ACEDEMIC YEAR 2022-2023

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

By the end of the educational experience our students will be able to:

1. The Cyber Security graduates are able to gain a thorough understanding of the Cyber Security landscape with its growing threats and vulnerabilities in the world of computing including software and hardware.
2. Attain skills to comprehend and anticipate future challenges and devise methods to meet them and also, be articulate and skilled to convince all the stakeholders.
3. The Cyber Security graduates are able to acquire and demonstrate the ability to use ethical standard tools, practices and technologies for the analysis, design, development, implementation and testing of innovative and optimal Cyber Security solutions without compromising the privacy needs of individual and entities and the security concerns of law enforcement agencies.

Mapping of PSOs to POs:

PSO Number	PO Number
PSO1	PO1, PO2, PO6,
PSO2	PO4, PO9, PO10,
POS3	PO3, PO5, PO7, PO8, PO11 PO12

Dr. Asha Durafe
Program Coordinator
Cyber Security Program

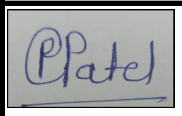
Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Sr. No	Title of Experiment	Page No.	Marks
1	One case study on building Data warehouse/Data Mart	4	13
2	Implementation of all dimension table and fact table based on experiment 1 case study	8	14
3	Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study.	14	14
4	Implementation of Bayesian algorithm	17	13
5	Implementation of Data Discretization (any one) & Visualization (any one).	22	14
6	Perform data Pre-processing taskOne case study on building Data warehouse/Data Mart and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool WEKA.	26	14
7	Implementation of K-means Clustering algorithm	40	13
8	Implementation of Single Link Agglomerative Hierarchical Clustering method	44	14
9	Implementation of Association Rule Mining algorithm (Apriori)	49	15
10	Implementation of Page rank algorithm.	51	13
11	Implement Linear regression using the R tool.	55	15
12	Assignment 01	61	17
13	Assignment 02	71	17

Experiment Number: 1					
Date of Performance:		**29-07-2022**			
Date of Submission:		**05-08-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	2	3	13	

Experiment No. 1

Aim: One case study on building Data warehouse/Data Mart

Laboratory Outcome: CSL 503.1: Design data warehouse and perform various OLAP operations.

Problem Statement: Write detailed problem statement and design dimensional modeling (creation of star and snowflake schema)

Related Theory:

When it comes to star schema vs snowflake schema, it's essential to remember their basic definitions: Star schemas offer an efficient way to organize information in a data warehouse, while Snowflake schemas are a variation of star schemas that allow for more efficient data processing.

Star Schema vs. Snowflake Schema: Differences

1. Star schema dimension tables are not normalized, snowflake schemas dimension tables are normalized.

2. Snowflake schemas will use less space to store dimension tables but are more complex.
3. Star schemas will only join the fact table with the dimension tables, leading to simpler, faster SQL queries.
4. Snowflake schemas have no redundant data, so they're easier to maintain.

5. Snowflake schemas are good for data warehouses, star schemas are better for data marts with simple relationships.
6. Both schemas improve the speed and simplicity of read queries and complex data analysis—especially when dealing with large data sets that pull information from diverse sources.

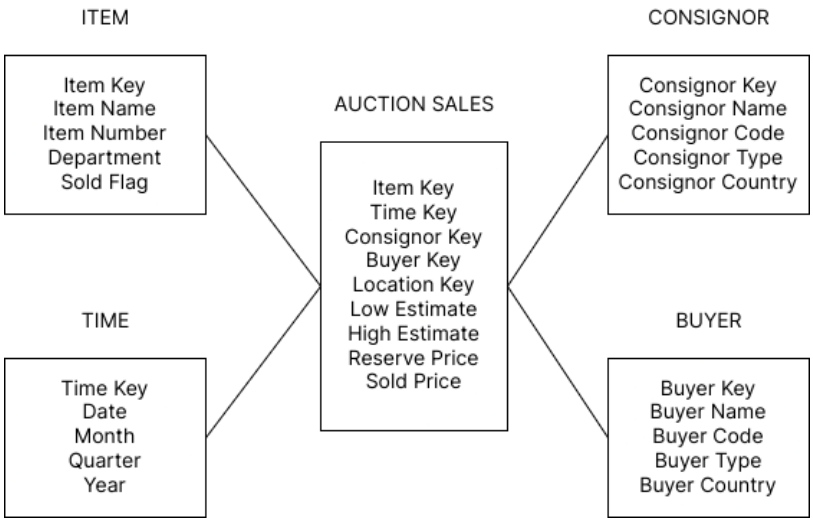
Despite their similarities, star schemas and snowflake schemas have key differences that every data scientist and data engineer needs to understand. To answer the question of star schema vs snowflake schema, we'll begin with an in-depth discussion of star schemas. Then, we'll move into snowflake schemas and explore a tutorial of what makes them unique.

Program Listing And Output:

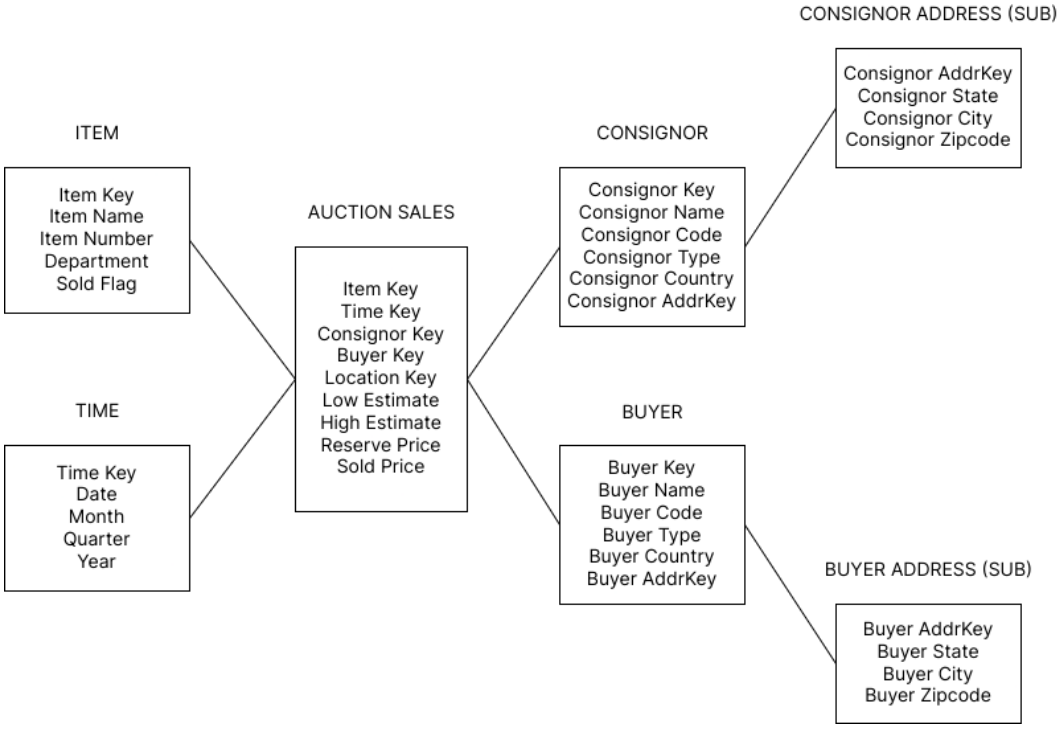
An auction company wants to design a data warehouse to record the sold price of items with their low estimate, high estimate, and reserve price.

Design Star Schema and Snowflake schema for the above problem

statement. Star Schema:



Snowflake Schema:



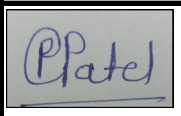
Conclusion: In this experiment, we implemented a case study on building a Data warehouse/Data Mart by designing Star Schema and Snowflake Schema for a problem statement.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 2					
Date of Performance:		**05-08-2022**			
Date of Submission:		**12-08-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
07	02	03	02	14	

Experiment No. 2

Aim: Implementation of all dimension table and fact table based on experiment 1 case study

Laboratory Outcome: CSL 503.1: Design data warehouse and perform various OLAP operations.

Problem Statement: Implement the Star Schema designed from experiment 1 in a MySQL database.

Related Theory:

A reality or fact table's record could be a combination of attributes from totally different dimension tables. The Fact Table or Reality Table helps the user to investigate the business dimensions that helps him in call taking to enhance his business.

On the opposite hand, Dimension Tables facilitate the reality table or fact table to gather dimensions on which the measures need to be taken.

The main difference between the fact table or reality table and the Dimension table is that the dimension table contains attributes on which measures are actually taken.

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table

```sql
CREATE TABLE IF NOT EXISTS Item(
    ItemKey INT PRIMARY KEY,
    ItemName varchar(50),
    ItemNumber INT NOT NULL,
    Department VARCHAR(50),
    SoldFlag INT NOT NULL );

INSERT INTO Item VALUES(001, 'Raspberry Pi', 011, 'Electronics',0);
INSERT INTO Item VALUES(002, 'Free in Freedom', 012, 'Books',1);
INSERT INTO Item VALUES(003, 'One Piece', 013, 'Antique',0);
INSERT INTO Item VALUES(004, 'Demon Dweller', 011, 'Weapon',1);

SELECT * FROM Item;
```

	itemkey [PK] integer	itemname character varying (50)	itemnumber integer	department character varying (50)	soldflag integer
1	1	Raspberry Pi	11	Electronics	0
2	2	Free in Freedom	12	Books	1
3	3	One Piece	13	Antique	0
4	4	Demon Dweller	11	Weapon	1

```sql
CREATE TABLE IF NOT EXISTS TimeTable(
    TimeKey INT PRIMARY KEY,
    _Date varchar(50),
    _Month varchar(50),
    _Quarter VARCHAR(50),
    _Year varchar(50) );

INSERT INTO TimeTable VALUES(001, '10-05-2022', 'May','Q2','2022' );
INSERT INTO TimeTable VALUES(002, '15-05-2022', 'May','Q2','2022');
INSERT INTO TimeTable VALUES(003, '20-05-2022', 'May','Q2','2022');
INSERT INTO TimeTable VALUES(004, '15-05-2022', 'May','Q2','2022');

SELECT * FROM TimeTable;
```

	timekey [PK] integer	_date character varying (50)	_month character varying (50)	_quarter character varying (50)	_year character varying (50)
1	1	10-05-2022	May	Q2	2022
2	2	15-05-2022	May	Q2	2022
3	3	20-05-2022	May	Q2	2022
4	4	15-05-2022	May	Q2	2022

```sql
CREATE TABLE IF NOT EXISTS Consigner(
    TimeKey INT PRIMARY KEY,
    ConsignerName varchar(50),
    ConsignerCode varchar(50),
    ConsignerType VARCHAR(50),
    ConsignerCountry varchar(50) );

INSERT INTO Consigner VALUES(001, 'Bakasta', 'BK11','Business','Clover' );
INSERT INTO Consigner VALUES(002, 'Mugiwara', 'MW12','Business','WindMill');
INSERT INTO Consigner VALUES(003, 'Copy Ninja', 'CN13','Business','Konoha');
INSERT INTO Consigner VALUES(004, 'Kaneki', 'KK14','Business','Tokyo');

SELECT * FROM Consigner;
```

	timekey [PK] integer	consignername character varying (50)	consignercode character varying (50)	consignertype character varying (50)	consignercountry character varying (50)
1	1	Bakasta	BK11	Business	Clover
2	2	Mugiwara	MW12	Business	WindMill
3	3	Copy Ninja	CN13	Business	Konoha
4	4	Kaneki	KK14	Business	Tokyo

```
CREATE TABLE IF NOT EXISTS Buyer(
    BuyerKey INT PRIMARY KEY,
    BuyerName varchar(50),
    CBuyerCode varchar(50),
    BuyerType VARCHAR(50),
    BuyerCountry varchar(50) );

INSERT INTO Buyer VALUES(001, 'Gojo', 'SS011','Business','Alag' );
INSERT INTO Buyer VALUES(002, 'Gojo', 'SS011','Business','Alag' );
INSERT INTO Buyer VALUES(003, 'Gojo', 'SS011','Business','Alag' );
INSERT INTO Buyer VALUES(004, 'Gojo', 'SS011','Business','Alag' );

SELECT * FROM Buyer;
```

	buyerkey [PK] integer	buyername character varying (50)	cbuyercode character varying (50)	buyertype character varying (50)	buyercountry character varying (50)
1	1	Gojo	SS011	Business	Alag
2	2	Gojo	SS011	Business	Alag
3	3	Gojo	SS011	Business	Alag
4	4	Gojo	SS011	Business	Alag

```sql
CREATE TABLE IF NOT EXISTS AuctionSales(
    ItemKey INT NOT NULL,
    TimeKey INT NOT NULL,
    ConsignerKey INT NOT NULL,
    BuyerKey INT NOT NULL,
    FOREIGN KEY (ItemKey) REFERENCES Item(ItemKey),
    FOREIGN KEY (TimeKey) REFERENCES TimeTable(TimeKey),
    FOREIGN KEY (ConsignerKey) REFERENCES Buyer(BuyerKey),

    LowEstimate INT Not NULL,
    HighEstimate INT NOT NULL,
    ReservePrice INT NOT NULL,
    SoldPrice INT NOT NULL );

INSERT INTO AuctionSales VALUES(001, 001, 001, 001, 2500, 5000, 3000, 4000 );
INSERT INTO AuctionSales VALUES(002, 002, 002, 002, 5000, 10000, 6000, 8000);
INSERT INTO AuctionSales VALUES(003, 003, 003, 003, 1250, 2500, 1500, 2000);
INSERT INTO AuctionSales VALUES(004, 004, 004, 004, 7500, 12500, 10000, 11000);

SELECT * FROM AuctionSales;
```

	itemkey integer	timekey integer	consignerkey integer	buyerkey integer	lowestimate integer	highestimate integer	reserveprice integer	soldprice integer
1	1	1	1	1	2500	5000	3000	4000
2	2	2	2	2	5000	10000	6000	8000
3	3	3	3	3	1250	2500	1500	2000
4	4	4	4	4	7500	12500	10000	11000

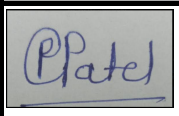
Conclusion: Here, we implemented the Star Schema designed in MySQL database.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 3					
Date of Performance:		**05-08-2022**			
Date of Submission:		**12-08-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
07	02	02	03	14	

Experiment No. 3

Aim: Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study.

Laboratory Outcome: CSL 503.1: Design data warehouse and perform various OLAP operations.

Problem Statement: Perform OLAP Operations on the database implemented in experiment 1 and 2.

Related Theory:

OLAP stands for Online Analytical Processing Server. It is a software technology that allows users to analyze information from multiple database systems at the same time.

It is based on a multi dimensional data model and allows the user to query on multi-dimensional data. OLAP databases are divided into one or more cubes and these cubes are known as Hyper-cubes.

There are four basic analytical operations that can be performed on an OLAP cube:

1. Drill down: In drill-down operation, the less detailed data is converted into highly detailed data.
2. Roll up: It is just opposite of the drill-down operation. It performs aggregation on the OLAP cube.
3. Dice: It selects a sub-cube from the OLAP cube by selecting two or more dimensions.
4. Slice: It selects a single dimension from the OLAP cube which results in a new sub-cube creation.

Program Listing And Output:

1) SLICE

```
auctionstarschema=# -- SLICE
auctionstarschema=# SELECT ItemName, SoldPrice
auctionstarschema-# FROM AuctionSales
auctionstarschema-# INNER JOIN Item ON AuctionSales.ItemKey=Item.ItemKey
auctionstarschema-# WHERE ItemName='One Piece';
 itemname  | soldprice
-----------+-----------
 One Piece |      2000
(1 row)
```

2) DRILLDOWN

```
auctionstarschema=# -- DRILLDOWN
auctionstarschema=# SELECT _Quarter, SUM(SoldPrice)
auctionstarschema-# FROM (AuctionSales NATURAL JOIN Item) JOIN TimeTable
auctionstarschema-# ON AuctionSales.TimeKey = TimeTable.TimeKey
auctionstarschema-# WHERE ItemName = 'One Piece' GROUP BY _Quarter;
 _quarter | sum
----------+------
 Q2       | 2000
(1 row)
```

3) DICE

```
auctionstarschema=# SELECT BuyerName,CBuyerCode
FROM (Buyer INNER JOIN AuctionSales ON Buyer.BuyerKey=AuctionSales.BuyerKey) JOIN TimeTable
ON AuctionSales.TimeKey=TimeTable.TimeKey
WHERE BuyerName = 'Gojo';
 buyername | cbuyercode
-----------+------------
 Gojo      | SS011
 Gojo      | SS011
 Gojo      | SS011
 Gojo      | SS011
```

4) ROLL UP

```
auctionstarschema=# SELECT _Date, _Quarter FROM (AuctionSales NATURAL Join Item) JOIN TimeTable ON
AuctionSales.TimeKey=TimeTable.TimeKey;
   _date    | _quarter
------------+----------
 10-05-2022 | Q2
 15-05-2022 | Q2
 20-05-2022 | Q2
 15-05-2022 | Q2
(4 rows)
```

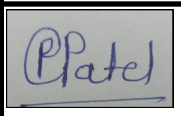
Conclusion: In this experiment, we performed OLAP Operations on the database implemented in experiment 1 and 2.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 4					
Date of Performance:	**12-08-2022**				
Date of Submission:	**26-08-2022**				
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
06	02	03	02	13	

Experiment No. 4

Aim: Implementation of Bayesian algorithm

Laboratory Outcome: CSL 503.2: Implement data mining algorithms like classification.

Problem Statement: Implement the Bayesian algorithm using any programming language of choice.

Related Theory:

Naive Bayes is among one of the very simple and powerful algorithms for classification based on Bayes Theorem with an assumption of independence among the predictors.

The Naive Bayes classifier assumes that the presence of a feature in a class is not related to any other feature.

Naive Bayes is a classification algorithm for binary and multi-class classification problems.

Bayes Theorem

- Based on prior knowledge of conditions that may be related to an event, Bayes theorem describes the probability of the event
- conditional probability can be found this way
- Assume we have a Hypothesis(H) and evidence(E),
  According to Bayes theorem, the relationship between the probability of Hypothesis before getting the evidence represented as P(H) and the probability of the hypothesis after getting the evidence represented as P(H|E) is:

  $$P(H|E) = P(E|H)*P(H)/P(E)$$

- Prior probability = P(H) is the probability before getting the evidence
  Posterior probability = P(H|E) is the probability after getting evidence
- In general,

  $$P(class|data) = (P(data|class) * P(class)) / P(data)$$

Program Listing And Output:

```
# Importing library
import math
import random
import csv


# the categorical class names are changed to numberic data # eg: yes and no
encoded to 1 and 0
def encode_class(mydata):
    classes = []
    for i in range(len(mydata)):
        if mydata[i][-1] not in classes: classes.append(mydata[i][-1])
    for i in range(len(classes)):
        for j in range(len(mydata)):
            if mydata[j][-1] == classes[i]:
                mydata[j][-1] = i
    return mydata


# Splitting the data
def splitting(mydata, ratio):
    train_num = int(len(mydata) * ratio)
    train = []
```

```
    # initially testset will have all the dataset
    test = list(mydata)
    while len(train) < train_num:
        # index generated randomly from range 0
        # to length of testset
        index = random.randrange(len(test))
        # from testset, pop data rows and put it in train
        train.append(test.pop(index))
    return train, test


# Group the data rows under each class yes or
# no in dictionary eg: dict[yes] and dict[no]
def groupUnderClass(mydata):
    dict = {}
    for i in range(len(mydata)):
        if mydata[i][-1] not in dict:
            dict[mydata[i][-1]] = []
        dict[mydata[i][-1]].append(mydata[i])
    return dict


# Calculating Mean
def mean(numbers):
    return sum(numbers) / float(len(numbers))


# Calculating Standard Deviation
def std_dev(numbers):
    avg = mean(numbers)
    variance = sum([pow(x - avg, 2) for x in numbers]) / float(len(numbers) - 1)
    return math.sqrt(variance)


def MeanAndStdDev(mydata):
    info = [(mean(attribute), std_dev(attribute)) for attribute in
            zip(*mydata)]  # eg: list = [ [a, b, c], [m, n, o], [x, y, z]]
    # here mean of 1st attribute =(a + m+x), mean of 2nd attribute = (b + n+y)/3 #
delete summaries of last class
    del info[-1]
    return info


# find Mean and Standard Deviation under each class
def MeanAndStdDevForClass(mydata):
    info = {}
    dict = groupUnderClass(mydata)
    for classValue, instances in dict.items():
        info[classValue] = MeanAndStdDev(instances)
```

```
    return info


# Calculate Gaussian Probability Density Function
def calculateGaussianProbability(x, mean, stdev):
    expo = math.exp(-(math.pow(x - mean, 2) / (2 * math.pow(stdev, 2))))
    return (1 / (math.sqrt(2 * math.pi) * stdev)) * expo


# Calculate Class Probabilities
def calculateClassProbabilities(info, test):
    probabilities = {}
    for classValue, classSummaries in info.items():
        probabilities[classValue] = 1
        for i in range(len(classSummaries)):
            mean, std_dev = classSummaries[i]
            x = test[i]
            probabilities[classValue] *= calculateGaussianProbability(x, mean,
std_dev)
    return probabilities


# Make prediction - the highest probability is the prediction
def predict(info, test):
    probabilities = calculateClassProbabilities(info, test)
    bestLabel, bestProb = None, -1
    for classValue, probability in probabilities.items():
        if bestLabel is None or probability > bestProb:
            bestProb = probability
        bestLabel = classValue
    return bestLabel


# returns predictions for a set of examples
def getPredictions(info, test):
    predictions = []
    for i in range(len(test)):
        result = predict(info, test[i])
        predictions.append(result)
    return predictions


# Accuracy score
def accuracy_rate(test, predictions):
    correct = 0
    for i in range(len(test)):
        if test[i][-1] == predictions[i]:
            correct += 1
    return (correct / float(len(test))) * 100.0
```

```
# add the data path in your system
filename = r'filedata.csv'

# load the file and store it in mydata list
mydata = csv.reader(open(filename, "rt"))
mydata = list(mydata)
mydata = encode_class(mydata)
for i in range(len(mydata)):
   mydata[i] = [float(x) for x in mydata[i]]

# split ratio = 0.7
# 70% of data is training data and 30% is test data used for testing
ratio = 0.7
train_data, test_data = splitting(mydata, ratio)
print('Total number of examples are: ', len(mydata))
print('Out of these, training examples are: ', len(train_data))
print("Test examples are: ", len(test_data))

# prepare model
info = MeanAndStdDevForClass(train_data)

# test model
predictions = getPredictions(info, test_data)
accuracy = accuracy_rate(test_data, predictions)
print("Accuracy of your model is: ", accuracy)
```

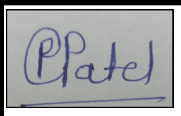
Conclusion: In this experiment, we implemented the Navie Bayes Algorithm using Python.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 5					
Date of Performance:		**12-08-2022**			
Date of Submission:		**26-08-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
07	02	03	03	14	

Experiment No. 5

Aim: Implementation of Data Discretization & Visualization.

Laboratory Outcome: CSL 503.2: Implement data mining algorithms like classification.

Problem Statement: Implement Data Discretization & Visualize it using any of the visualizations in Python Jupyter Notebook.

Related Theory:

Data discretization:

It refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data becomes easy. In other words, data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss.

There are two forms of data discretization: first is supervised discretization, and the second is unsupervised discretization. Supervised discretization refers to a method in which the class data is used. Unsupervised discretization refers to a method depending upon the way which operation proceeds. It means it works on the top-down splitting strategy and bottom-up merging strategy.

Data visualization:

It is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.

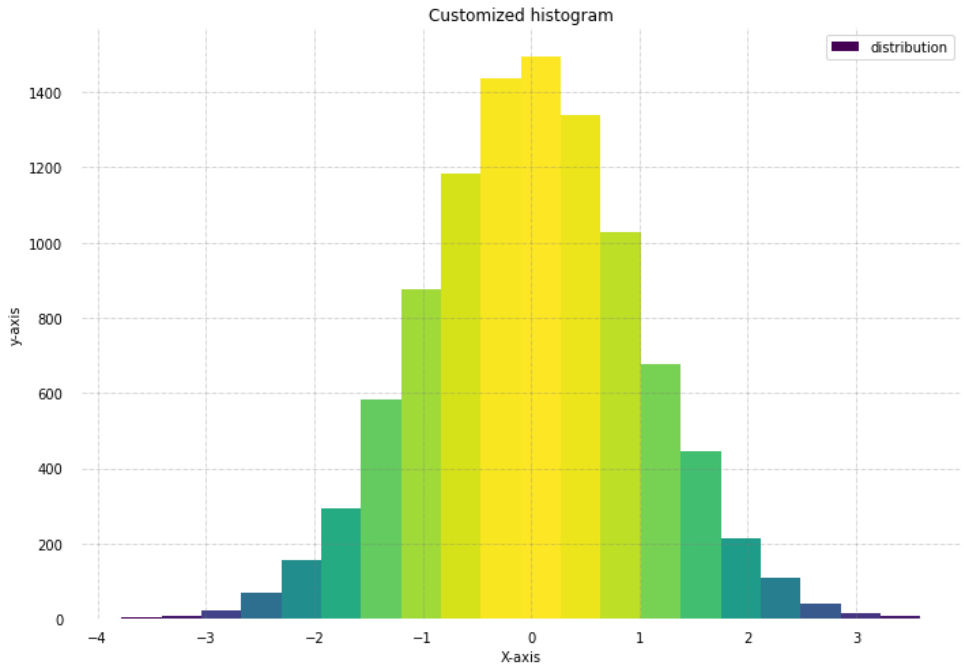
Histogram: is basically used to represent data provided in a form of some groups.It is an accurate method for the graphical representation of numerical data distribution.It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

Program Listing And Output:

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter

# Creating dataset
np.random.seed(23685752)
N_points = 10000
n_bins = 20

# Creating distribution
x = np.random.randn(N_points)
y = .8 ** x + np.random.randn(10000) + 25
legend = ['distribution']

# Creating histogram
fig, axs = plt.subplots(1, 1,
                        figsize =(10, 7),
                        tight_layout = True)


# Remove axes splines
for s in ['top', 'bottom', 'left', 'right']:
    axs.spines[s].set_visible(False)

# Remove x, y ticks
axs.xaxis.set_ticks_position('none')
axs.yaxis.set_ticks_position('none')
```

```
# Add padding between axes and labels

axs.xaxis.set_tick_params(pad = 5)
axs.yaxis.set_tick_params(pad = 10)

# Add x, y gridlines
axs.grid(b = True, color ='grey',
        linestyle ='-.', linewidth = 0.5,
        alpha = 0.6)

# Creating histogram
N, bins, patches = axs.hist(x, bins = n_bins)

# Setting color
fracs = ((N**(1 / 5)) / N.max())
norm = colors.Normalize(fracs.min(), fracs.max())

for thisfrac, thispatch in zip(fracs, patches):
   color = plt.cm.viridis(norm(thisfrac))
   thispatch.set_facecolor(color)

# Adding extra features
plt.xlabel("X-axis")
plt.ylabel("y-axis")
plt.legend(legend)
plt.title('Customized histogram')

# Show plot plt.show()
```

Output():



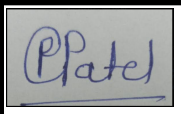
Conclusion: We successfully implemented Data Discretization & Visualization.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 6					
Date of Performance:		**26-08-2022**			
Date of Submission:		**16-09-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
02	02	02	03	14	

Experiment No. 6

Aim: Perform data Pre-processing task and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool WEKA.

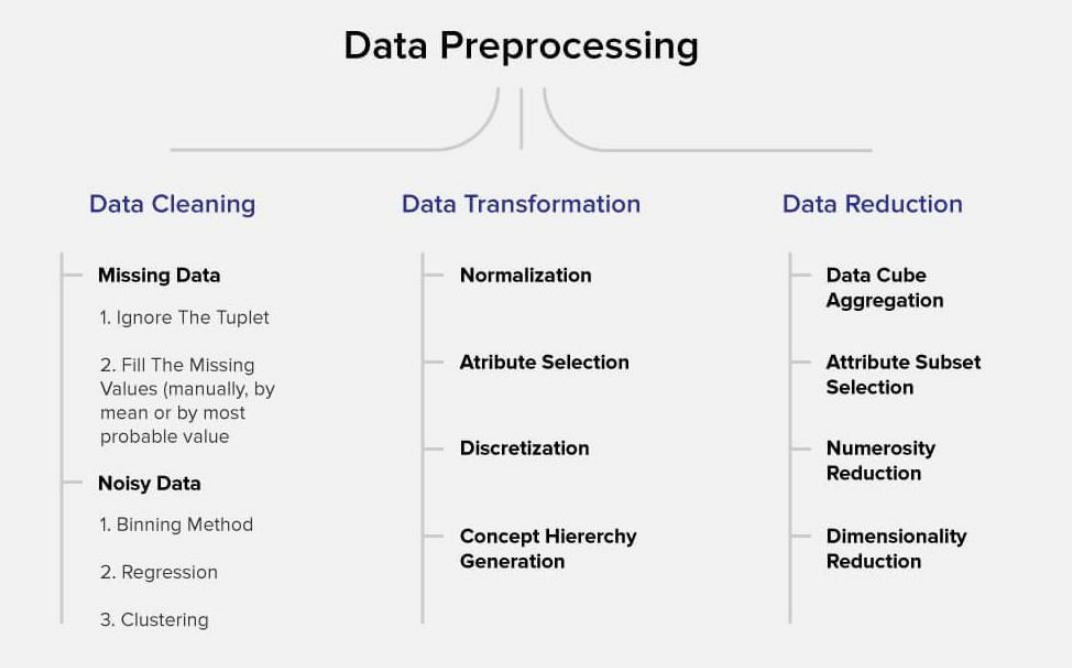
Laboratory Outcome: CSL 503.2: Implement data mining algorithms like classification.
CSL 503.3: Implement clustering algorithms on a given set of data samples.
CSL 503.4: Implement Association rule mining and web mining algorithms.

Problem Statement: Perform data Pre-processing task and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool WEKA.

Related Theory:

Preprocessing in Data Mining:

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.



Data Classification :

Data classification is broadly defined as the process of organizing data by relevant categories so that it may be used and protected more efficiently. On a basic level, the classification process makes data easier to locate and retrieve. Data classification is of particular importance when it comes to risk management, compliance, and data security.

Data classification involves tagging data to make it easily searchable and trackable. It also eliminates multiple duplications of data, which can reduce storage and backup costs while speeding up the search process. Though the classification process may sound highly technical, it is a topic that should be understood by your organization's leadership.

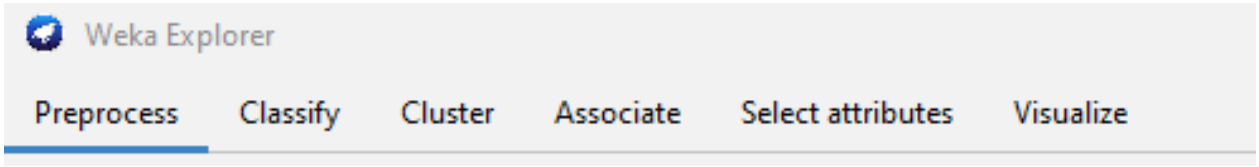
Program Listing & Output:

Weka App GUI:

The GUI Chooser application allows you to run five different types of applications as listed here

- Explorer
- Experimenter
- KnowledgeFlow
- Workbench
- Simple CLI

When you click on the **Explorer button** in the Applications selector, it opens the following screen



At the very top of the window, just below the title bar, is a row of tabs.

The tabs are as follows:

Preprocess: Choose and modify the data being acted on.

Classify: Train and test learning schemes that classify or perform regression.

Cluster: Learn clusters for the data.

Associate: Learn association rules for the data.

Select attributes: Select the most relevant attributes in the data.

Visualize: View an interactive 2D plot of the data.

Loading Data:

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

Open file: This brings up a dialog box allowing you to browse for the data file on the local file system.

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

Open DB: Reads data from a database. (Note that to make this work you might have to edit the file in weka/experiment/DatabaseUtils.props.)

Experimenter

Allows users to execute different experimental variations on data sets.

Knowledge Flow

Explorer with drag and drop functionality.

Supports incremental learning from previous results.

Simple CLI

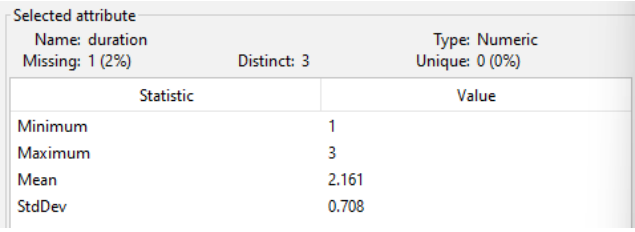
Command Line Interface. Simple interface for executing commands from a terminal.

Workbench

Combines all GUI interfaces into one.

Program Listing And Output:

Before Pre-Processing :

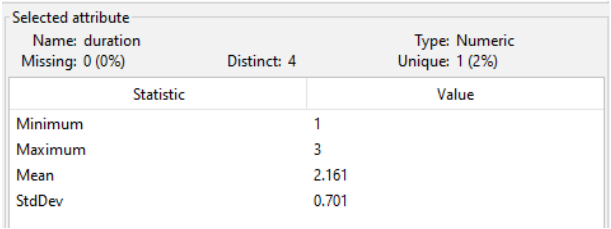


Selected attribute

Name: duration Type: Numeric
Missing: 1 (2%) Distinct: 3 Unique: 0 (0%)

Statistic	Value
Minimum	1
Maximum	3
Mean	2.161
StdDev	0.708

After Pre-Processing :



Selected attribute

Name: duration Type: Numeric
Missing: 0 (0%) Distinct: 4 Unique: 1 (2%)

Statistic	Value
Minimum	1
Maximum	3
Mean	2.161
StdDev	0.701

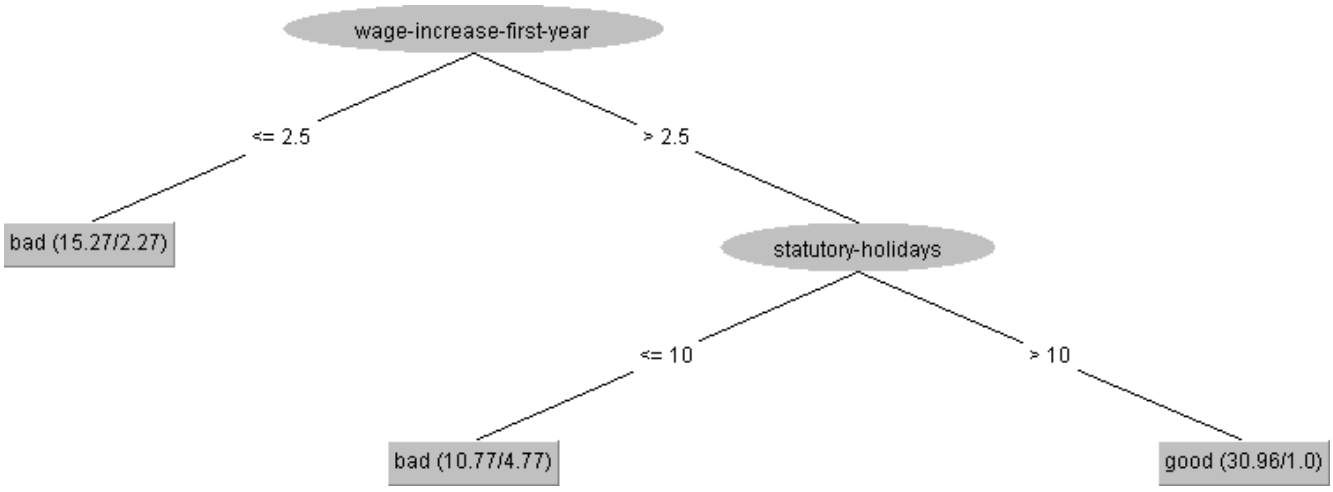
Classifier (J48):

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     labor-neg-data
Instances:    57
Attributes:   17
              duration
              wage-increase-first-year
              wage-increase-second-year
              wage-increase-third-year
              cost-of-living-adjustment
              working-hours
              pension
              standby-pay
              shift-differential
              education-allowance
              statutory-holidays
              vacation
              longterm-disability-assistance
              contribution-to-dental-plan
              bereavement-assistance
              contribution-to-health-plan
              class
```

```
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
------------------

wage-increase-first-year <= 2.5: bad (15.27/2.27)
wage-increase-first-year > 2.5
|   statutory-holidays <= 10: bad (10.77/4.77)
|   statutory-holidays > 10: good (30.96/1.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds
```

```
=== Summary ===

Correctly Classified Instances          50               87.7193 %
Incorrectly Classified Instances         7               12.2807 %
Kappa statistic                          0.745
Mean absolute error                      0.195
Root mean squared error                  0.304
Relative absolute error                 42.6664 %
Root relative squared error             63.6959 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                 0.950    0.162    0.760      0.950    0.844      0.758    0.918     0.809
                 0.838    0.050    0.969      0.838    0.899      0.758    0.918     0.933
Weighted Avg.    0.877    0.089    0.896      0.877    0.880      0.758    0.918     0.890

=== Confusion Matrix ===

  a  b   <-- classified as
 19  1 |  a = bad
  6 31 |  b = good
```

J48 tree:



Naive Bayes:

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     labor-neg-data-weka.filters.unsupervised.attribute.ReplaceMissingValues
Instances:    57
Attributes:   17
              duration
              wage-increase-first-year
              wage-increase-second-year
              wage-increase-third-year
              cost-of-living-adjustment
              working-hours
              pension
              standby-pay
              shift-differential
              education-allowance
              statutory-holidays
              vacation
              longterm-disability-assistance
              contribution-to-dental-plan
              bereavement-assistance
              contribution-to-health-plan
              class
Test mode:    evaluate on training data
```

```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                                 Class
Attribute                          bad    good
                                (0.36)  (0.64)
=================================================
duration
  mean                              2  2.1622
  std. dev.                    0.4714  0.4494
  weight sum                       20      37
  precision                    0.6667  0.6667

wage-increase-first-year
  mean                         2.7059  4.3959
  std. dev.                     0.835  1.1563
  weight sum                       20      37
  precision                    0.2941  0.2941
```

Classifier output

```
  none                            8.0     3.0
  half                           14.0    23.0
  full                            1.0    14.0
  [total]                        23.0    40.0

bereavement-assistance
  yes                            18.0    38.0
  no                              4.0     1.0
  [total]                        22.0    39.0

contribution-to-health-plan
  none                            9.0     1.0
  half                            3.0     8.0
  full                           11.0    31.0
  [total]                        23.0    40.0



Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds
```

```
=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances          55               96.4912 %
Incorrectly Classified Instances         2                3.5088 %
Kappa statistic                          0.923
Mean absolute error                      0.0351
Root mean squared error                  0.1125
Relative absolute error                  7.68   %
Root relative squared error             23.5624 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                 0.950    0.027    0.950      0.950    0.950      0.923    0.999     0.998
                 0.973    0.050    0.973      0.973    0.973      0.923    0.999     0.999
Weighted Avg.    0.965    0.042    0.965      0.965    0.965      0.923    0.999     0.999

=== Confusion Matrix ===

  a  b   <-- classified as
 19  1 |  a = bad
  1 36 |  b = good
```

Cluster:

Clusterer output

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 137.79496140158423

Initial starting points (random):

Cluster 0: 1,5.7,3.971739,3.913333,none,40,empl_contr,7.444444,4,no,11,generous,ye
Cluster 1: 1,2,3.971739,3.913333,tc,40,ret_allw,4,0,no,11,generous,no,none,no,none

Missing values globally replaced with mean/mode
```

```
Final cluster centroids:
                                                Cluster#
Attribute                               Full Data           0           1
                                           (57.0)      (48.0)      (9.0)
=============================================================================
duration                                   2.1607      2.2533      1.6667
wage-increase-first-year                   3.8036      3.9834      2.8444
wage-increase-second-year                  3.9717      4.0209      3.7097
wage-increase-third-year                   3.9133      3.9511      3.7119
cost-of-living-adjustment                    none        none        none
working-hours                             38.0392     37.7541     39.5599
pension                                 empl_contr  empl_contr        none
standby-pay                                7.4444      7.7431      5.8519
shift-differential                          4.871      5.2298       2.957
education-allowance                            no          no          no
statutory-holidays                        11.0943      11.237     10.3333
vacation                            below_average below_average below_average
longterm-disability-assistance                yes         yes          no
contribution-to-dental-plan                  half        half        none
bereavement-assistance                        yes         yes         yes
contribution-to-health-plan                  full        full        none
class                                        good        good         bad



Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      48 ( 84%)
1       9 ( 16%)
```

Associator:

```
Scheme:       weka.associations.FilteredAssociator -F "weka.filters.MultiFilter -F \"weka.filters.unsupervised
Relation:     labor-neg-data-weka.filters.unsupervised.attribute.ReplaceMissingValues
Instances:    57
Attributes:   17
              duration
              wage-increase-first-year
              wage-increase-second-year
              wage-increase-third-year
              cost-of-living-adjustment
              working-hours
              pension
              standby-pay
              shift-differential
              education-allowance
              statutory-holidays
              vacation
              longterm-disability-assistance
              contribution-to-dental-plan
              bereavement-assistance
              contribution-to-health-plan
              class
```

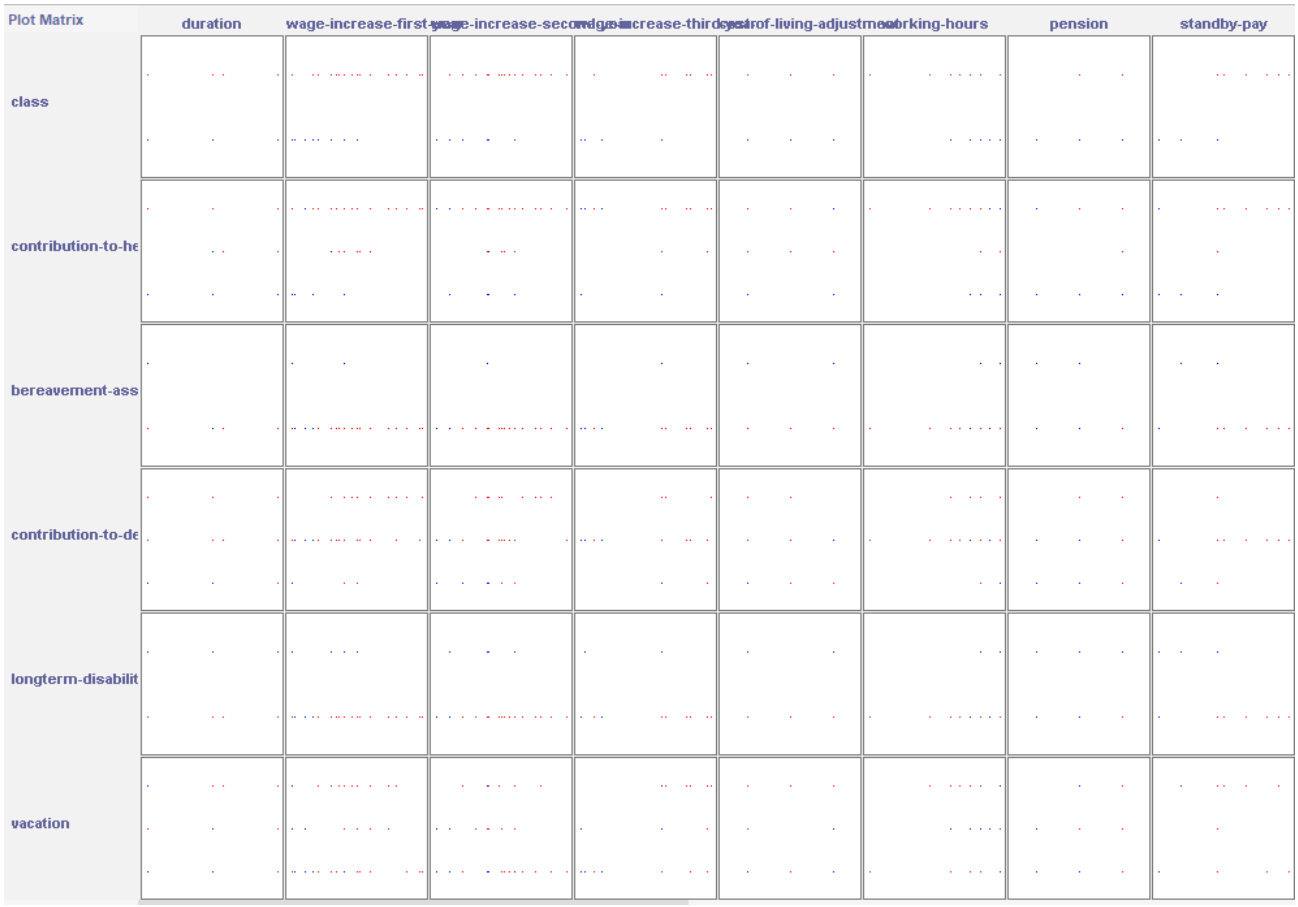
Attribute Selection:

Attribute selection output

```
              wage-increase-third-year
              cost-of-living-adjustment
              working-hours
              pension
              standby-pay
              shift-differential
              education-allowance
              statutory-holidays
              vacation
              longterm-disability-assistance
              contribution-to-dental-plan
              bereavement-assistance
              contribution-to-health-plan
              class
Evaluation mode:    evaluate on all training data
```

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 17 class):
        Information Gain Ranking Filter

Ranked attributes:
 0.38571   7 pension
 0.3068    2 wage-increase-first-year
 0.24487  16 contribution-to-health-plan
 0.24447  13 longterm-disability-assistance
 0.22977  14 contribution-to-dental-plan
 0.19494  11 statutory-holidays
 0.19467   3 wage-increase-second-year
 0.12353   4 wage-increase-third-year
 0.11327   8 standby-pay
 0.08349  15 bereavement-assistance
 0.07592   5 cost-of-living-adjustment
 0.07182  12 vacation
 0.00178  10 education-allowance
 0         9 shift-differential
 0         6 working-hours
 0         1 duration

Selected attributes: 7,2,16,13,14,11,3,4,8,15,5,12,10,9,6,1 : 16
```

Attribute Visualizer:



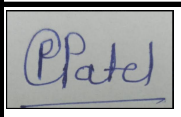
Conclusion: In this experiment, we performed data Pre-processing tasks and demonstrated Classification, Clustering, Association algorithm on data sets using data mining tool WEKA.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 7					
Date of Performance:		**26-08-2022**			
Date of Submission:		**16-09-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
06	02	03	02	13	

Experiment No. 7

Aim: Implementation of K-means Clustering Algorithm

Laboratory Outcome: CSL 503.3: Implement clustering algorithms on a given set of data samples.

Problem Statement: Implement the K-means Clustering Algorithm using the programming language of your choice.

Related Theory:

We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups.

To achieve this, we will use the kMeans algorithm; an unsupervised learning algorithm. 'K' in the name of the algorithm represents the number of groups/clusters we want to classify our items into.

(It will help if you think of items as points in an n-dimensional space). The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, we will use the euclidean distance as measurement.

The algorithm works as follows:

1. First, we initialize k points, called means or cluster centroids, randomly.
2. We categorize each item to its closest mean and we update the mean's coordinates, which are the averages of the items categorized in that cluster so far.
3. We repeat the process for a given number of iterations and at the end, we have our clusters.

The "points" mentioned above are called means because they are the mean values of the items categorized in them. To initialize these means, we have a lot of options.

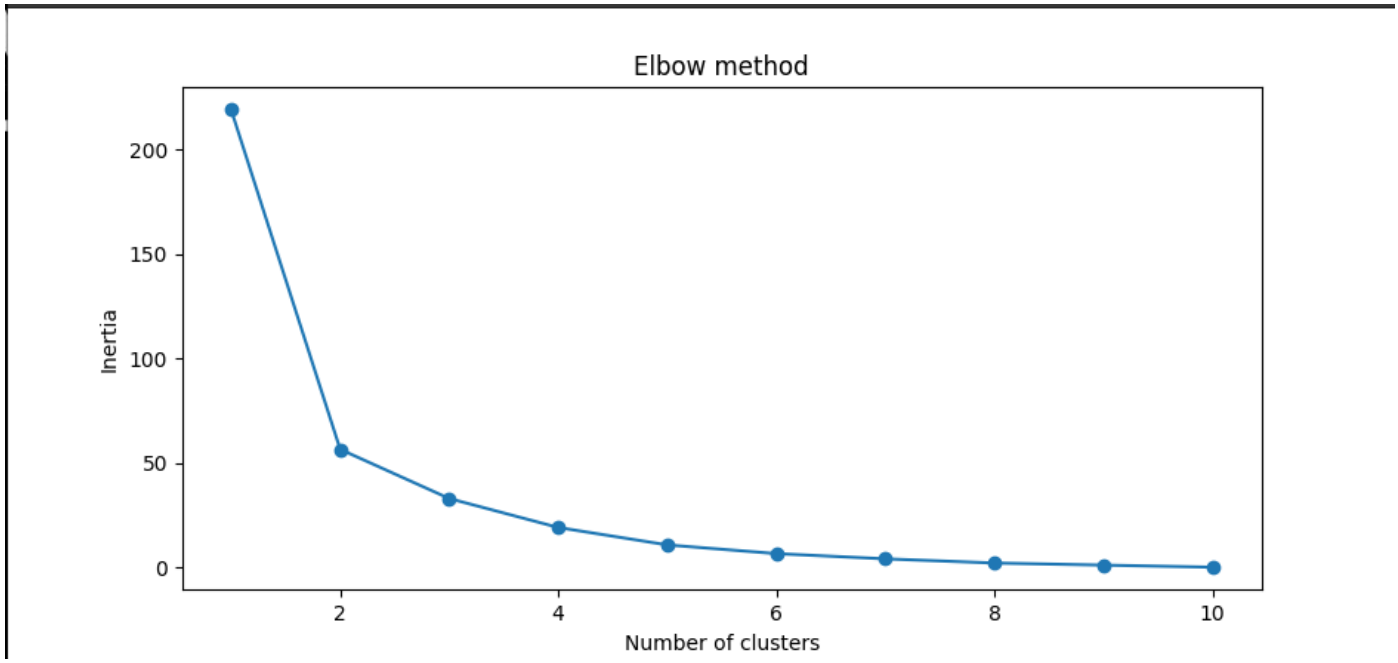
An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set (if for a feature x the items have values in [0,3], we will initialize the means with values for x at [0,3]).

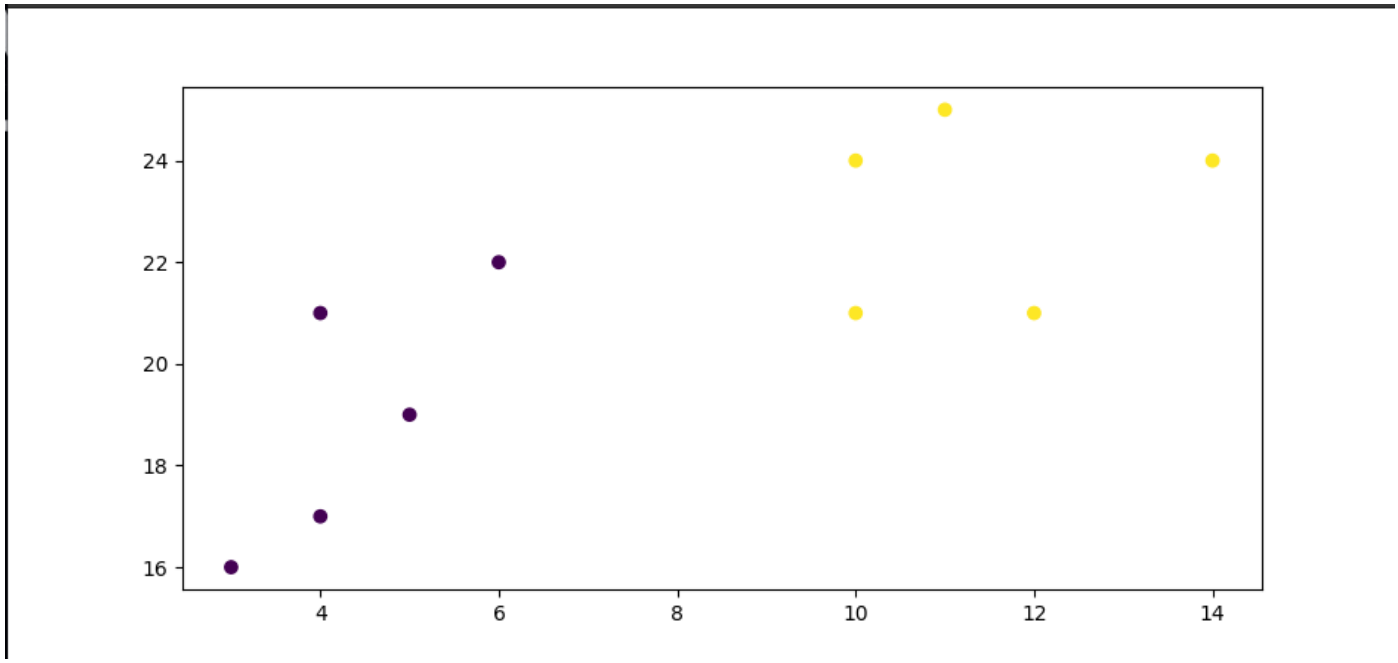
Program Listing And Output:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

x = [4, 5, 10, 4, 3, 11, 14 , 6, 10, 12]
y = [21, 19, 24, 17, 16, 25, 24, 22, 21, 21]

data = list(zip(x, y))
print(data)


inertias = []

for i in range(1,11):
   kmeans = KMeans(n_clusters=i)
   kmeans.fit(data)
   inertias.append(kmeans.inertia_)

plt.plot(range(1,11), inertias, marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.show()

kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

plt.scatter(x, y, c=kmeans.labels_)
```

```
plt.show()
```

Output :





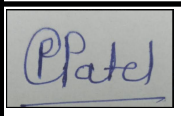
Conclusion: In this experiment, we successfully implemented the K-means algorithm in Python.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 8					
Date of Performance:	**16-09-2022**				
Date of Submission:	**30-09-2022**				
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
07	02	02	03	14	

Experiment No. 8

Aim: Implementation of Single Link Agglomerative Hierarchical Clustering method

Laboratory Outcome: CSL 503.3: Implement clustering algorithms on a given set of data samples.

Problem Statement: Write a Python Code in Jupyter Notebook to demonstrate the Implementation of Single Link Agglomerative Hierarchical Clustering method.

Related Theory:

Agglomerative Clustering:

Also known as bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering.

This clustering algorithm does not require us to pre-specify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then

successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

The process of Hierarchical Clustering involves either clustering sub-clusters (data points in the first iteration) into larger clusters in a bottom-up manner or dividing a larger cluster into smaller sub-clusters in a top-down manner.

During both the types of hierarchical clustering, the distance between two sub-clusters needs to be computed.

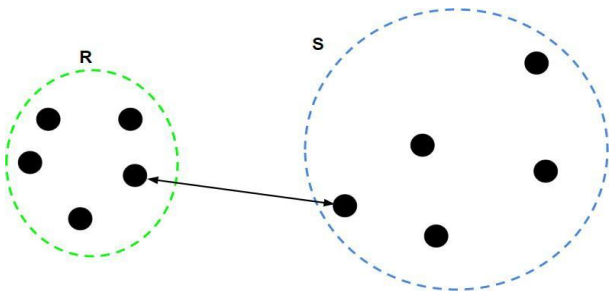
The different types of linkages describe the different approaches to measure the distance between two sub-clusters of data points.

Single Linkage:

For two clusters R and S, the single linkage returns the minimum distance between two points i and j such that i belongs to R and j belongs to S.

$$L(R,S) = max(D(i,j)), i \epsilon R, j \epsilon S$$

This is illustrated in the figure below:

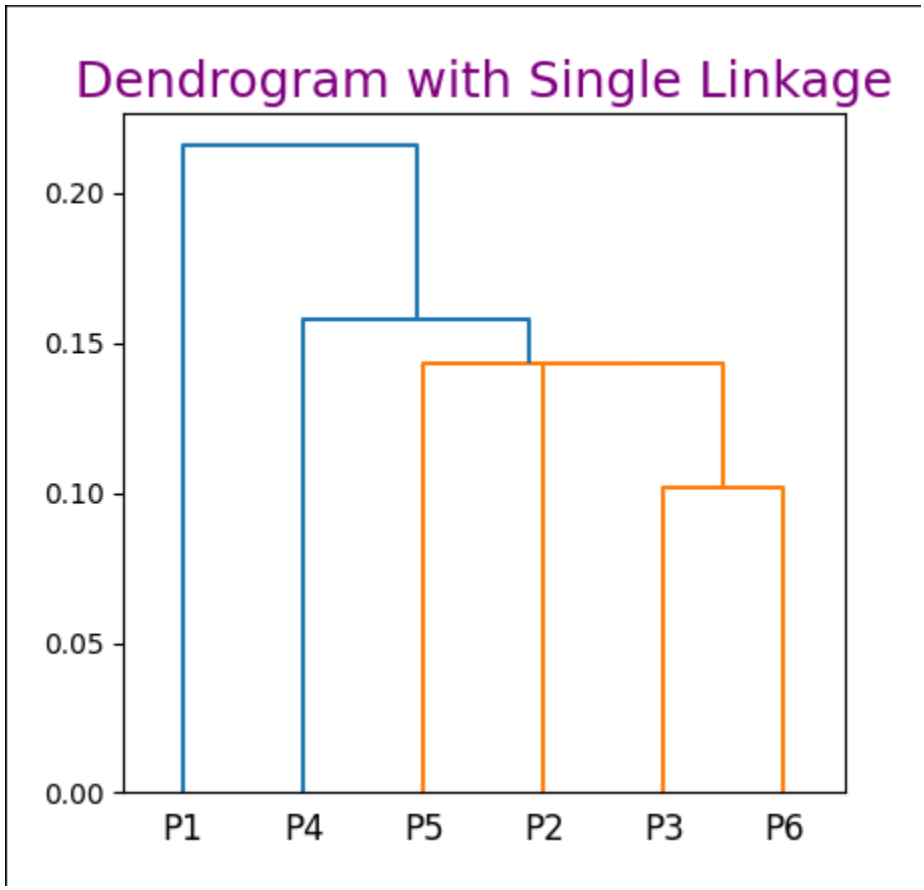


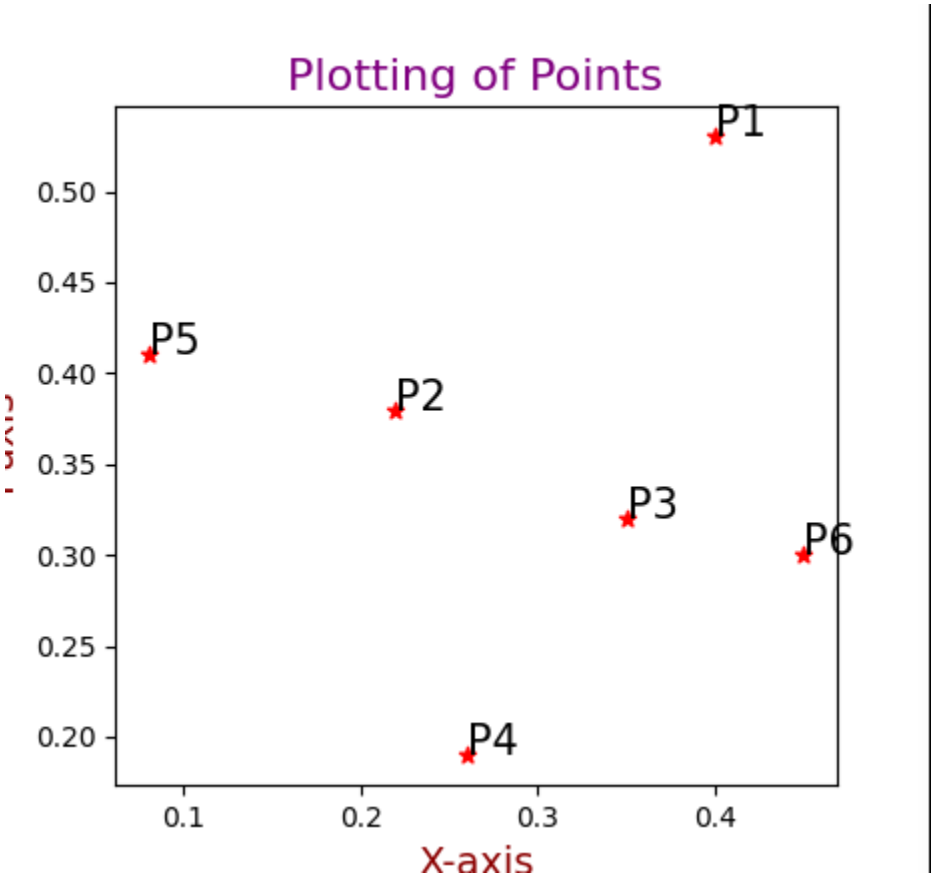
Program Listing And Output:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.cluster.hierarchy as shc
from scipy.spatial.distance import squareform, pdist

# X-axis of point
x = [0.40, 0.22, 0.35, 0.26, 0.08, 0.45]

# Y-axis of point
y = [0.53, 0.38, 0.32, 0.19, 0.41, 0.30]

point = ['P1', 'P2', 'P3', 'P4', 'P5', 'P6']
data = pd.DataFrame({'Point': point, 'x': np.round(x, 2),
                     'y': np.round(y, 2)})
data = data.set_index('Point')

# Printing all points
print(data)
print('\n\n')

# Plotting the points
plt.figure(figsize=(8, 5))
plt.scatter(data['x'], data['y'], c='r', marker='*')
plt.xlabel('X-axis', fontsize=14, color='darkred')
plt.ylabel('Y-axis', fontsize=14, color='darkred')
plt.title('Plotting of Points', fontsize=16, color='purple')

for j in data.itertuples():
   plt.annotate(j.Index, (j.x, j.y), fontsize=15)
dist = pd.DataFrame(squareform(pdist(data[['x', 'y']]), 'euclidean'),
                    columns=data.index.values, index=data.index.values)  #
Displaying the dendrogram
plt.figure(figsize=(12, 5))
plt.title("Dendrogram with Single Linkage", fontsize=18, color='purple')

dend = shc.dendrogram(shc.linkage(data[['x', 'y']], method='single'),
labels=data.index)

plt.show()
```

Output() :

Point	x	y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30





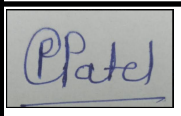
Conclusion: In this experiment, we implemented the Single Link Agglomerative Hierarchical Clustering method.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 9					
Date of Performance:	**16-09-2022**				
Date of Submission:	**30-09-2022**				
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
07	02	03	03	15	

Experiment No. 9

Aim: Implementation of Association Rule Mining algorithm (Apriori)

Laboratory Outcome: CSL 503.4: Implement Association rule mining and web mining algorithms.

Problem Statement: Write a Python Code in Jupyter Notebook to demonstrate the Implementation of Association Rule Mining algorithm

Related Theory:

Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule learning that analyzes that people who bought product A also bought product B.

The primary objective of the apriori algorithm is to create the association rule between different objects. The association rule describes how two or more objects are related to one another. Apriori algorithm is also called frequent pattern mining. Generally, you operate the Apriori algorithm on a database that consists of a huge number of transactions.

Program Listing And Output:

```python
import numpy as np
import pandas as pd
from apyori import apriori

store_data = pd.read_csv('store_data.csv', header=None)
print("Dataset :-\n", store_data)
print("\nShape of Dataset :", store_data.shape)
records = []
for i in range(0, 10):
    records.append([str(store_data.values[i,j]) for j in range(0, 6)])

association_rules = apriori(records, min_support=0.5, min_confidence=0.9,
                            min_lift=1.3, min_length=2)

association_results = list(association_rules)

print("\nNumber of Association Results :",
      len(association_results))
print("\n" + str(association_results))
```

Output():

```
uwu@pop-os:~/clgtp/socket_prog/server$ python apriori.py
Dataset :-
                  0                  1            2                  3              4                   5  ...            14      15                 16               17       18         19
0           shrimp            almonds      avocado     vegetables mix   green grapes   whole weat flour  ...  mineral water  salmon  antioxydant juice  frozen smoothie  spinach  olive oil
1          burgers          meatballs         eggs                NaN            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
2          chutney                NaN          NaN                NaN            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
3           turkey            avocado          NaN                NaN            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
4    mineral water               milk   energy bar  whole wheat rice      green tea                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
...            ...                ...          ...                ...            ...                 ...  ...           ...     ...                ...              ...      ...        ...
7496        butter         light mayo  fresh bread                NaN            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
7497       burgers  frozen vegetables         eggs       french fries      magazines           green tea  ...           NaN     NaN                NaN              NaN      NaN        NaN
7498       chicken                NaN          NaN                NaN            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
7499     escalope          green tea          NaN                NaN            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN
7500          eggs    frozen smoothie  yogurt cake     low fat yogurt            NaN                 NaN  ...           NaN     NaN                NaN              NaN      NaN        NaN

[7501 rows x 20 columns]

Shape of Dataset : (7501, 20)

Number of Association Results : 0

[]
```

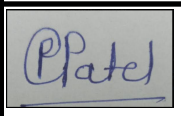
Conclusion: In this Experiment, we implemented the Association Rule Mining algorithm (Apriori).

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 10					
Date of Performance:	**30-09-2022**				
Date of Submission:	**07-10-2022**				
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
06	02	02	03	13	

Experiment No. 10

Aim: Implementation of Page rank algorithm.

Laboratory Outcome: CSL 503.4: Implement Association rule mining and web mining algorithms.

Problem Statement: Write a Python Code in Jupyter Notebook to demonstrate the Implementation of PageRank Algorithm

Related Theory:

PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages.

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

It is not the only algorithm used by Google to order search engine results, but it is the first algorithm that was used by the company, and it is the best-known.

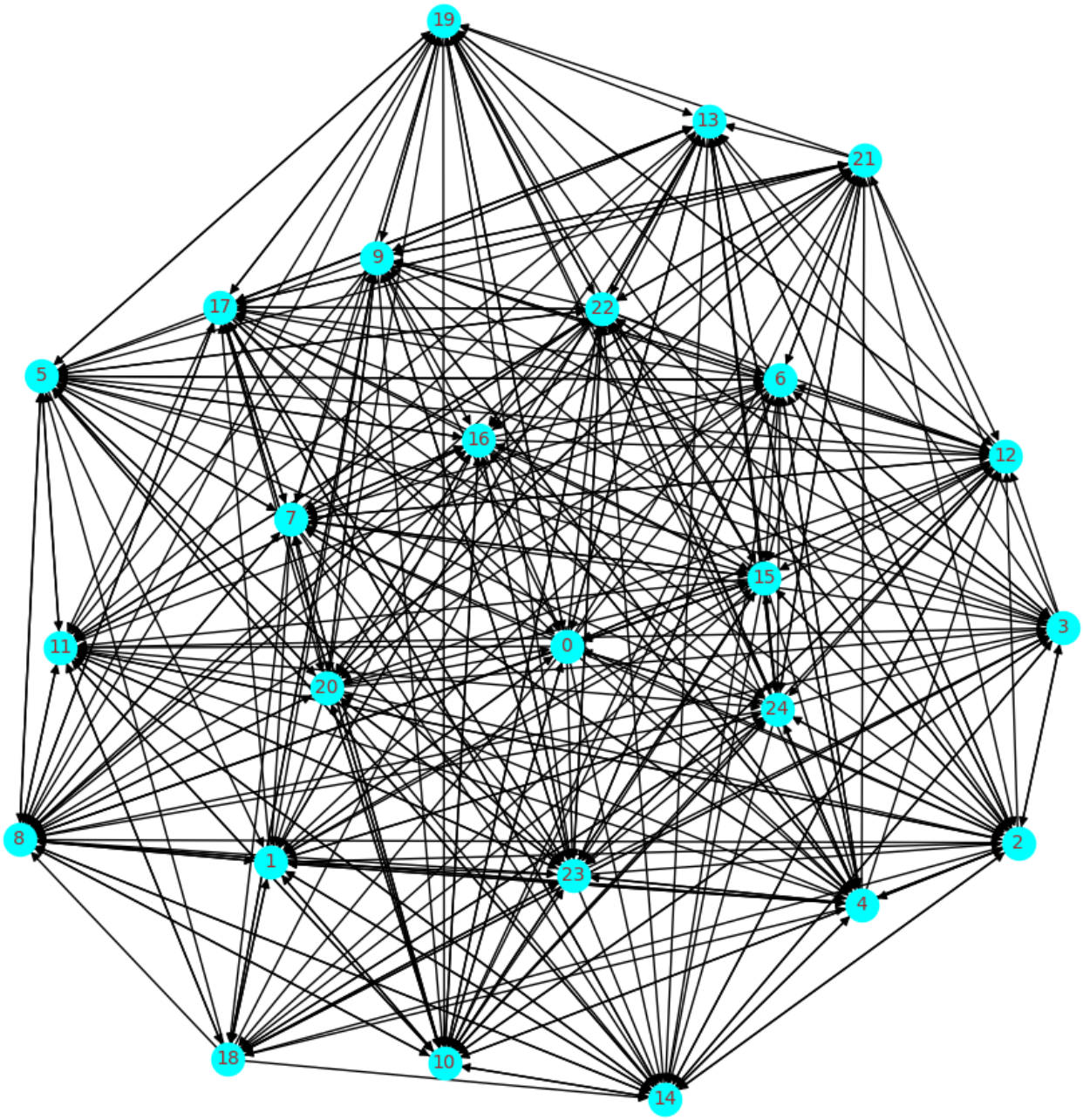
The above centrality measure is not implemented for multi-graphs.

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

Program Listing And Output:

```
import networkx as nx
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import operator
import random as rd

graph = nx.gnp_random_graph(25, 0.6, directed=True)
nx.draw(graph, with_labels=True, font_color='BROWN', font_size=10,
node_color='CYAN')
plt.show()

count = graph.number_of_nodes()
print(list(graph.neighbors(1)))

rank_dict = {}
x = rd.randint(0, 25)

for j in range(0, 25):
   rank_dict[j] = 0

rank_dict[x] = rank_dict[x] + 1

for i in range(600000):
   list_n = list(graph.neighbors(x))

if len(list_n) == 0:
   x = rd.randint(0, 25)
   rank_dict[x] = rank_dict[x] + 1
else:
```

```
    x = rd.choice(list_n)
    rank_dict[x] = rank_dict[x] + 1
print("Updated Random Walk Score:")

for j in range(0, 25):
    rank_dict[j] = rank_dict[j] / 600000

pagerank = nx.pagerank(graph)

pagerank_sorted = sorted(pagerank.items(), key=lambda
    v: (v[1], v[0]), reverse=True)
print(pagerank_sorted)

rank_dict_sorted = sorted(rank_dict.items(), key=lambda v: (v[1], v[0]),
reverse=True)

print(rank_dict_sorted)
print("Order generated by implementation algorithm:\n")

for i in rank_dict_sorted:
    print(i[0], end=" ")
print("\n\nOrder generated by networkx library:\n")
for i in pagerank_sorted:
    print(i[0], end=" ")
```



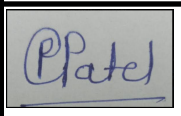
Conclusion: In this experiment, we successfully demonstrated the Implementation of PageRank Algorithm.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 11					
Date of Performance:	**30-09-2022**				
Date of Submission:	**07-10-2022**				
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
07	02	03	03	15	

Experiment No.

11 Aim: Implement Linear regression using the R tool.

Laboratory Outcome: CSL 503.2: Implement data mining algorithms like classification.

Problem Statement: Write commands in R Language to implement Linear Regression using the R tool.

Related Theory:

The R Foundation, a nonprofit focused on supporting the continued development of R through the R Project, describes R as "a language and environment for statistical computing and graphics." But, if you're familiar with R for data science, you probably know it's a lot more than that.

R was created in the 1990s by Ross Ihaka and Robert Gentleman at the University of Auckland in New Zealand. The R language was modeled based on the S language developed at Bell Laboratories by John Chambers and other employees.

Today, R is an open-source language; it's accessible as a free software compatible with many systems and platforms.

R vs. Python

Python and R are both open-source software languages that have been around for a while. When comparing R vs. Python, some feel that Python is a more general programming language. Python is often taught in introductory programming courses and is the primary language for multiple machine learning workflows, RStudio reports. R is typically used in statistical computing. RStudio notes that R is often taught in statistics and data science courses. It adds that many machine learning interfaces are written in Python, while many statistical methods are written in R.

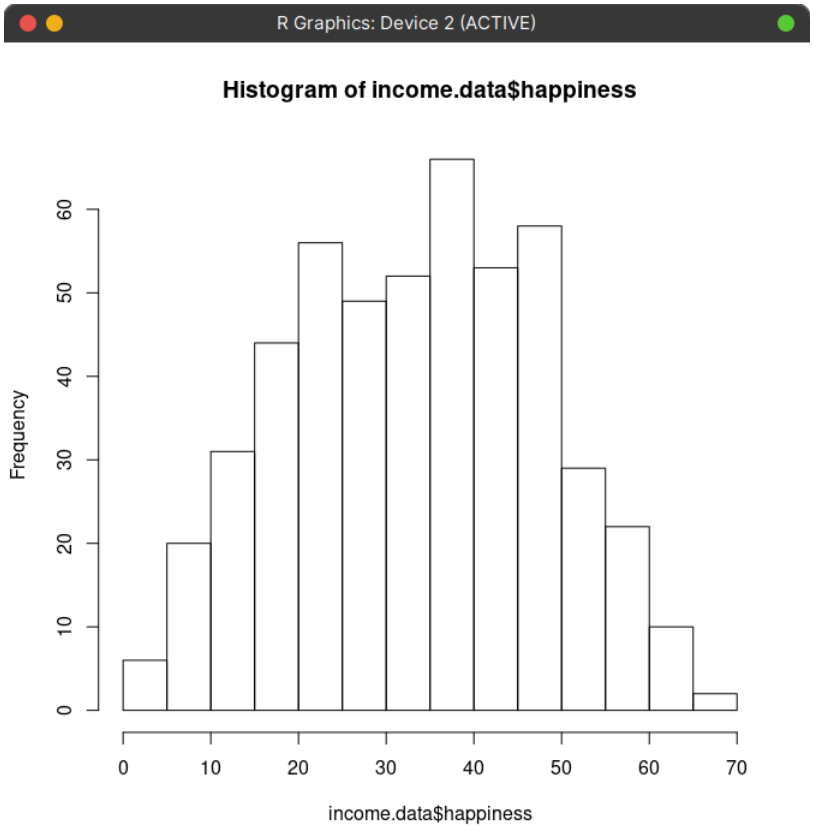
In terms of R vs. Python environments, the R environment is ideal for data manipulation and graphing. Some Python applications include web development, numeric computing and software development. Additionally, while R has numerous packages, Python has many libraries devoted to data science.

Program Listing And Output:

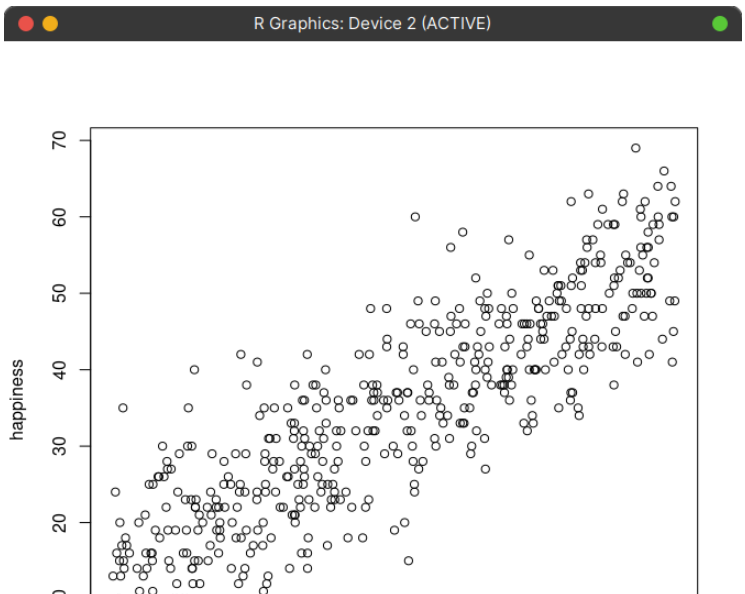
```
> income.data <- read.csv("income.data.csv") (income.data)>
+ summary(income.data)
```

```
       pp              income       happiness
 Min.   :  1.0   Min.   : 506   Min.   : 3.00
 1st Qu.:125.2   1st Qu.:2006   1st Qu.:23.00
 Median :249.5   Median :3424   Median :35.00
 Mean   :249.5   Mean   :3467   Mean   :33.95
 3rd Qu.:373.8   3rd Qu.:4992   3rd Qu.:45.00
 Max.   :498.0   Max.   :6482   Max.   :69.00
```

```
> hist(income.data$happiness)
```




```
> plot(happiness ~ income, data = income.data)
```

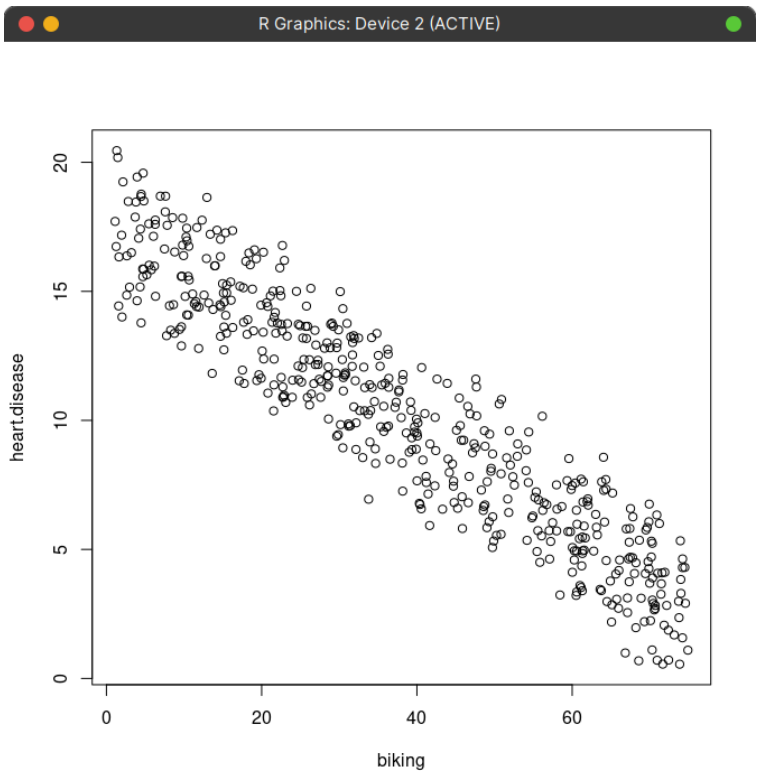


Multiple Regression:

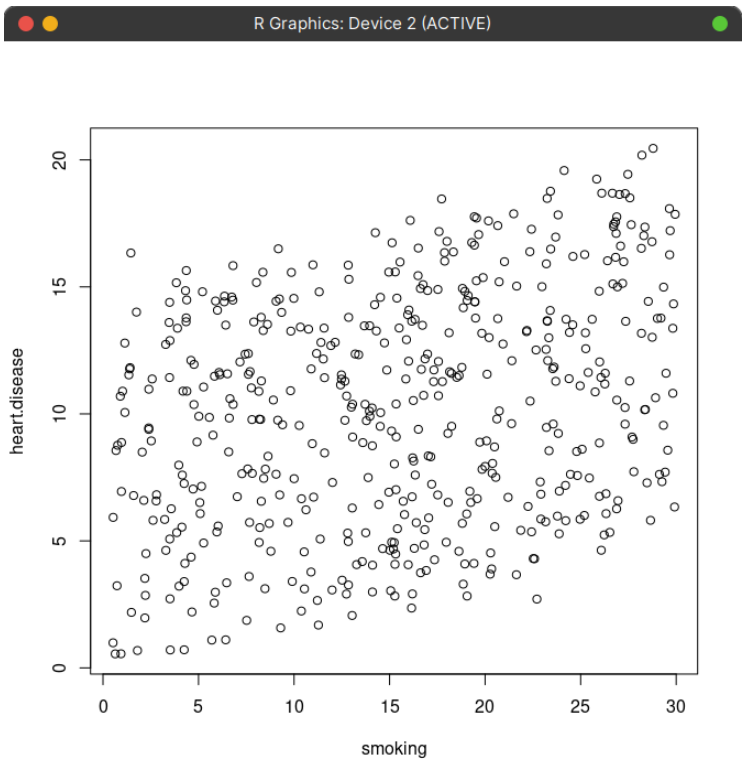
```
>          heart.data <-
read.csv("heart.data.csv")
summary(heart.data)
```

```
       X              biking          smoking        heart.disease
 Min.   :  1.0   Min.   : 1.119   Min.   : 0.5259   Min.   : 0.5519
 1st Qu.:125.2   1st Qu.:20.205   1st Qu.: 8.2798   1st Qu.: 6.5137
 Median :249.5   Median :35.824   Median :15.8146   Median :10.3853
 Mean   :249.5   Mean   :37.788   Mean   :15.4350   Mean   :10.1745
 3rd Qu.:373.8   3rd Qu.:57.853   3rd Qu.:22.5689   3rd Qu.:13.7240
 Max.   :498.0   Max.   :74.907   Max.   :29.9467   Max.   :20.4535
```

```
> plot(heart.disease ~ biking, data=heart.data)
```



```
> plot(heart.disease ~ smoking, data=heart.data)
```




```
> income.happiness.lm <- lm(happiness ~ income, data = income.data)
> summary(income.happiness.lm)
```

```
Call:
lm(formula = happiness ~ income, data = income.data)

Residuals:
     Min       1Q   Median       3Q      Max
-20.2455  -4.8548   0.2277   4.5859  24.2481

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 9.2269182  0.7203696   12.81   <2e-16 ***
income      0.0071323  0.0001858   38.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.197 on 496 degrees of freedom
Multiple R-squared:  0.7482,    Adjusted R-squared:  0.7477
F-statistic:  1474 on 1 and 496 DF,  p-value: < 2.2e-16
```

Conclusion: Here, we implemented Linear Regression using the R tool.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Assignment 1

Marks : 17/20

4+4+5+4 = 17/20 Partly

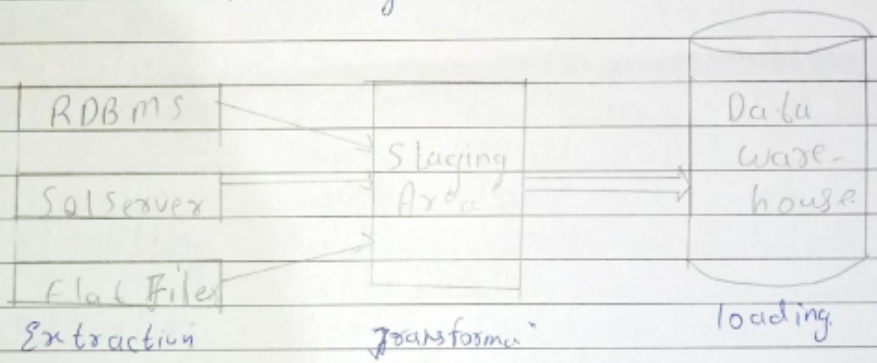
30/8/2024

Name: Kaivalya Mungase
Roll No: 24
Subject: Data Warehousing & Mining.

Assignment 1

Q.1

→ ① ETL is process in which ETL tool extracts data from various data source systems, transforms it in staging area, and then finally, loads it into Data warehouse system.



Following are steps of ETL process:

1) Extraction.
- First step is extraction. In this step, data from various source systems is extracted which can be in various formats like relational database.
- It is important to extract data types from various source systems & store it into staging area first & not directly into data warehouse.

4

- Hence loading it directly into data warehouse may damage it and rollback will be much more difficult.
- Therefore, this is one of most important steps of ETL process.

② Transformation:

- The second step of ETL process is transformation
- In this step, a set of rules or function are applied on extracted data to convert it into single standard format
- It may involve following processes:
  a) filtering - loading certain attributes
  b) Cleaning - filling up the null values with some default values
  c) joining - join multiple attributes
  d) splitting - splitting single into multiple attributes
  e) Sorting - Sorting tuples on basis of some attribute.

③ Loading:

- This is final process.
- transformed data is loaded into data warehouse.
- data is updated by loading into data warehouse very frequently & sometimes after longer interval.

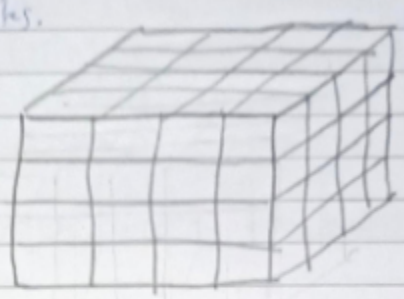
Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Q.2)
→ Dimensions: Course, student & time.
fact: Aggregate marks.

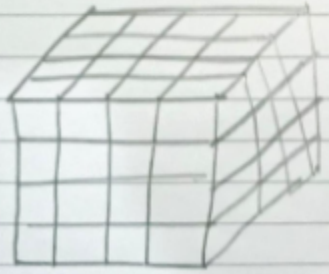


→ draw

c) Operations: There are five basic analytical operations that can be performed on an OLAP cube.

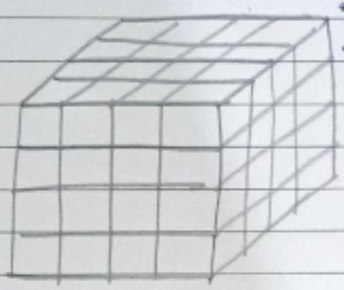
① Drill down: In drill down operation the less detailed data is converted into highly detailed data.
It can be done by:
- Moving down by the concept hierarchy
- Adding new dimension in cube

given in overview section.
In drill down operation is performed by moving down in concept hierarchy of time dimension.

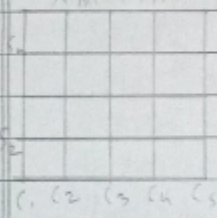


② Roll up: It is just opposite of drill down operation. It performs aggregation on OLAP cube. It can be done by:
- Climbing up in concept hierarchy
- Reducing dimensions

roll up operation is performed by climbing up in the concept hierarchy of student dimension



③ Slice: It selects a single dimension from OLAP cube which results in new sub-cube. In cube given obtaining aggregate marks of all students for all characters for sem 1.



④ Dices: It selects a sub-cube from OLAP cube by selecting two or more dimensions.

⑤ Pivot: It is also known as rotation operation as it rotates the current to get a new views of r represen".

draw.

Q.3)

→ Data mining is proces of extracting info. to identify patterns, trends & useful data that could allow business to to be data driven decision from huge sets of data is known as data mining

① Classification: This technique is used to obtain important and relevent info about data & metadata. This data mining technique helps to classify data in different classes. Data mining can be classified by different classes. Data ~~mining~~ mining can be classified by different criteria as follows:

i) Classificn of data mining frameworks as per the type of data sources mined.

ii) Classificn of data mining framework as per database involved

iii) Classificaⁿ of data mining frameworks according to data mining techniques used.

② Clustering: Clustering analyzes data objects without consulting an identified class label. In general, the labels do not exist in training data simply because they are not known to begin with. Clustering can be used to generate these labels. The object are clustered based on principle of max intra-class similarity & minimizing the interclass similarity.

③ Regressions: Regression analysis is data mining process is used to identify and analyze relationship betⁿ variable because of presence of other factor. It is used to define the probability of specific variable. Regression, primarily form of planning & modeling.

④ Association Rule: This data mining technique helps to discover link between more items. It hidden pattern in data set association rule are if then statement that support to shows probability of interaction betⁿ data item within large data set in different types of databases.

⑤ Outer detection: This type of data mining technique relates to observation of data items.

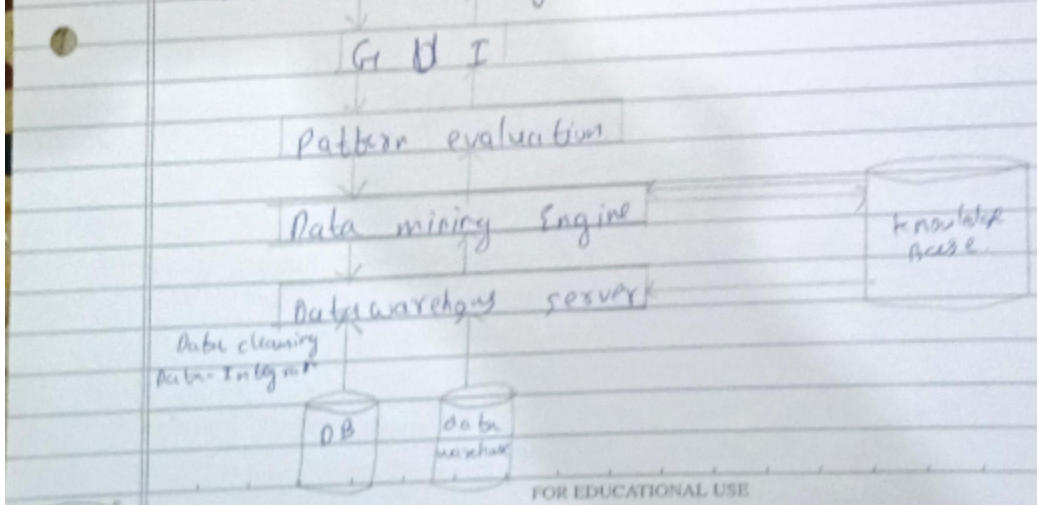
In data set, which do not match any expected pattern or expected behaviour. This technique may be used in various domains like intrusion, detection etc.

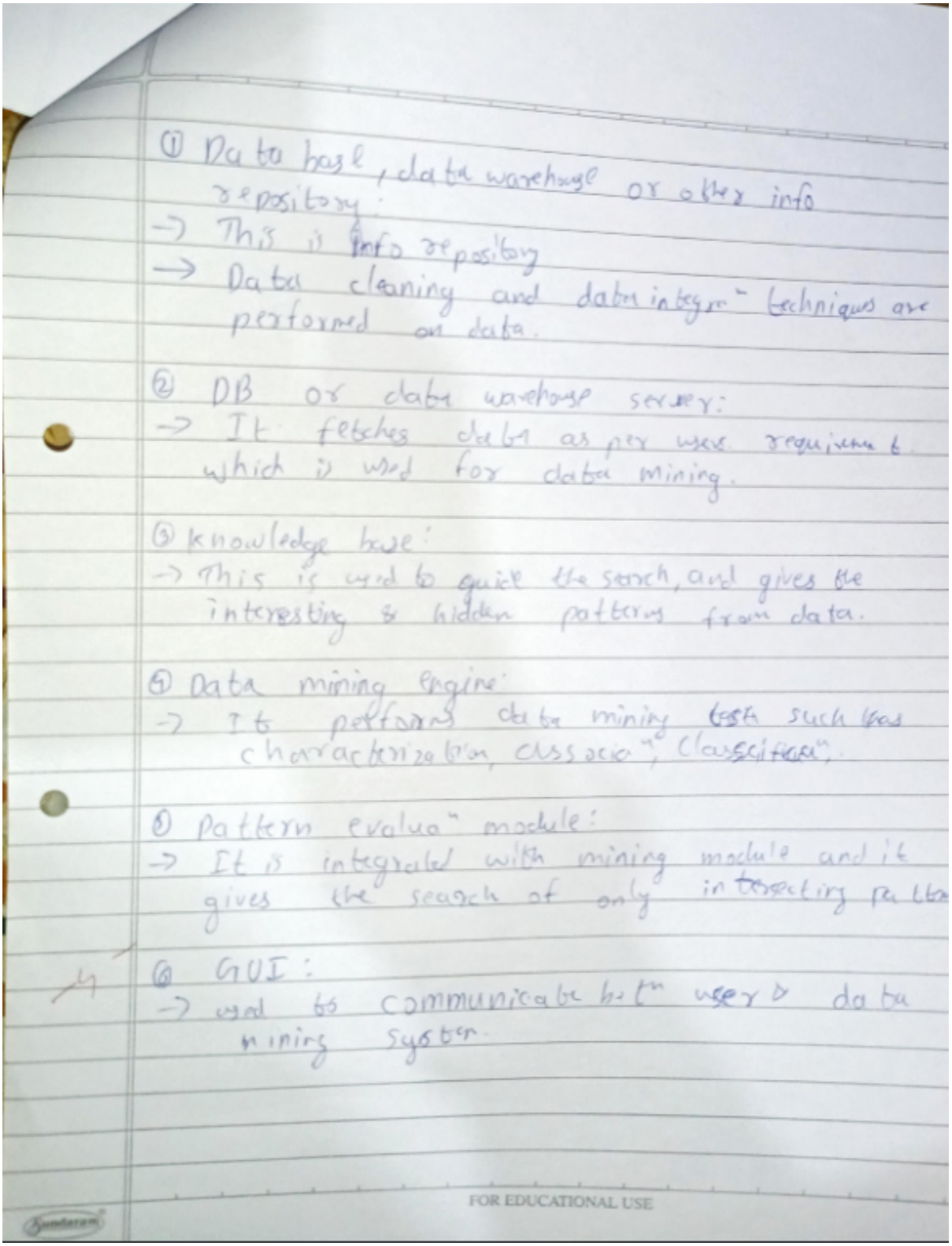
Sequential patterns: The sequential pattern is data mining technique specialized for evaluating sequential data to discover sequential patterns. It comprises of finding interesting subsequences in set of sequences, where state of sequence can be measured in terms of different criteria.

Prediction: Prediction used a combination of other data mining techniques such as trends, clustering classification etc. It analyze past event or instances in right sequence to predict future event.

4)

→ Architecture of typical data mining system





① Data base, data warehouse or other info repository.
→ This is info repository
→ Data cleaning and data integration techniques are performed on data.

② DB or data warehouse server:
→ It fetches data as per user requirement & which is used for data mining.

③ knowledge base:
→ This is used to guide the search, and gives the interesting & hidden patterns from data.

④ Data mining engine:
→ It performs data mining tasks such as characterization, association, classification.

⑤ Pattern evaluation module:
→ It is integrated with mining module and it gives the search of only interesting patterns

⑥ GUI:
→ used to communicate b/t user & data mining system.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Assignment 2

Marks : 17/20

$4+5+4+4 = \frac{17}{20}$ Daily

Kaiwalya Mungase Roll No: 24 TE IS Date: 30/09/22

DWM Assignment - 2

Q.1 Describe Applications of cluster analysis.

→ Cluster: Given data is divided into different groups by combining similar objects into group This group is nothing but cluster.

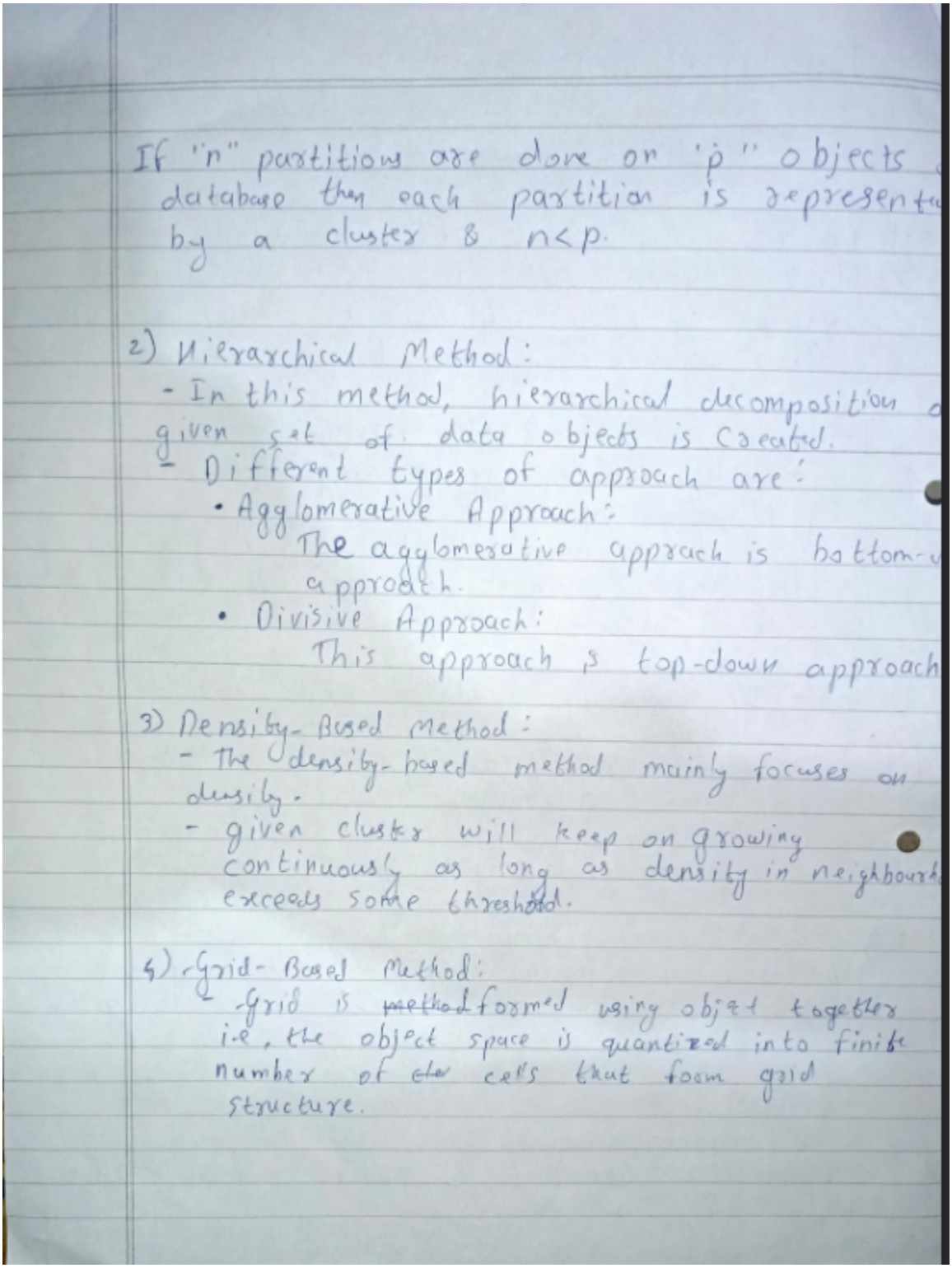
* Applications of Cluster Analysis:
- Widely used in img processing, data analysis, and pattern recognition
- It helps marketers to find distinct groups in their customer base and they can characterize their customer base & can characterize their customer.
- It can be used in field in field of biology, by deriving animal and plant taxonomies & identifying genes with the same capabilities.
- It helps in information discovery by classifying documents on web.

Q.2 Classify clustering methods.

→ Cluster: Given data is divided into different groups by combining similar objects into group.

* Clustering Methods:

i) Partioning method: It is used to make partitions on data in order to form clusters.



If "n" partitions are done on "p" objects database then each partition is represented by a cluster & $n<p$.

2) Hierarchical Method:
- In this method, hierarchical decomposition of given set of data objects is created.
- Different types of approach are:
  - Agglomerative Approach:
    The agglomerative approach is bottom-up approach.
  - Divisive Approach:
    This approach is top-down approach

3) Density-Based method:
- The density-based method mainly focuses on density.
- given cluster will keep on growing continuously as long as density in neighbourhood exceeds some threshold.

4) Grid-Based Method:
- Grid is ~~method~~ formed using object together i.e, the object space is quantized into finite number of ~~the~~ cells that form grid structure.

5) Model-Based Method :
- In this method, all clusters are hypothesized in order to find data which is best suited for model.

6) Constraint-Based Method :
- The constraint-based clustering method is performed by incorporation of application or user-oriented constraints.

(5)

Q.2 →

Diff bet^n classification & Clustering.

Classification	Clustering
- Supervised learning approach where a specific label is provided to machine to classify new observations	- Unsupervised learning approach where ~~grouping~~ learning approach where grouping is done on similarities basis.
- Uses training dataset	- ~~Uses~~ do not use training dataset.
- labels for training data.	- In clustering there are no labels for training data.
- uses algo to categorize new data as per the observations of training set.	- uses statistical concepts in which the data set is divided int subsets. with same feature.
- more complex as compared to clustering	- less complex as compared to classification.

(4)

Q.4 → Explain Web Structure Mining.

- Web structure mining is a tool that can recognize the relationship between web pages linked by data or direct link connection. This structured data is discoverable by provision of web structure schema through datasets techniques for web pages.
- This connection enables a search engine to pull data associated with search query directly to connecting web page from website content rests upon.
- This completion takes place through need of Spiders scanning website, fetching home page, URL and connecting data through reference connection to bring forth specific page including desired information.

(h)

- Web mining can widely be viewed as the application of adapted data mining methods to web, whereas data mining method to web
- It has distinctive property to support a collection of multiple data types.
- web has several aspects that yield multiple approaches for mining process, such as web pages including text, web pages are connected via hyperlinks, & user activity can be monitored via web server logs.
- Structure mining uses minimize two main problems of world wide web because of its large amount of data.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security