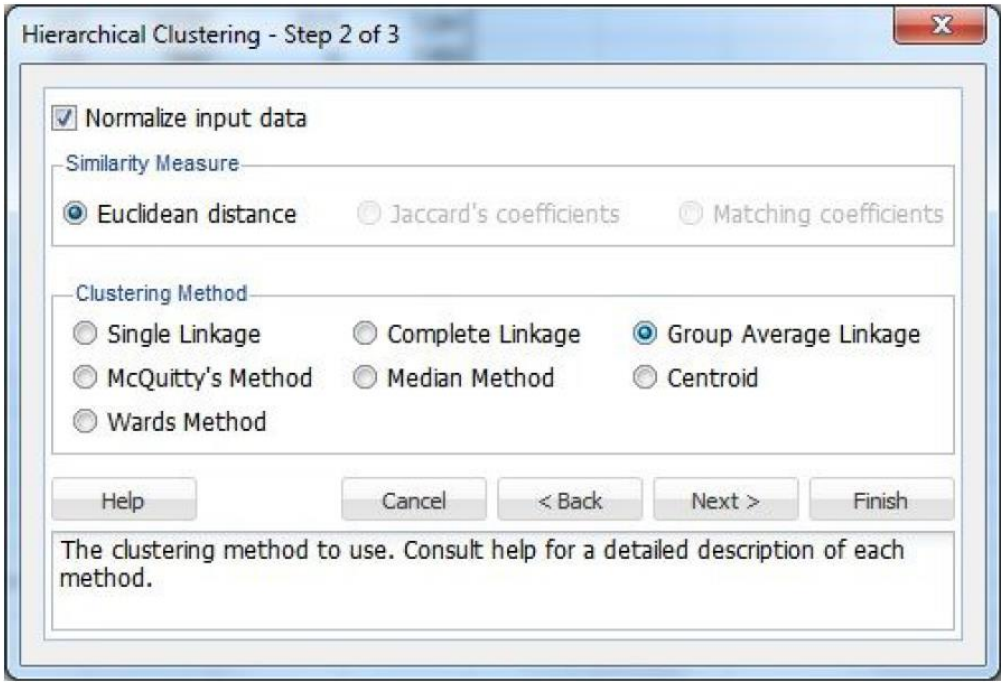
Here we will create Hierarchical clustering.



Preprocessing the data



Setting the number of cluster 4 and also normalizing input data.



After clicking on finish our model will be created and started training after then we can see the analysis of created model.

XLMiner : Hierarchical Clustering

Date: 24-Ma

Output Navigator			
Predicted Clusters	Dendrogram	Inputs	Clustering Stages

Elapsed Times in Milliseconds		
Clustering Time	eport Tim	Total
0	0	0

Inputs

Data	
Workbook	Utilities.xlsx
Worksheet	Data
Range	$A$1:$J$23
# Records in the input data	22
Input variables normalized	Yes
Data Type	Raw Data

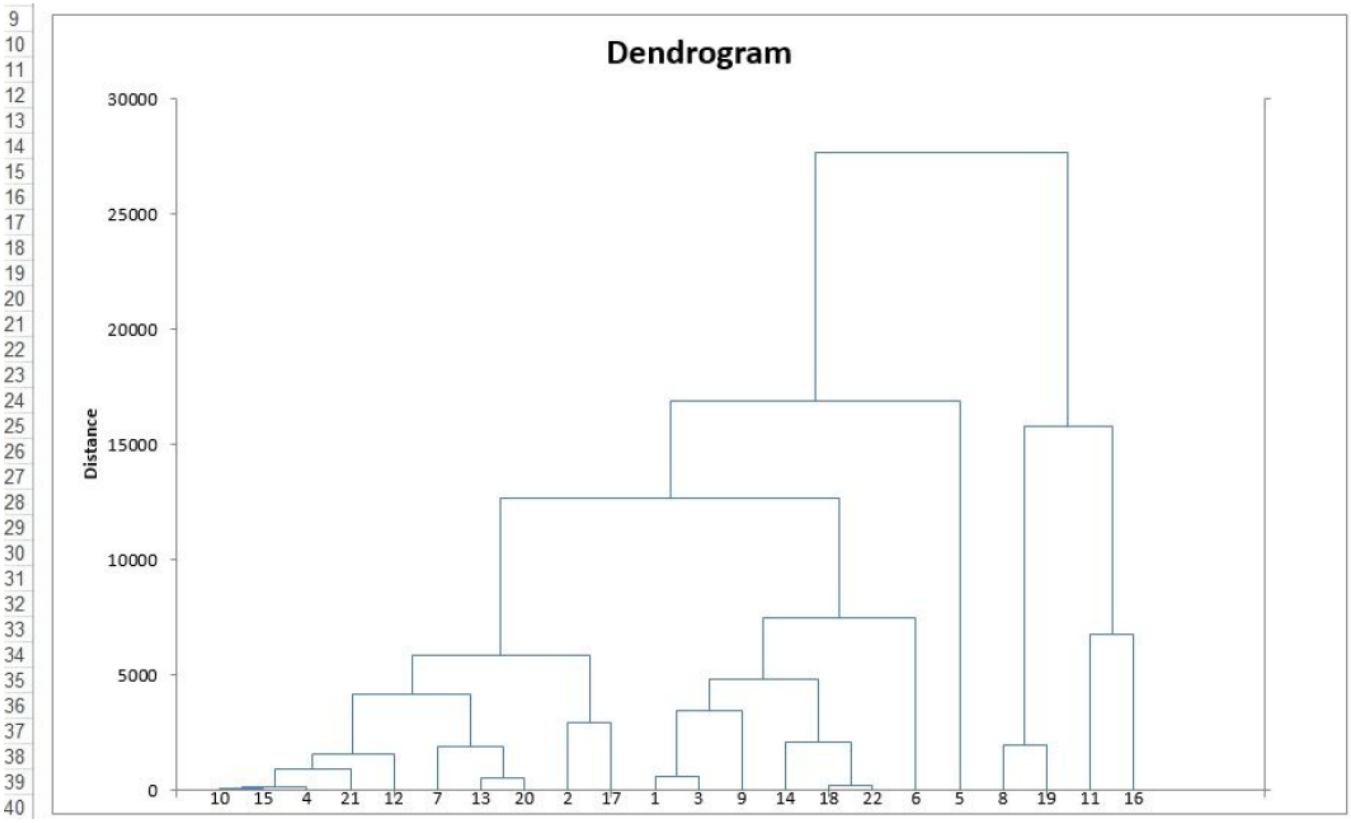
Variables								
# Selected Variables	8							
Selected Variables	x1	x2	x3	x4	x5	x6	x7	x8

Parameters/Options	
Draw Dendrogram	Yes
Selected Similarity Measure	Euclidean Distance
Selected Clustering Method	Group Average Linkage
Show Cluster Membership	Yes
# Clusters	4

Clustering Stages

Stage	Cluster 1	Cluster 2	Distance
1	12	21	1.384124
2	10	13	1.407032
3	4	20	1.816465
4	14	19	1.876051
5	1	18	1.877248
6	4	10	2.087329
7	7	12	2.167725
8	8	16	2.201457
9	1	14	2.324667
10	2	22	2.421916
11	7	15	2.452042
12	2	4	2.724784
13	3	9	2.752623
14	1	6	3.127672
15	1	3	3.265603
16	8	11	3.446275
17	7	17	3.642377
18	1	2	3.64586
19	1	7	4.074331
20	1	5	4.368398
21	1	8	4.608095

Visualizing Dendrogram of the created clusters



Practicals-6

Aim: To perform hand on experiments with sample data sets on Weka.

What is Weka?

Weka is a collection of machine learning algorithms for data mining tasks.
The algorithms can either be applied directly to a dataset or called from your own Java code.
Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Features of Weka

1. machine learning
2. data mining
3. preprocessing
4. classification
5. regression
6. clustering
7. association rules
8. attribute selection
9. experiments
10. workflow
11. visualization

❖ Perform Apriori Algorithm on Existing Dataset

1. Start Weka tool, click on “Explorer” button from Application group box.

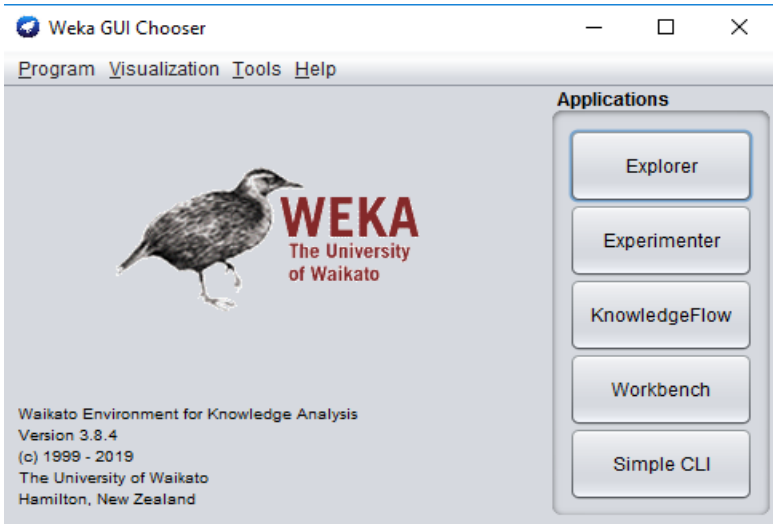


Fig-6.1

2. It will load up the Explorer window of weka tool.

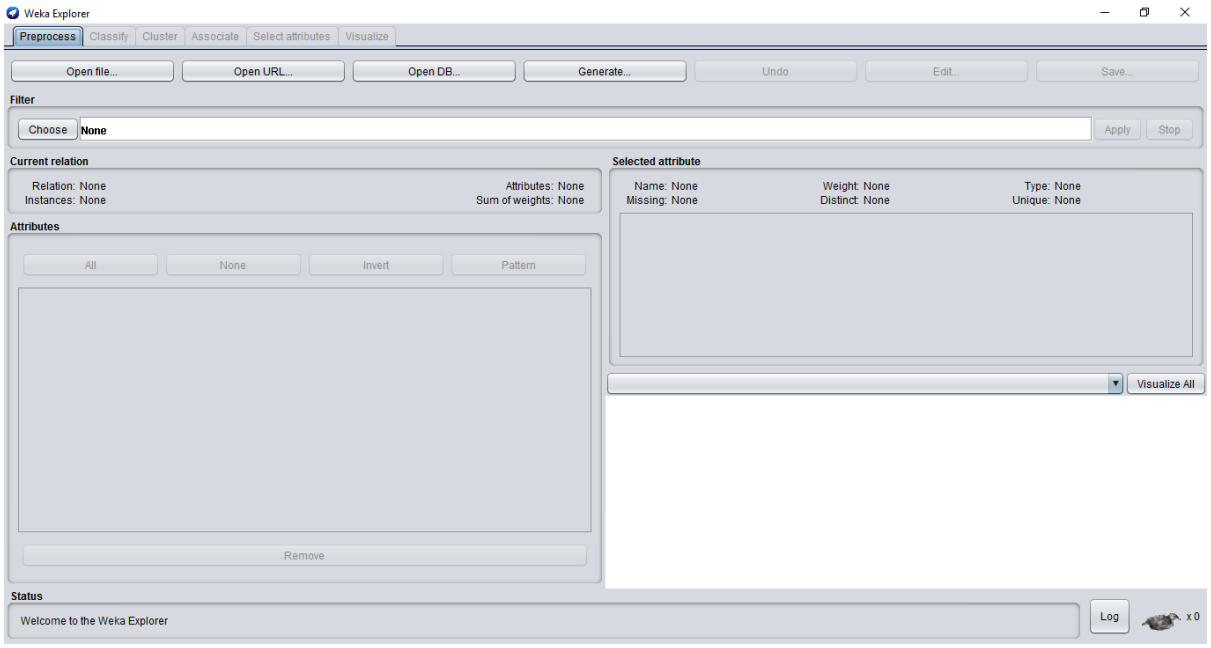


Fig-6.2

3. Then click on **Open file** button**,** it will **File Chooser** window, in that paste the following path as **file name**field and click open.

C:\Program Files\Weka-3-8-4\data\supermarket.arff

This file **supermarket.arff** is a sample databae bundled and shipped with the weka, to parform and test your skills, the location "*C:\Program Files\Weka-3-8-4\data*" has many sample databases, you can choose yours from there.

4. It will load the database into Weka.

5. After loading the database we can see the different data like attributes, relation name, instances and visualization (Shown in below figure).

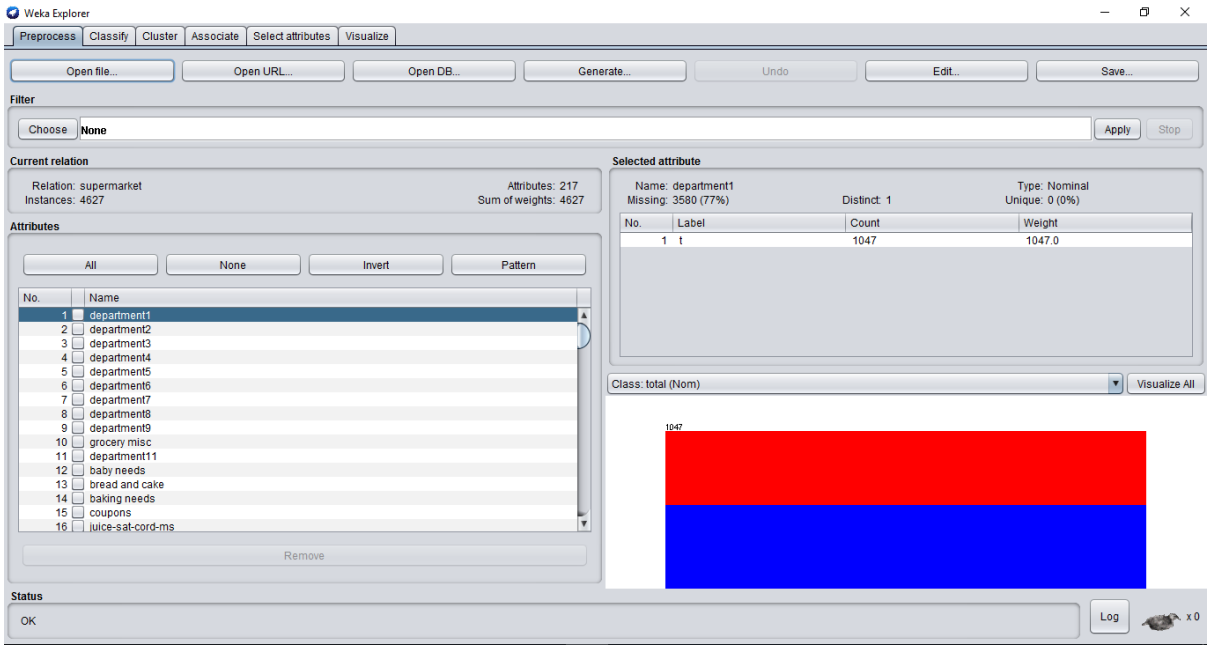


Fig-6.3

6. Now click on "**Associate**" tab.

7. It will display "Associator" window.

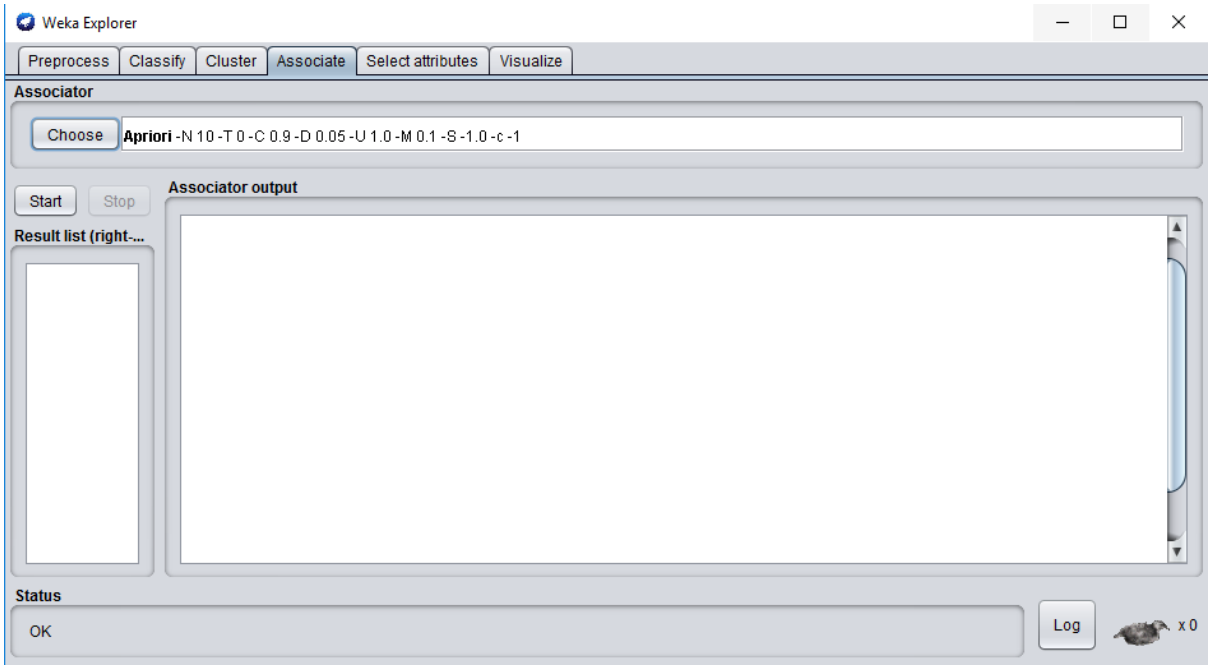


Fig-6.4

8. In that windows by default ***Apriori*** is chosen, we only need to press start to run algorithm, we can choose other algorithm by pressing that **Choose** button.

9. Press **start** btton.
10. Weka tool will start processing.

11. Once the processing finishes, it will display the result (shown in below fig.)

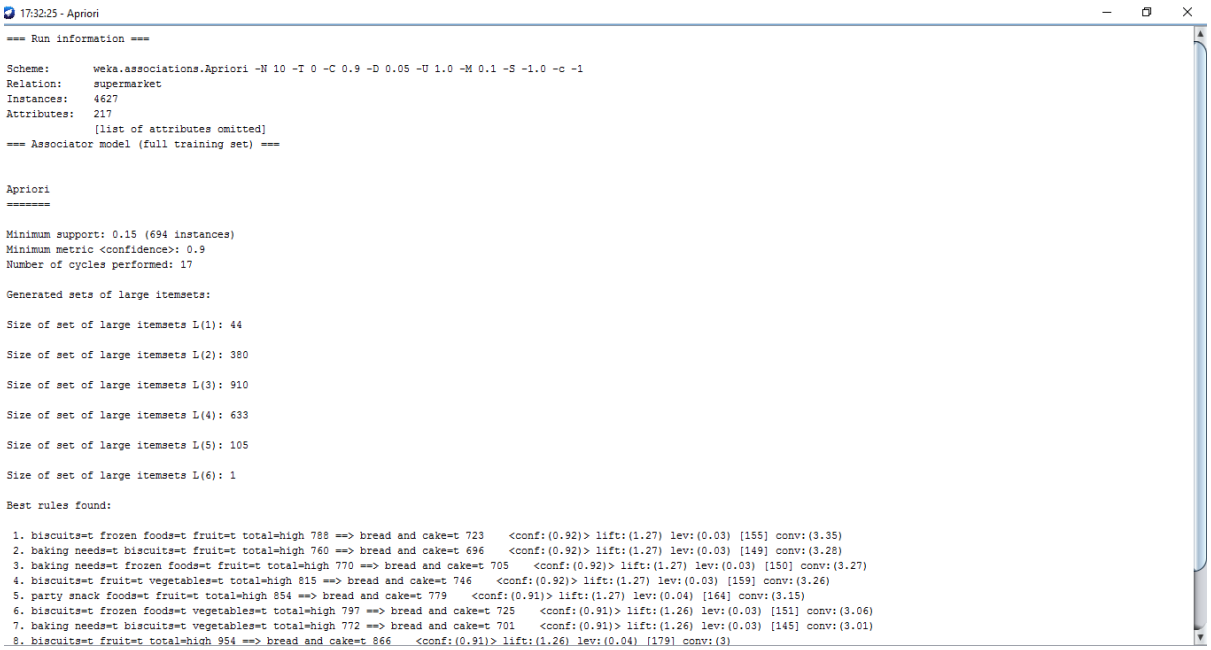


Fig-6.5 (output)

The Output contains:
- Scheme name
- No. Instances
- No. Attributes
- Minimum Support
- Minimum Confidence

This is how apriori algorithm can be performed on custom/existing data sets.

Practical-7

Aim: Design and Create cube by identifying measures and dimensions for Star Schema, Snowflake schema by SQL Server Analysis Service.
Software Required: Analysis services- SQL Server-2005.
Knowledge Required: Data cube

Theory/Logic:

Creating a Data Cube

To build a new data cube using BIDS, you need to perform these steps:
• Create a new Analysis Services project
• Define a data source
• Define a data source view
• Invoke the Cube Wizard
We'll look at each of these steps in turn.

Creating a New Analysis Services Project
To create a new Analysis Services project, you use the New Project dialog box in BIDS.
This is very similar to creating any other type of new project in Visual Studio.

To create a new Analysis Services project, follow these steps:

1. Select Microsoft SQL Server 2005 ⇒ SQL Server Business intelligence. Development Studio from the Program s 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.

   Figure 1 shows the Solution Explorer window of the new project, ready to be populated with objects.

Fig. 1 New Analysis Services project

Defining a Data Source

To define a data source, you'll use the Data Source Wizard. You can launch this wizard by right-clicking on the Data Sources folder in y our new Analysis Services project. The wizard will walk you through the process of defining a data source for your cube, including choosing a connection and specifying security credentials to be used to connect to the data source.

To define a data source for the new cube, follow these steps:

1. Right-click on the Data Sources folder in Solution Explorer and select New Data Source
2. Read the first page of the Data Source Wizard and click next
3. You can base a data source on a new or an existing connection. Because you don't have any existing connections, click New.
4. In the Connection Manager Dialog box, select the server containing your analysis services sample database from the Server Name combo box.
5. Fill in your authentication information.
6. Select the Native OLE DB\SQL Native Client provider (this is the default provider).
7. Select the Adventure Works DW database. Figure 2 shows the filled in Connection Manager Dialog box.
8. Click OK to dismiss the Connection Manager Dialog box.
9. Click Next.

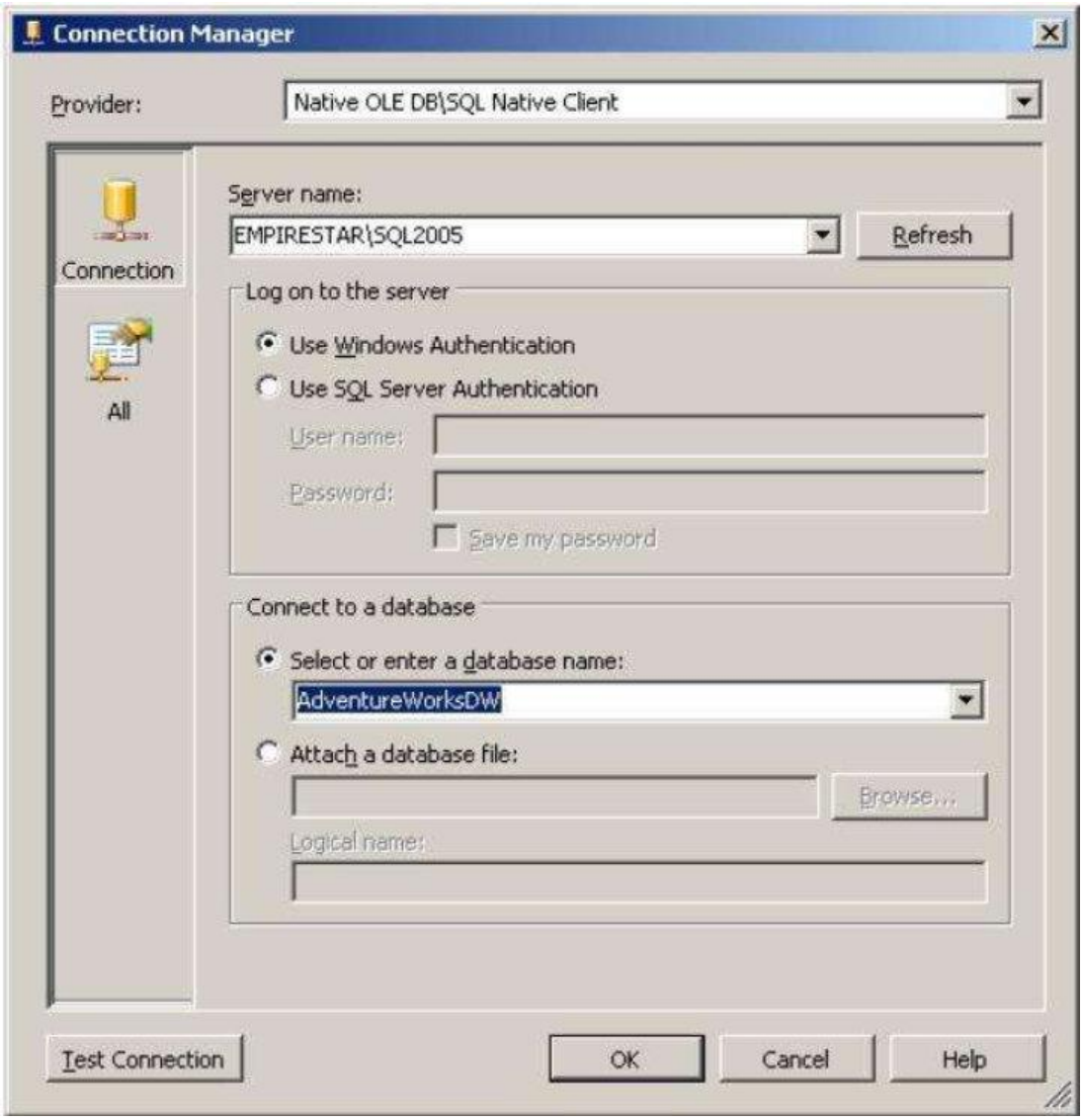


Fig. 2 Setting up a connection

10. Select Default impersonation information to use the credentials you just supplied for the connection and click Next.
11. Accept the default data source name and click Finish.

Defining a Data Source View

A data source view is a persistent set of tables from a data source that supply the data for a particular cube. BIDS also includes a wizard for creating data source views, which you can invoke by right-clicking on the Data Source Views folder in Solution Explorer.

To create a new data source view, follow these steps:

1. Right-click on the Data Source Views folder in Solution Explorer and select New Data Source View.
2. Read the first page of the Data Source View Wizard and click Next.
3. Select the Adventure Works DW data source and click Next. Note that you could also launch the Data Source Wizard from here by clicking New Data Source.
4. Select the dbo.FactFinance table in the Available Objects list and click the ⇒ button to move it to the Included Object list. This will be the fact table in the new cube.
5. Click the Add Related Tables button to automatically add all of the tables that are directly related to the dbo.FactFinance table. These will be the dimension tables for the new cube. Figure 3 shows the wizard with all of the tables selected.
6. Click Next.
7. Name the new view Finance and click Finish. BIDS will automatically display the schema of the new data source view, as shown in Figure 4.

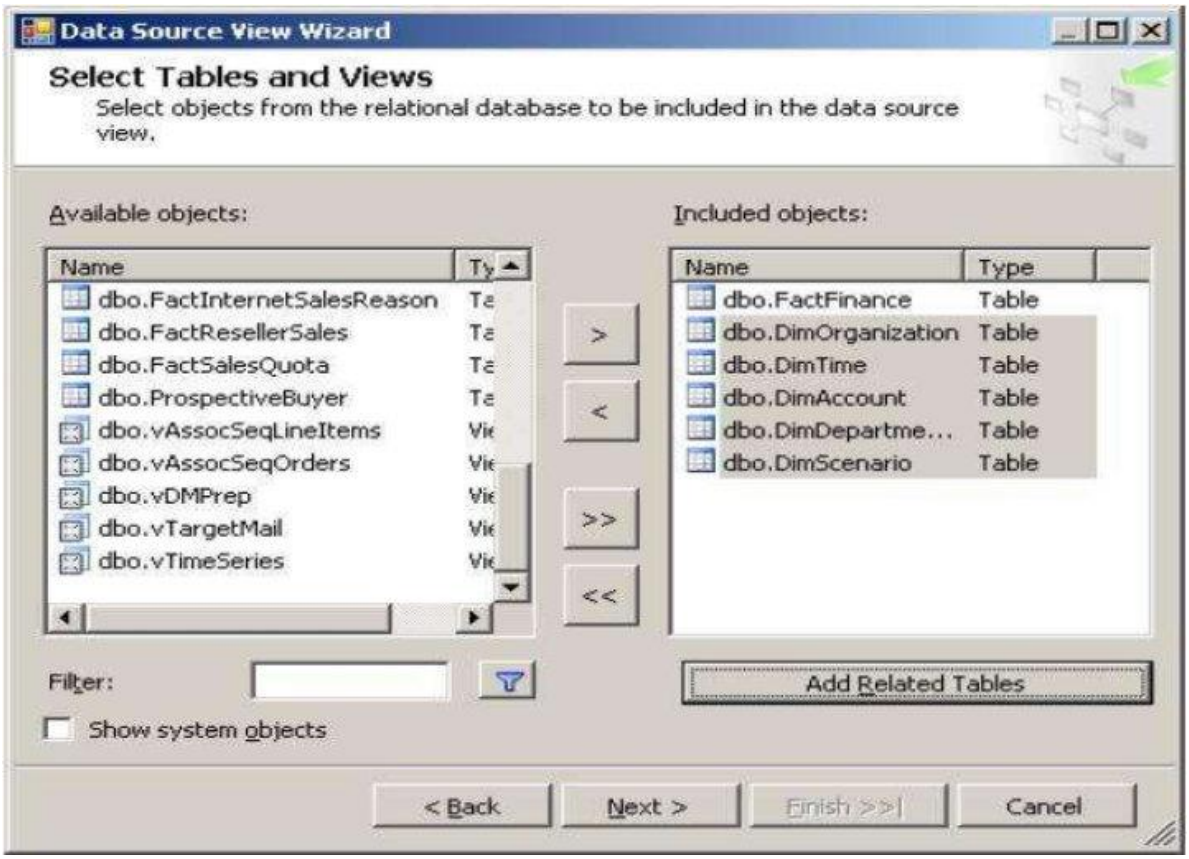


Fig. 3 Selecting tables for the data source view

Analysis Services

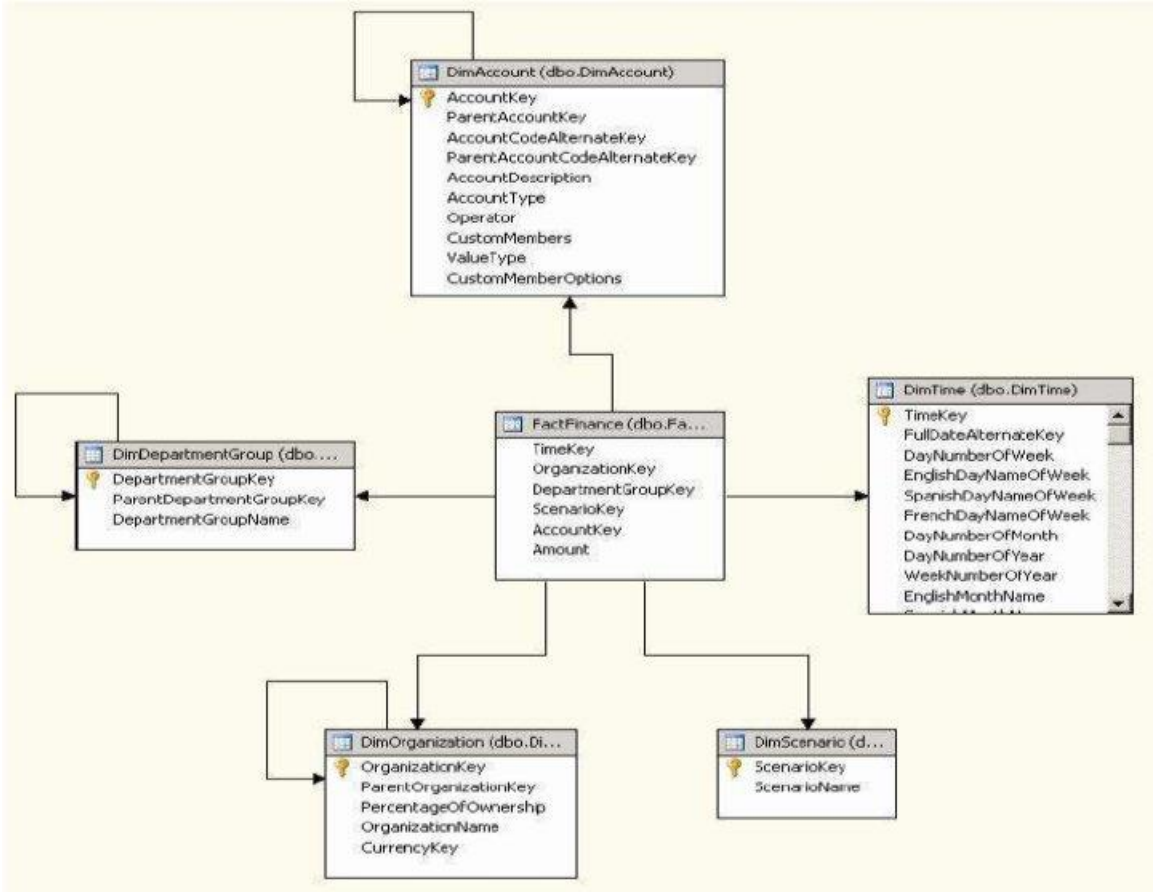


Fig. 4 The Finance data source view

Invoking the Cube Wizard

As you can probably guess at this point, you invoke the Cube Wizard by right clicking on the Cubes folder in Solution Explorer. The Cube Wizard interactively explores the structure of your data source view to identify the dimensions, levels, and measures in your cube.

To create the new cube, follow these steps:

1. Right-click on the Cubes folder in Solution Explorer and select New Cube.
2. Read the first page of the Cube Wizard and click Next2
3. Select the option to build the cube using a data source.
4. Check the Auto Build checkbox.
5. Select the option to create attributes and hierarchies.
6. Click Next.
7. Select the Finance data source view and click next.
8. Wait for the Cube Wizard to analyze the data and then click next.
9. The Wizard will get most of the analysis right, but you can fine-tune it a bit. Select DimTime in the Time Dimension combo box. Uncheck the Fact checkbox on the line for the dbo.Dim Time table. This will allow you to analyse this dimension using standard time periods.

10. Click Next.
11. On the Select Time Periods page, use the combo boxes to match time Property names to time columns according to Table 1.

Time Property Name	Time Column
Year	CalendarYear
Quarter	CalendarQuarter
Month	MonthNumberOfYear
Day of Week	DayNumberOfWeek
Day of Month	DayNumberOfMonth
Day of Year	DayNumberOfYear
Week of Year	WeekNumberOfYear
Fiscal Quarter	FiscalQuarter
Fiscal Year	FiscalYear

Fig. 5 Time columns for financial cube

12. Click Next.
13. Accept the default measures and click next.
14. Wait for the Cube Wizard to detect hierarchies and then click next.
15. Accept the default dimension structure and click next.
16. Name the new cube Finance Cube and click Finish.

Deploying and Processing a Cube

At this point, you've defined the structure of the new cube - but there's still more work to be done. You still need to deploy this structure to an Analysis Services server and then process the cube to create the aggregates that make querying fast and easy. To deploy the cube you just created, select Build Deploy Adventure WorksCube1. This will deploy the cube to your local Analysis Server, and also process the cube, ⇒ building the aggregates for you. BIDS will open the Deployment Progress window, as shown in Figure 5, to keep you informed during deployment and processing.

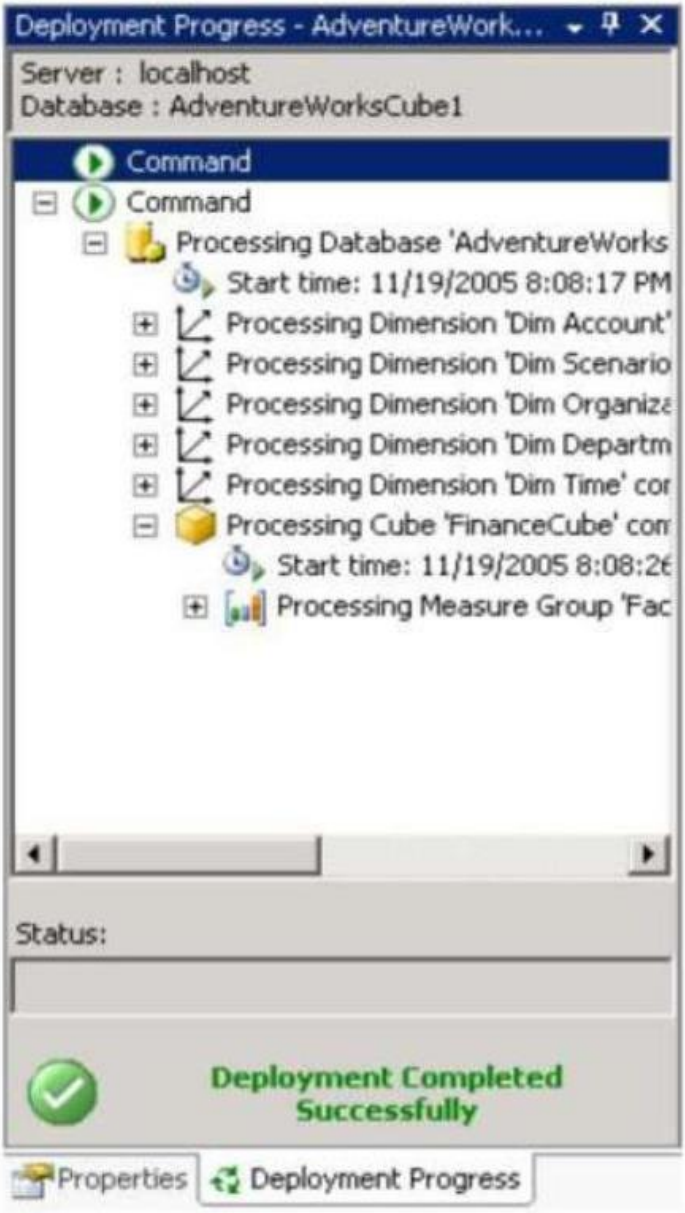


Fig. 6 Deploying a cube

Exploring a Data Cube

At last you're ready to see what all the work was for. BI DS includes a built-in Cube Browser that lets you interactively explore the data in any cube that has been deployed and processed. To open the Cube Browser, right-click on the cube in Solution Explorer and select Browse. Figure 1-6 shows the default state of the Cube Browser after it's just been opened. The Cube Browser is a drag and-drop environment. If you've worked with pivot tables in Microsoft Excel, you should have no trouble using the Cube browser. The pane to the left includes all of the measures and dimensions in your cube, and the pane to the right gives you drop targets for these measures and dimensions. Among other operations, you can:

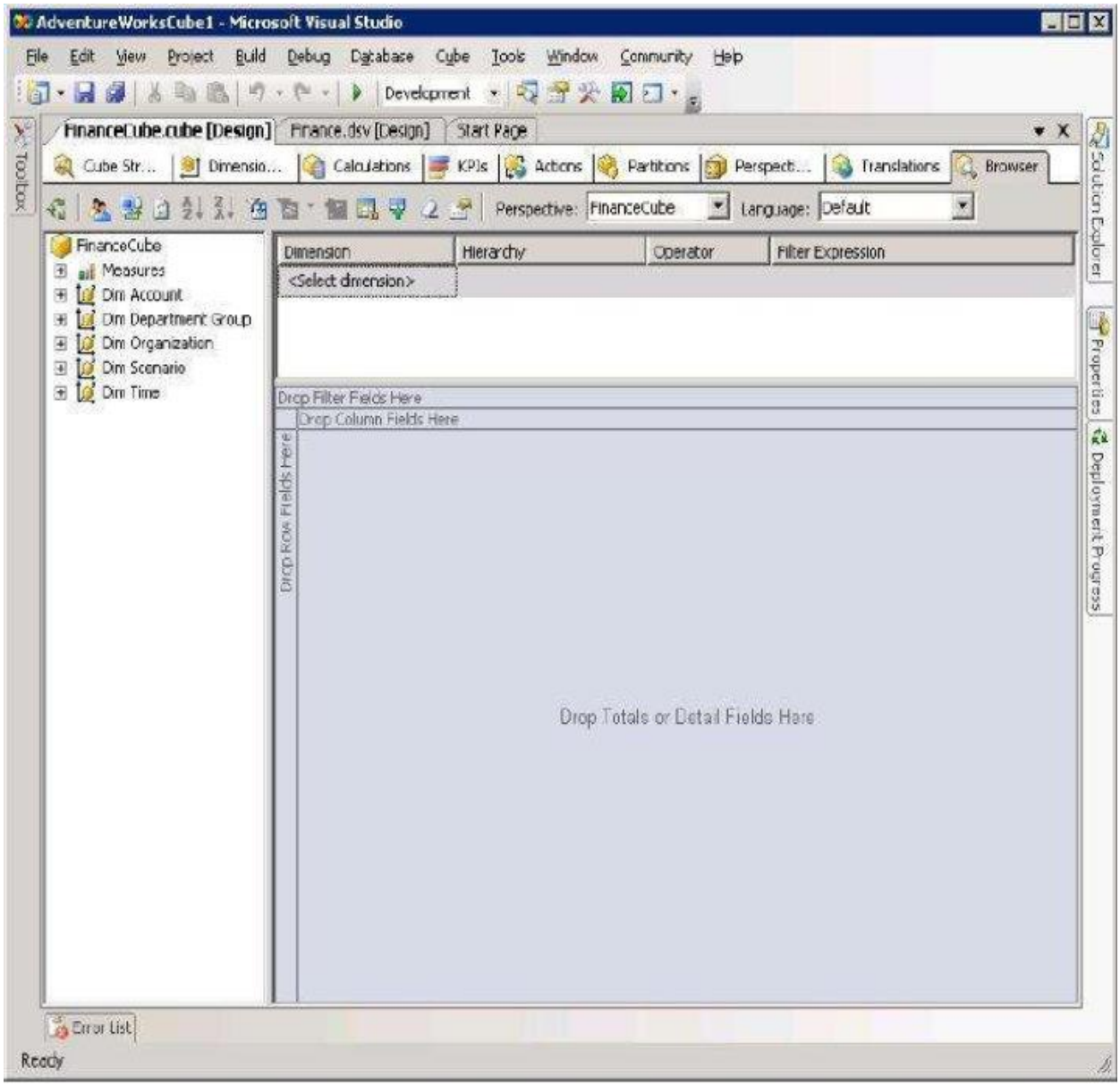


Fig. 7 The cube browser in BIDS

- Drop a measure in the Totals /Detail area to see the aggregated data for that measure.
- Drop a dimension or level in the Row Fields area to summarize by that level or dimension on rows.
- Drop a dimension or level in the Column Fields area to summarize by that level or dimension on columns
- Drop a dimension or level in the Filter Fields area to enable filtering by members of that dimension or level.
- Use the controls at the top of the report a rea to select additional filtering expressions.

To see the data in the cube you just created, follow these steps:

1. Right-click on the cube in Solution Explorer and select Browse.
2. Expand the Measures node in the metadata panel (the area at the left of the user interface).

3. Expand the Fact Finance node.
4. Drag the Amount measure and drop it on the Totals/Detail area.
5. Expand the Dim Account node in the metadata panel.
6. Drag the Account Description property and drop it on the Row Fields area.
7. Expand the Dim Time node in the metadata panel.
8. Drag the Calendar Year-Calendar Quarter-Month Number of Year hierarchy and drop it on the Column Fields area.
9. Click the + sign next to year 2001 and then the + sign next to quarter 3.
10. Expand the Dim Scenario node in the metadata panel.
11. Drag the Scenario Name property and drop it on the Filter Fields area.
12. Click the dropdown arrow next to scenario name. Uncheck all of the checkboxes except for the one next to the Budget name.

Figure 7 shows the result. The Cube Browser displays month-by-month budgets by account for the third quarter of 2001. Although you could have written queries to extract this information from the original source data, it's much easier to let Analysis Services do the heavy lifting for you.

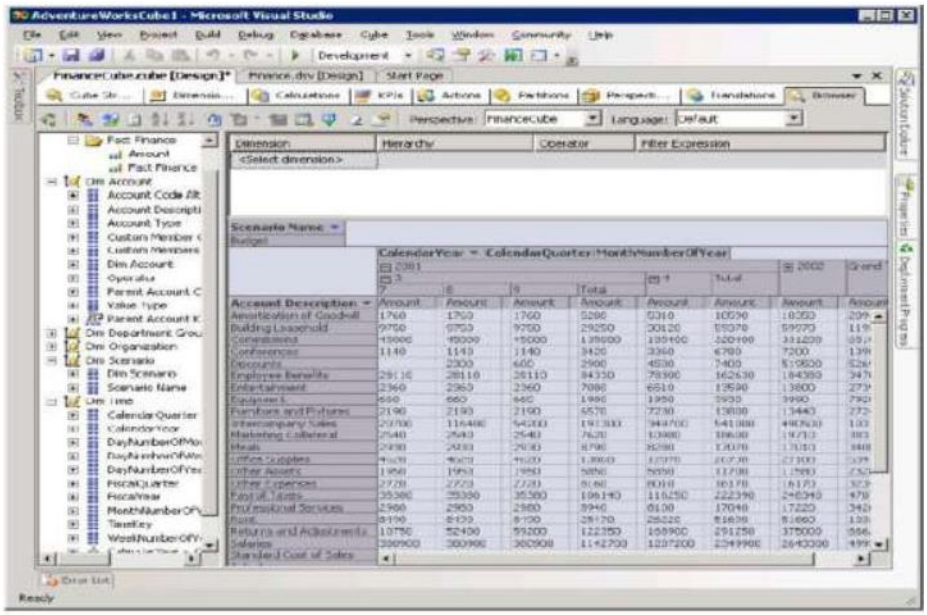


Fig. 8 Exploring cube data in the cube browser

Practical-8

Aim: Design and Create cube by identifying measures and dimensions for Design storage for cube using storage mode MOLAP, ROLAP and HOALP.

Software Required: Analysis services- SQL Server-2005.
Knowledge Required: MOLAP, ROLAP, and HOLAP
Theory/Logic:
Partition Storage (SSAS)
Physical storage options affect the performance, storage requirements, and storage locations of partitions and their parent measure groups and cubes. A partition can have one of three basic storage modes:

- Multidimensional OLAP (MOLAP)
- Relational OLAP (ROLAP)
- Hybrid OLAP (HOLAP)

Microsoft S QL Server 2005 Analysis Services (SSAS) supports all three basic storage modes. It also supports proactive caching, which enables you to com bine the characteristics of ROLAP and MOLAP storage for both immediacy of data and query performance. You can configure the storage mode and proactive caching options in one of three ways.

Storage Configuration Method	Description
Storage Settings dialog	You can configure storage settings for a partition or configure default storage settings for a measure group
Storage Design Wizard	You can configure storage settings f or a partition at the same time that you design aggregations. You can also define a filter to restrict the source data that is read into the partition using any of the three storage modes.
Usage-Based Optimization Wizard	You can select a storage m ode and optimize aggregation design based on queries that have been sent to the cube

MOLAP

The MOLAP storage mode causes the aggregations of the partition and a copy of its source data to be stored in a multidimensional structure in Analysis Services, which structure is highly optimized to maximize query performance. This can be storage on the computer where the partition is defined or on another Analysis Services computer. Storing data on the computer where the partition is defined creates a local partition. Storing data on another Analysis Services computer creates are mote partition. The multidimensional structure that stores the partition's data is located in a sub folder of the Data f older of the Analysis Service s program files or another location specified during setup of Analysis Services.

Because a copy of the source data resides in the Analysis Services data folder, queries can be resolved without accessing the partition's source data even when the results cannot

be obtained from the partition's aggregations. The MOLAP storage mode provides the most rapid query response times, even without aggregations, but which can be improved substantially through the use of aggregations.

As the source data changes, objects in MOLAP storage must be processed periodically to incorporate those changes. The time between one processing and the next creates a latency period during which data in OLAP objects may not match the current data. You can incrementally update objects in MOLAP storage without down time. However, there may be some downtime required to process certain changes to OLAP objects, such as structural changes. You can minimize the downtime required to update MOLAP storage by updating and processing cubes on a staging server and using database synchronization to copy the processed objects to the production server. You can also use proactive caching to minimize latency and maximize availability while retaining much of the performance advantage of MOLAP storage.

ROLAP

The ROLAP storage mode causes the aggregations of the partition to be stored in tables in the relational database specified in the partition's data source. Unlike the MOLAP storage m ode, ROLAP does not cause a copy of the source data to be stored in the Analysis Services data folders. When results cannot be derived from the aggregations or query cache, the fact table in the data source is accessed to answer queries. With the ROLAP storage m ode, query response is generally slower than that available with the other MOLAP or HOLAP storage modes. Processing time is also typically slower. Real time ROL AP is typically used when clients need to see changes immediately. No aggregations are stored with real-time ROLAP. ROLAP is also used to save storage space for large datasets that are infrequently queried, such as purely historical data.

Note: When using ROLAP, Analysis Services m ay return incorrect information related to the unknown m ember if a join is combined with a group by, which eliminates relational integrity errors rather than returning the unknown member value. If a partition uses the R OLAP storage m ode and its source data is stored in SQL Serve r 2005 Analysis Services (SSAS), Analysis Services attempts to create indexed views to contain aggregations of the partition. If Analysis Services cannot create indexed views, it does not create aggregation tables. While Analysis Services handles the session requirements for creating indexed views on SQ L Server 2005 Analysis Services (SSAS) , the creation and use of indexed views for aggregations requires the following conditions to be met by the ROLAP partition and the tables in its schema:

- The partition cannot contain measures that use the **Min** or **Max** aggregate functions.
- Each table in the schema of the ROLAP partition must be used only once. For example, the schema cannot contain "dbo"."address" AS "Customer Address" and "dbo"."address" AS "SalesRep Address".
- Each table must be a table, not a view.
- All table names in the partition's schema must be qualified with the owner name, for example, "dbo"."customer".

- All tables in the partition's schema must have the same owner; for example, you cannot have a From Clause like : "tk"."Customer", "john"."store", or "dave"."sales_fact_2004".
- The source columns of the partition's measures must not be null able.
- All tables used in the view must have been created with the following options set to ON:
    1. ANSI_NULLS
    2. QUOTED_IDENTIFIER
- The total size of the index key, in SQL Server 2005, cannot exceed 900 bytes. SQL Server 2005 will assert this condition based on the fixed length key columns when the CREATE INDEX statement is processed. However, if there are variable length columns in the index key, SQL Server 2005 will also assert this condition for every update to the base tables. Because different aggregations have different view definitions, ROLAP processing using indexed views can succeed or fail depending on the aggregation design.
- The session creating the indexed view must have the following options on: ARITHABORT, C ONCAT_NULL_YEILDS_NULL, QUOTE D_IDENTIFIER, ANSI_NULLS, ANSI_ PADDING, and ANSI_ WARNING. This setting can be made in SQL Server Management Studio.
- The session creating the indexed view must have the following option off: NUMERIC_ROUNDABORT. This setting can be made in SQL Server Management Studio.

HOLAP

The HOLAP storage mode combines attributes of both MOLAP and ROLAP. Like MOLAP, HOLAP causes the aggregations of the partition to be stored in a multidimensional structure on an Analysis Services server computer. HOLAP does not cause a copy of the source data to be stored. F or queries that access only summary data contained in the aggregations of a partition, HOLAP is the equivalent of MOLAP. Queries th at access source data, such as a drilldown to an atomic cube cell for which there is no aggregation data, must retrieve data from the relational database and will not be as fast as if the source data were stored in the MOLAP structure.

Partitions stored as HOLAP are smaller than equivalent MOLAP partitions and respond faster than ROLAP partitions for queries involving summary data. HOLAP storage mode is generally suitable for partitions in cubes that require rapid query response for summaries based on a large amount of source data. However, where users generate queries that must touch leaf level data, such as for calculating median values, MOLAP is generally a better choice.

Steps:

1. In the Analysis service object explorer tree pan e, expand the Cubes folder, right click the created cube, and then click **Property.**

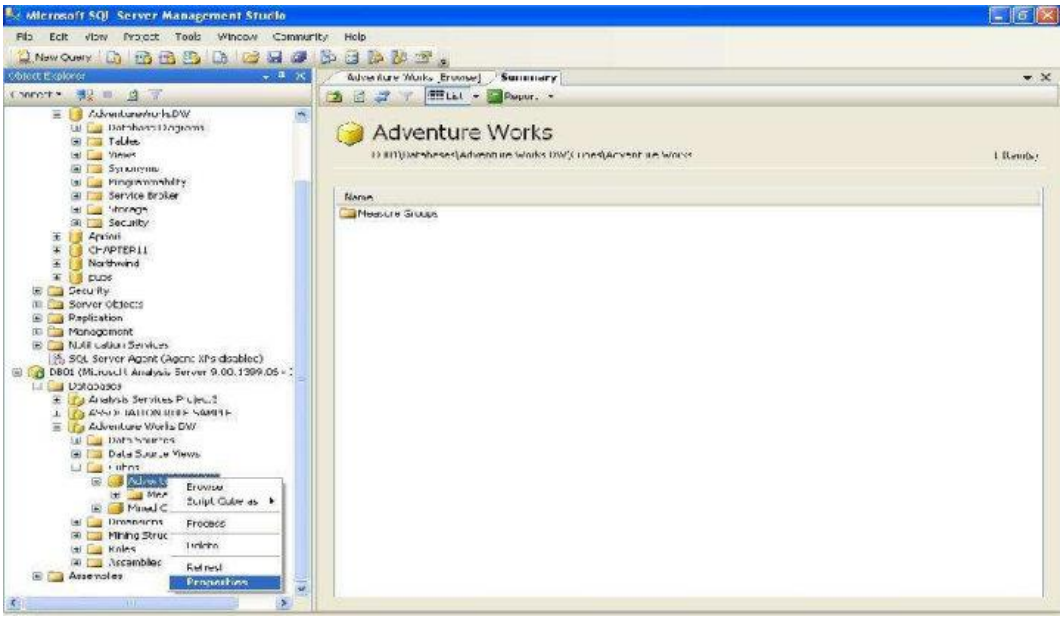


Fig. 1

2. In the property wizard, select **proactive caching and** then select **option button.**

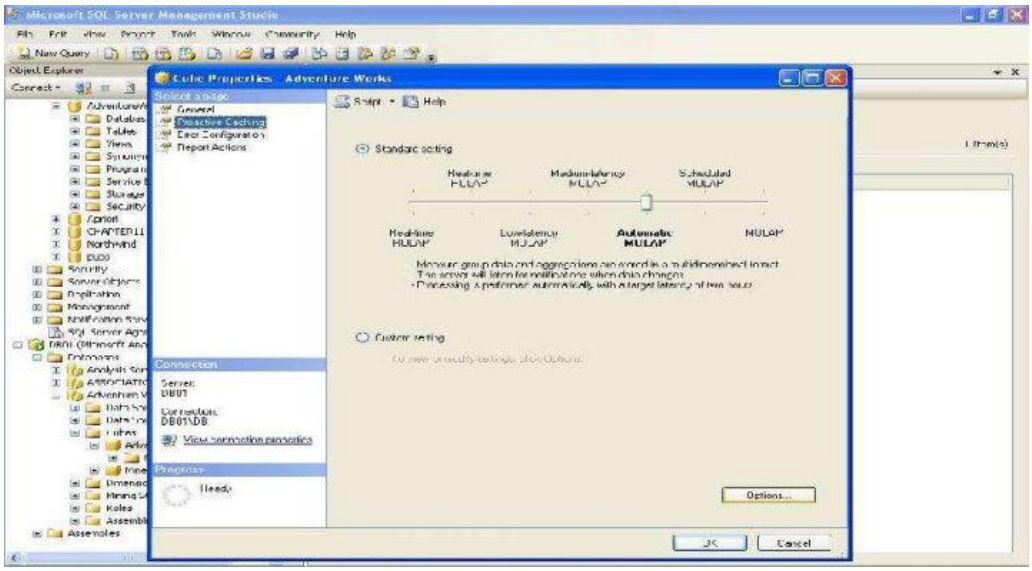


Fig. 2

3. Select MOLAP/HOLAP/ROLAP as your data storage type, and then click **Next**

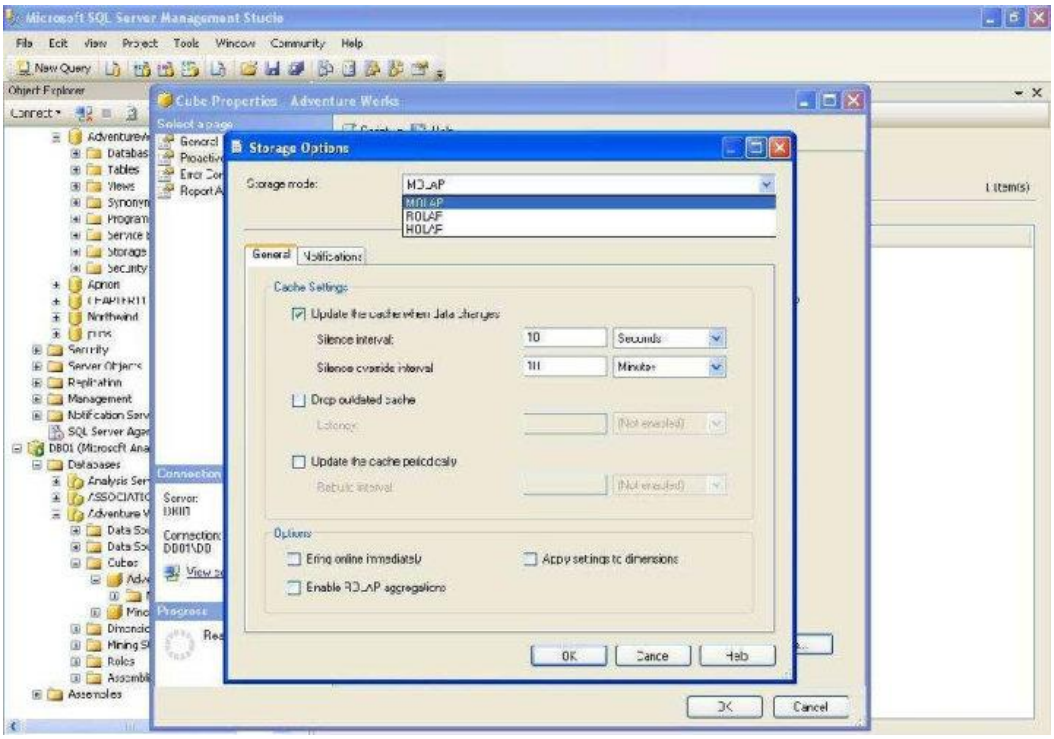


Fig. 3

4. After setting required parameters, **click ok button**.
5. After that right click on created cube and then select **Process.**

Practical - 9

Aim: Design and create data mining models using Analysis Services of SQL Server.

Software Required**:** Analysis services- SQL Server-2005.
Knowledge Required**:** Data Mining Models

Theory/Logic:
The tutorial is broken up into three sections.

1. Preparing the SQL Server Database,
2. Preparing the Analysis Services Database, and
3. Building and Working with the Mining Models.

1. Preparing the SQL Server Database

The AdventureWorksDW database, which is the basis for this tutorial, is installed with SQL Server (not by default, but as an option at installation time) and already contains views that will be used to create the mining models. If it was not installed at the installation time, you can add it by selecting Change button from Control Panel Æ Add/Remove Programs Æ Microsoft SQL Server 2005.

2. Preparing the Analysis Services Database

Before you begin to create and work with mining models, you must perform the following tasks:

1. Create a new Analysis Services project
2. Create a data source.
3. Create a data source view.

a) Creating an Analysis Services Project

Each Analysis Services project defines the schema for the objects in a single Analysis Services database. The Analysis Services database is defined by the mining models, OLAP cubes, and supplemental objects that it contains.

To create an Analysis Services project

1. Open Business Intelligence Development Studio.
2. Select **New** and **Project** from the **File** menu.
3. Select Analysis Services Project as the type for the new project and name it **Adventure Works.**

b) Creating a Data Source

A data source is a data connection that is saved and managed within your project and deployed to your Analysis Services database. It contains the server name and database where your source data resides, as well as any other required connection properties.

To create a data source

1. Right-click the **Data Source** project item in Solution Explorer and select **New Data Source**
2. On the Welcome page, click **next**
3. Click **New** to add a connection to the **AdventureWorksDW** database.
4. The **Connection Manager** Dialog box opens. In the **Server name** drop-down box, select the server where **AdventureWorksDW** is hosted (for example, localhost), enter your credentials, and then in the **Select the database on the server** drop-down box select the **AdventureWorksDW** database.
5. Click **OK** to close the **Connection Manager D**ialog box.
6. Click **Next**.
7. By default the data source is named Adventure Works DW. Click **Finish** The new data source, Adventure Works DW, appears in the Data Sources folder in Solution Explorer.

c) Creating a Data Source View

A data source view provides an abstraction of the data source, enabling you to modify the structure of the data to make it more relevant to your project. Using data source views, you can select only the tables that relate to your particular project, establish relationships between tab les, and add calculated columns and named views without modifying the original data source.

To create a data source view

1. In Solution Explorer, right-click **Data Source View**, and then click **New Data Source View**.
2. On the Welcome page, click **next**.
3. The **Adventure Works DW** data source you created in the last step is selected by default in the **Relational data sources** window. Click **Next**.
4. If you want to create a new data source, click **New Data Source** to launch the Data Source Wizard.
5. Select tables in the following list and click the right arrow button to include them in the new data source view:

6. **vAssocSeqLineItems**

7. **vAssocSeqOrders**

8. **vTargetMail**

9. **vTimeSeries**

10. Click **Next**

11. By default the data source view is named Adventure Works DW. Click **Finish**. Data Source View Editor opens to display the Adventure Works DW data source view, as shown in Figure Solution Explorer is also updated to include the new data source view.

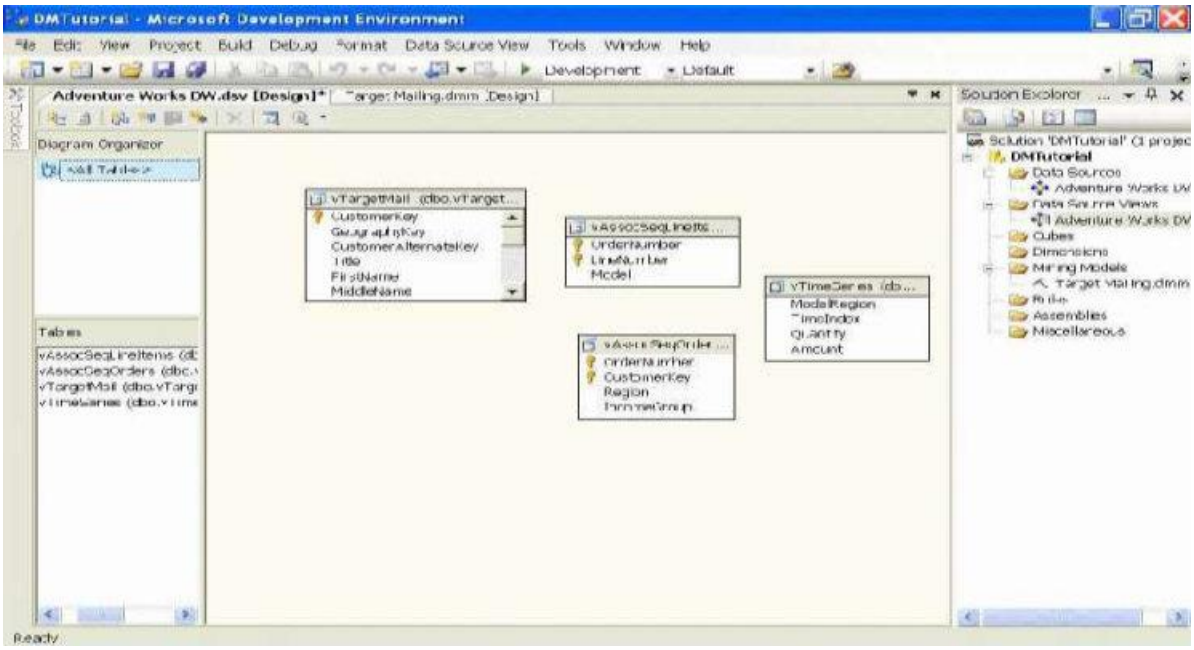


Fig. 1

Practical-10

Aim: Open Ended Problem : Perform hands on experiment on data mining tool like how that tool is uses and advance data mining technique on data set.

I've used RapidMiner Studio!

Why RapidMiner?

RapidMiner has a very large ML algorithms library and excellent tools for automated optimization of those algorithms. Is one of the best tools I know for text mining and analytics. It's not only very powerful but also very intuitive and easy to use.

Rapid Miner is a platform for data scientists and big data analysts to quickly analyse their data. Rapid Miner has taken a huge leap in the AI community since it is most popularly used by non-programmers and researchers. The platform provides a vast number of options in terms of plugins and data analysis techniques. Apart from this, it is also compatible with iOs, Android, and web application tools like Node JS and flask. This platform is useful for anyone with an idea they would like to experiment with without spending much time or effort on it.

You can download RapidMiner from www.rapidminer.com It is open source.

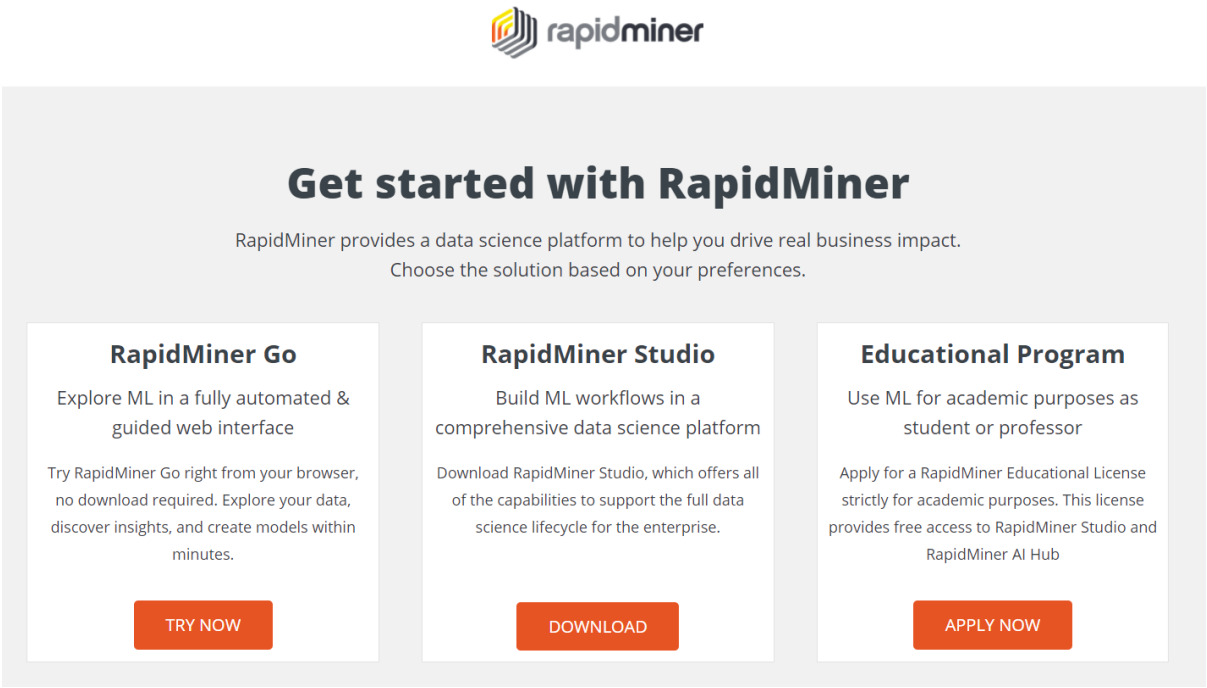


Fig. 10.1

In above image you can see there are various types of versions. I've used RapidMiner Studio.

Step by step representation of tool RapidMiner Studio:

1. Download the RapidMiner Studio and run it. You will see the below screen.

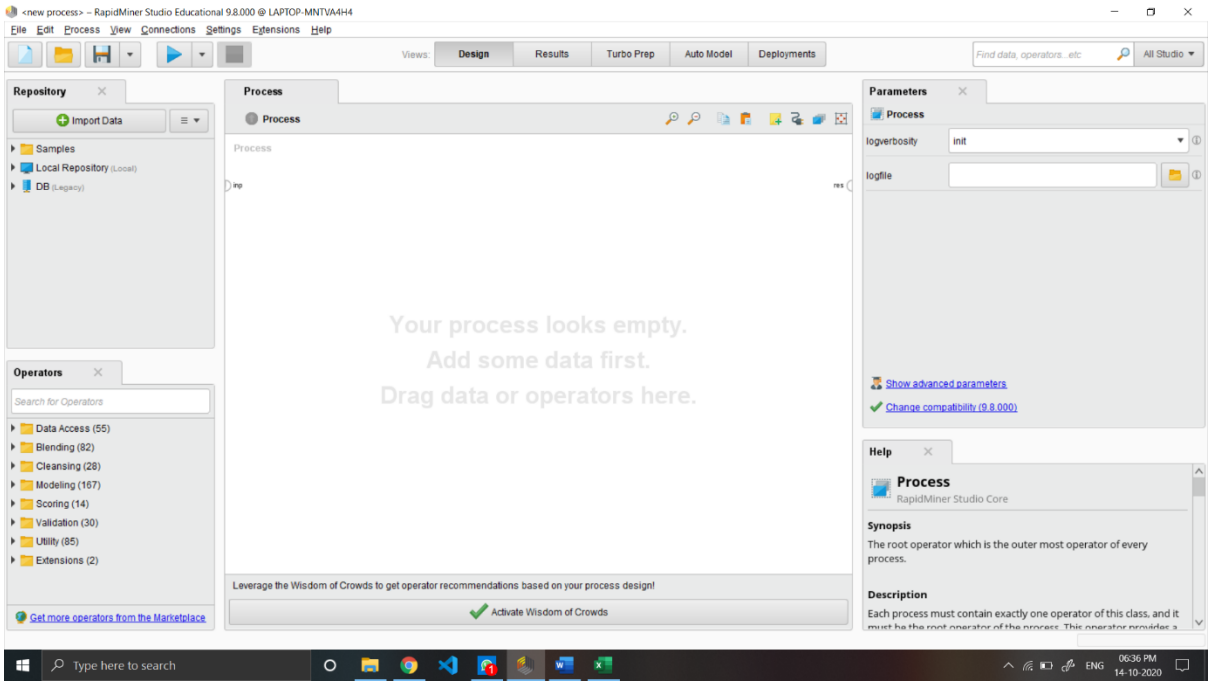


Fig. 10.2

2. Now you have to import the data. Click on the **Import Data** button and select the dataset.

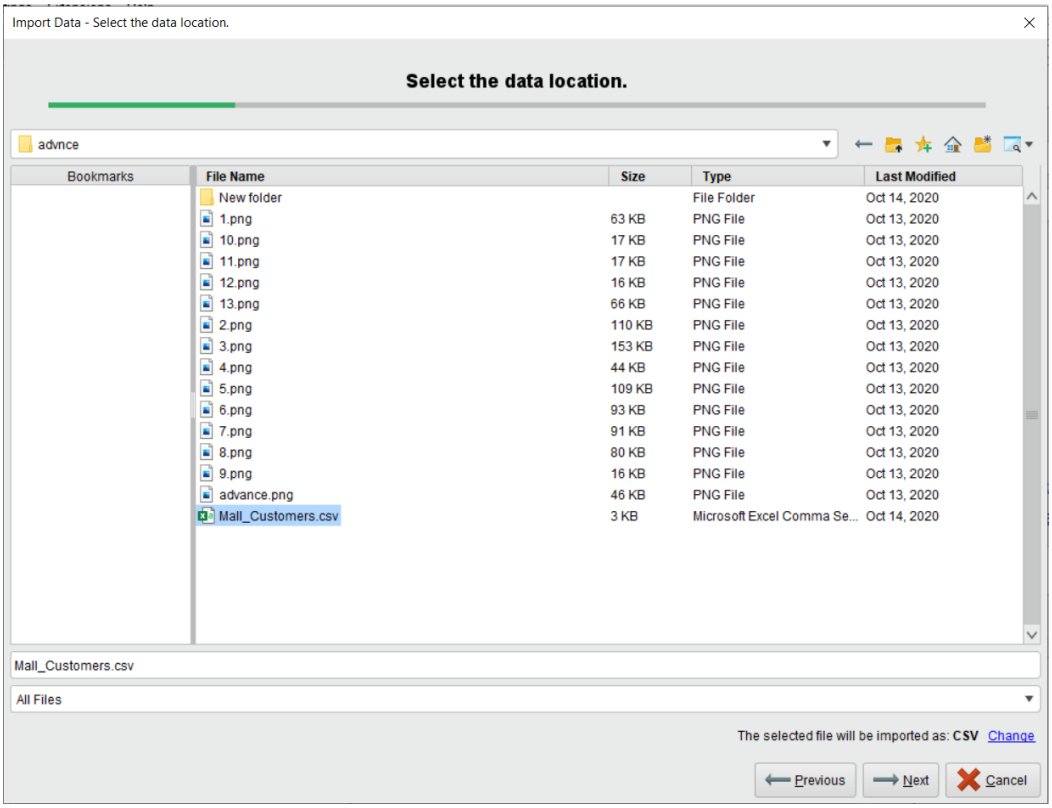
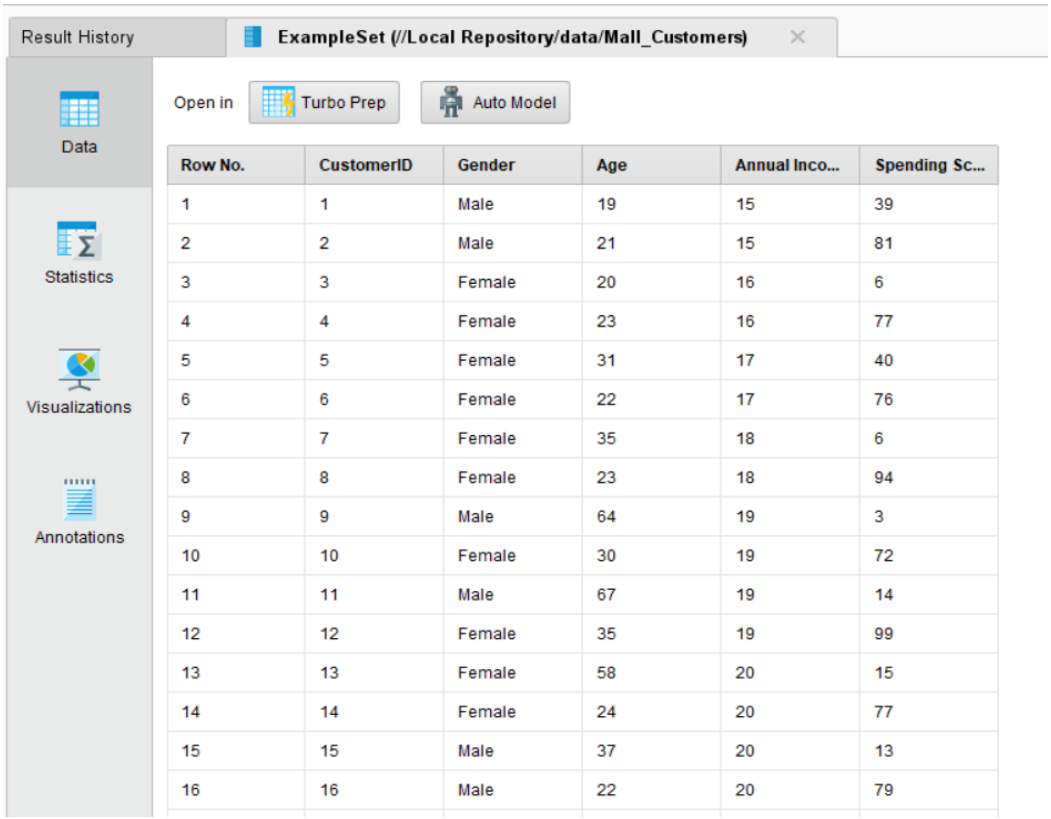


Fig. 10.3

Here, I select Mall_Customers.csv

3. After selecting the dataset you can see the dataset on the dashboard.



Row No.	CustomerID	Gender	Age	Annual Inco...	Spending Sc...
1	1	Male	19	15	39
2	2	Male	21	15	81
3	3	Female	20	16	6
4	4	Female	23	16	77
5	5	Female	31	17	40
6	6	Female	22	17	76
7	7	Female	35	18	6
8	8	Female	23	18	94
9	9	Male	64	19	3
10	10	Female	30	19	72
11	11	Male	67	19	14
12	12	Female	35	19	99
13	13	Female	58	20	15
14	14	Female	24	20	77
15	15	Male	37	20	13
16	16	Male	22	20	79

Fig. 10.4

4. For visualize the data in graphical representation Click on the Visualization button.

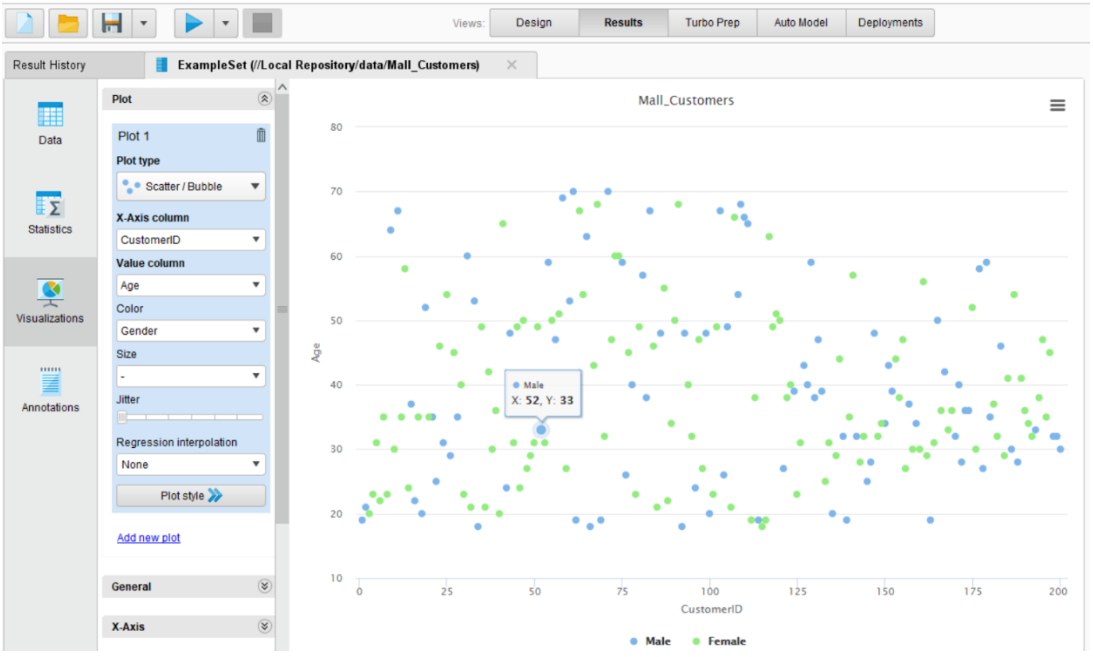


Fig. 10.5 (Scatter/bubble)

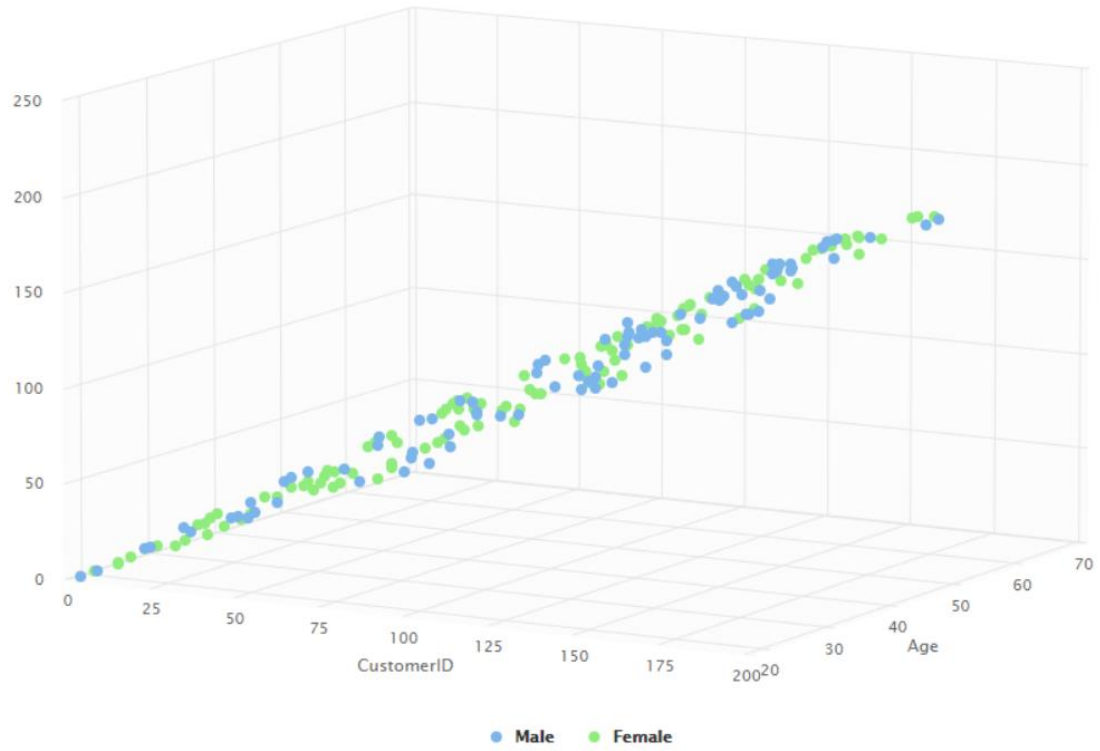


Fig. 10.6 (Scatter 3D)

There are many types of visualization formats available for data.

5. There are many options for data processing like Transform, Cleanse, Generate, Pivot & Merge.

Fig. 10.7

6. Here we choose Cleanse. The cleanse option automatically understands and cleans your dataset for you. The auto cleanses option first asks you to select the target column.

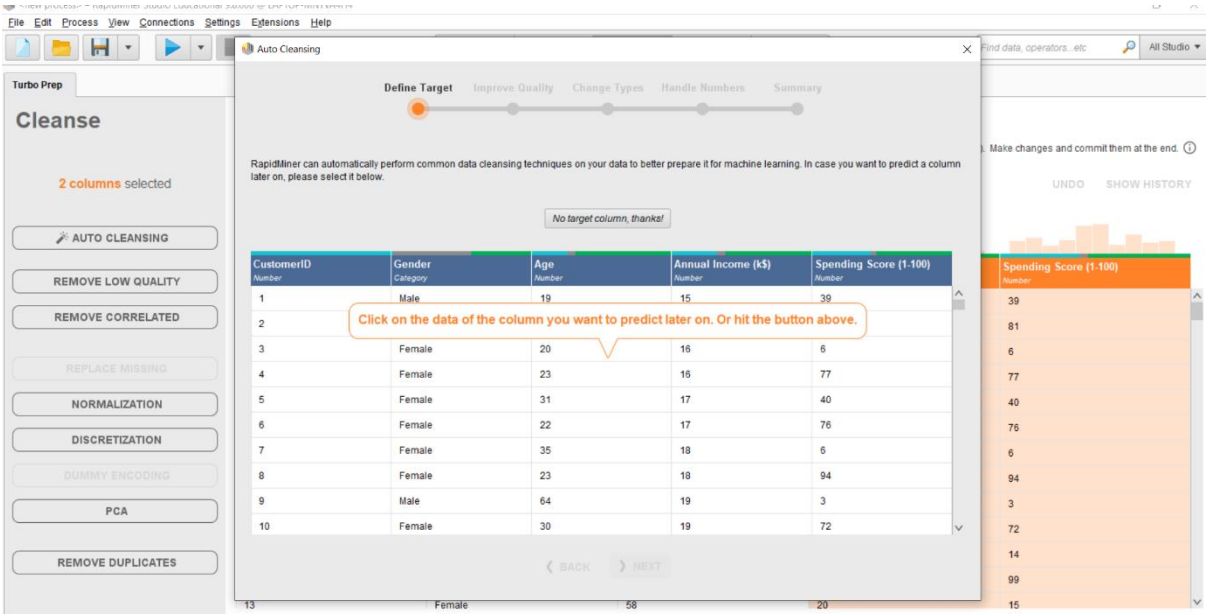


Fig. 10.8

Now you have the clean data that is ready to be used for modelling.

7. Once we have cleaned the data, we can start the modelling process. Select the option of auto-model and select the dataset that has just been processed. You can see three modeling options Predict, Clusters and Outliers. Here I choose Clusters.

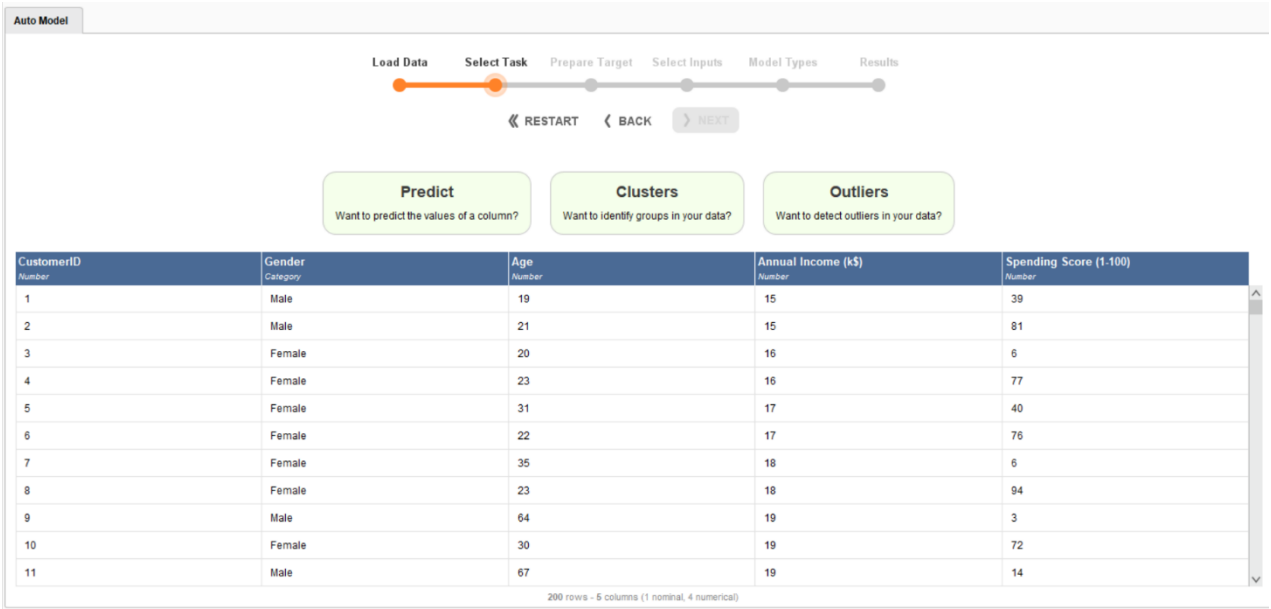


Fig. 10.9

8. Now you need to select input columns from the dataset.

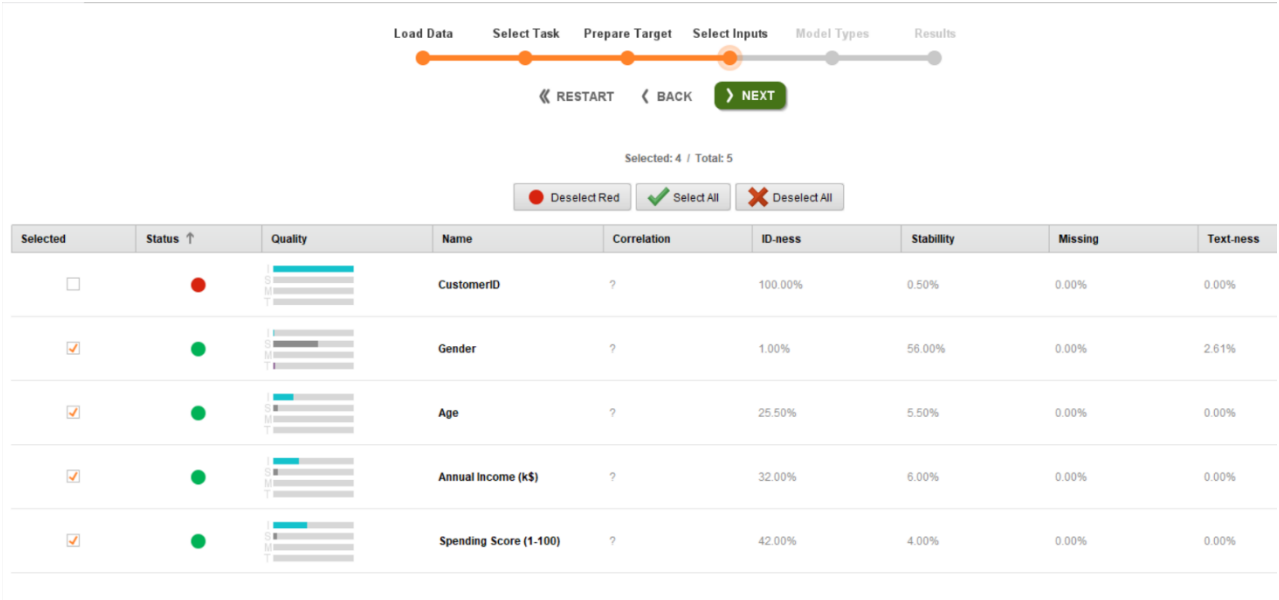


Fig. 10.10

9. Now choose the model type that you want to perform on dataset. Here I select **K-means clustering** with two clusters.

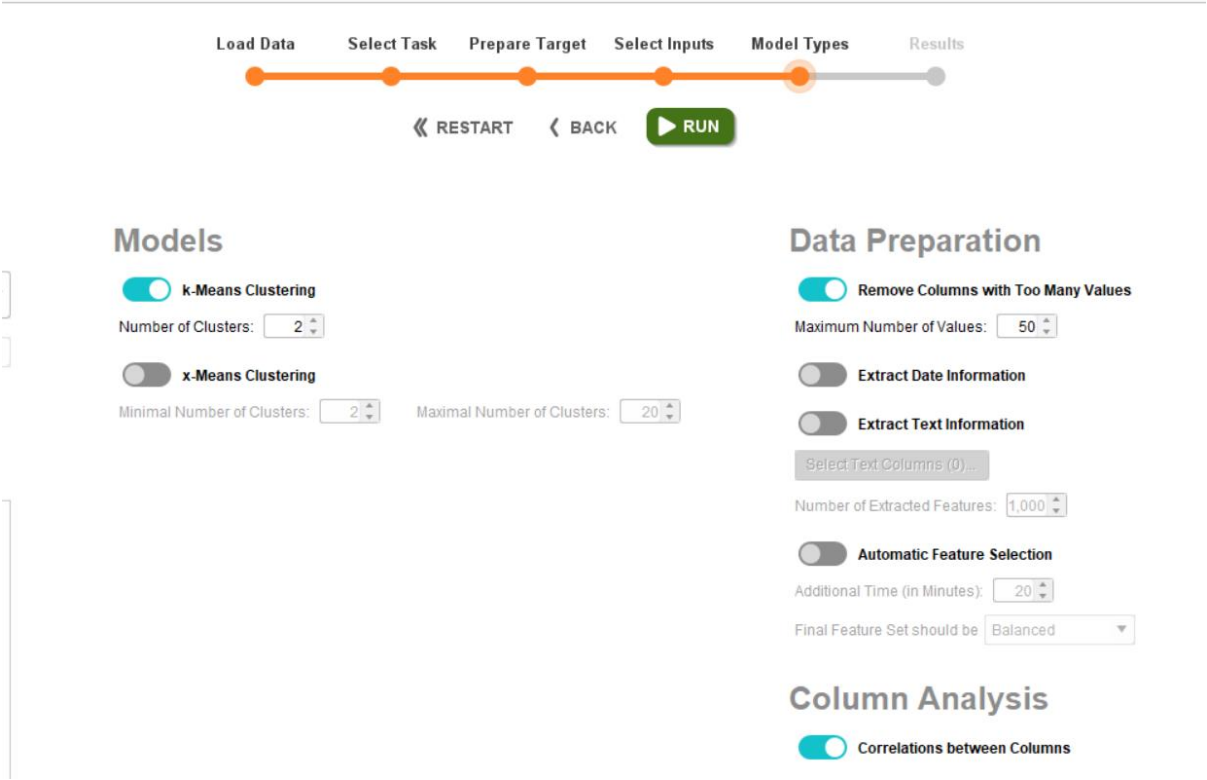


Fig. 10.11

10. Now let the K-means performed on the data. And you will see the result of **K-means clustering** model.

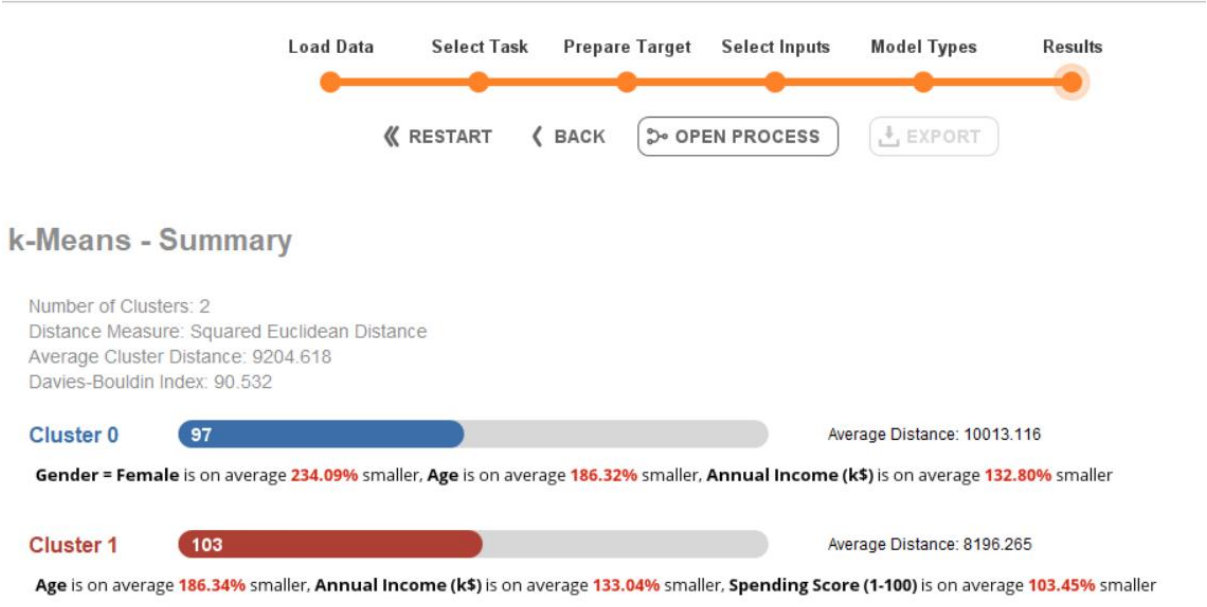


Fig. 10.12 (Summary)

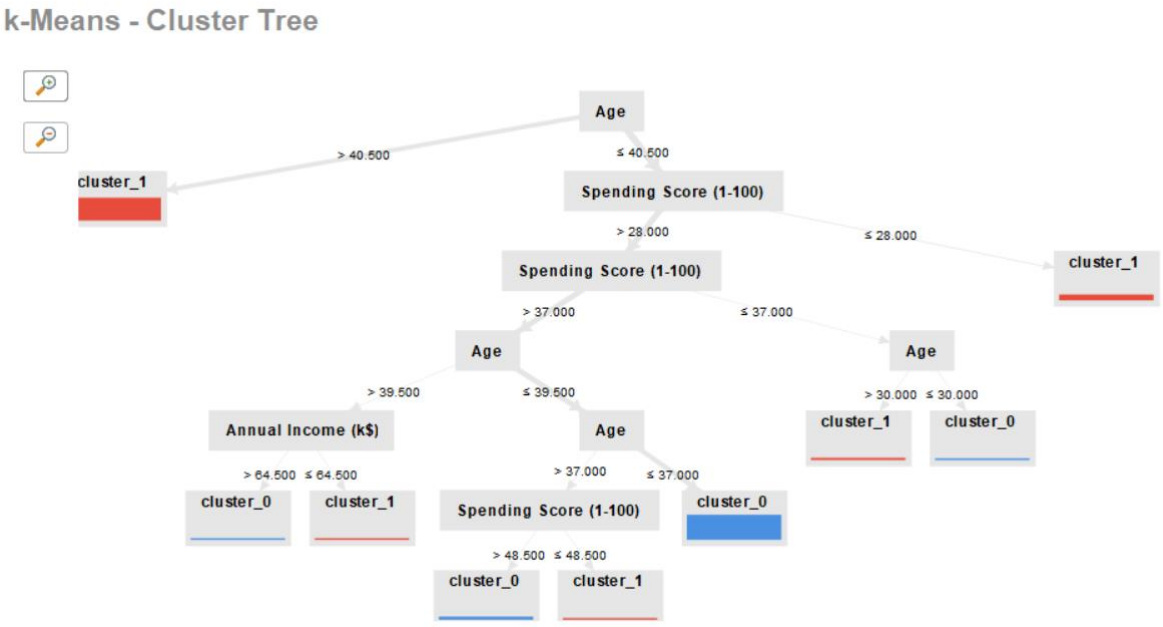


Fig. 10.13 (Cluster Tree)

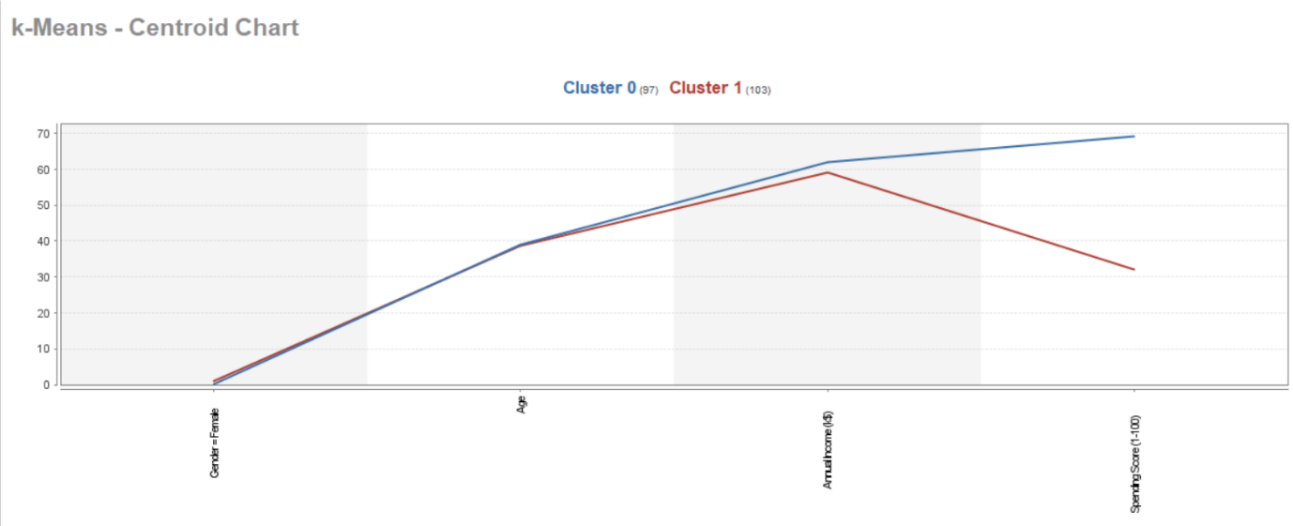


Fig. 10.14 (Centroid Chart)

k-Means - Clustered Data

Row No.	id	cluster	Gender = Female	Age	Annual Income (k$)	Spending Score (1-100)
1	1	cluster_0	0	19.000	15.000	39.000
2	2	cluster_0	0	21.000	15.000	81.000
3	3	cluster_1	1	20.000	16.000	6.000
4	4	cluster_0	1	23.000	16.000	77.000
5	5	cluster_0	1	31.000	17.000	40.000
6	6	cluster_0	1	22.000	17.000	76.000
7	7	cluster_1	1	35.000	18.000	6.000
8	8	cluster_0	1	23.000	18.000	94.000
9	9	cluster_1	0	64.000	19.000	3
10	10	cluster_0	1	30.000	19.000	72.000
11	11	cluster_1	0	67.000	19.000	14.000
12	12	cluster_0	1	35.000	19.000	99.000
13	13	cluster_1	1	58.000	20.000	15.000

Fig. 10.15 (Clustered Data)