Data Warehousing and Mining Lab (DWML)-

Case study on OLAP & ETL process tools

Data Warehousing - OLAP

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. This chapter covers the types of OLAP, operations on OLAP, differences between OLAP, and statistical databases and OLTP.

>>Types of OLAP Servers

We have four types of OLAP servers –

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)
- Specialized SQL Servers

>Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.

- ☐ ROLAP includes the following –
- ☐ Implementation of aggregation navigation logic.
- ☐ Optimization for each DBMS back end.
- ☐ Additional tools and services.

>Multidimensional OLAP

MOLAP uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

>Hybrid OLAP

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allow large data volumes of detailed information. The aggregations are stored separately in the MOLAP store.

>Specialized SQL Servers

Specialized SQL servers provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

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

Here is the list of OLAP operations –

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

>>Top OLAP Marketing Tools

You can use OLAP tools to analyze large volumes of multidimensional data from different perspectives. They make it easy to filter, analyze, and visualize key data insights. These tools are often part of a Business Intelligence Suite.

OLAP marketing tools should have the following features:

1. The ability to analyze large volumes of (big) data
2. The ability to perform analytical operations
3. A high degree of interactivity
4. Fast response times
5. Different types of data visualizations
6. The ability to analyze why things happen

Some OLAP tools used in marketing include:

>IBM Cognos

IBM Cognos is a web-based reporting and analytical tool to help you understand your organizational data. It's used to view or create detailed business reports, analyze data, and help you make effective business decisions.

>MicroStrategy

MicroStrategy is a business analytics platform that helps enterprises build and deploy analytics and mobile apps to transform their business. The MicroStrategy platform provides interactive dashboards, highly formatted reports, ad hoc queries, and automated report distribution. The software's ROLAP architecture is a key differentiator from other vendors who offer full-featured solutions.

>Palo OLAP Server

Palo is a MOLAP (Multidimensional Online Analytical Processing) server typically used as a BI tool for controlling and budgeting. It is a Jedox AG product. Palo enables multiple users to share one centralized data storage. It works with real-time data. Data can then be consolidated or written back with the help of multidimensional queries. Palo stores run-time data in its memory to give faster data access to users.

>Sisense

Sisense is an agile business intelligence (BI) solution that provides advanced tools to manage big data in marketing analytics. It helps you simplify complex data and transform it into powerful analytic apps to give you a more comprehensive understanding of your data.

>icCube

icCube owns a business intelligence software that offers an end-to-end BI solution. This is great for software companies looking to embed data analytics, visualization, and reporting into their product. icCube sells an online analytical processing server that is implemented in Java as per J2EE standards. It's an in-memory OLAP server and is compatible with any data source that holds its data in tabular form.

>SAP NetWeaver Business Warehouse

SAP NetWeaver Business Warehouse provides a high-performance infrastructure that helps you evaluate and interpret data. It provides reporting, analysis, and interpretation of business data quickly and in line with market needs.

>Oracle Business Intelligence Enterprise Edition (OBIEE)

Oracle Business Intelligence Enterprise Edition helps customers discover new data insights and make faster business decisions by offering interactive dashboards, powerful operational reporting, and real-time alerts. It reduces the total cost of ownership and increases return on investment for the entire organization.

>Apache Kylin

Apache Kylin is an open-source, distributed Analytical Data Warehouse for Big Data. It provides an SQL interface and MOLAP combined with Hadoop and Spark to support large data. In addition, Kylin reduces query processing time and quickly filters billions of data rows.

>Final Thoughts

Businesses continuously need to plan, analyze, and report on sales and marketing activities to maximize efficiency. OLAP applications can help increase the productivity of business managers, developers, marketing analysts, and whole organizations. In addition, they can also help you transform data into actionable insights.

What is ETL

ETL stands for **Extract Transform and Load**. ETL combines all the three database functions into one tool to fetch data from one database and place it into another database.

>Extract: Extract is the process of fetching (reading) the information from the database. At this stage, data is collected from multiple or different types of sources.

>Transform: Transform is the process of converting the extracted data from its previous form into the required form. Data can be placed into another database. Transformation can occur by using rules or lookup tables or by combining the data with other data.

>Load: Load is the process of writing the data into the target database.

>>ETL is used to integrate the data with the help of three steps **Extract, Transform, and Load**, and it is used to blend the data from multiple sources. It is often used to build a **data warehouse**.In the ETL process, data is extracted from the source system and converted into a format that can be examined and stored into a **data warehouse** or any other system. ETL is an alternative but a related approach which is designed to push processing down to the database to improve the performance.

>>Types of ETL Tools

ETL tools can be categorized into the following main types:

>Batch ETL Tools

In this type of ETL tool, batch processing is used to acquire data from the source systems. The data is extracted, transformed, and loaded into the repository in batches of ETL jobs. It's a cost-effective method because it uses limited resources in a time-bound way.

>Real-Time ETL Tools

Data is extracted, cleansed, enriched, and loaded to the target system in real-time ETL tools. These tools offer you faster access to information and improve time to insights.

As the need to gather and analyze the data in the shortest possible time has augmented, these ETL tools are becoming more popular among businesses.

>On-Premise ETL Tools

Many companies operate legacy systems that have both the data and the repository configured on-premise. The main reason behind such an implementation is data security. That's why companies prefer having an ETL tool deployed on-site.

>Cloud ETL Tools

As the name suggests, these tools are deployed on the cloud as various cloud-based applications form an essential part of enterprise architecture. Companies opt for cloud ETL tools to manage data transfer from these applications. Cloud-based ETL tools let businesses leverage flexibility and agility in the ETL process.

>Oracle

Oracle is the industry-leading database. It offers a wide range of choice of Data Warehouse solutions for both on-premises and in the cloud. It helps to optimize customer experiences by increasing operational efficiency.

STEPS:

1. Installing WEKA

   Open Terminal

   > sudo apt update

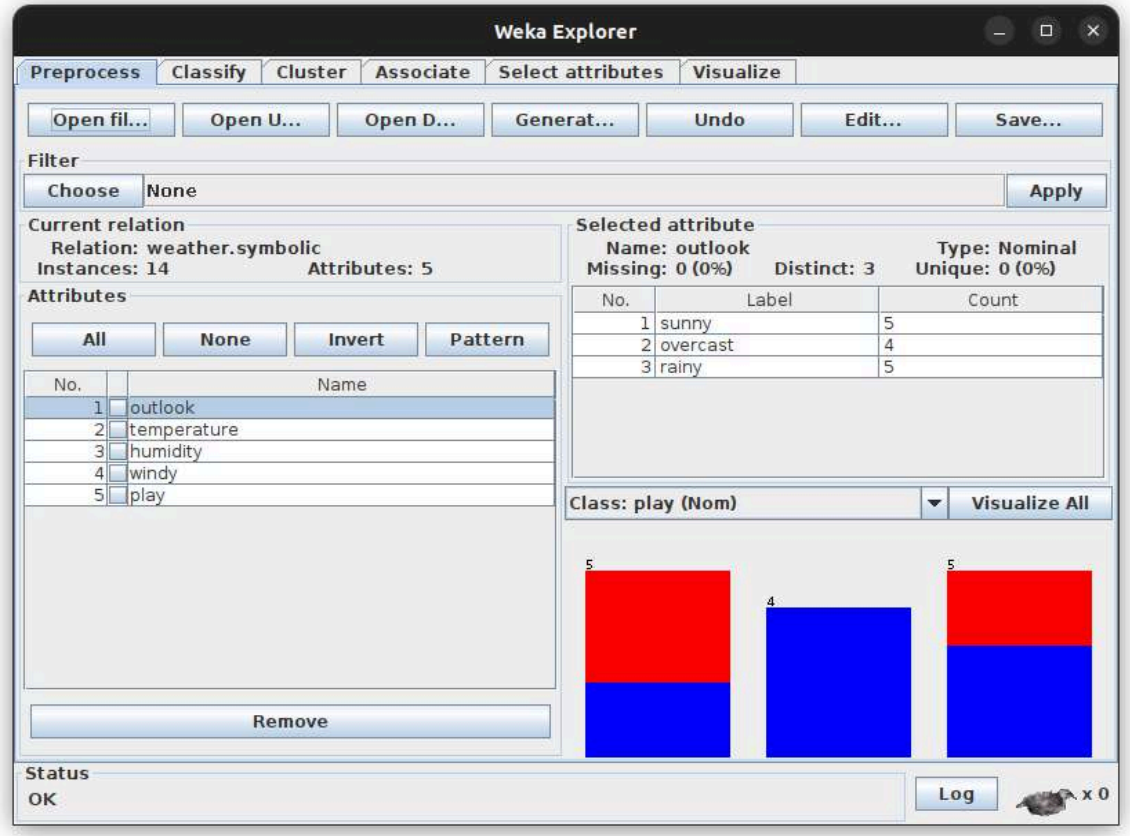
   > sudo apt -y install weka

```
computer@computer-ThinkCentre: ~
(base) computer@computer-ThinkCentre:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 https://ppa.launchpadcontent.net/gns3/ppa/ubuntu jammy InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:6 http://packages.microsoft.com/repos/code stable InRelease
Hit:7 https://deb.nodesource.com/node_18.x jammy InRelease
Hit:8 https://dl.google.com/linux/chrome/deb stable InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
250 packages can be upgraded. Run 'apt list --upgradable' to see them.
(base) computer@computer-ThinkCentre:~$ sudo apt -y install weka
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
weka is already the newest version (3.6.14-3).
0 upgraded, 0 newly installed, 0 to remove and 250 not upgraded.
(base) computer@computer-ThinkCentre:~$ weka
[warning] /usr/bin/weka: Unable to locate libsvm.jar in /usr/share/java
---Registering Weka Editors---
Trying to add database driver (JDBC): jdbc.idbDriver - Error, not in CLASSPATH?
```

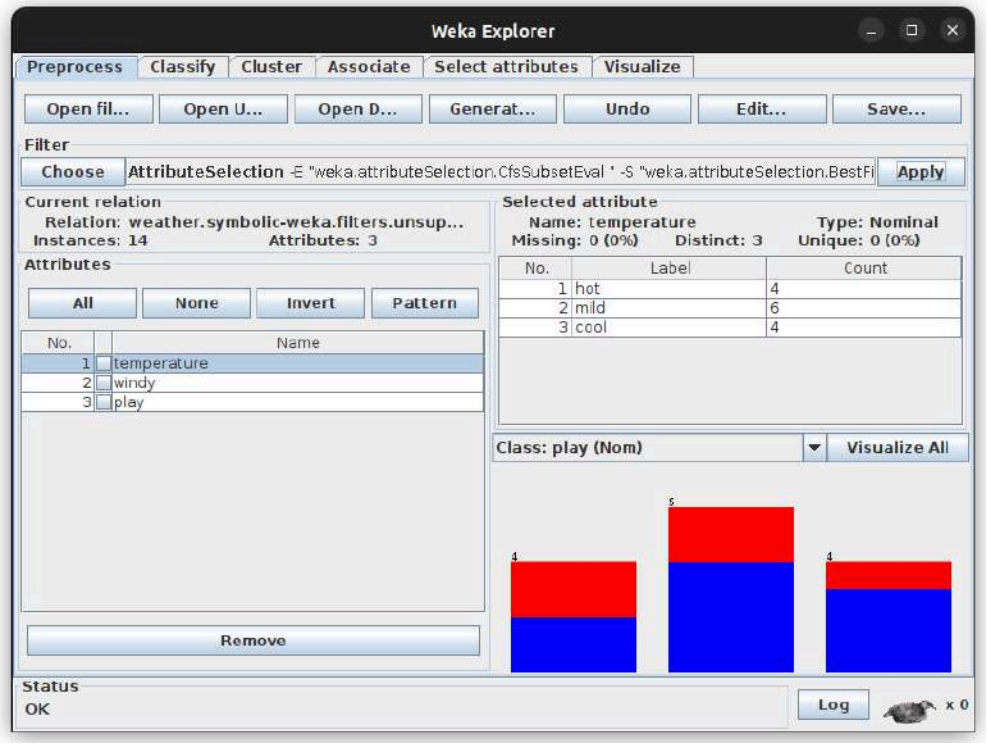
2. Open WEKA.

   > weka

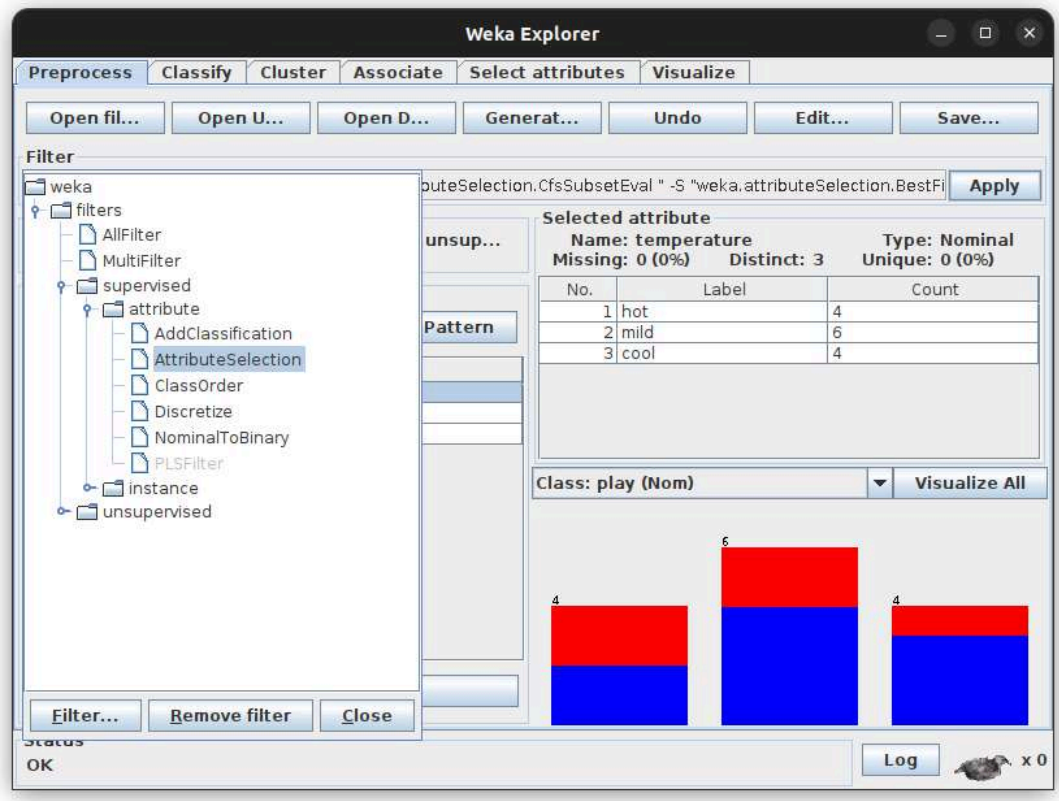
3. Open file: /->usr->share>doc-> weka->examples->weather.nominal.arff



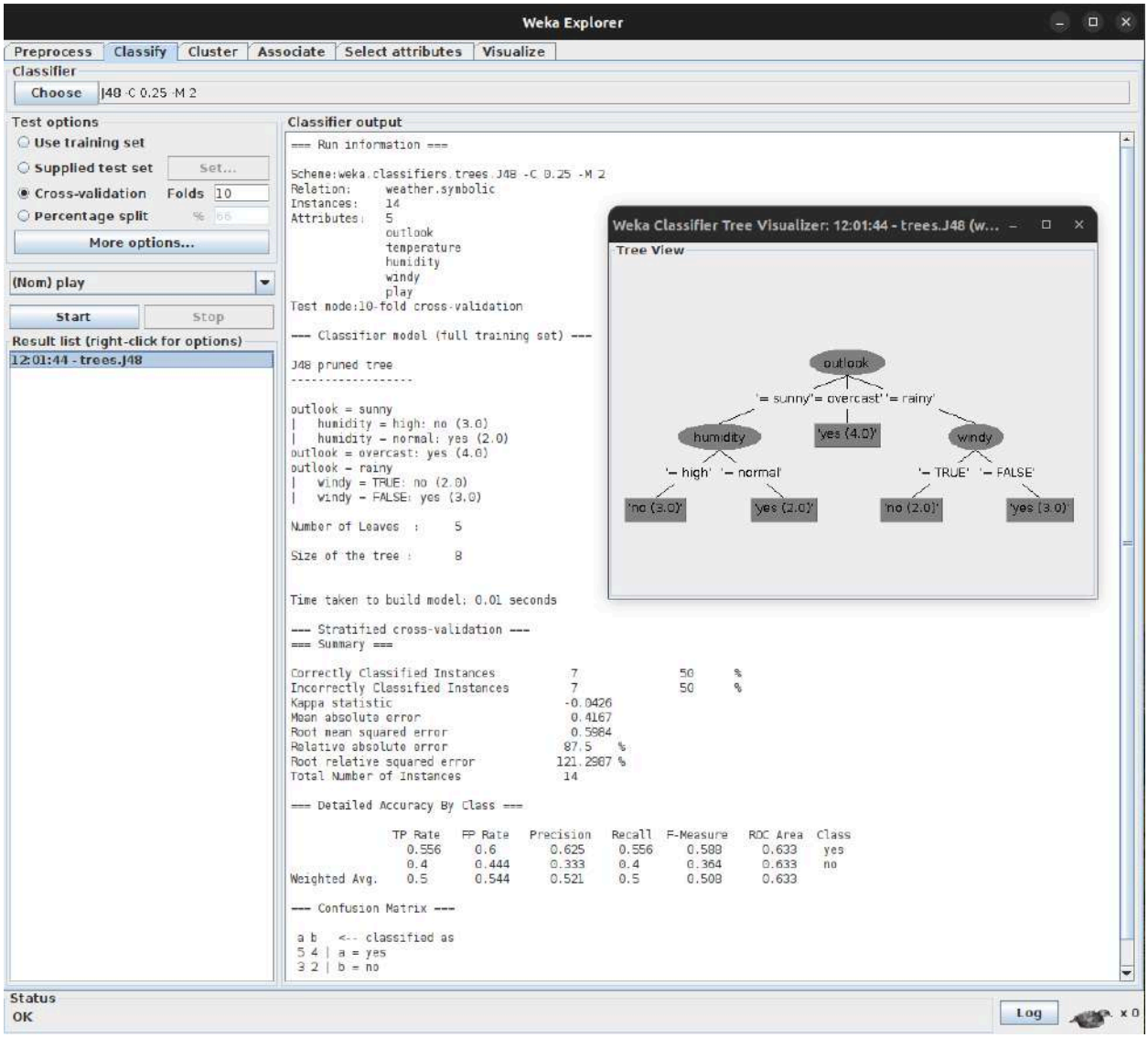
4. Removing attributes: To remove Attribute/s select them and click on the Remove button at the bottom.



5. Applying FIlters: Click on the Choose button in the Filter subwindow and select the following filter >weka>filters>supervised>attribute>AttributeSelection

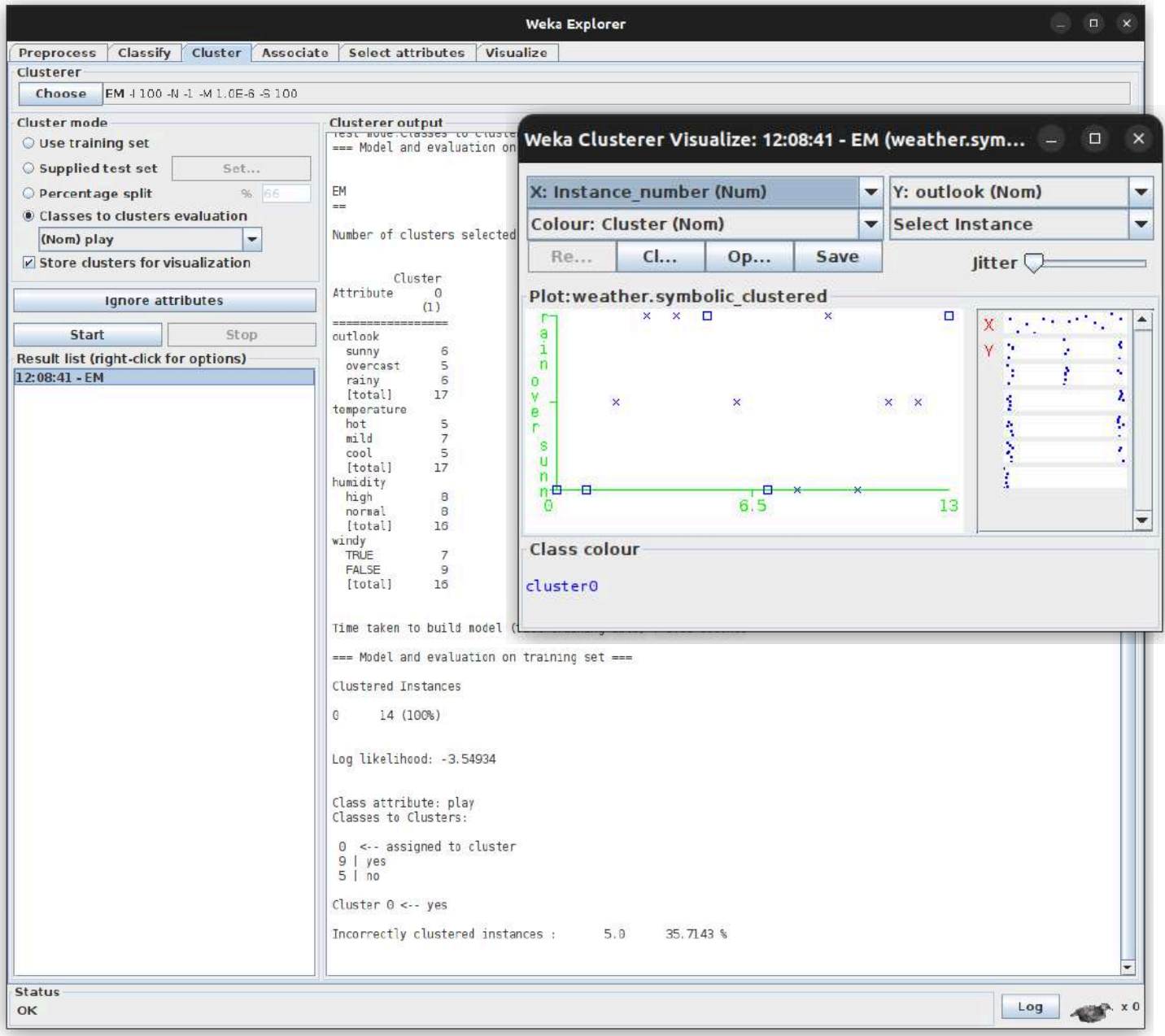


6. Selecting Classifier: Click on the Choose button and select the following classifier −weka→classifiers>trees>J48



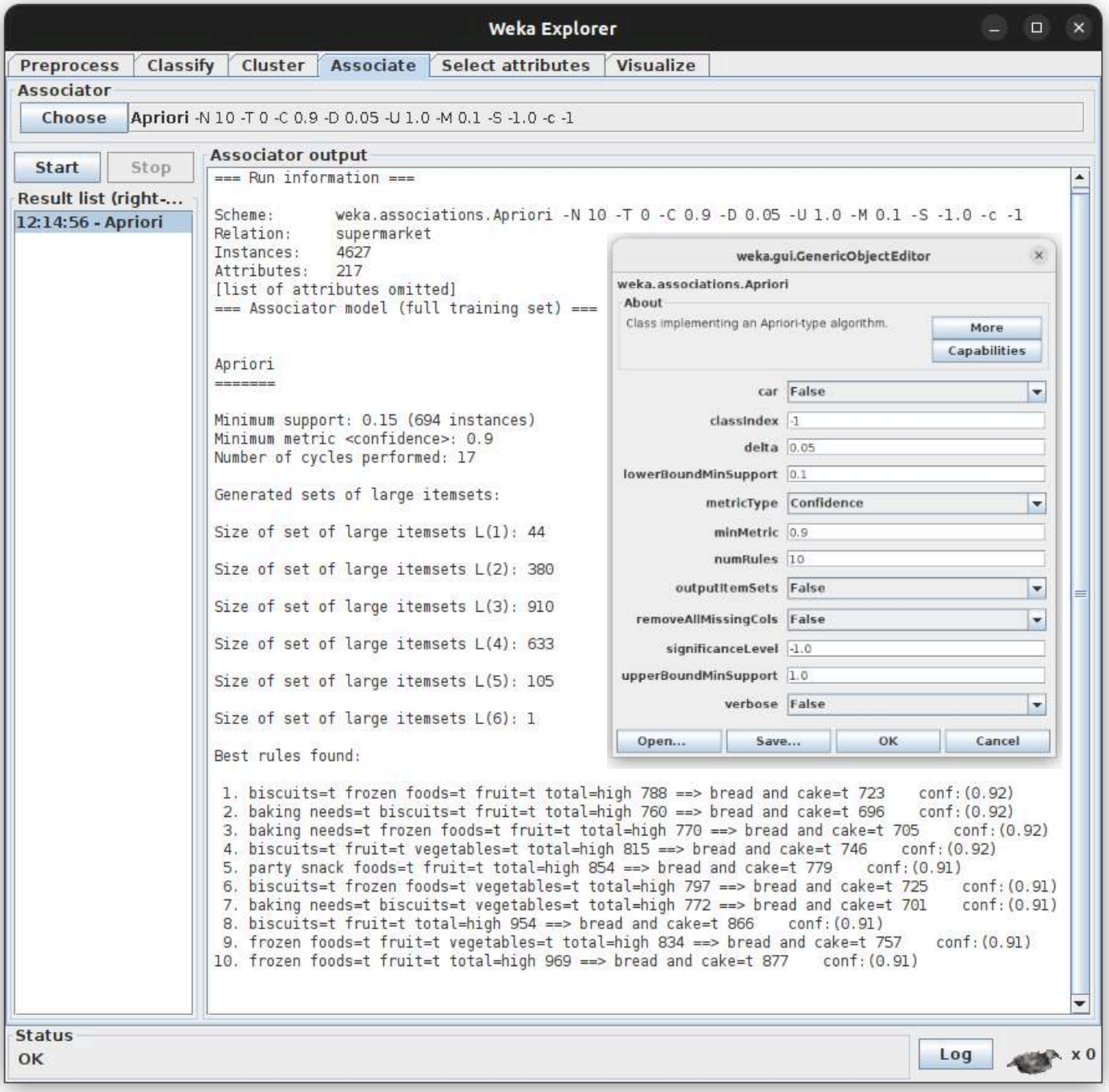
STEPS:

1. Open file - weather.arff
2. CLUSTER tab
3. Choose
4. Select EM
5. Start



STEPS:

1. Open supermarket.arff
2. Open Associate Tab
3. Choose
4. Select Apriori association
5. Start



CODE:

```
data = [
      ['T100',['I1','I2','I5']],
      ['T200',['I2','I4']],
      ['T300',['I2','I3']],
      ['T400',['I1','I2','I4']],
      ['T500',['I1','I3']],
      ['T600',['I2','I3']],
      ['T700',['I1','I3']],
      ['T800',['I1','I2','I3','I5']],
      ['T900',['I1','I2','I3']]
      ]
init = []
for i in data:
  for q in i[1]:
      if(q not in init):
          init.append(q)
init = sorted(init)
print(init)
sp = 0.4
s = int(sp*len(init))

from collections import Counter
c = Counter()
for i in init:
  for d in data:
      if(i in d[1]):
          c[i]+=1
print("C1:")
for i in c:
  print(str([i])+": "+str(c[i]))
print()
l = Counter()
for i in c:
  if(c[i] >= s):
      l[frozenset([i])]+=c[i]
print("L1:")
for i in l:
  print(str(list(i))+": "+str(l[i]))
print()
pl = l
pos = 1
for count in range (2,1000):
  nc = set()
```

```
    temp = list(l)
    for i in range(0,len(temp)):
        for j in range(i+1,len(temp)):
            t = temp[i].union(temp[j])
            if(len(t) == count):
                nc.add(temp[i].union(temp[j]))
    nc = list(nc)
    c = Counter()
    for i in nc:
        c[i] = 0
        for q in data:
            temp = set(q[1])
            if(i.issubset(temp)):
                c[i]+=1
    print("C"+str(count)+":")
    for i in c:
        print(str(list(i))+": "+str(c[i]))
    print()
    l = Counter()
    for i in c:
        if(c[i] >= s):
            l[i]+=c[i]
    print("L"+str(count)+":")
    for i in l:
        print(str(list(i))+": "+str(l[i]))
    print()
    if(len(l) == 0):
        break
    pl = l
    pos = count
print("Result: ")
print("L"+str(pos)+":")
for i in pl:
    print(str(list(i))+": "+str(pl[i]))
print()
from itertools import combinations
for l in pl:
    c = [frozenset(q) for q in combinations(l,len(l)-1)]
    mmax = 0
    for a in c:
        b = l-a
        ab = l
        sab = 0
        sa = 0
```

```
    sb = 0
    for q in data:
        temp = set(q[1])
        if(a.issubset(temp)):
            sa+=1
        if(b.issubset(temp)):
            sb+=1
        if(ab.issubset(temp)):
            sab+=1
    temp = sab/sa*100
    if(temp > mmax):
        mmax = temp
    temp = sab/sb*100
    if(temp > mmax):
        mmax = temp
    print(str(list(a))+" -> "+str(list(b))+" = "+str(sab/sa*100)+"%")
    print(str(list(b))+" -> "+str(list(a))+" = "+str(sab/sb*100)+"%")
curr = 1
print("choosing:", end=' ')
for a in c:
    b = l-a
    ab = l
    sab = 0
    sa = 0
    sb = 0
    for q in data:
        temp = set(q[1])
        if(a.issubset(temp)):
            sa+=1
        if(b.issubset(temp)):
            sb+=1
        if(ab.issubset(temp)):
            sab+=1
    temp = sab/sa*100
    if(temp == mmax):
        print(curr, end = ' ')
    curr += 1
    temp = sab/sb*100
    if(temp == mmax):
        print(curr, end = ' ')
    curr += 1
print()
print()
```

OUTPUT:

```
(base) computer@computer:~/Desktop$ python apriori.py
['I1', 'I2', 'I3', 'I4', 'I5']
C1:
['I1']: 6
['I2']: 7
['I3']: 6
['I4']: 2
['I5']: 2

L1:
['I1']: 6
['I2']: 7
['I3']: 6
['I4']: 2
['I5']: 2

C2:
['I2', 'I4']: 2
['I2', 'I3']: 4
['I3', 'I1']: 4
['I5', 'I4']: 0
['I4', 'I1']: 1
['I5', 'I1']: 2
['I3', 'I4']: 0
['I2', 'I5']: 2
['I5', 'I3']: 1
['I2', 'I1']: 4

L2:
['I2', 'I4']: 2
['I2', 'I3']: 4
['I3', 'I1']: 4
['I5', 'I1']: 2
['I2', 'I5']: 2
['I2', 'I1']: 4

C3:
['I3', 'I2', 'I4']: 0
['I2', 'I5', 'I1']: 2
['I2', 'I4', 'I1']: 1
['I5', 'I3', 'I1']: 1
['I2', 'I5', 'I4']: 0
['I2', 'I3', 'I1']: 2
```

['I2', 'I5', 'I3']: 1

L3:
['I2', 'I5', 'I1']: 2
['I2', 'I3', 'I1']: 2

C4:
['I5', 'I1', 'I2', 'I3']: 1

L4:

Result:
L3:
['I2', 'I5', 'I1']: 2
['I2', 'I3', 'I1']: 2

['I2', 'I5'] -> ['I1'] = 100.0%
['I1'] -> ['I2', 'I5'] = 33.33333333333333%
['I2', 'I1'] -> ['I5'] = 50.0%
['I5'] -> ['I2', 'I1'] = 100.0%
['I5', 'I1'] -> ['I2'] = 100.0%
['I2'] -> ['I5', 'I1'] = 28.57142857142857%
choosing: 1 4 5

['I2', 'I3'] -> ['I1'] = 50.0%
['I1'] -> ['I2', 'I3'] = 33.33333333333333%
['I2', 'I1'] -> ['I3'] = 50.0%
['I3'] -> ['I2', 'I1'] = 33.33333333333333%
['I3', 'I1'] -> ['I2'] = 50.0%
['I2'] -> ['I3', 'I1'] = 28.57142857142857%
choosing: 1 3 5

CODE:

```
dataset = [
          [0,0,1,0,0],
          [0,0,1,1,0],
          [1,0,1,0,1],
          [2,1,1,0,1],
          [2,2,0,0,1],
          [2,2,0,1,0],
          [1,2,0,1,1],
          [0,1,1,0,0],
          [0,2,0,0,1],
          [2,1,0,0,1],
          [0,1,0,1,1],
          [1,1,1,1,1],
          [1,0,0,0,1],
          [2,1,1,1,0]
          ]
mp = dict()
for i in range(len(dataset)):
   row = dataset[i]
   y = row[-1]
   if (y not in mp):
       mp[y] = list()
   mp[y].append(row)

for label in mp:
   print(label)
   for row in mp[label]:
       print(row)

test = [2,1,0,1]

probYes = 1

count = 0
total = 0
for row in dataset:
   if(row[-1] == 1):
       count+=1
   total+=1
```

```
print("Total yes: "+str(count)+" / "+str(total))
probYes *= count/total
for i in range(len(test)):
    count = 0
    total = 0
    for row in mp[1]:
        if(test[i] == row[i]):
            count += 1
        total += 1
    print('for feature '+str(i+1))
    print(str(count)+" / "+str(total))
    probYes *= count/total

probNo = 1
count = 0
total = 0
for row in dataset:
    if(row[-1] == 0):
        count+=1
    total+=1
probNo *= count/total
print("Total no: "+str(count)+" / "+str(total))
for i in range(len(test)):
    count = 0
    total = 0
    for row in mp[0]:
        if(test[i] == row[i]):
            count += 1
        total += 1
    print('for feature '+str(i+1))
    print(str(count)+" / "+str(total))
    probNo *= count/total

print(probYes)
print(probNo)

prob = probYes/(probYes+probNo)
print("Probability of playing golf: "+str(prob*100)+"%")
```

OUTPUT:

```
(base) computer@computer-ThinkCentre:~$ python NaivesBayes.py
0
[0, 0, 1, 0, 0]
[0, 0, 1, 1, 0]
[2, 2, 0, 1, 0]
[0, 1, 1, 0, 0]
[2, 1, 1, 1, 0]
1
[1, 0, 1, 0, 1]
[2, 1, 1, 0, 1]
[2, 2, 0, 0, 1]
[1, 2, 0, 1, 1]
[0, 2, 0, 0, 1]
[2, 1, 0, 0, 1]
[0, 1, 0, 1, 1]
[1, 1, 1, 1, 1]
[1, 0, 0, 0, 1]
Total yes: 9 / 14
for feature 1
3 / 9
for feature 2
4 / 9
for feature 3
6 / 9
for feature 4
3 / 9
Total no: 5 / 14
for feature 1
2 / 5
for feature 2
2 / 5
for feature 3
1 / 5
for feature 4
3 / 5
0.021164021164021163
0.006857142857142859
 Probability of playing golf: 75.5287009063444%
```

CODE:

```
data = [
[5,2],[2,4],[9,5],[4,6],[5,2],[1,5],[6,7],[4,2],[6,4],[9,2],[4,5],[1,6],[4,7],[3,6],[1,1],[8,4],[8,7],
          [7,2],[2,2],[2,1],[1,2],[1,4],[2,6],[7,7],[7,4],[3,4],[1,4]
        ]
x = [i[0] for i in data]
y = [i[1] for i in data]
import math
def dist(center, point):
  d = 0.0
  for i in range(0,len(point)):
    d += (center[i]-point[i])**2
  return math.sqrt(d)

def assignCenters(centers, dataset):
  clusters = []
  for i in range(len(dataset)):
    distances = []
    for center in centers:
      distances.append(dist(center, dataset[i]))
      temp = [z for z, val in enumerate(distances) if val==min(distances)]
    clusters.append(temp[0])
  return clusters

def mean_center(k, dataset, clusters):
  nCenters = []
  for i in range(k):
    x = 0.0
    y = 0.0
    count = 0
    for j in range(len(clusters)):
      if(i == clusters[j]):
        x += dataset[j][0]
        y += dataset[j][1]
        count += 1
    x = x/count
    y = y/count
    nCenters.append([x,y])
  return nCenters
print("enter k")
```

```
k = int(input())
centers = []
for i in range(k):
  print("enter center "+str(i))
  temp = [int(x) for x in input().split()]
  centers.append(temp)
  print("Initial centers: ")
  print(centers)
  print("Initial clusters: ")
  clusters = assignCenters(centers, data)
for i in range(k):
  print("cluster "+str(i))
  for j in range(len(clusters)):
      if(i == clusters[j]):
          print(data[j],end=' ')
print()
print()
for itr in range(10):
  print("Iteration "+str(itr))
  centers = mean_center(k,data,clusters)
  print("Updated centers: ")
  print(centers)
  clusters = assignCenters(centers, data)
  print("Updated clusters: ")
for i in range(k):
  print("cluster "+str(i))
  for j in range(len(clusters)):
      if(i == clusters[j]):
          print(data[j],end=' ')
print()
print()
```

OUTPUT:

```
computer@computer-ThinkCentre:~/Documents/CSE-AIML/TE/AIML11$ python
-u "/home/computer/Documents/CSE-AIML/TE/AIML11/kMeans.py"
enter k
2
enter center 0
6 4
Initial centers:
[[6, 4]]
Initial clusters:
enter center 1
9 2
Initial centers:
[[6, 4], [9, 2]]
Initial clusters:
cluster 0
[5, 2] [2, 4] [4, 6] [5, 2] [1, 5] [6, 7] [4, 2] [6, 4] [4, 5] [1, 6]
[4, 7] [3, 6] [1, 1] [8, 4] [8, 7] [2, 2] [2, 1] [1, 2] [1, 4] [2, 6]
[7, 7] [7, 4] [3, 4] [1, 4] cluster 1
[9, 5] [9, 2] [7, 2]

Iteration 0
Updated centers:
[[3.6666666666666665, 4.25], [8.333333333333334, 3.0]]
Updated clusters:
Iteration 1
Updated centers:
[[2.9, 4.0], [7.857142857142857, 4.4285714285714295]]
Updated clusters:
Iteration 2
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 3
Updated centers:
```

```
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 4
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 5
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 6
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 7
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 8
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
Iteration 9
Updated centers:
[[2.5555555555555554, 3.8333333333333335], [7.444444444444445,
4.666666666666667]]
Updated clusters:
cluster 0
[5, 2] [2, 4] [4, 6] [5, 2] [1, 5] [4, 2] [4, 5] [1, 6] [4, 7] [3, 6]
[1, 1] [2, 2] [2, 1] [1, 2] [1, 4] [2, 6] [3, 4] [1, 4] cluster 1
[9, 5] [6, 7] [6, 4] [9, 2] [8, 4] [8, 7] [7, 2] [7, 7] [7, 4]
```

CODE:

```
import networkx as nx
import numpy as np
from numpy import array
import matplotlib.pyplot as plt
with open('hits.txt') as f:
    lines = f.readlines()

G = nx.DiGraph()

for line in lines:
    t = tuple(line.strip().split(','))
    G.add_edge(*t)

h, a = nx.hits(G, max_iter=100)
h = dict(sorted(h.items(), key=lambda x: x[0]))
a = dict(sorted(a.items(), key=lambda x: x[0]))

print(np.round(list(a.values()), 3))
print(np.round(list(h.values()), 3))

pr = nx.pagerank(G)
pr = dict(sorted(pr.items(), key=lambda x: x[0]))
print(np.round(list(pr.values()), 3))


sim = nx.simrank_similarity(G)
lol = [[sim[u][v] for v in sorted(sim[u])] for u in sorted(sim)]
sim_array = np.round(array(lol), 3)
print(sim_array)

nx.draw(G, with_labels=True, node_size=2000, edge_color='#eb4034', width=3,
font_size=16, font_weight=500, arrowsize=20, alpha=0.8)
plt.savefig("graph.png")
```

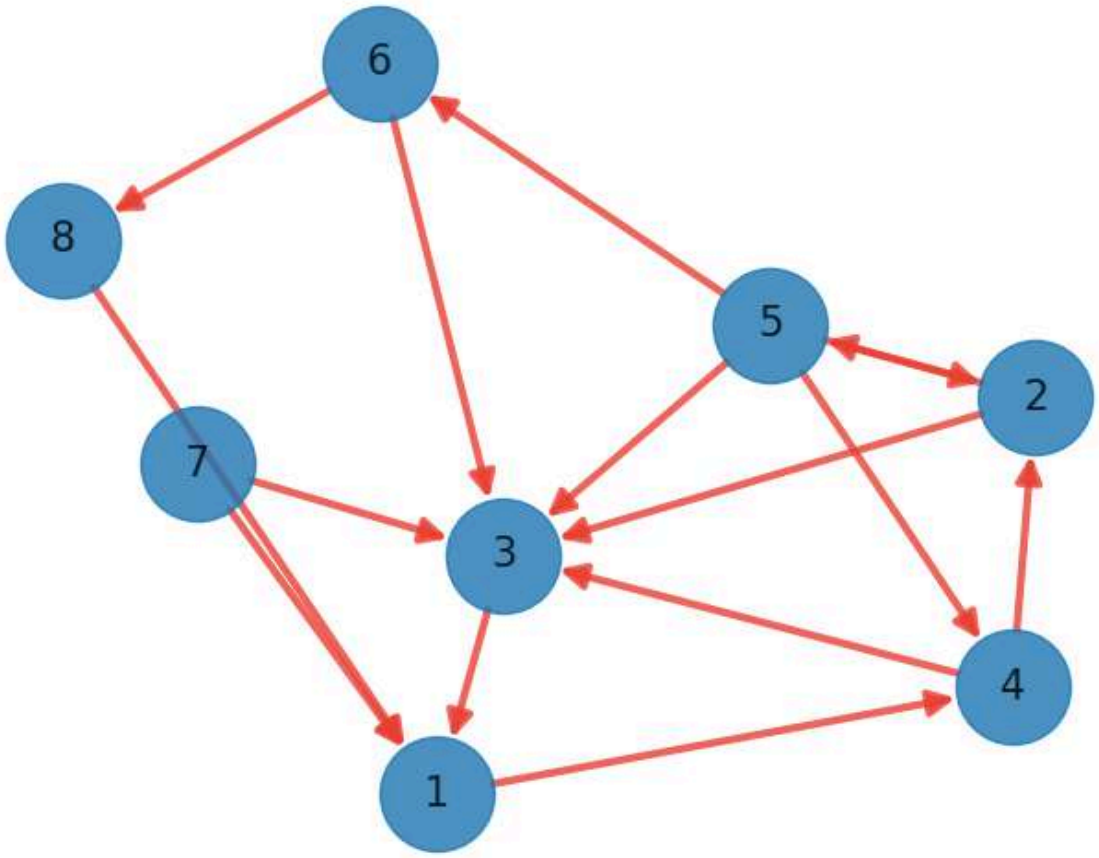
hits.txt

```
1,4
2,3
2,5
3,1
4,2
4,3
5,3
5,2
5,4
5,6
6,3
6,8
7,1
7,3
8,1
```

OUTPUT:

```
computer@computer-ThinkCentre:~/Documents/CSE-AIML/TE/AIML11$ python -u
"/home/computer/Documents/CSE-AIML/TE/AIML11/pageRank.py"
/home/computer/anaconda3/lib/python3.9/site-packages/networkx/algorithms/l
ink_analysis/hits_alg.py:78: FutureWarning: adjacency_matrix will return a
scipy.sparse array instead of a matrix in Networkx 3.0.
  A = nx.adjacency_matrix(G, nodelist=list(G), dtype=float)
[0.088 0.187 0.369 0.128 0.059 0.11  0.    0.059]
[0.043 0.144 0.03  0.187 0.268 0.144 0.154 0.03 ]
[0.241 0.137 0.218 0.24  0.077 0.035 0.019 0.034]
[[1.    0.207 0.221 0.193 0.217 0.269 0.    0.171]
 [0.207 1.    0.355 0.369 0.302 0.553 0.    0.369]
 [0.221 0.355 1.    0.242 0.4   0.324 0.    0.427]
 [0.193 0.369 0.242 1.    0.229 0.548 0.    0.243]
 [0.217 0.302 0.4   0.229 1.    0.271 0.    0.498]
 [0.269 0.553 0.324 0.548 0.271 1.    0.    0.244]
 [0.    0.    0.    0.    0.    0.    1.    0.   ]
 [0.171 0.369 0.427 0.243 0.498 0.244 0.    1.   ]]
```

graph.png



CODE(pageHit.py):

```
import networkx as nx
import matplotlib.pyplot as plt
G = nx.DiGraph()
G.add_edges_from([('A', 'D'), ('B', 'C'), ('B', 'E'), ('C', 'A'), ('D', 'C'), ('E', 'D'), ('E', 'B'), ('E', 'F'), ('E', 'C'), ('F', 'C'), ('F', 'H'), ('G', 'A'), ('G', 'C'), ('H', 'A')])
plt.figure(figsize =(10, 10))
nx.draw_networkx(G, with_labels = True)
hubs, authorities = nx.hits(G, max_iter = 50, normalized = True)
print('Hub  Scores: ')
for i in hubs:
    print("{}: {},".format(i, hubs[i]))
print('\nAuthority Scores: ')
for i in authorities:
    print("{}: {},".format(i, authorities[i]))
```

OUTPUT:

```
computer@computer-ThinkCentre:~/Documents/CSE-AIML/TE/AIML11$ python -u
"/home/computer/Documents/CSE-AIML/TE/AIML11/pageHit.py"
   A = nx.adjacency_matrix(G, nodelist=list(G), dtype=float)
 Hub  Scores:
 A: 0.04642540403219996,
 D: 0.13366037526115382,
 B: 0.15763599442967322,
 C: 0.03738913224642651,
 E: 0.2588144598468665,
 F: 0.15763599442967322,
 H: 0.03738913224642651,
 G: 0.17104950750758036,

 Authority Scores:
 A: 0.10864044011724336,
 D: 0.13489685434358004,
 B: 0.11437974073336449,
 C: 0.388372800387618,
 E: 0.06966521184241475,
 F: 0.11437974073336449,
 H: 0.06966521184241477,
 G: 0.0,
```

Name :- Prathamesh S. Chikankar

Roll No. :- AIML11 Branch :- CSE-(AI & ML)

Year :- TE Subject :- DW&M [Data warehousing and mining]

Topic :- Assignment No. 01

Sign :- Pratik Date :- August'22

1. Explain Data Warehouse Architecture in detail.
Architecture is the proper management / arrangement of the components.

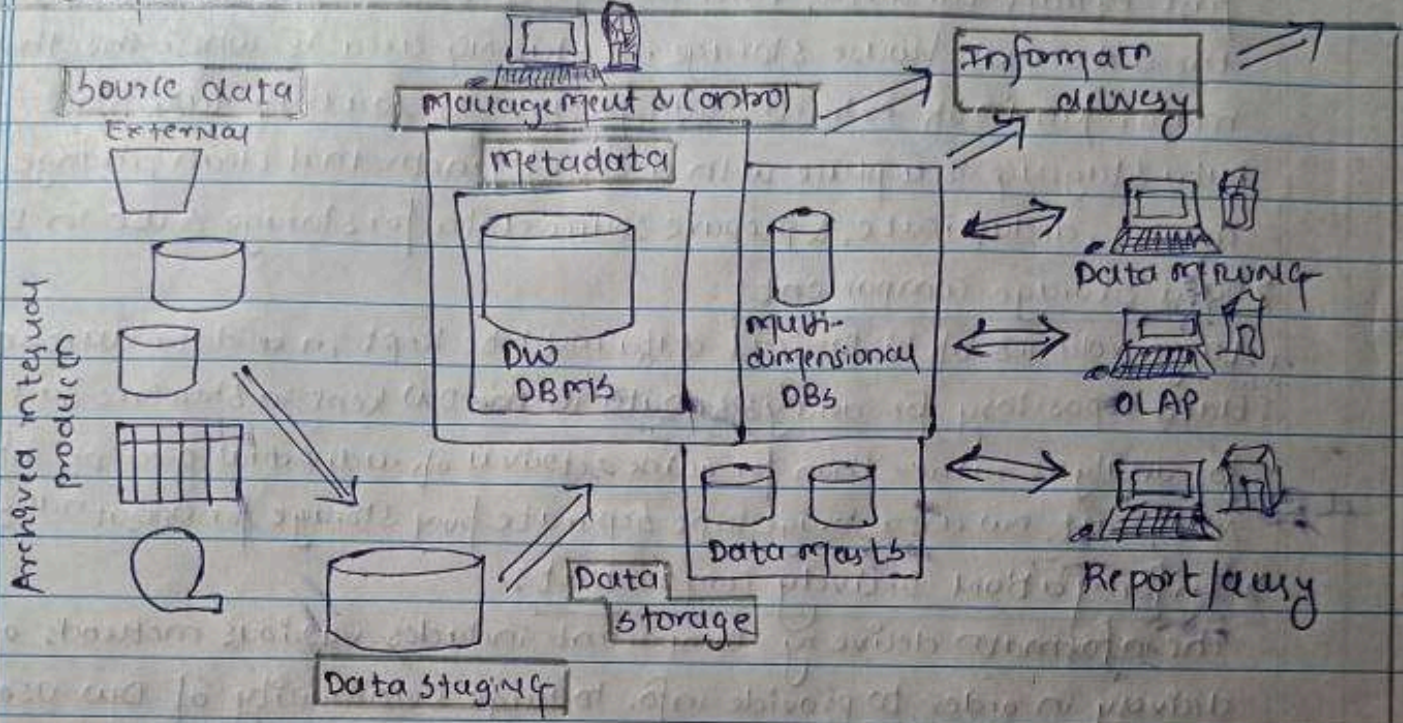


fig. Data Warehouse Architecture

> Source data component
Source data coming into the data warehouse may be grouped in four broad categories, as discussed here.
- Production data: This type of information originates from the company's numerous operational systems. (data from various operational systems)
- Internal data: Users preserve their "private" spreadsheets, documents customer profiles, & sometimes even departmental DB in every company.
- External data: most executives rely on data from outside sources for a large portion of their information they use. Rely on competitor market share data. Use standard values of financial indicators.
- Archived data: operational systems are largely designed to run a company's existing operation. Data kept in operational system DB.

PTO

> Data staging component : (ETL process)

To get the data ready, three major functn must be completed. you must first extract the data, then transform it and at last load it ful to the data wasehouse storage. A staging area is where the three major functn of extractn, transformatn & loading take place.

Data staging is a place with a set of functn that clean, change, combine convert, deduplicate, & prepare source data for storage & use in DW.

> Data Storage component:

Large volumes of historical data must be kept in a data warehouse's data repository for analysis. Data in the DW kept in structure suitable for analysis rather than for quick retrival of individual piece of informatn. As a result, DW data storage kept seperate from storage for operatnal systms.

> Information Delivery component:

The informatn delivery component includes various methods of info. delivery in order to provide info. to large community of DW users. this informatn can be delivered through online, intranet, internet or email mode.

> metadata components:

Metadata is "data that describes other data". It is similar to data dictionary or data catalog in DBMS. Keeps the informatn about logical data structure, info. about files & addresses, info about indices & so on. Types → operational metadata, Extractn and Transformatn metadata and End-user metadata. eg:- web page, document, etc

> Management and control component:

This component of DW sits on the top of all other components. The managment & control component coordinates services & activities within the DW. Controls data transformatn. Monitors info. delivery. The management & control component integrates with the metadata component to perform the management & control functn.

2. Explain Data warehouse schemas in detail.

Schema is a logical description of entire database. A database uses a relational model, while DW uses star and snowflake schema.

> Star schema:

- A star schema is the elementary form of a dimensional model, in which data are organised into facts and dimensions.
- A fact is an event that is counted or measured, such as sale or dealer.
- It is a relational schema (represents multidimensional data model)
- It is known as star schema because the entity-relationship diagram of this schemas simulates a star, with points, diverge from central table.
- Centre consists of a large fact table & points of star are dimension Table.

Fact Table:

- It is a table in a star schema which contains facts and connected to dimensions. A fact has two types of columns: those that include fact and those that are foreign key to the dimension table. primary key of the fact table is generally a composite key that is made up of all its foreign key.

Dimension Tables:

- A dimension is an architecture usually composed of one or more heirarchies that categorize data. If a dimension has not got heirarchies and levels, it is called a flat dimension or list. Small in size than fact table.

characteristics:

It creates denormalized database that can quickly provide query response
It provides a flexible design that can be changed easily.
It reduces the complexity of metadata for both developers & end users.
Keys: primary keys, surrogate keys, foreign key.

Advantages:

- Query performance & query as well, - Built in referential integrity

Disadvantages:

- Data integrity is not enforced well since in a highly denormalize schema.

PTO

>Snowflake Schema:
- Snowflake schema is variant of star schema.
- Here the centralized fact table is connected to multiple dimensions.
- In the snowflake schema, dimensions are present in a normalized form in multiple related table.
- Snowflake schema affects only the dimension tables & not fact tables.

Characteristics of snowflake schema:
- uses small disk space.
- Easy to implement dimension that is added to schema.
- Multiple tables, so performance is reduced.

Advantages:
- It provides structured data which reduces the problem of integrity.
- Uses small disk space because data are highly structured.

Disadvantages:
- Reduces space consumed by dimension tables, but compared with the entire DW the saving is usually insignificant.
- Avoid snowflaking or normalization of dimensions tables, unless required and appropriate. - complex structure
- Do not snowflake hierarchies of one-dimensional table into separate table. - query performance degrades.

---

3. Why is entity-relationship modelling technique not suitable for the datawarehouse? How is dimensional modelling different.

- Dimensional Modelling (DM) is a DS technique optimized for data storage in a DW. The purpose of dimensional modelling is to optimize the DB for faster retrieval of data. The concept of DM was developed by Ralph Kimball and consists of "fact" & dimension" tables. Dimension table records information on each

PTO

dimension, and fact records all the "fact", or measures.

- A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a DW. In contrast, relational models are optimized for addition, updating and deletion of data in real-time online transaction systems.
- These dimensional and relational models have their unique way of data storage that has specific advantages.
- for instance, in relational model, normalization and ER models reduce redundancy in data. On the contrary, dimensional model in DW arranges data in such a way that is easier to retrieve information and generates reports.
- Hence, Dimensional models are used in the DW systems and not a good fit for relational systems.

4. What is Metadata? Explain different types of metadata.

Metadata is "data that describes other data" (data dictionary)
Metadata in a data warehouse is similar to the data dictionary or the data catalog in a DBMS.
In data dictionary, you keep the information about the logical data structures, the information about files, addresses, info. about indexes. and so on, so data dictionary contains data about data in DB. Similarly, metadata component is data about data in the DW.

Types of Metadata:

> Operational metadata.

Data for the DW comes from various operational systems of enterprise. These source systems include different data structures. The data elements selected for the DW have various fields lengths & data types. When we deliver information from the source system to the end-users,

PTO

we must be able to tie that back to the source data sets. Operational metadata contains all this info. about the operational data sources.

> Extract & Transformation Metadata -

Extract & transformation metadata include data about the removal of data from the source systems, namely, the extract frequencies, extract methods, and business rules for the data extract. contains info. about all data transformation takes place in data staging.

> End-User Metadata -

The end-user metadata is the navigational map of the DW. It enables the end-users to find data from the DW. The end-user metadata allows the end-users to use their business technology terminology and look for the info. in those ways in which they usually think of the business.

---

5. A dimension Table is wide, the fact table is deep. Explain.

- fact table is a table in a star schema which contains facts and connected to dimensions, has two types of columns: those that include fact & those that use foreign keys to dimension table.
- whereas in other hand dimension is an architecture usually composed of one or more hierarchies that categorize data. If a dimension has not good got hierarchies and levels, it is called a flat dimension or list.

> fact tables are deep whereas dimension tables are wide as fact tables will have a higher no. of rows and a lesser number of columns. A primary key defined in the fact table is primary key to identify each row seperately. The primary key is also called a composite key in fact table. A dimension table contains a higher granular information so have less no. of records and it needs to have all necessary details (more columns) related to the grain of the table. On the other side,

PTO

a fact table has the lowest level grain of a subject area. Lower grain causes more number of rows in the fact table.

6. for a supermarket chain, consider the following dimensions namely product, store, time and promotion. The schema contains a central fact table for sales with three measures unit_sales, dollars_sales and dollar_cost.

i. Draw star schema

ii. Calculate the maximum no. of base fact table records for warehouse with the following values given below:

→ Time period - 5 years

→ Store - 300 stores reporting daily sales.

→ product - 40,000 products in each store (about 4000 sell in each store daily)

→ promotn - a sold item may be in only one promotn in a store on given day.

> Star Schema:

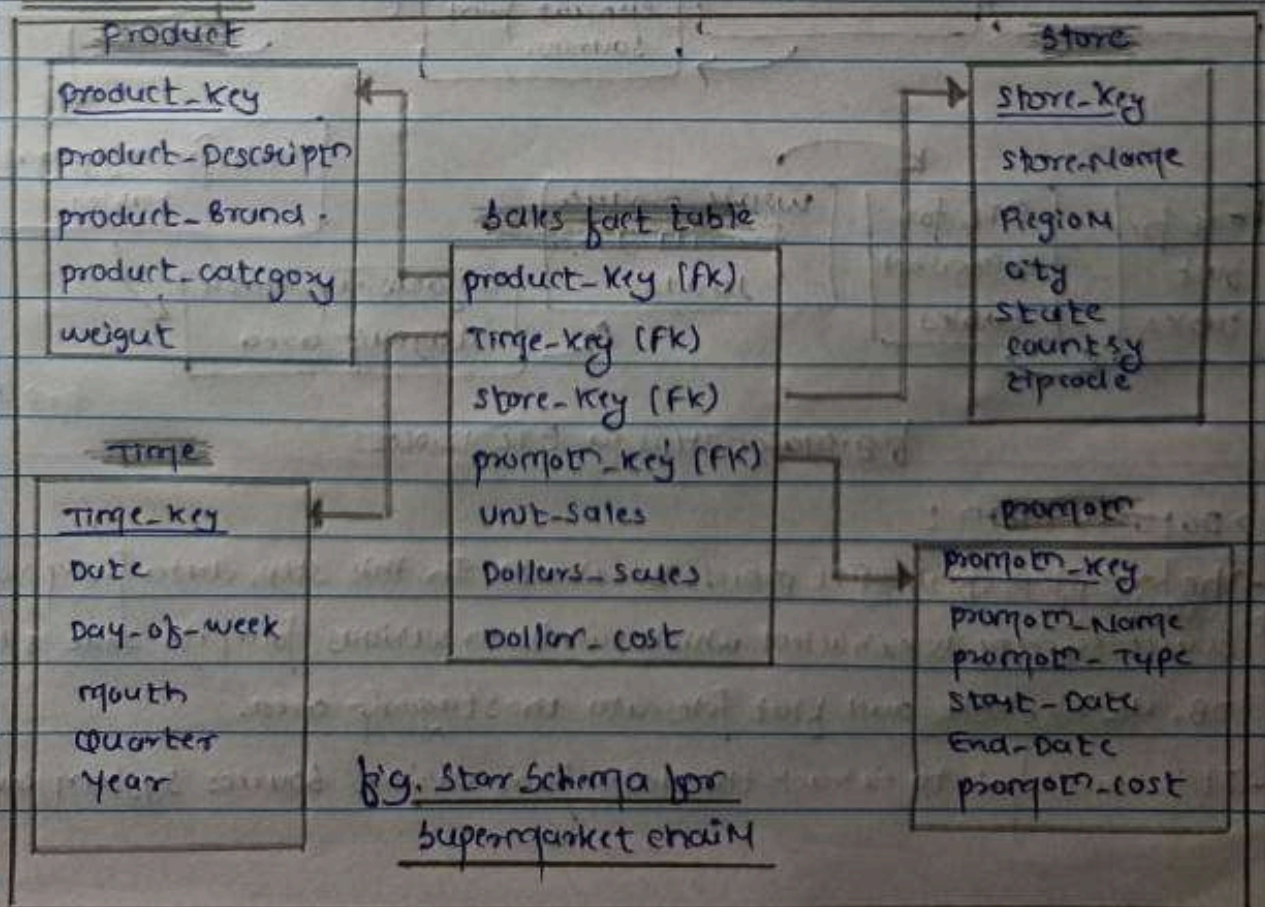


fig. Star Schema for supermarket chain

PTO

> fact table records

→ Time period = 5 years = 5 × 365 days = 1825 days

→ No. of stores = 300

→ product sell in each store daily = 4000

→ promotn = 1

∴ maximum no. of fact table records = 1825 × 300 × 4000 × 1

= 2190000000 records.

7. Explain the major steps in ETL process:

The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL, which stands for extraction, transformation, and loading.

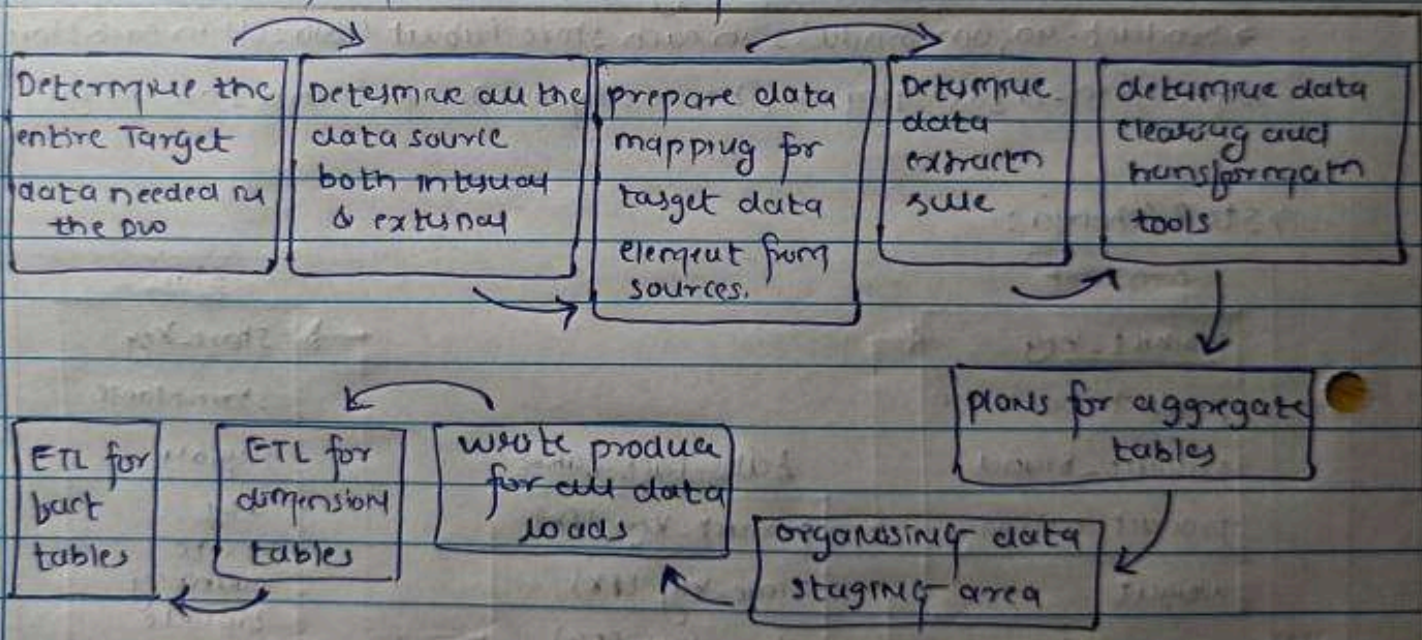


fig. major steps in ETL process

> Data extraction:

- The first step of the ETL process is extraction. In this step, data from various source system is extracted which can be in various formats like relational DB, NoSQL, XML and flat files into the staging area.
- It is important to extract the data from various sources system and

PTO

Store it into the staging area first and not directly into the DW because the extracted data is in various formats and can be corrupted also.
- Hence, loading it directly into the data warehouse may damage it & rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.

⇒ Identifying sources → Deciding methods → Extraction frequency → deciding the time window → Exceptional concern (which will handeled) → Identification of data sources.

Techniques:

⇒ Immediate data Extraction Technique: (Real time extraction)
→ capture through transaction logs → database triggers → Source application

⇒ deffered data Extraction
→ capture based on date and time stamp → comparing files → without transaction logs for data & time stamp.

> Data transformation:

- The second step of ETL process is transformation. In this step, a set of rules or function are applied on the extracted data to convert it into single standerised format.

⇒ Task involved in Data Transformation
→ format revision (data types, data field lengths)
→ decoding of fields → splitting of fields → Merging information
→ character set conversion → date & time conversion
→ summarization → key restructuring → de duplication

Techniques

⇒ Data Smoothing: used for removing the Noise from a dataset.
⇒ Data aggregation: collecting data from variety of sources & storing it in a single format. This method helps in collecting vast data.
⇒ Discretization: process of converting continuous data into set of data intervals
⇒ Generalization: low-level data attributes pass into high-level attribute.
⇒ Normalization: data transformation in data mining.

PTO

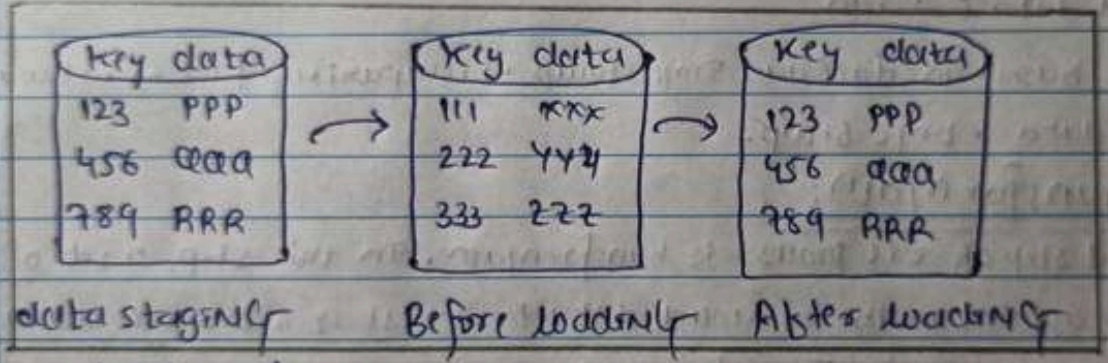
7 Data loading:

- The third and final step of the ETL process is loading. In this step, the transformed data is finally loaded into the data warehouse.
- Sometimes the data is updated by loading into the data warehouse very frequently & sometimes it is done after longer but regular intervals.
- The rate & period of loading solely depends on the requirements and varies from system to system.

⇒ Initial load, Incremental load, full refresh

Techniques:

data staging



⇒ load mode

⇒ append mode of data loading

⇒ destructive merge (w.r.t. p.k)

⇒ constructive merge.

8. Describe the steps involved in Data mining when viewed as a process of Knowledge Discovery.

- Knowledge Discovery in the database (KDD) is the process of searching for hidden knowledge in the massive amounts of data that we are technically capable of generating and storing.
- The Basic task of KDD is to extract knowledge for information from a lower level data (database).

PTO

- The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

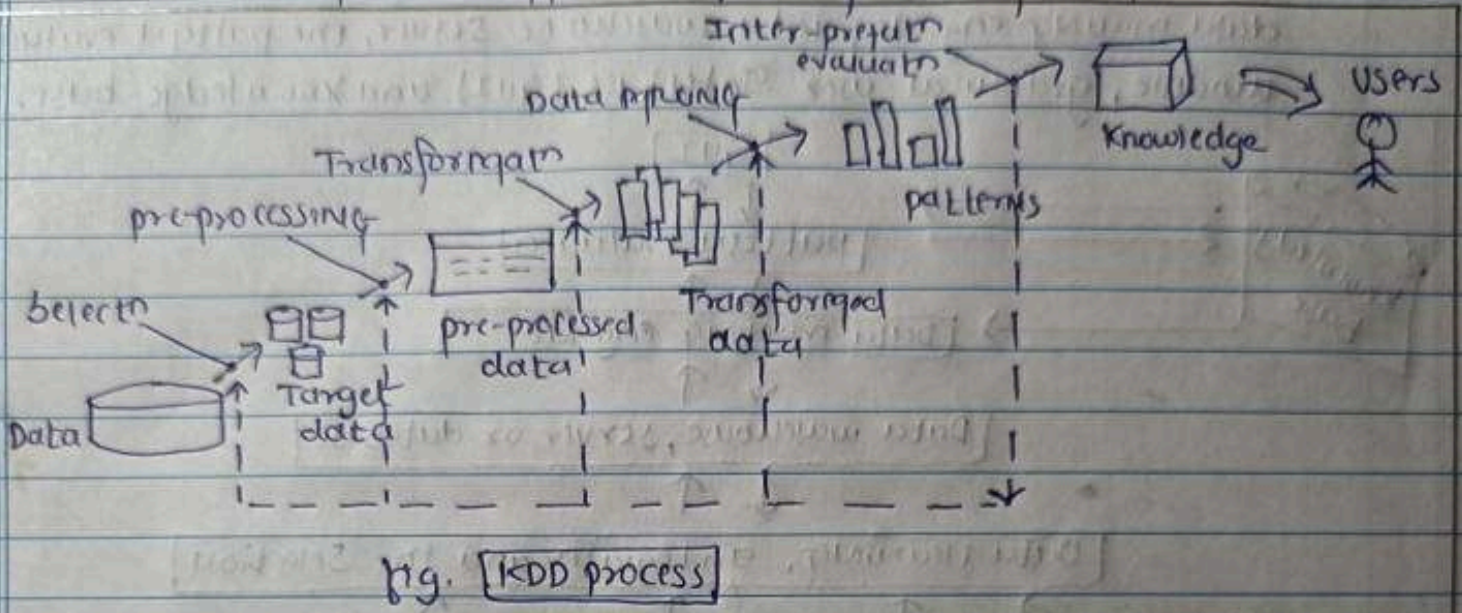


fig. KDD process

> Data cleaning: Removal of noise, inconsistance data & outliers.
> Data integration: Data from various sources such as Databases, data warehouse, and transactional data are integrated.
> Data Selection: Data relevant to the analysis task is retrieved from DB. Collecting only necessary information to the model.
> Data transformation: Transformed & consolidated into forms appropriate for mining by performing summary aggregation operation.
> Data mining: essential process where intelligent methods are applied to extract data patterns. (which model/parameters may appropriate)
> pattern Evaluation: To identify truly interesting patterns representing knowledge based on interesting measures.
> Knowledge presentation: visualization and knowledge representation techniques are used to present mined knowledge to users. visualization can be in form of graphs, charts or table.

This all are valuable or enlightening in its discovery.

9. Explain Data mining architecture in detail.

The significant components of data mining systems are data sources, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface (GUI) and knowledge base.

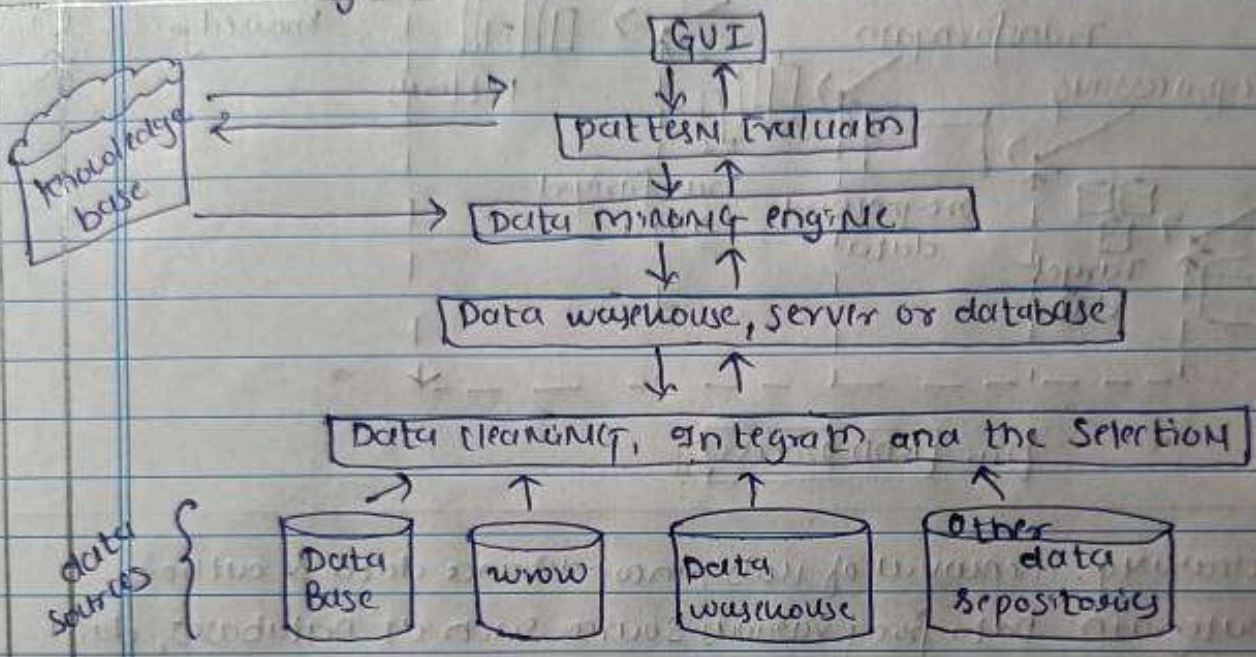


fig. Data mining architecture

> Data sources: Actual sources of data is the Database, data warehouse, www, txt files, and other documents. you need a huge amount of historical data for data mining successful

> Data cleaning, integration and selection: Before passing data to the database or DW server, data must clean, integrated & selected.

> Database or Data warehouse server: The DB or DW server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per the user request.

> Data mining engine: Major component of any data mining system. Contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series,

PTO

analysis, etc..

> pattern Evaluation module: The pattern evaluation module is primarily responsible for the measure of investigation of the patterns by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.
It might utilize a stake threshold to filter out discovered patterns.

> Graphical user interface: The graphical user interface (GUI) module communicate b/w the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process. Cooperates with data mining system when user specifies a query or a task & displays the results.

> Knowledge Base: Helpful in the entire process of data mining. Contains user view and data from user experiences that might be helpful in the reliable, data mining process. The pattern assessment module regularly interacts with knowledge base to get inputs & also update it.

10. At last.

11. Discuss the types of attributes & basic statistical descriptn of data in detail

Types of attributes

> Data objects: compose to form data sets. Represents entity. Typically described by attributes. If stored in DB, the data objects are referred to as data tuples.

> Attribute types: represent characteristic or feature of data object.

- Nominal attribute: qualitative attribute related to names. Nominal attributes are refered as categorical attributes and there is no order (rank position) among values of nominal attribute.

eg:-

Own House	: 1. yes 2. NO
Marital status	: 1. Unmarried 2. married

PTO

- Binary Attribute: quantative attribute. Data has only 2 values / state
for eg:- Yes or no, true or false.
Symmetric Binary attribute: Both values are equally important
eg:- Gender
Assymmetric Binary attribute: not equally important (e.g. Result)

Example:

Gender :	Male, Female
Cancer Detected :	Yes, No
Result :	pass, fail

- Ordinal attribute: Qualitative attribute
meaningful sequence or ranking (order) b/w them, but the magnitude b/w values is not actually known.

Example:

Grade :	A, B, C, D, E, F, O
Income :	Low, medium, High
product rating :	0, 1, 2, 3, 4, 5

- Numeric attribute:
quantitive because, it is measurable quantity, represent in integers or real values. Types → real interval or ratio scale.
→ interval - data can be added or subtracted but cannot be multiplied or divided. (Don't have correct reference point / zero point)
→ ratio-scaled attribute - fix zero-point (can be multiplied), compute difference b/w values, mean, median, mode, quantile-range, five no. summary can be given.
- Discrete attribute: quantitive attribute, can be numerical and also be in categorical form; have finite values.

Example:

profession :	physician, Teacher, clerk
zipcode :	400050, 400051, 400052

- Continuous attribute: quantitive; can take values b/w two specified values

Example:

Height :	5.2, 5.4, 5.6, ..
weight :	50.33, ..

PTO

Basic statistical description of data

essential to have an overall picture of the data, if data preprocessing to be made successful.

following are the basic statistical description of data.

⇒ Measures of central Tendency: used to represent centre of set data.

- Mean($\bar{x}$): mean of n no. sum of no. divided by n.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \dots x_n}{n}$$

for weighted arithmetic mean

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} = \frac{w_1 x_1 + w_2 x_2 + \dots w_n x_n}{w_1 + w_2 + \dots w_n}$$

- Median: Median of n no. is the middle no. when no. are written in order. If n is even, median is avg/mean of two middle no.

$$\text{median} = L_1 + \left[\frac{n/2 - (\sum \text{freq})_1}{\text{freq}_{\text{median}}}\right] \text{width}$$

$L_1 \rightarrow$ lower_boundary of median interval; $n \rightarrow$ no. of values in entire set

$(\sum \text{freq})_1 \rightarrow$ sum of freq; $\text{freq}_{\text{median}} \rightarrow$ median interval; width...

- Mode: Mode of n no. is the no. or no.s that occur most frequently. may be one mode, no mode, or more than one mode.

for unimode, numeric data are asymmetrical.

mean - median ≈ 3 × (mean - median)

- Midrange: Average of smallest & largest value in the set.

⇒ Dispersion of data: statistic that tells you how dispersed, or spread out, data values, are

- Quantiles: values that divides your values into quantals

PTO

As quantiles divides no. up according to when their position is on no. line, you have to put the no. in order before you can fig. out where quantile are.

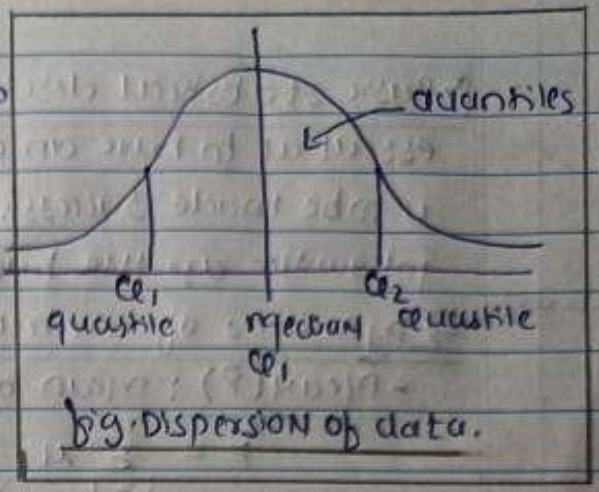
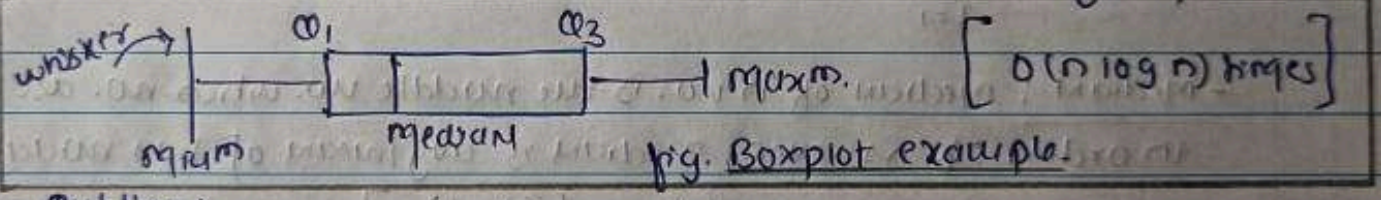


fig. Dispersion of data.

- Range: difference btw the upper and lower quantile value in set of data. commonly referred as IQR

$IQR = Q_3 - Q_1$

- Five no. summary: Rough data about your data set looks like. Includes five items: minimum value, Q_1 (first quantile, median, third quantile (Q_3), and the maximum value.
- Boxplot: (or whisker plot) defined as graphical method of displaying variation in set of data. Incorporates the five-summary as follow.



[O(n log n) times]

fig. Boxplot example.

- Outlier: value higher or lower than 1.5 x IQR (Inter-quantile range)
- variance and standard deviation: Measures of data dispersion for data set $x_1, x_2 \ldots x_n$ variance is calculated as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right) - (\bar{x})^2$$

The standard deviation is square root of variance σ^2.
when all observation have same values, $\sigma = 0$; otherwise $\sigma > 0$.

⇒ Graphic Displays of Basic Statistical Description of data:
Helpful for visual inspection of data, useful for data preprocessing.
Includes quantile, quantile-quantile plots, histograms & scatter plots.

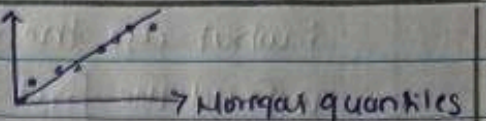
- Quantile plot: Graphical way of checking whether your data are normally distributed i.e. invariant distribution.

PTO

we compute for each data value, the normal quantile value as

$$z = \frac{x_i - \bar{x}}{\rho}$$

Rough fig:-



- quantile-quantile plot: plots of two quantiles against each other.
Point where certain values fall below that quantile.
purpose of q-q plots is to find out if two sets of data comes from the same distribution. [comparison]
If two data sets come from common distribution, points will fall on single reference line.

- Histograms: usually shows the distribution of values of single variable divides values into bins & shows a bar plot of no. of object in each bin.
height of each bar indicates no. of objects.
shape depend on no. of bins.

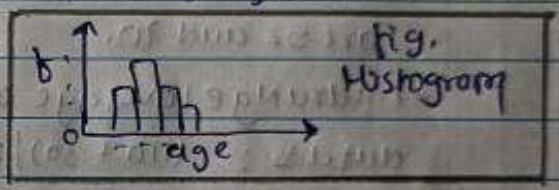


fig. Histogram

- Scatter plot: determining if there is relationship, pattern, or trend existing btw two numeric attributes.
Bivariate distribution of data
Comparison of two attributes (correlation). amount of data in cluster
correlation can be, +ve, -ve or zero.

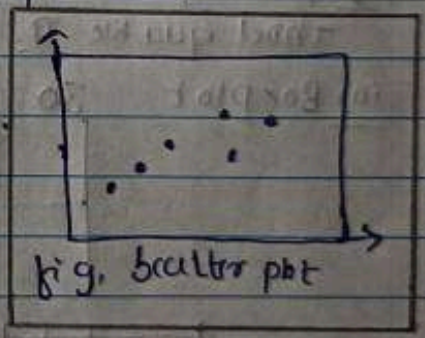


fig. Scatter plot

10. college wants to record the marks for the courses completed by students using the dimensions. i) course, ii) student, iii) Time & a measure Aggregate marks. Create a cube and perform following OLAP operation: i) ROLLUP ii) DRILL down, iii) SLICE, iv) DICE, v) pivot.
pg.no.

12. Suppose that the data for analysis includes the attribute salary (in thousand of dollars), shown in increasing order.

PTO

30, 36, 47, 50, 52, 56, 60, 63, 70, 70, 110.

i. what are the mean, median, mode & midrange of data?

ii. find the first quartile (Q_1) & third quartile (Q_3) of data.

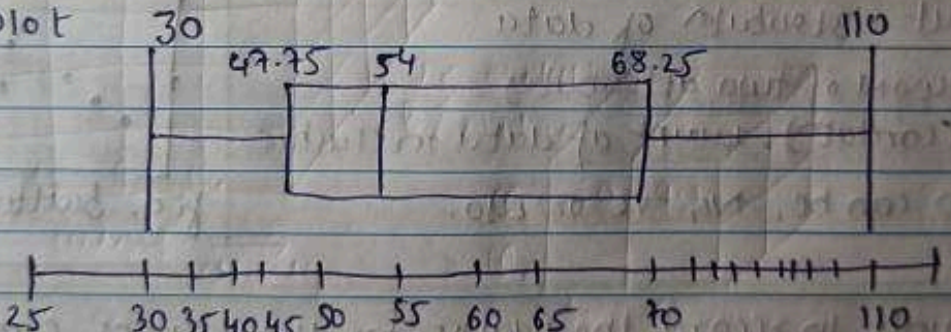
iii. Show the boxplot of the data.

i) Mean $= \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{696}{12} = 58$

- Median (mean of two middle value of ordered set, as the no. of values in the set is even) of the data is 54.
- This value set has two values that occur with the same highest frequency and is, therefore Bimode.
  are modes (values occurring with greatest frequency) of the data are 52 and 70.
- Midrange (average of largest & smallest value in data set) of the data is : $(110 + 30)/2 = 70$.

ii) first quartile Q_1 (corresponding to 25th percentile) of data is : 47.75
third quartile Q_3 (— 1 — 75th percentile) of data is : 68.25

iii) Boxplot



Boxplot for salary attribute.

OLAP Operation :

1. Slice : for single students get the marks for all courses and all semester is slice operation as cube is generated with respect to one dimension Student-Dim

2. Dice : Get the marks of single (or multiple) students for one (or more)

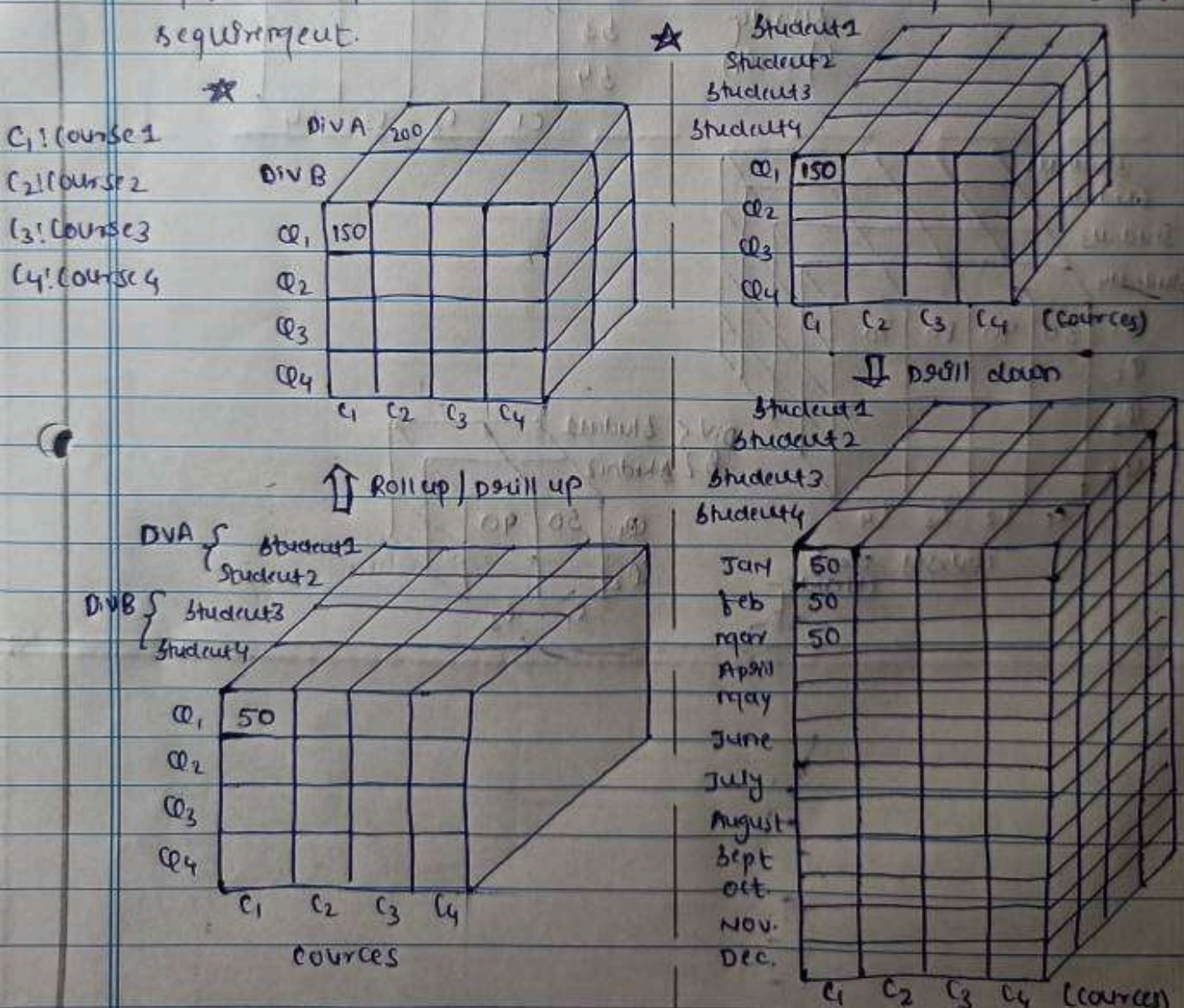
PTO

Semester but for all courses is dice operation on cube is generated with respect to two dimensions i.e. Student_Dim and Time_Dim.

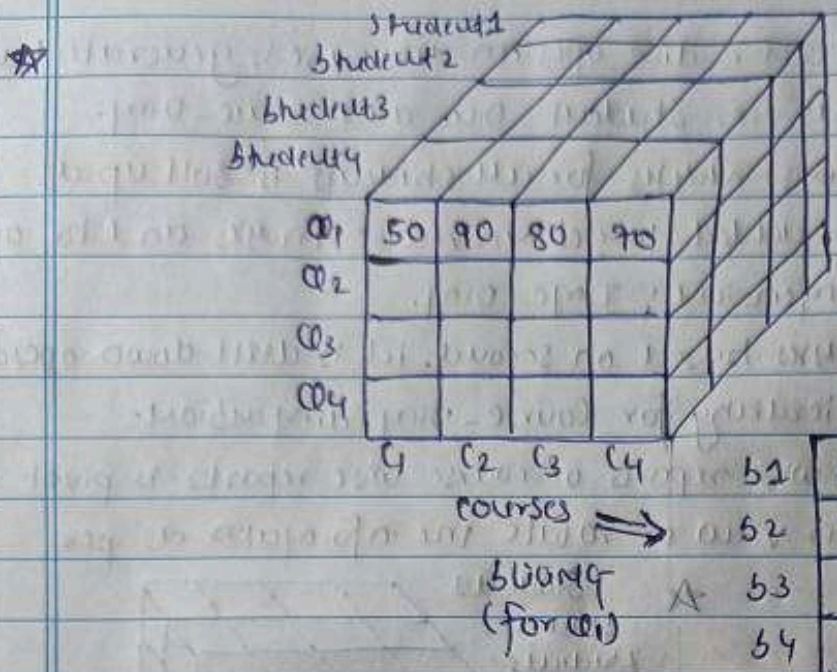
3. Roll up: Get the year wise marks for all students is roll up as semester wise marks are added to get year wise marks and it is one level up Hierarchy of dimension: Time_Dim.

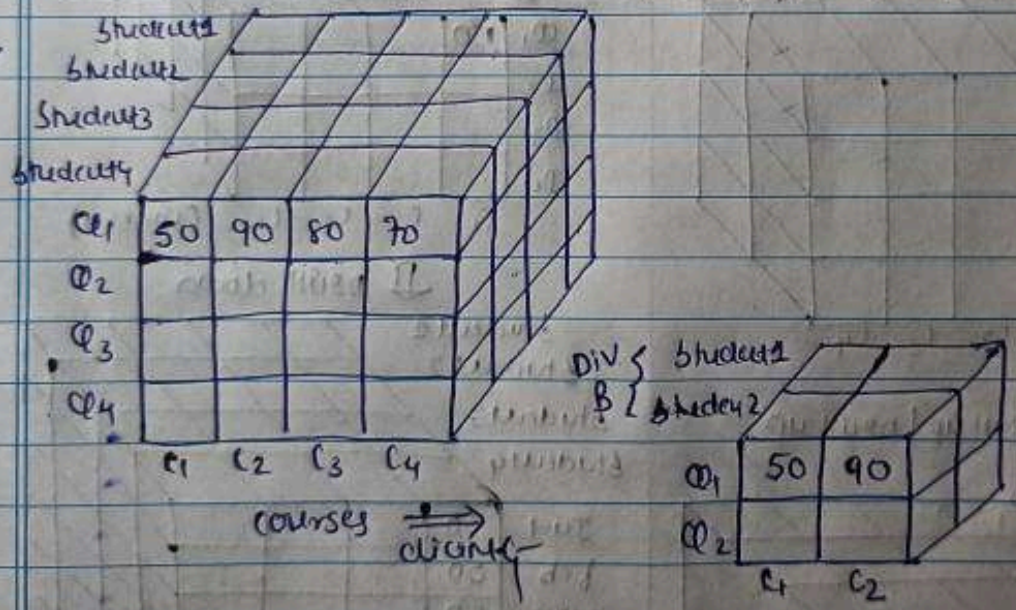
4. Drill Down: Get the marks based on second_id is drill down operation as its one level down hierarchy for Course_Dim. dimension.

5. pivot: Getting the room wise reports or course wise reports is pivot operation. In pivot one can view or rotate the information as per requirement.



PTO


Student1
Student2
Student3
Student4
Q1 50 90 80 70
Q2
Q3
Q4
C1 C2 C3 C4
courses
Student
(for Q1)
S1 50 90 80 70
S2
S3
S4
C1 C2 C3 C4


Student1
Student2
Student3
Student4
Q1 50 90 80 70
Q2
Q3
Q4
C1 C2 C3 C4
courses
student
DIV B
Student1
Student2
Q1 50 90
Q2
C1 C2

Name :- Prathamesh S. Chikankar

Roll No. :- AIML11 Branch :- CSE-(AI & ML)

Year :- TE Subject :- DWM

Topic :- Assignment No. 02

Sign :- Prathmesh Date :- 06th Oct '22

1. Explain Market Basket Analysis with an example.

- Market Basket Analysis is a data mining technique that is used to uncover patterns in any retail setting.
- The goal of Market Basket analysis is to understand consumer behaviour by identifying relationships b/w the items that people buy.
- A co-occurrence is when two or more things take place together.
- Creates If-then scenario rules, for eg, if item A is purchased then item B is likely to be purchased.
- The rules could be written as: If {A} then {B}
- Association rule has three measures that express degree of confidence in the rule. They are support, confidence and lift.

> The support is the no. of transaction that include items in the {A} and {B} parts of the rule as percentage of total no. of transn. Also called as occurrence frequency, frequency, support count or count.

$$\text{Support } (A \to B) = \frac{\text{No. of transaction containing both A and B}}{\text{Total no. of transaction.}}$$

> The confidence of the rule is the ratio of no. of transaction that include all items in {B} as well as the {B A} to the no. of {A}.

$$\text{Confidence } (A \to B) = \frac{\text{no. of transaction containing both A and B}}{\text{No. of transaction containing A}}$$

> The lift or lift ratio, is the ratio of confidence to expected confidence. Expected confidence is the confidence divided by the frequency of B. Greater lift values indicate stronger association.

$$\text{Lift } (A \to B) = \frac{\text{no. of transaction containing both A \& B}}{\text{no. of transaction containing A}} \Big/ \frac{\text{no. of transaction containing B}}{\text{total no. of transaction}}$$

- Example: Consider there are nine laptops containing varying combination of Asus, Acer, MSI, and Microsoft:

PTO

laptop	product 1	produt 2	product 3
1	Acer	ASUS	
2	Acer	MSI	ASUS
3	MSI	Microsoft	
4	Acer	ASUS	
5	MSI	Microsoft	
6	Acer	ASUS	Microsoft
7	Acer	ASUS	
8	ASUS	Microsoft	
9	ASUS	Acer	

> support (Acer → ASUS)

$$= \frac{\text{no. of baskets containing both Acer \& ASUS}}{\text{Total no. of laptops}}$$

$$= \frac{6}{9} \Rightarrow 0.67$$

> confidence (Acer → ASUS)

$$= \frac{\text{no. of laptops containing both Acer and ASUS}}{\text{No. of laptops containing Acer}}$$

$$= \frac{6}{6} \Rightarrow 1.00$$

> Lift (A → B) =

$$= \frac{\text{no. of baskets containing both Acer and ASUS}}{\text{no. of laptops containing Acer}} \Big/ \frac{\text{no. of laptops containing ASUS}}{\text{Total no. of laptops.}}$$

$$= \frac{6/6}{7/9} = 1.29$$

- Applications of Market Basket Analysis:

• Retail, • Telecommunications, • Banks, • Insurance & • Medical.

Q. 2. Discuss Association Rule Mining and Apriori Algorithm

Association rule mining finds interesting association and correlation among large sets of data items.

This rule shows how frequently a itemset occurs in a transaction.

A typical example is Market Basket Analysis.

An implication expression of the form $X \rightarrow Y$, where X and Y are any 2 itemsets.

Example : {Milk, Diaper} → {Beer}

This rule must satisfy the confidence value.

Rule evaluation matrics -

- Support (s) -
- Confidence (c)
- Support = σ(X+Y) ÷ total
- Lift (L)
- Conf (X => Y) = Supp (X ∪ Y) ÷ Supp (X)
- Lift (X => Y) = Conf (X => Y) ÷ Supp (Y)

The association rule is very useful in analysing datasets.

Apriori algorithm was the first algorithm that was proposed for frequent itemset mining.

It is an iterative approach to discover the most frequent itemsets.

This algorithm uses two steps "join" and "prune" to reduce the search space.

- Join step : this step generate (K+1) itemset from K-itemsets by joining each item with itself.
- Prune step : this step scans the count of each item in the database.

- Advantages: Easy to understand.
  Join & prune steps are easy to implement on large itemset in large DB.
- Disadvantages: Entire database needs to be scanned
  Requires high comparison if the itemsets are very large and minimum support is kept very low.

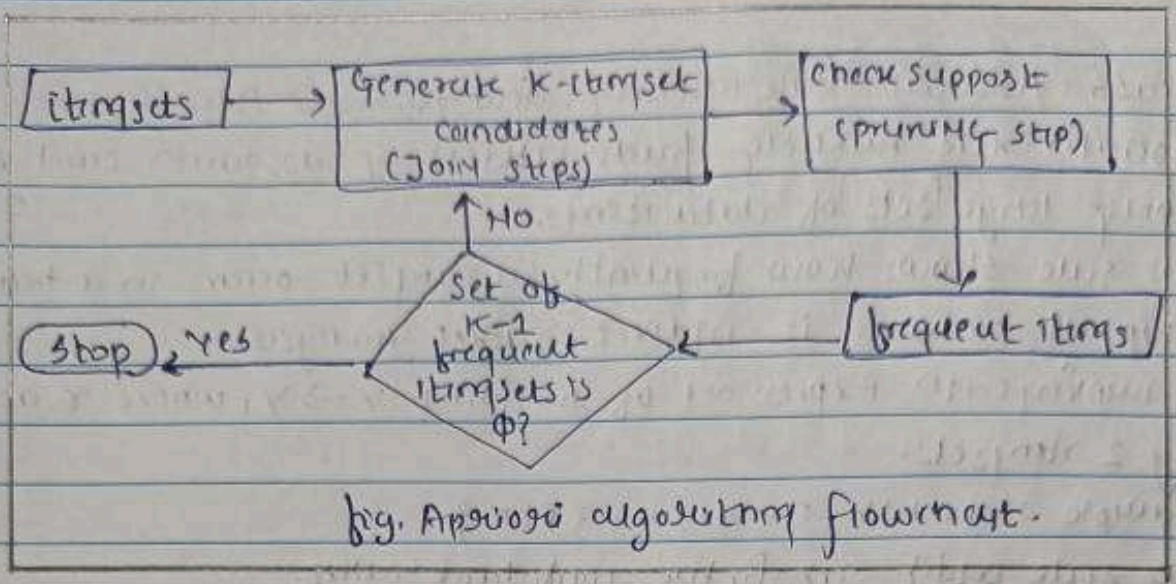


fig. Apriori algorithm flowchart.

3. A database has four transactn. Let minimum support = 50% and minimum confidence = 50%.

Transaction_id	items bought
T100	A, B, C
T200	A, C
T300	A, D
T400	B, E, F

find all frequent item sets using apriori algorithm.
List strong- association rules
The given minimum support = 50%
The no. of transactn = 4
∴ minimum support = 0.5 × 4 = 2
Step 1: Calculate the minimum count of each item.

∴

items	minimum count
A	3
B	2
C	2
D	1
E	1
F	1

PTO

Step 2 - Delete the items that do not have a minimum support count of 2.
∴ Items D, E and F are deleted.

Items	minimum cost
A	3
B	2
C	2

Step 3 - Combine 2-items and find out the minimum cost count of the occurences of the 2-items.

items	minimum cost
A, B	1
A, C	2
B, C	1

Now, only got the item-set {A, C} that is frequent.
Step 4 - Generate Association Rules form the frequent itemset discovered in the above step-3.

Rule 1 - {A} ⇒ {C}

$$\text{Confidence} = \frac{\text{Support } \{A, C\}}{\text{Support } \{A\}} = \frac{2}{3} \times 100 = \underline{66.67\%}$$

Rule 2 - {C} ⇒ {A}

$$\text{confidence} = \frac{\text{Support } \{A, C\}}{\text{support } \{C\}} = \frac{2}{2} \times 100 = \underline{100\%}$$

the minimum given confidence = 50%.
∴ this shows that all the above association rules are strong.

Apriori is an algorithm for frequent item set mining and association rule learning over the given dataset.
It works by identifying the frequent individual items in the

dataset and executing them to large and larger item sets as long as those item sets appear sufficiently often in the dataset.

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

D: a database of transactions;

min_sup: the minimum support count threshold.

output: L: frequent itemsets in D.

Method:

1) L1 = find_frequent_1-itemsets(D);
2) for (k=2; Lk-1 ≠ ϕ; k++) {
3) Ck = apriori_gen (Lk-1);
4) for each transaction t ∈ D {
5) Ct = subset (Ck, t);
6) for each candidate c ∈ Ct
7) c.count ++;
8) }
9) Lk = { c ∈ Ck | c.count ≥ min_sup}
10) }
11) return L = ∪k Lk;

procedure apriori_gen (Lk-1: frequent (k-1) -itemsets)

for each itemset l1 ∈ Lk-1

for each itemset l2 ∈ Lk-1

if ((l1[1] = l2[1]) ∧ (l1[2] = l2[2])

∧ ... ∧ (l1[k-2] = l2[k-2]) ∧ (l1[k-1] < l2[k-1]))

then {

c = l1 ⋈ l2;

if has_infrequent_subset (c, Lk-1) then

delete c;

```
        else add c to Lk
    }
return Lk;
procedure has_frequent_subset (c: candidate k-itemset;
                Lk-1: frequent (k-1) -itemsets);
for each (k-1)-subset s of c
    if s ∈ Lk-1 then
        return TRUE;
return FALSE;
```

Q.4. A database has five transactn. Let min_sup = 60% & min_conf = 80%. Find all the frequent itemsets using F-P growth.

TID	items_bought	ordered_itemsets
T100	{M, O, N, K, E, Y}	K, E, M, O, Y
T200	{D, O, N, K, E, Y}	K, E, O, Y
T300	{M, A, K, E}	K, E, M
T400	{M, U, C, K, Y}	K, M, Y
T500	{C, O, O, K, I, E}	K, E, O

GIVEN: min_sup = 60%

∴ sup_count to be satisfied = 5 × 0.6 = 3

Step 1: Scan the databases for count of each itemset.

Itemset	sup_count
{A}	1
{C}	2
{D}	1
{E}	4
{I}	1
{K}	5

Itemset	sup_count
{M}	3
{N}	2
{O}	4
{U}	1
{Y}	3

Step 2: Sort the set of frequent itemsets in the order of descending support count and denote that usb as a L.

L:

Itemset	Sup_count
{K}	5
{E}	4
{O}	4
{M}	3
{Y}	3

Step 3: Scan the database for second time and sort items in each transaction according to descending support count ie. ordered_itemset

TID	List of items
T100	{K, E, M, O, Y}
T200	{K, E, O, Y}
T300	{K, E, M}
T400	{K, M, Y}
T500	{K, E, O}

Step 4: construct the FP tree

4.1. Create a node with label 'NULL'.

4.2. Scan T100, T200, T300, T400 T500, T600.

4.3. Construct Branch with their respective nodes

4.4. Also increment its count by 1

4.5. Connect all similar nodes.

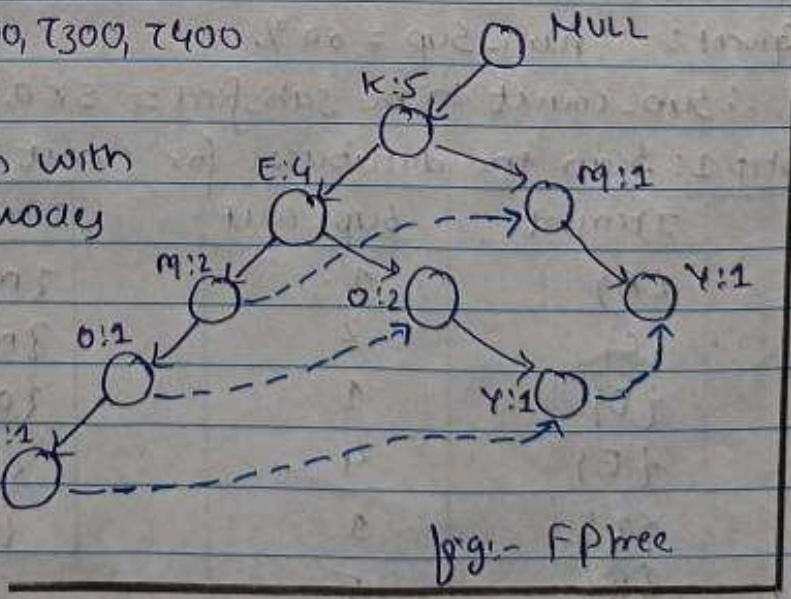


fig:- FP tree

PTO

Step 5: Mining FP-tree

Itemset	Conditional pattern base	Conditional FP-tree	Frequent patterns generated
{Y}	{{K, E, M, O:1}, {K, M:1}, {K, E, O:1}}	{K:3}	{K, Y:3}
{O}	{{K, E, M:1}, {K, E:2}}	{K:3, E:3}	{K, O:3}, {E, O:3}, {K, E, O:3}
{M}	{{K, E:2}, {K:1}}	{K:3}	{K, M:3}
{E}	{{K:4}}	{K:4}	{K, E:4}
{K}	-	-	-

FP tree

Frequent pattern Tree is a tree-like structure that is made with the recurring itemsets of the dataset.

The purpose of the FP tree is to mine the most frequent Data pattern.

Each node of the FP tree represents an item of the itemset.

Algorithm:

FP growth: mine frequent itemsets using FP tree by pattern fragment growth.

Input:

D, a transaction DB;

min_sup, the minimum support count threshold.

output:

Complete set of frequent patterns.

Method:

Scan

mscit_tree ([PIP], T)

mscit-tree (P, N)

FP_growth (FP tree, null)

procedure FP-growth (Tree, α)

PTO

if tree contains a single path P then
for each combination
else for each a_i in the header of Tree2
{
;
If Tree$_B$ = ϕ then
call FP_growth (Tree, β)}}

Q.5. Demonstrate multidimensional & multilevel Association rule mining with suitable examples.

Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules.

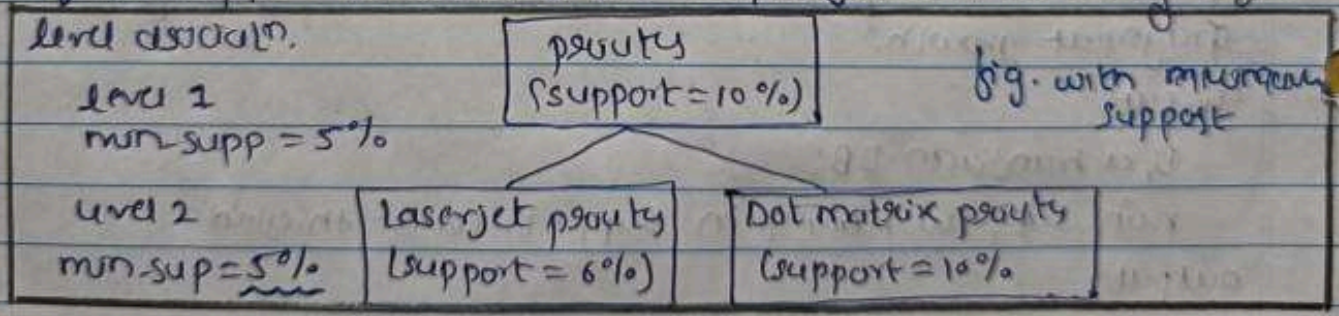
> Support and confidence of multilevel association rules.

Support of rules increases specialized and decreases generalized itemsets.
Confidence - Support is below threshold value, rule becomes invalid.

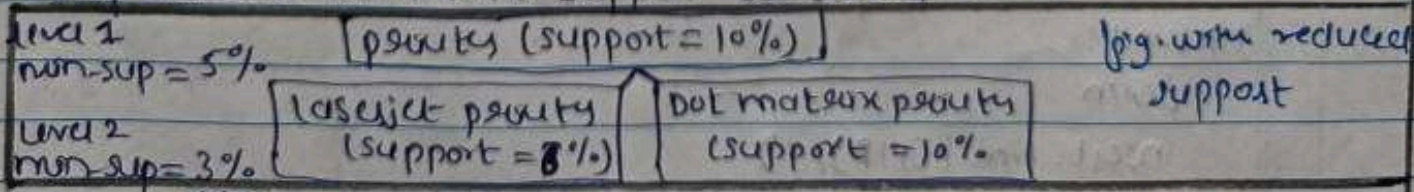
> Approaches of multilevel association rules:

-Using uniform support level for all levels.

If the support threshold is too low, it generate too many high level association.



-Using reduced minimum support at lower level.



at every abstraction level, there is its own minimum support threshold; so minimum support at lower levels reduces.

-Using item or group-based minimum support.

PTO

Multidimensional associaton rules. (database)
> Single-dimension rule : buys (X, "milk") → buy (X, "bread")
> Multi-dimensional rule : contain more than one predicate.
> categorical attribute : finite no. of values, eg:- color.
> quantitative attribute : numeric values, eg:- age
Techniques: can be categorical or quantitative
- Static Discretisation of quantitative attributes.
We refer to this as MRUNG- multidimensional associaton rules using static discretisation of quantitative attributes.

The discretized numeric attributes, with their interval labels, can then be treated as categorical attributes (where each interval is category considered)

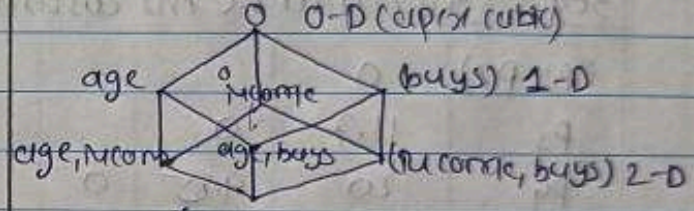


fig. Lattice of cuboids to form 3-D data cube.

- Dynamic quantitative associaton rules

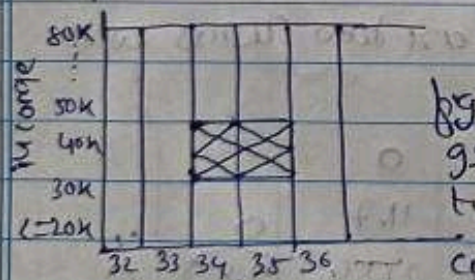


fig. A 2-D grid for tuples representing customers purchasing SUV

the strong associaton rules obtained are mapped a 2-D grid as shown. This can further combine

Q.6. Differentiate b/w Simple linkage, average linkage and complete linkage algorithms. Use a complete linkage algorithm to find the clusters from the following dataset.

X	4	8	15	24	724
Y	4	4	8	4	12

Step 1: Calculate distance from each object (point) using Euclidean distance

PTO

Distance $[(x,y),(a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$

Distance matrix:

P1	0				
P2	(4)	0			
P3	11.7	8.06	0		
P4	20	16	9.85	0	
P5	21.54	17.89	9.85	8	0
	P1	P2	P3	P4	P5

P1 and P2 are two clusters with shortest distance 4
merge P1 and P2

Step2: Recalculate the distance of P3 from (P1,P2):

(P1,P2)	0			
P3	11.7	0		
P4	20	9.85	0	
P5	21.54	9.85	(8)	0
	(P1,P2)	P3	P4	P5

$dist((P1,P2),P3) = max(dist(P1,P3), dist(P2,P3))$
$= max(11.7, 8.06)$
matrix 11.7

Step3: In above matrix P4 and P5 are two clusters with shortest distance 8, merge P4 and P5

Recompute the distance matrix

(P1,P2)	0		
P3	11.7	0	
(P4,P5)	21.54	(9.85)	0
	(P1,P2)	(P3)	(P4,P5)

Step4:

P3 and (P4,P5) are two clusters with shortest distance 9.85
MERGING
Recompute the distance matrix

(P1,P2)	0	
(P3,P4,P5)	21.54	0
	(P1,P2)	(P3,P4,P5)

PTO

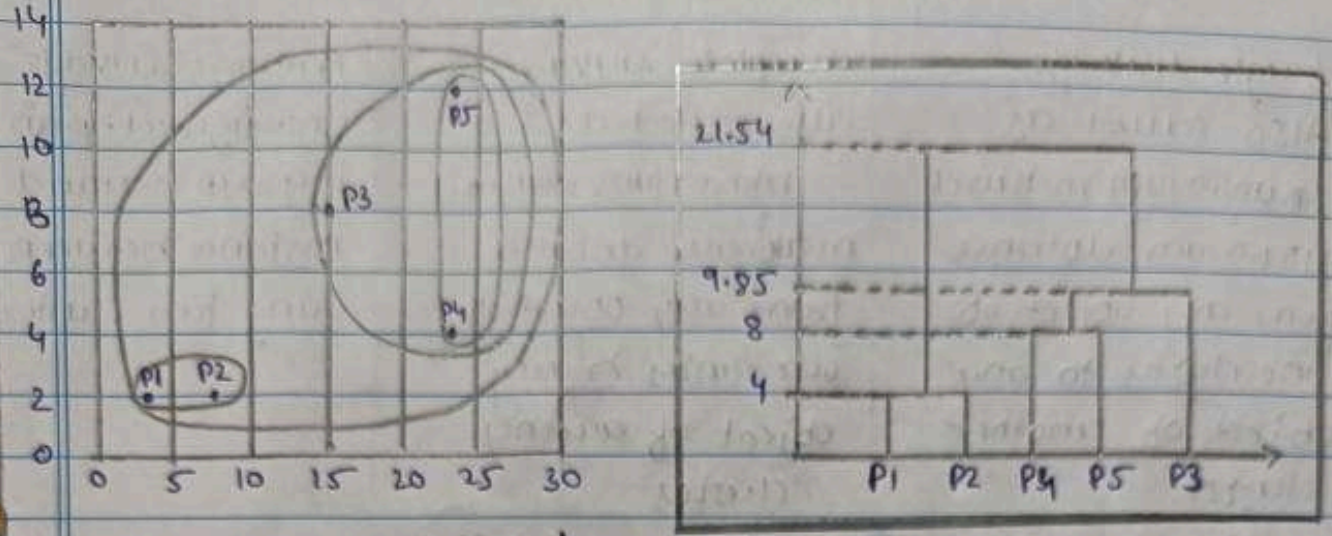


Fig. Dendogram

Step5: Looking at the above distance matrix in Step4, we see that (P1, P2) and (P3, P4, P5) have the smallest distance 21.54 (only one left) so, we merge those two single values clusty. No need to re-compute. as there are no more clusters to merge.

Hierarchical clustering

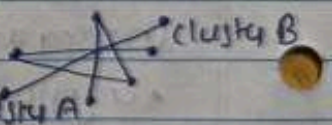
- produces set of nested clusters organized as hierarchical tree
- can be visualized as dendogram - a tree like diagram that records the sequence of merges or split.
  - agglomerative
  - DIVSE

Algorithm:

1. compute the distance matrix b/w input data points
2. Let each data point be a clusty
3. Repeat
4. merge the two closest clusters
5. update the distance matrix
6. Until only a single clusty remain.

Advantage: no. prior info. required about clusters.

disadvantage: can never undo what was done previously.

	Single linkage	Complete-linkage	Average-linkage
>	Also called as minimum method	also called as maximum method	unweighted-pair group method
>	minimum distance from any object of one cluster to any object of another cluster.	maximum distance from any object of one cluster to any object of another clusters	Distance between any two clusters
>	D(A,B) computed as D(A,B) = min{d(i,j)} : i→ in cluster A j→ in cluster B	D(A,B) = max{d(i,j)} i→ in cluster A j→ in cluster B	D(A,B) = mean{d(i,j)} i→ in cluster A j→ in cluster B
>	'loose' clusters	"tight" clusters	"in blw" clusters
>	can handle non-elliptical shape	less susceptible to noise and outliers	less susceptible to noise & outliers.
>	Sensitive to noise and outliers	Tends to break large clusters	Biased towards globular clusters.
>			

Q7. Explain Naives Bayesian classification in detail:

Naives Bayes classifiers are a collection of classification algorithms based on 'Bayes' theorem. It is not a single algorithm but a family of algorithms, where all of them share a common principle, i.e. every pair of features being classified is independent of each other. To start with, let us consider a dataset. Consider a fictional dataset that describes the weather conditions for playing a game of golf. Given the weather conditions, each tuple classifies that conditions as

PTO

fit ("Yes") or unfit ("No") for playing golf.
classify using Naives bayes classification

outlook	temp	Humadity	windy	play
Rainy	cool	high	true	?

Bayes theorm: find probability of an event occuring given the probability of another event that has already occured.
Stated mathematically as:

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

.. event B is true ie. evidence
... event A prior

regards to our dataset

$$P(y|x) = \frac{P(x|y)\,P(y)}{P(x)}$$

.. y → class variable
.. x → dependent feature $(x_1, x_2, \ldots x_n)$

Naives assumption

$$P(y|x_1, \ldots x_n) = \frac{P(x_1|y)\,P(x_2|y) \ldots P(x_n|y)]\,P(y)}{P(x_1)\,P(x_2) \ldots P(x_n)}$$

also mathematically

$y = \text{argmax}_y P(y) \prod_{i=1}^{?} P(x_i|y)$

to calculate $P(y)$ & $P(x_i|y)$ ... class prob. & conditional prob.

Steps:-

importing libraries
importing datasets
Reading data
Splitting data
Creating model
fitting model
making predictn
Evaluating result

Apply the Naives Bayes classifier algorithm to classify an unknown sample X (outlook = Sunny, temperature = cool, humidity = high,

PTO

windy = false) The sample dataset is as follows:

outlook	Temperature	Humidity	windy	class
sunny	Hot	High	false	N
sunny	Hot	High	true	N
overcast	Hot	High	false	P
RaiN	mild	High	false	P
RaiN	cool	Normal	false	P
RaiN	cool	Normal	true	N
overcast	cool	Normal	true	P
sunny	Mild	High	false	N
sunny	cool	Normal	false	P
RaiN	Mild	Normal	false	P
sunny	Mild	Normal	true	P
overcast	Mild	High	true	P
overcast	Hot	Normal	false	P
RaiN	Mild	High	true	P

P c (class = Yes) = 10/14 = ~~0.64~~

N c (class = No) = 4/14 = ~~0.36~~

outlook	Y	N
sunny	2/9	3/5
overcast	4/9	0
rain	3/9	2/5

Humidity	Y	N
High	3/9	4/5
normal	6/9	1/5

Temperature		
Hot	2/9	2/5
mild	4/9	2/5
cool	3/9	1/5

windy			
strong	3/9	3/5	true
weak	6/9	2/5	false

(outlook = sunny, Temperature = cool, Humidity = high
windy = true)
windy = false)

RTO

Soln:-

Consider the tuple X to classify.

X = [(outlook) = sunny], temperature = cool, humidity = high, windy = false]

The data tuples described by the attribute look (outlook), temperature humidity, windy

The class label class which has 2 distinct values P and N

Let C1 corresponds to the class P & C2 corresponds to the class N

X = (outlook = sunny, temperature = cool, humidity = high, windy = false)

We need to calculate position probability

P(X|Ci) P(Ci) for i = 1, 2

P(class = P) = 10/14 = 0.174

P(class = N) = 4/14 = 0.285

To compute P(X|Ci) for i = 1, 2, we compute the following conditional probability.

P(outlook = sunny / class = P) = 2/10 = 0.2

P(outlook = sunny / class = N) = 3/4 = 0.75

P(temperature = cool / class = P) = 3/10 = 0.3

P(temperature = cool / class = N) = 1/4 = 0.25

P(humidity = high / class = P) = 4/10 = 0.4

P(humidity = high / class = N) = 3/4 = 0.75

P(windy = false / class = P) = 6/10 = 0.6

P(windy = false / class = N) = 2/4 = 0.5

P(X / class P) = P(outlook = sunny / class = P) × P(temperature = cool / class = P) × P(humidity = high / class = P) × P(windy = false / class = P)

= 0.2 × 0.3 × 0.4 × 0.6 ⇒ 0.0144

PTO

$P(X \mid class=N) = P(outlook=sunny \mid class=N) \times P(temperature=cool \mid class=N) \times$
$P(humidity=high \mid class=N) \times P(windy=false \mid class=N)$
$= 0.75 \times 0.25 \times 0.75 \times 0.5 \Rightarrow 0.0703$

To find the class i that maximizes $P(X \mid C_i) P(C_i)$ we compute
$P(X \mid class=P)\, P(class=P) = 0.0144 \times 0.0714 \Rightarrow 0.0028$ 0.01028.
$P(X \mid class=N)\, P(class=N) = 0.0703 \times 0.285 \Rightarrow 0.0200.$

The Naive Bayesian classification algorithm predicts class N for tuple X.

Q. 8. Explain K-Means clustering algorithm.

> K-Means: Each cluster is represented by the center of cluster.
> K-Means clustering is simple unsupervised learning algorithm
> K-means tries to put passed n data points into the set of K clusters where each data point is assigned to its closest cluster.
> This method is defined by the objective function which tries to minimize the sum of all squared distances within a cluster, for all the clusters.
> objective function: $\arg_S \min \sum_{x_i \in S_i} \| x_j - \mu_i \|^2$ ∴ x_i = data point
S_i = cluster.
μ_i = cluster mean.
> K-Means clustering Algorithm

Algorithm: K-means. → Each cluster is represented by mean value.
Input: K: the no. of clusters.
D: dataset containing n objects
Output: A set of K clusters.
Method:
(1) arbitrarily choose k objects from D as initial cluster centers,
(2) Repeat
(3) (re) assign each object to the cluster to which the object is most

PTO

similar, based on the mean value for the objects in the clusters.
(4) update the cluster means, calculate mean value of object
(5) until no changes.

Explain K-Medoids clustering algorithm.

> K-medoids (also called as partitioning around medoid) algorithm.
> A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum
> The dissimilarity of the medoid (Ci) and object (Pi) is calculated by using $E = |P_i - C_i|$
> The cost in k-medoids algorithm given by

$$c = \sum_{C_i} \sum_{P_i \in C_i} |P_i - C_i|$$

> Algorithm: K-medoids. PAM .. based on medoid or central objects.

Input: K: no. of clusters
D: data set containing n objects

Output: A set of K clusters.

Method:

(1) arbitrarily choose k objects in D as initial representative seeds.
(2) Repeat
(3) assign each object to nearest representative object.
(4) randomly select non-representative object, O_{random};
(5) compute total cost, S, object O_j with O_{random};
(6) If $S<0$ then swap O_j with O_{random} to form new set of k
(7) until no change;

> Time complexity $O(K \times (n-k)^2)$
> Advantage: Fast K-medoid & less sensitive PAM
> disadvantage: may obtain different results.

PTO

Find clusters using K-means clustering algorithm, if we have several objects (4 types of medicines) and each object have two attributes or features as shown in table below. The mission is to group these objects into K=2 group of medicine based on the two features (pH and weight index)

Object	Attribute 1 (x) weight index	Attribute 2 (y) pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Soln:-

No. of clusters K=2

Initial cluster centres be $C_1 = (1,1)$ and $C_2 (2,1)$

We will check distance b/w data points and all cluster centres.

We will use Euclidean distance formula for finding distance.

Distance $(x, a) = \sqrt{(x-a)^2}$ OR

Distance $[(x,y), (a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$

As given data in pair, we will use second formula of Euclidean dist.

Iteration 1:

following notation: D_1 = distance from cluster $C_1(1,1)$

D_2 = distance from cluster $C_2(2,1)$

- Data point (1,1): $D_1 = 0$ and $D_2 = 1$

  $D_1 = [(1,1), (1,1)] = \sqrt{(1-1)^2 + (1-1)^2} = 0$

  $D_2 = [(1,1), (2,1)] = \sqrt{(1-2)^2 + (1-1)^2} = 1$

  o smallest distance, so point (1,1) belong to cluster C_1
- Data point (2,1): $D_1 = 1$ and $D_2 = 0$ ∴ (2,1) belongs to C_2
- Data point (4,3): $D_1 = 3.6$ and $D_2 = 2.83$ ∴ (4,3) belongs to C_2
- Data point (5,4): $D_1 = 5$ and $D_2 = 4.24$ ∴ (5,4) belongs to C_2

Hence clusters are $C_1 : \{A(1,1)\}$

$C_2 : \{B(2,1), C(4,3), D(5,4)\}$

PTO

Recalculating centr of clusty C_2 as $= \frac{(2+4+5)}{3}, \frac{(1+3+4)}{3} \Rightarrow (3.67, 2.67)$

7 Itsyam 2:

D_1 = Distance form clusty C1 (1,1)

D_2 = Distance form clusty C2 (3.67, 2.67)

- data point (1,1) : $D_1 = 0$ & $D_2 = 3.14$ ∴ (1,1) belongs to C1
- data point (2,1) : $D_1 = 1$ & $D_2 = 2.75$ ∴ (2,1) belongs to C1
- data point (4,3) : $D_1 = 3.6$ & $D_2 = 0.47$ ∴ (4,3) belongs to C2
- data point (5,4) : $D_1 = 6$ & $D_2 = 1.88$ ∴ (5,4) belongs to C2

Recalculating centr of clusty C1 & C2 ∴ $C_1 = (1.5, 1)$ & $C_2 = (4.5, 3.5)$

7 Itsyam 3:

D_1 = Distance form clusty C_1 (1.5, 1)

D_2 = distance form clusty C2 (4.5, 3.5)

- data point (1,1) : $D_1 = 0.5$ & $D_2 = 4.3$ ∴ (1,1) belongs to C1
- data point (2,1) : $D_1 = 0.5$ & $D_2 = 3.54$ ∴ (2,1) belongs to C1
- data point (4,3) : $D_1 = 3.2$ & $D_2 = 0.71$ ∴ (4,3) belongs to C2
- data point (5,4) : $D_1 = 4.6$ & $D_2 = 0.71$ ∴ (5,4) belongs to C2

Recalculating centr of clusty C1 & C2 C_1 ~ C_2 :~

Comparing stsyam 2 & itsyam 3

C1 : { A(1,1), B(2,1) }

C2 : { C(4,3), D(5,4) }

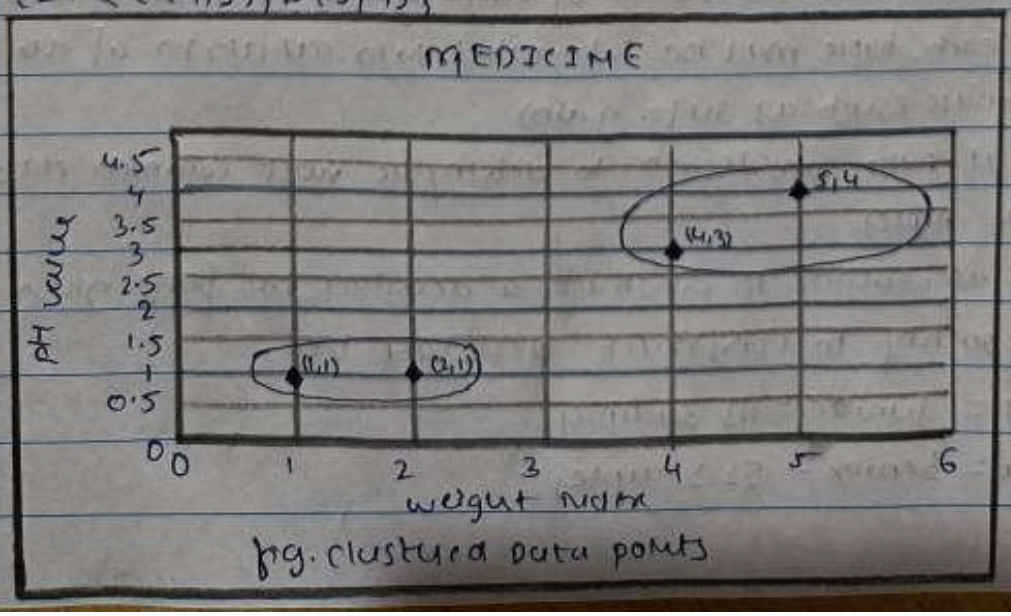


fig. clustered data points

9. The following table consist of training data from the employee DB. The data have been generalized. for example, '31...35" for age represents the age range of 31 to 35. For a given row entry, count represents the no. of data tuples having values for department, status, age and salary given in that row.

department	status	Age	salary (in k)	count
sales	senior	21..35	46K...50K	30
sales	junior	26..30	26K..30K	40
sales	junior	31..35	31K..35K	40
systems	junior	26..30	46K...50K	20
systems	senior	31..35	66K..70K	5
systems	junior	26..30	46K...50K	3
systems	senior	41..45	66K..70K	3
marketing	senior	36..40	46K...50K	10
marketing	junior	31..35	41K..45K	4
secretary	senior	46..50	36K..40K	4
secretary	junior	26..30	26K..30K	6

a) How would you modify the Basic decision tree algorithm to take into consideration the count of each generalized data tuple (row entry)?

The basic decision tree algorithm should be modified as follows to take into consideration the count of each generalized data tuple.

- Count of each tuple must be integrated into calculation of the attribute selection measure (such as info. gain)
- Take count into consideration to determine most common class among the tuples.

b) Use your algorithm to construct a decision tree from given data.

ID3 algorithm to construct decision tree

C1 = status = junior = 113 samples

C2 = status = senior = 52 samples

$\therefore P(C1) = 113/165$ and $P(C2) = 52/165$

(1) Entropy before split for given database D:

$H(D) = \sum_{i=1}^{s} p_i \log(1/p_i)$ $\therefore H(D) = \frac{113}{165}\log_2\frac{165}{113} + \frac{52}{165}\log_2\frac{165}{52}$

$\therefore H(D) = 0.3740 + 0.5250 \Rightarrow 0.899$

(2) splitting attribute

> department as C1: status = junior & C2: status = senior with H.

$\therefore H(\text{department}) = \frac{110}{165} \times H(\text{sales}) + \frac{31}{165} \times H(\text{systems}) + \frac{145}{165} \times H(\text{marketing}) + \frac{10}{165} \times H(\text{operations})$

$= \frac{110}{165} \times 0.8454 + \frac{31}{165} \times 0.8238 + \frac{14}{165} \times 0.8631 + \frac{10}{165} \times 0.9709 = 0.8504$

$\therefore H(\text{department}) = 0.8504$

Gain(department) = H(D) − H(department) = 0.899 − 0.8504 ⇒ 0.0486

> age as C1: status = junior & C2: status = senior with entropy H

$\therefore H(\text{age}) = \frac{20}{165} \times 0 + \frac{49}{165} \times 0 + \frac{79}{165} \times 0.9906 + \frac{10}{165} \times 0 + \frac{3}{165} \times 0 + \frac{4}{165} \times 0 = 0.4743$

Gain(age) = H(D) − H(age) = 0.899 − 0.4743 ⇒ 0.4247

> salary

$\therefore H(\text{salary}) = \frac{46}{165} \times 0 + \frac{40}{165} \times 0 + \frac{4}{165} \times 0 + \frac{4}{165} \times 0 + \frac{63}{165} \times 0.9468 + \frac{8}{165} \times 0$

H(salary) = 0.3615

Gain(salary) = H(D) − H(salary) = 0.899 − 0.3615 ⇒ 0.5375

summary: Gain(department) = 0.0486

Gain(age) = 0.4247 and Gain(salary) = 0.5375

salary has highest attribute gain

∴ it is used as decision tree having 6 possible values.

from the dataset D, the root node has 6 branches

All the branches are given except only one branch,

i.e. 46k...50k is not giving a unique class label attribute

PTO

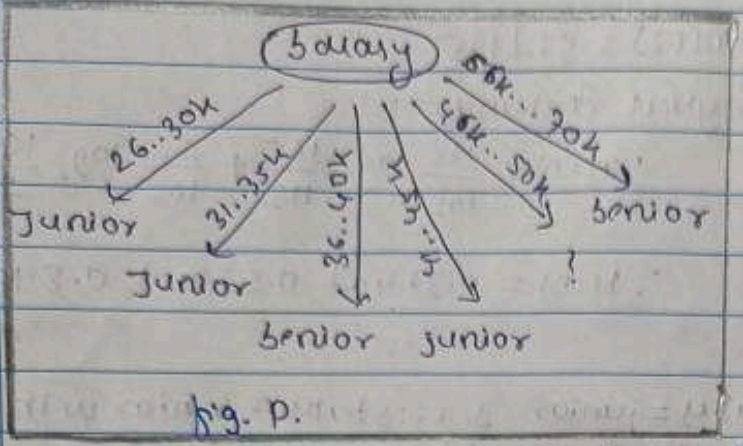


fig. P.

Now consider salary 46k...50k and count the no. of tuples from the original dataset D. Let us denote it as D1.

Let the class label attribute be as follows:

C_1 = Status = junior / salary = 46k... 50k = 23 samples

C_2 = Status = senior / salary = 46k... 50k = 40 samples.

∴ $\in$ / 46k $P(C1) = 23/63$ and $P(C2) = 40/63$

① Entropy before split

$$H(D1) = \sum_{i=1}^{s} p_i \log_2\left(\frac{1}{p_i}\right) = \frac{23}{63}\log_2\frac{63}{23} + \frac{40}{63}\log_2\frac{63}{40} = 0.9468$$

② Splitting attribute

> department

$$H(\text{department}) = \frac{30}{63}\times 0 + \frac{23}{63}\times 0 + \frac{10}{63}\times 0 + \frac{0}{63}\times 0 \Rightarrow 0$$

Gain(department) = H(D1) − H(department) = 0.9468 − 0 = 0.9468

> age

$$H(\text{age}) = \frac{20}{63}\times 0 + \frac{3}{63}\times 0 + \frac{30}{63}\times 0 + \frac{10}{63}\times 0 + \frac{0}{63}\times 0 + \frac{0}{63}\times 0 = 0$$

Gain(age) = H(D1) − H(age) = 0.9468 − 0 ⇒ 0.9468

Summary

Gain(Depart | D1) = 0.9468

Gain(age | D1) = 0.9468

Both attribute have same gain

∴ we choose one of them, & will place it below "Salary = 46k..50k".

we choose department as attribute below "salary = 46k..50k".

from the dataset 1,

PTO

If salary = 46k .. 50k AND department
= sales Then status = junior.
also = systems Then status = junior.
also = marketing Then status = senior.

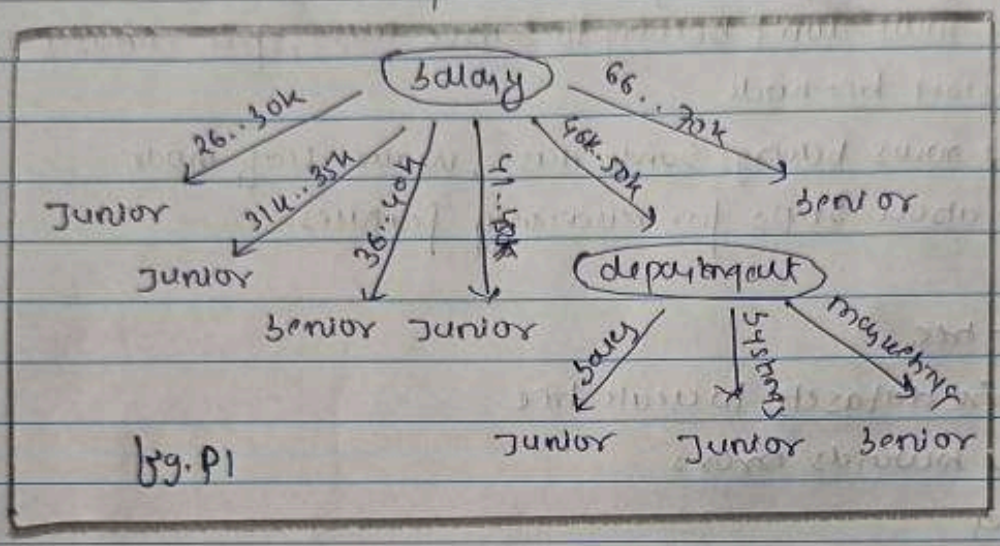


fig.P1

7 **ID3**: ID3 stands for Iterative Dichotomiser 3 and is named such because the algorithm iteratively (repeatedly) dichotomizes (divides) features into two or more groups, at each steps.
Uses a top-down greedy approach to build decision tree.
used to ID3 is for classification problems with nominal features.

7 **Metrics in ID3**
- uses gain and just gain to find best feature.
- Information Gain
- Entropy
- dataset as D
- ie. $Entropy(D) = \sum_{i=1}^{n} p_i \log_2 1/p_i$

⇓
highest info gain selected.
- randomly drawn object is entropy
- Binary classification

.. n = total no. of classes
p_i = probability of class 'i'
ratio of no. of rows to no. of rows.

PTO

Information Gain

$$Gain(A) = Entropy(D) - Entropy(A)$$

> ID3 steps:

1. Calculate Info. gain of each feature
2. consider all rows don't belong to same class, split dataset D
3. make decision tree node
4. if all only rows belongs same class, make leaf node
5. Repeat the above steps for remaining features.

> Advantages:

→ builds short tree
→ Searches entire datasets to create tree
→ less sensitive towards errors

> disadvantages

→ doesn't perform backtracking while searching
→ may converge in locally optimal soln.

10. write a short note on the following:

a) DBSCAN:

Density Based Spatial clustering of application with Noise)
↓
population.

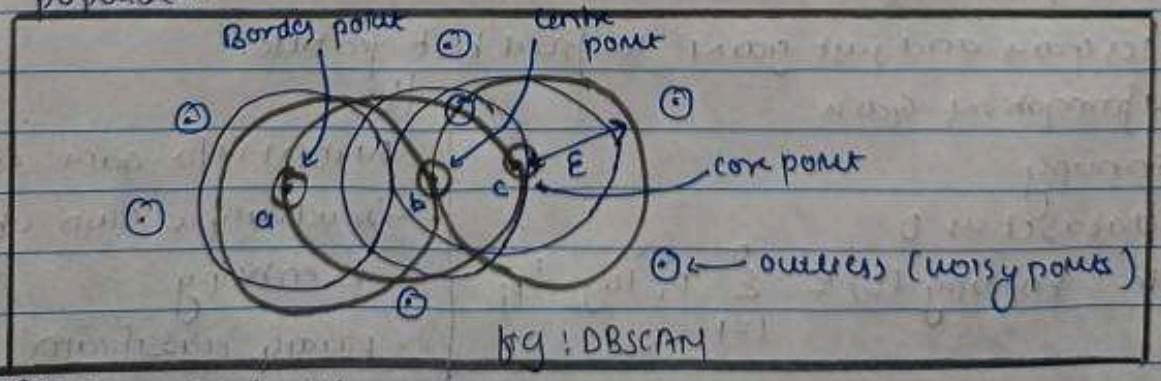


fig: DBSCAN

(1) ε - neighbourhood
(2) min-points amust = 3
(3) core points => strong constituent of the clusters

PTO

④ Border points
⑤ Noise points → Travesbal only

> directly density reachable
Varifying ⑥ neighbour or core point
> possible through core point
> for example c → b, b → a ∴ c → a
a → b, b → c but a ✗ c because a isn't core point
> this method is based on the main notion of density. The basic idea is to continue growing the given clusters as long as the density in the neighbourhood exceeds some threshold, i.e. for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum no. of points.

b) Web structure mining:

> Web structure mining is used for creating a model of web organization. It is the process of analyzing the nodes and connection structure of a website using graph theory.
> There are two things that can be obtained from this:
① the structure of website in website in terms of how it connected to other sites.
② document structure of website itself
> The web structure mining can be used to discover link structure of hyperlink. It is used to identify that the web pages are either linked by information or direct link connection.
> example: page Rank algorithm, CLEVER, HITS).

- **Page Rank:**

> the page rank (PR) algorithm is applicable in web pages.
> page Rank is an algorithm used by Google search to rank websites in their search engine results.
> page rank algorithm is designed to increase the effectiveness of

PTO

search engines and improve that their efficiency.
> It is a way of measuring the importance of website pages
> page Rank is used to prioritize the pages returned from a traditional search engine using keyword searching.
> page rank is calculated based on the no. of pages that point to it.

- Hyperlink-Induced Topic Search (HITS)

> HITS, also known as Hubs and authorities, developed by John Kleinberg is a link analysis algorithm that rates web pages. It was precursor to page Rank
> The idea behind Hubs and Authority stemmed from a particular insight into the creation of web pages when the Internet was originally forming.
> The HITS algorithm treats www as directed graph G(V,E), where V is the set of vertices representing pages and E is a set of edges corresponds to link.
> Based on mutually recursive facts: Hubs point to lots of authorities Authorities are pointed to by lots of hubs.
a) Authority: valuable informative webpages usually pointed to by many hyperlinks.
b) Hub: A webpage that points to many authority pages is itself a resource and is called a hub.
> Authorities and hubs reinforce one another.

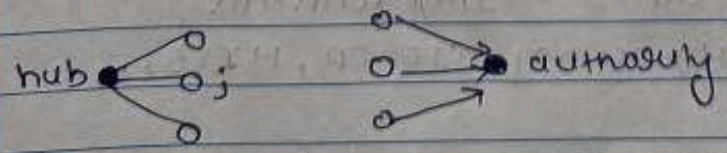


fig: Hub and Authority

A good authority pointed by many good hubs.
A good hub points to many good authorities.