lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

DATAMING AND DATA WARE HOUSING

LABORATARY MANUAL

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

---

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLGOY

VISION

To excel in computing arena and to produce globally competent computer science and Information Technology graduates with Ethical and Human values to serve the society

MISSION

- To impart strong theoretical and practical background in computer science and information technology discipline with an emphasis on software development.
- To provide an open environment to the students and faculty that promotes professional growth
- To inculcate the skills necessary to continue their education and research for contribution to society.

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

---

COURSE OUTCOMES (CO's)

CO1: Understand the Environment of weka tool and prepare Data sets.

CO2: Understand various pre-processing Techniques.

CO3: Analyze Various classification Algorithms.

CO4: Apply the Association rule mining to various data sets to Extract Patterns.

CO5: Analyze various clustering Algorithms.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates of Computer Science and Information Technology will acquire strong knowledge to analyze, design, and develop computing products and solutions for real-life problems utilizing the latest tools, techniques, and technologies.

PEO2: Graduates of Computer Science and Information Technology shall have interdisciplinary approach, professional attitude and ethics, communication, teamwork skills and leadership capabilities to solve social issues through their Employment, Higher Studies and Research.

PEO3: Graduates will engage in life-long learning and professional development to adapt to dynamically computing environment.

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

---

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge
PO2: Problem Analysis
PO3: Design & Development
PO4: Investigations
PO5: Modern Tools
PO6: Engineer & Society
PO7: Environment & Sustainability
PO8: Ethics
PO9: Individual & Team Work
PO10: Communication Skills
PO11: Project Mgt & Finance
PO12: Life Long Learning

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Ability to solve contemporary issues utilizing skills.
PSO2: To acquire knowledge of the latest tools and technologies to provide technical solutions
PSO3: To qualify in national and international competitive examinations for successful higher studies and employment.

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

LAB SYLLABUS

1. Demonstration of preprocessing on dataset student.arff
2. Demonstration of preprocessing on dataset labor.arff
3. Demonstration of classification rule process on dataset student.arff using j48 algorithm
4. Demonstration of classification rule process on dataset employee.arff using j48 algorithm
5. Demonstration of classification rule process on dataset employee.arff using id3 algorithm
6. Demonstration of classification rule process on dataset employee.arff using naïve bayes algorithm
7. Demonstration of Association rule process on dataset contactlenses.arff using a priori algorithm
8. Demonstration of Association rule process on dataset test.arff using apriori algorithm
9. Demonstration of clustering rule process on dataset iris.arff using simple k-means
10. Demonstration of clustering rule process on dataset student.arff using simple means

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

---

COURSE OUTCOMES Vs PO's & PSO's

SNO	DESCRIPTION	PO(1..12) MAPPING	PSO(1..3) MAPPING
1	Understand the Environment of weka tool and prepare Data sets.	PO2,PO3	PSO1,PSO2
2	Understand various pre-processing Techniques	PO3,PO11	PSO1
3	Analyze Various classification Algorithms.	PO3,PO11	PSO2
4	Apply the Association rule mining to various data sets to Extract Patterns	PO1,PO6	PSO1
5	Analyze various clustering Algorithms.	PO3,PO6	PSO2
COURSE OVERALL PO/PSO MAPPING:			

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

---

SYLLABUS INDEX

PROGRAMS BEYOND SYLLABUS				
11	Implement clustering procedure using python?	45-46	CO1,CO2,CO3,CO4,CO5,CO6	PO1,PO2,PO3,PO11 PSO1,PSO2
12	Implement classification procedure using python?	46-49	CO1,CO2,CO3,CO4,CO5,CO6	PO1,PO2,PO3,PSO1 PSO2
OPEN ENDED EXPERIMENTS				
13	Implement Exploratory data analysis using python	50		
14	Implement Classification: Predicting Customer Churn using python	51		

lendi Institute of Engineering & Technology
An Autonomous Institution
Accredited by NAAC with '' A'' Grade, Accredited by NBA (ECE, CSE.EEE & MECH)
Approved by A.I.C.T.E. & Permanently Affiliated to J. N. T. U. Gurajada, VIZIANAGARAM
Via 5th APSP Battalion, Jonnada (V), Denkada (M), NH-3, Vizianagaram Dist - 535005, A.P. Website : www.lendi.org
Ph : 08922-241111, 241666, Cell No : 9490344747, 9490304747,e-mail : lendi_2008@yahoo.com

Instructions to students

Pre-lab activities:

- Prepare observation note book which contains the following :
  - Procedure/algorithm/program to solve the problems discussed in the theory class
  - Solutions to the exercises given in the previous lab session
- Refer the topics covered in theory class

In-lab activities:

- Note down errors observed while executing program and remedy for that.
- Note down corrections made to the code during the lab session
- Answer to vivo-voce
- Get the observation corrected
- Note down inferences on the topic covered by the programs executed

Post-lab activities:

- Solve the given exercises
- Devise possible enhancements that can be made to the solved problem to simplify the logic
- Executed programs should be recorded in the lab record and corrected within one week after completion of the experiment.
- After completion of every module, a test will be conducted, and assessment results will have weight in the final internal marks.

General Instructions:

- Student should sign in the log register before accessing the system.
- Student is only responsible for any damage caused to the equipment in the laboratory during his session.
- Usage of pen drives is not allowed in the lab.
- If a problem is observed in any hardware equipment, please report to the lab staff immediately; do no attempt to fix the problem yourself.
- Systems must be shut down properly before leaving the lab.
- Please be considerate of those around you, especially in terms of noise level. While labs are a natural place for conversations regarding programming, kindly keep the volume turned down

Experiment 1

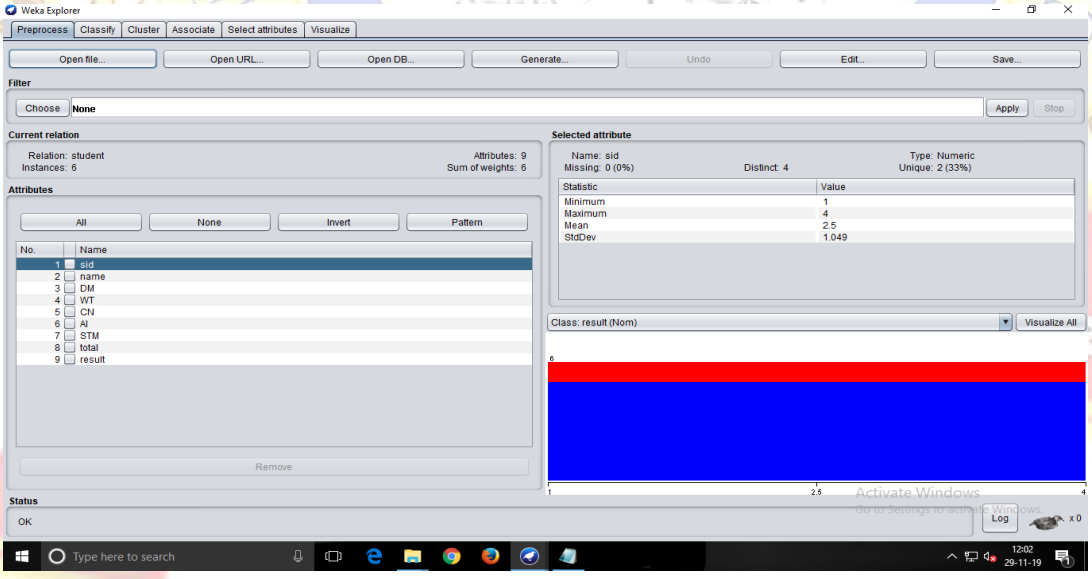
1) **AIM:** Demonstration of preprocessing on dataset student.arffData set

@relation student
@attribute sid numeric
@attribute name {usha,hari,rajesh,kiran,giri,manash}
@attribute DM numeric
@attribute WT numeric
@attribute CN numeric
@attribute AI numeric
@attribute STM numeric
@attribute total numeric
@attribute result {pass,fail}

@data
1,usha,60,55,45,50,40,250,pass
2,hari,60,55,45,40,40,240,pass
3,rajesh,60,55,40,50,40,240,pass
4,kiran,60,55,30,50,40,235,fail
2,giri,60,55,45,60,40,260,pass
3,manash,60,55,65,50,40,270,pass

Ex1:

Result:



DESCRIPTION:

Add:

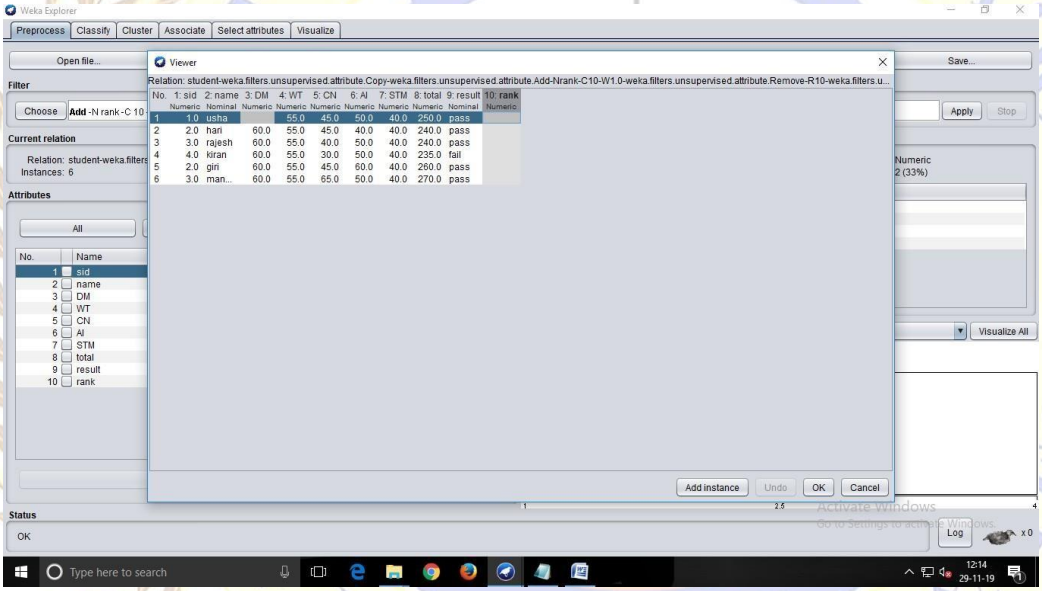
NAME

weka.filters.unsupervised.attribute.Add

SYNOPSIS

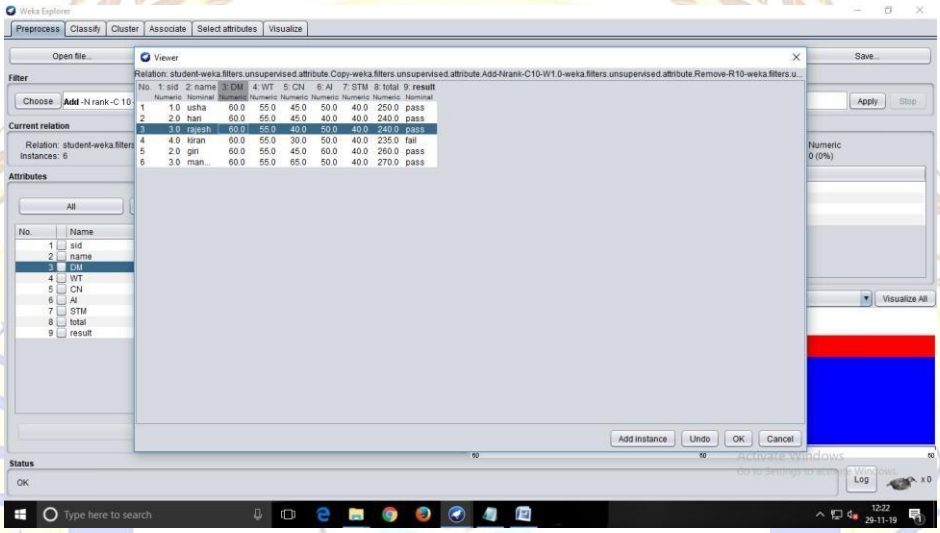
An instance filter that adds a new attribute to the dataset. The new attribute will contain all missingvalues.

OPTIONS
nominalLabels -- The list of value labels (nominal attribute creation only). The list must be comma-separated, eg: "red,green,blue". If this is empty, the created attribute will be numeric.
debug -- If set to true, filter may output additional info to the console.
attributeName -- Set the new attribute's name
attributeIndex -- The position (starting from 1) where the attribute will be inserted (first and last are valid indices).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built.weight -- The weight for the new attribute.
dateFormat -- The format of the date values (see ISO-8601). attributeType -- Defines the type of the attribute to generate.

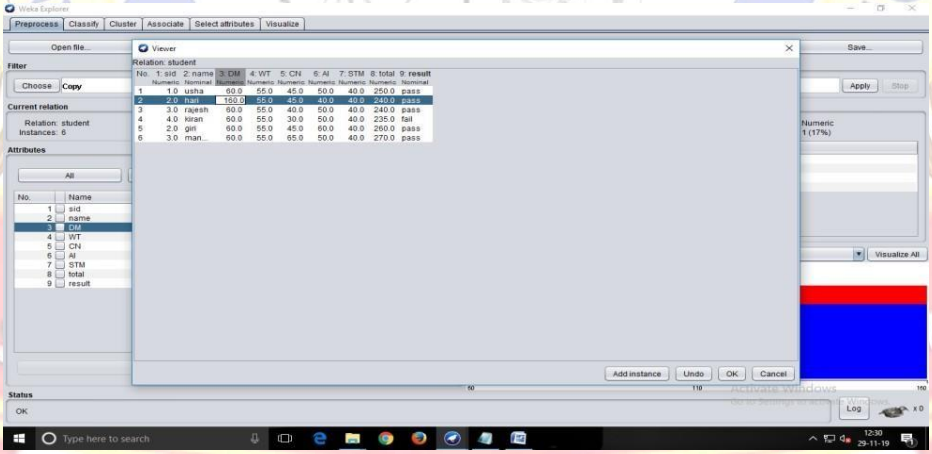


Remove:
NAME
weka.filters.unsupervised.attribute.Remove
SYNOPSIS
A filter that removes a range of attributes from the dataset. Will re-order the remaining attributes if invert matching sense is turned on and the attribute column indices are not specified in ascending order.
OPTIONS
debug -- If set to true, filter may output additional info to the console.
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use withcaution to reduce runtime.)
attributeIndices -- Specify range of attributes to act on. This is a comma separated list of attribute indices, with "first" and "last" valid values. Specify an inclusive range with "-". E.g: "first-3,5,6-10,last". invertSelection -- Determines whether action is to select or delete. If set to true, only the specified attributes will be kept; If set to false, specified attributes will be deleted.

missing attributes:



ErrornicData:



CONCLUSION :

Data preprocessing in Data Mining and Data Warehousing (DMDW) is essential for ensuring data quality, consistency, and relevance, which enables more accurate analysis and decision-making. By cleaning, transforming, and organizing raw data, preprocessing helps to improve the efficiency and effectiveness of data mining models and ensures reliable insights from large datasets.

VIVA QUESTIONS:

1. What is data preprocessing in machine learning?

.

2. Why is data cleaning important in data preprocessing?

3. What is the difference between normalization and standardization?

5. What is the role of outlier detection in data preprocessing?

6. How do you encode categorical variables into numerical data?

7. What are the common techniques for feature scaling?

8. What is the purpose of data transformation in preprocessing?

9. **Explain the difference between data discretization and binning.**

10. **What is feature selection, and why is it important?**

.

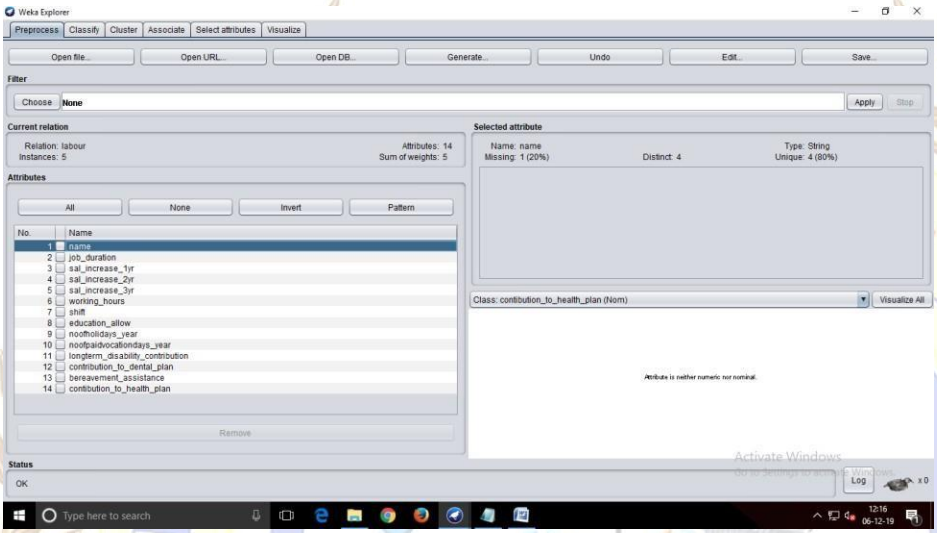
Experiment 2

2) **AIM:** Demonstration of preprocessing on dataset labor.arffData set

```
@relation labour
@attribute name
string
@attribute job_duration numeric
@attribute sal_increase_1yr
numeric @attribute
sal_increase_2yr numeric
@attribute sal_increase_3yr
numeric @attribute working_hours
numeric @attribute shift
{day,night} @attribute
education_allow {yes,no}
@attribute noofholidays_year
numeric
@attribute noofpaidvocationdays_year numeric
@attribute longterm_disability_contribution
{yes,no} @attribute contribution_to_dental_plan
{none,half,full}@attribute bereavement_assistance
{yes,no}
@attribute contibution_to_health_plan {none,half,full}

@data
?,3,2,2,2,6,day,no,5,5,no,half,no,none
harish,2,2,2,3,5,night,no,5,4,no,none,no,no
ne
ramesh,8,2,3,3,7,day,no,5,5,yes,full,yes,hal
f
suresh,5,2,3,2,6,night,no,5,3,no,half,no,hal
f
Naresh,5,2,2,2,6,day,no,5,4,no,half,no,half
```

Result:



DESCRIPTION

Copy:

NAME

weka.filters.unsupervised.attribute.C opySYNOPSIS

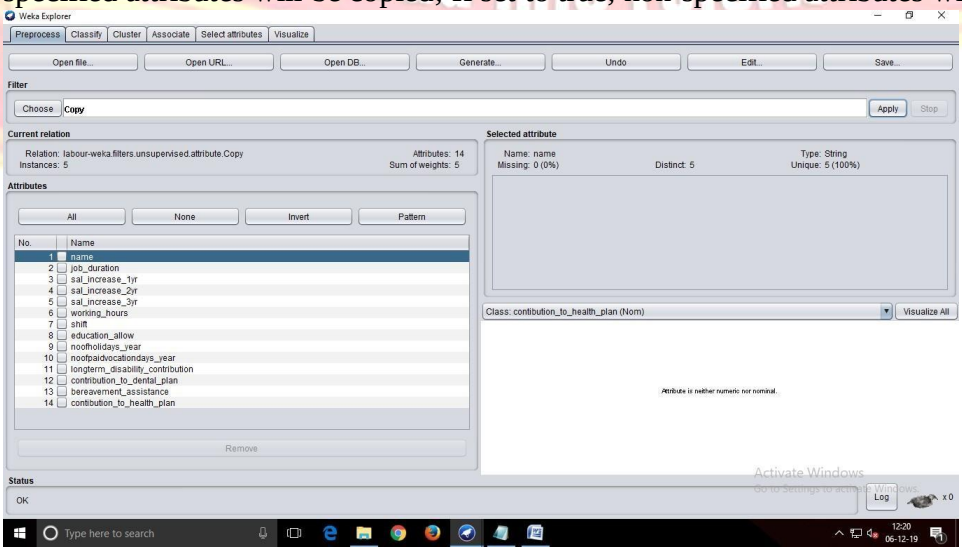
An instance filter that copies a range of attributes in the dataset. This is used in conjunction with other filters that overwrite attribute values during the course of their operation -- this filter allows the originalattributes to be kept as well as the new attributes.

OPTIONS

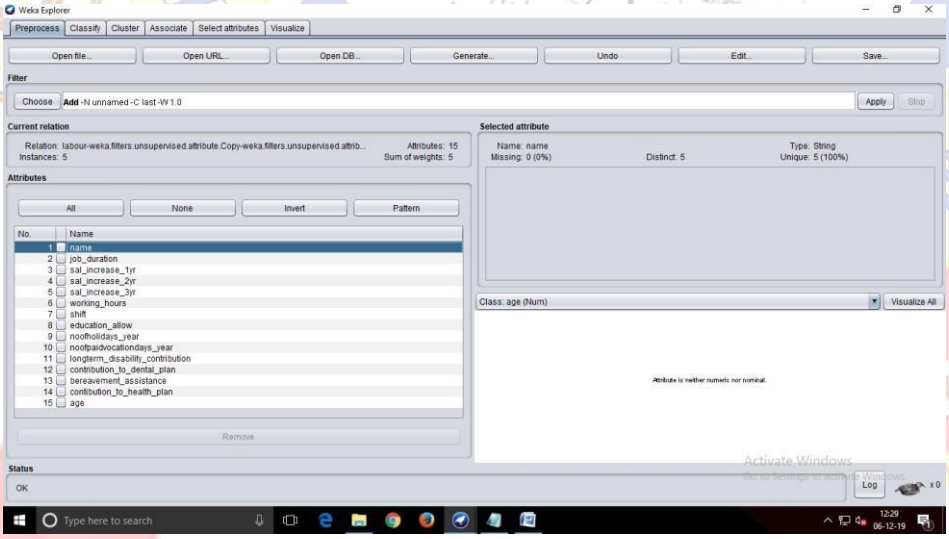
debug -- If set to true, filter may output additional info to the console.

doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use withcaution to reduce runtime.)

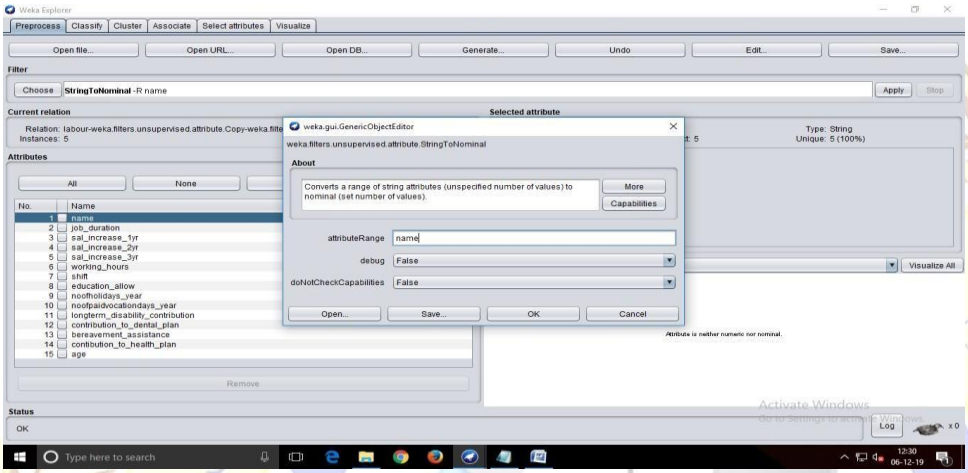
attributeIndices -- Specify range of attributes to act on. This is a comma separated list of attribute indices, with "first" and "last" valid values. Specify an inclusive range with "-". E.g: "first-3,5,6-10,last". invertSelection -- Sets copy selected vs unselected action. If set to false, only the specified attributes will be copied; If set to true, non-specified attributes will be copied.



Add:
NAME
weka.filters.unsupervised.attribute.Add
SYNOPSIS
An instance filter that adds a new attribute to the dataset. The new attribute will contain all missingvalues.
OPTIONS
nominalLabels -- The list of value labels (nominal attribute creation only). The list must be comma-separated, eg: "red,green,blue". If this is empty, the created attribute will be numeric.
debug -- If set to true, filter may output additional info to the console.
attributeName -- Set the new attribute's name.
attributeIndex -- The position (starting from 1) where the attribute will be inserted (first and last are validindices).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built.weight -- The weight for the new attribute.
dateFormat -- The format of the date values (see ISO-8601). attributeType -- Defines the type of the attribute to generate.



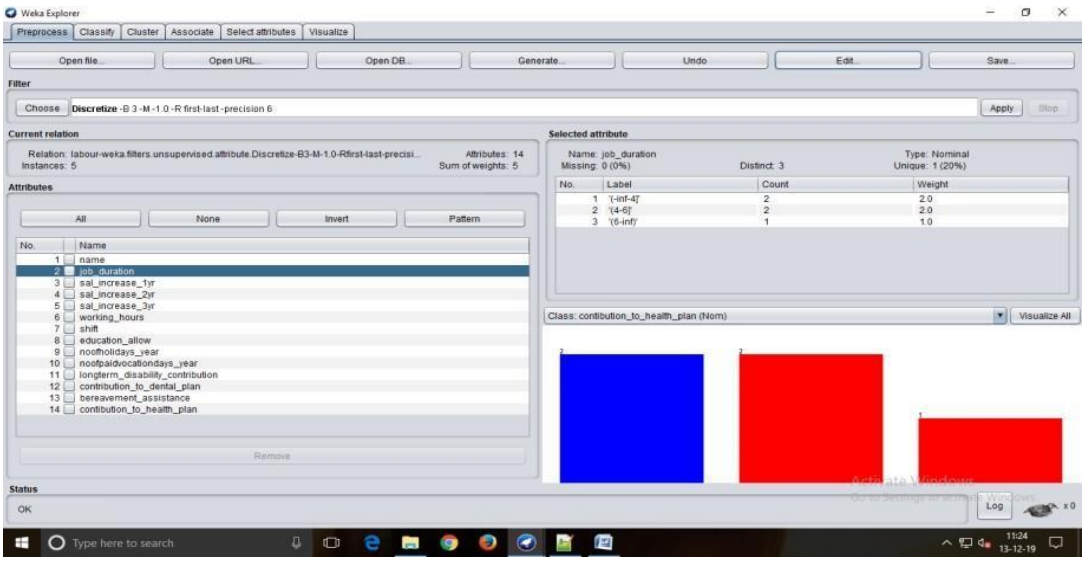
String-to-Nominal:
NAME
weka.filters.unsupervised.attribute.StringToNomi
nalSYNOPSIS
Converts a range of string attributes (unspecified number of values) to nominal (set number of values). You should ensure that all string values that will appear are represented in the first batch of the data.
OPTIONS
debug -- If set to true, filter may output additional info to the console.
attributeRange -- Sets which attributes to process ("first" and "last" are valid values and ranges and listscan also be used).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use withcaution to reduce runtime.)

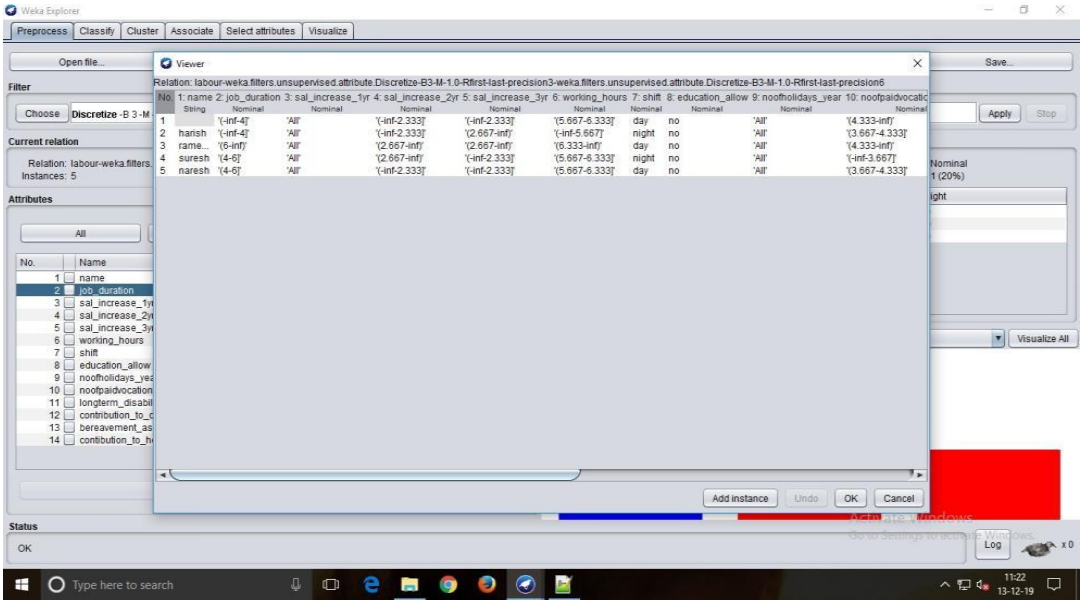


Normalize:
NAME
weka.filters.unsupervised.attribute.Normalize
SYNOPSIS
Normalizes all numeric values in the given dataset (apart from the class attribute, if set). By default, the resulting values are in [0,1] for the data used to compute the normalization intervals. But with the scale and translation parameters one can change that, e.g., with scale = 2.0 and translation = -1.0 you get valuesin the range [-1,+1].

OPTIONS
debug -- If set to true, filter may output additional info to the console.
translation -- The translation of the output range (default: 0).
doNotCheckCapabilities -- If set, the filter's capabilities are not checked before it is built. (Use withcaution to reduce runtime.)
scale -- The factor for scaling the output range (default: 1).
ignoreClass -- The class index will be unset temporarily before the filter is applied.

Discretize:
NAME
weka.filters.unsupervised.attribute.Discretize
SYNOPSIS
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.Discretization is by simple binning. Skips the class attribute if set.
OPTIONS
spreadAttributeWeight -- When generating binary attributes, spread weight of old attribute across newattributes. Do not give each new attribute the old weight.
makeBinary -- Make resulting attributes binary.
debug -- If set to true, filter may output additional info to the console.





CONCLUSION:

Preprocessing the LABOR.ARFF dataset involved cleaning the data by handling missing values, removing irrelevant features, and normalizing the numerical attributes to ensure consistency and improve model performance. These steps enhance the quality of the dataset, leading to more accurate and reliable outcomes in data mining tasks such as classification or clustering.

VIVA QUESTIONS:

1. What is a dataset in the context of data mining?

2. What is the difference between structured and unstructured datasets?

3. What is a training dataset and a testing dataset in data mining?

.

4. What is the role of data preprocessing in a dataset before applying data mining techniques?

5. What are noisy data and how do you handle them in a dataset?

6. What is a sparse dataset in data mining?

Experiment 3

3) **AIM :** Demonstration of Association rule process on dataset contact lenses.arff using apriori algorithmData set
@relation contact-lenses
@attribute age {young, pre-presbyopic,presbyopic} @attribute spectacle-prescrip {myope, hypermetrope}@attribute astigmatim {no,yes}
@attribute tear-prod-rate {reduced,
normal}@attribute contact-lenses
{soft,hard,none} @data
young,myope,no,reduced,none young,myope,no,normal,soft young,myope,yes,reduced,none young,myope,yes,normal,hard young,hypermetrope,no,reduced,none young,hypermetrope,no,normal,soft young,hypermetrope,yes,reduced,none young,hypermetrope,,yes,normal,hard pre-presbyopic,myope,no,reduced,none pre-presbyopic,myope,no,normal,soft pre-presbyopic,myope,yes,reduced,nonepre-presbyopic,myope,yes,normal,soft pre-presbyopic,myope,yes,reduced,nonepre-presbyopic,myope,yes,normal,hard
pre-presbyopic,hypermetrope,no,reduced,none pre-presbyopic,hypermetrope,no,normal,soft pre-presbyopic,hypermetrope,,yes,reduced,nonepre-presbyopic,hypermetrope,yes,normal,none presbyopic,myope,no,normal,none presbyopic,myope,yes,reduced,none presbyopic,myope,yes,normal,hard presbyopic,hypermetrope,no,reduced,none presbyopic,hypermetrope,no,,normal,soft Presbyopic,hypermetrope,yes,reduced,none

NAME
weka.associations.Apriori

SYNOPSIS
Class implementing an Apriori-type algorithm. Iteratively reduces the minimum support until it finds therequired number of rules with the given minimum confidence.
The algorithm has an option to mine class association rules. It is adapted as explained in the secondreference.

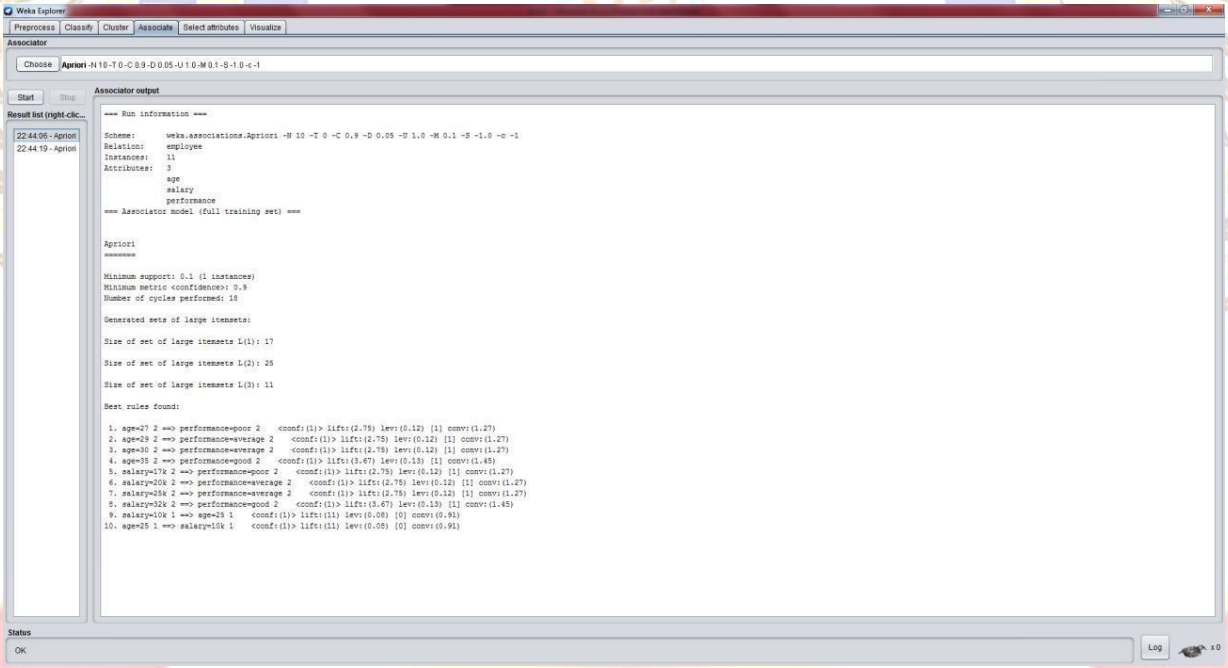
OPTIONS
minMetric -- Minimum metric score. Consider only rules with scores higher than this value.verbose -- If enabled the algorithm will be run in verbose mode.
numRules -- Number of rules to find.
lowerBoundMinSupport -- Lower bound for minimum support.
classIndex -- Index of the class attribute. If set to -1, the last attribute is taken as class attribute.
outputItemSets -- If enabled the itemsets are output as well.
car -- If enabled class association rules are mined instead of (general) association rules.
doNotCheckCapabilities -- If set, associator capabilities are not checked before associator is built (Usewith caution to reduce runtime).

removeAllMissingCols -- Remove columns with all missing values.
significanceLevel -- Significance level. Significance test (confidence metric only).
treatZeroAsMissing -- If enabled, zero (that is, the first value of a nominal) is treated in the same way as amissing value.

DESCRIPTION :

Steps:

1. Open the datafile in wekaexplorer.It is presumed that the required data fields have beendiscretized.In this example it is age attribute.
2. Clicking on the associate tab will bring up the interface for association rule algorithm.
3. We will use apriori algorithm. This is the default algorithm.
4. In order to change the parameters for the run we click on the text box immediately to the right ofthe chosen button.



CONCLUSION:

Applying the Apriori algorithm to the Contact Lenses.arff dataset effectively uncovered frequent itemsets and association rules that reveal patterns in the relationships between features like age, prescription, and lenses type. This process provides valuable insights for decision-making, enabling better recommendations for lens usage based on customer attributes.

VIVA QUESTIONS:

1. What is an association rule in data mining?

2. What are the key components of an association rule?

3. What is the difference between support and confidence in association rules?

4. What is the "Apriori algorithm" used for in association rule mining?

5. What is the concept of "lift" in association rule mining?

6. How would you handle rules that are too general or too specific in association rule mining?

Experiment-4

4)**AIM:** Demonstration of Association rule process on dataset test.arff using apriori algorithm.

Data set

@relation attribute
@attribute
bread{y,n}
@attribute
jelly{y,n}
@attribute
butter{y,n}
@attribute
milk{y,n}
@attribute
sugar{y,n}@data
yyy n n
y n y n
ny n y
y ny n
n y yyy
n y n

NAME

weka.associations.Apriori

SYNOPSIS

Class implementing an Apriori-type algorithm. Iteratively reduces the minimum support until it finds therequired number of rules with the given minimum confidence.

The algorithm has an option to mine class association rules. It is adapted as explained in the secondreference.

OPTIONS

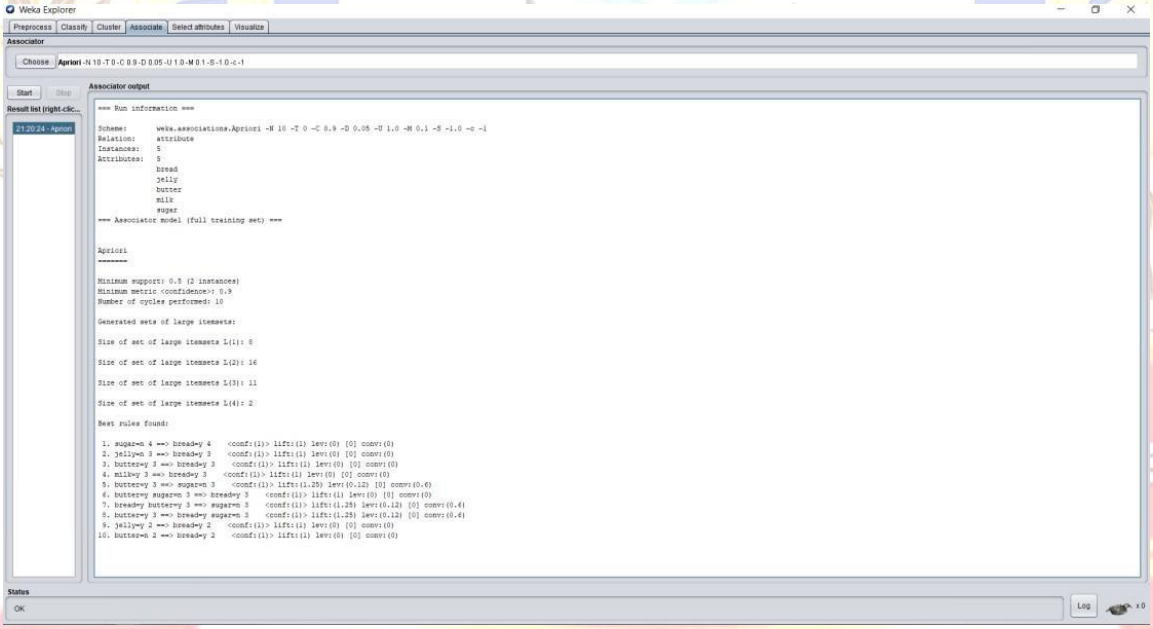
minMetric -- Minimum metric score. Consider only rules with scores higher than this value.verbose -- If enabled the algorithm will be run in verbose mode.
numRules -- Number of rules to find.
lowerBoundMinSupport -- Lower bound for minimum support.
classIndex -- Index of the class attribute. If set to -1, the last attribute is taken as class attribute.
outputItemSets -- If enabled the itemsets are output as well.
car -- If enabled class association rules are mined instead of (general) association rules.
doNotCheckCapabilities -- If set, associator capabilities are not checked before associator is built (Usewith caution to reduce runtime).

removeAllMissingCols -- Remove columns with all missing values.
significanceLevel -- Significance level. Significance test (confidence metric only).
treatZeroAsMissing -- If enabled, zero (that is, the first value of a nominal) is treated in the same way as amissing value.

DESCRIPTION:

Steps:

1. Open the datafile in wekaexplorer.It is presumed that the required data fields have beendiscretized.In this example it is age attribute.
2. Clicking on the associate tab will bring up the interface for association rule algorithm.
3. We will use apriori algorithm. This is the default algorithm.
4. In order to change the parameters for the run we click on the text box immediately to the right ofthe chosen button.



CONCLUSION:

Applying the Apriori algorithm to the Contact TEST.arff dataset effectively uncovered frequent itemsets and association rules that reveal patterns in the relationships between features like age, prescription, and lenses type. This process provides valuable insights for decision-making, enabling better recommendations for lens usage based on customer attributes.

VIVA QUESTIONS :

1. **What is the Apriori algorithm in data mining?**

2. What is the apriori property in the context of the Apriori algorithm?

3. What are support, confidence, and lift in the Apriori algorithm?

4. What is the role of the "min_support" threshold in the Apriori algorithm?

5. How does the Apriori algorithm generate candidate itemsets?

.

6. **What is the main limitation of the Apriori algorithm?**

Experiment 5

5)**AIM :**Demonstration of classification rule process on dataset student.arff using j48 algorithm.

Data set

@relation
student
@attribute age {<30,30-40,>40}
@attribute income
{low,medium,high}@attribute
student {yes,no}
@attribute credit-rating
{fair,excellent}@attribute buyspc
{yes,no}
@data
%
<30, high, no, fair, no
<30, high, no, excellent,
no30-40, high, no, fair,
yes
>40, medium, no, fair, yes
>40, low, yes, fair, yes
>40, low, yes, excellent, no
30-40, low, yes, excellent,
yes
<30, medium, no, fair, no
<30, low, yes, fair, no
>40, medium, yes, fair, yes
<30, medium, yes, excellent,
yes 30-40, medium, no,
excellent, yes30-40, high, yes,
fair, yes
>40, medium, no, excellent, no
%

NAME
weka.classifiers.trees.J48

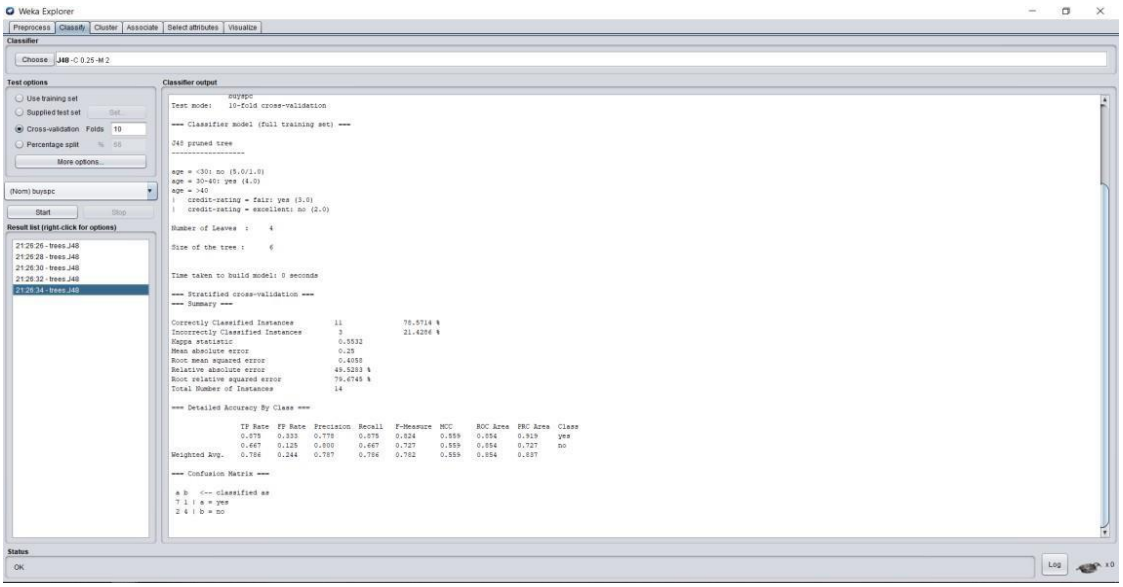
SYNOPSIS
Class for generating a pruned or unpruned C4.5 decision
tree.OPTIONS
seed -- The seed used for randomizing the data when reduced-error pruning is used.
unpruned -- Whether pruning is performed.
confidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning).

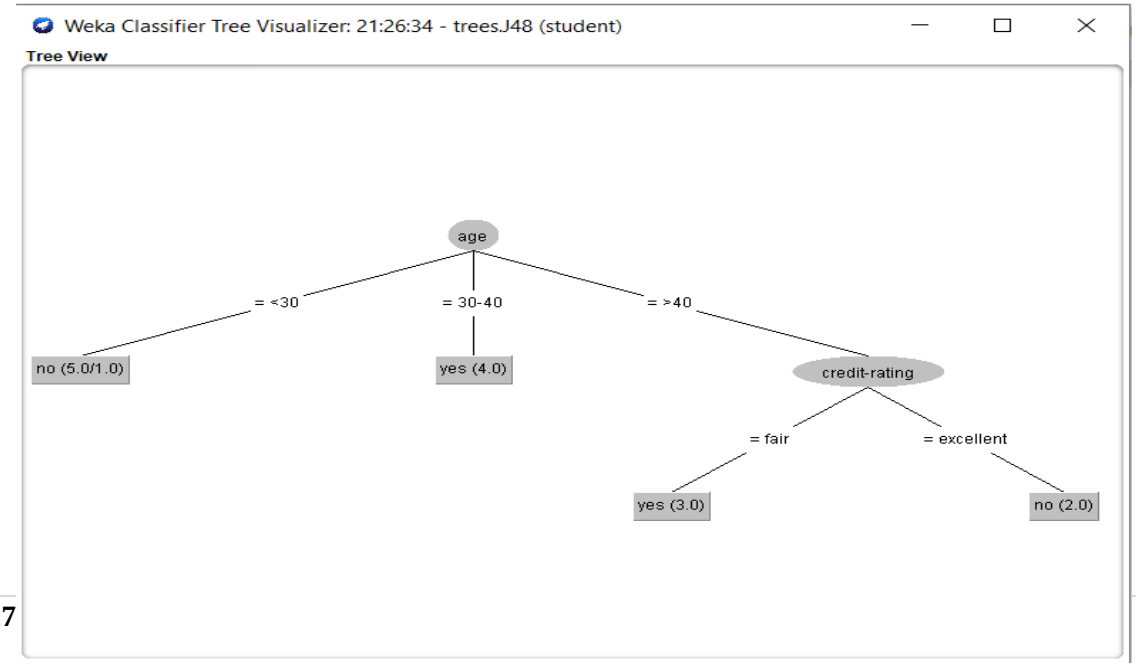
numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning,the rest for growing the tree.
numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.reducedErrorPruning -- Whether reduced-error pruning is used instead of C.4.5 pruning.
useLaplace -- Whether counts at leaves are smoothed based on Laplace.
doNotMakeSplitPointActualValue -- If true, the split point is not relocated to an actual data value. Thiscan yield substantial speed-ups for large datasets with numeric attributes.

DESCRIPTION:

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose the J48 algorithm in classify.
3. Click on start for each attribute to apply the algorithm on data.
4. Discover the highest percentage of correctly classified instances.
5. Generate a tree by clicking visualize tree on that particular attribute.





CONCLUSION :

Using the J48 algorithm on the Student.arff dataset successfully generated classification rules that identify key patterns and relationships between student attributes and their academic performance. This process aids in predicting student outcomes, providing insights for educational interventions and personalized support strategies.

VIVA QUESTIONS:

1. What is the J48 algorithm in data mining?

2. How does the J48 algorithm work?

3. What is information gain in the context of the J48 algorithm?

4. What is pruning in the J48 algorithm and why is it important?

5. What are the advantages of using the J48 algorithm?

6. What is the stopping criterion for building a tree in J48?

Experiment 6

6)**AIM :**Demonstration of classification rule process on dataset employee.arff using j48 algorithm.

Data set

@relation employee
@attribute age{25,27,28,29,30,35,48}
@attribute
salary{10k,15k,17k,20k,25k,30k,35k,32k}
@attribute performance{good,average,poor}
@data
25 10k poor
27 15k poor
27 17k poor
28 17k poor
29 20k average
30 25k average
29 25k average
30 20k average
35 32k good
35 35k good
48 32k good

NAME
weka.classifiers.trees.J48

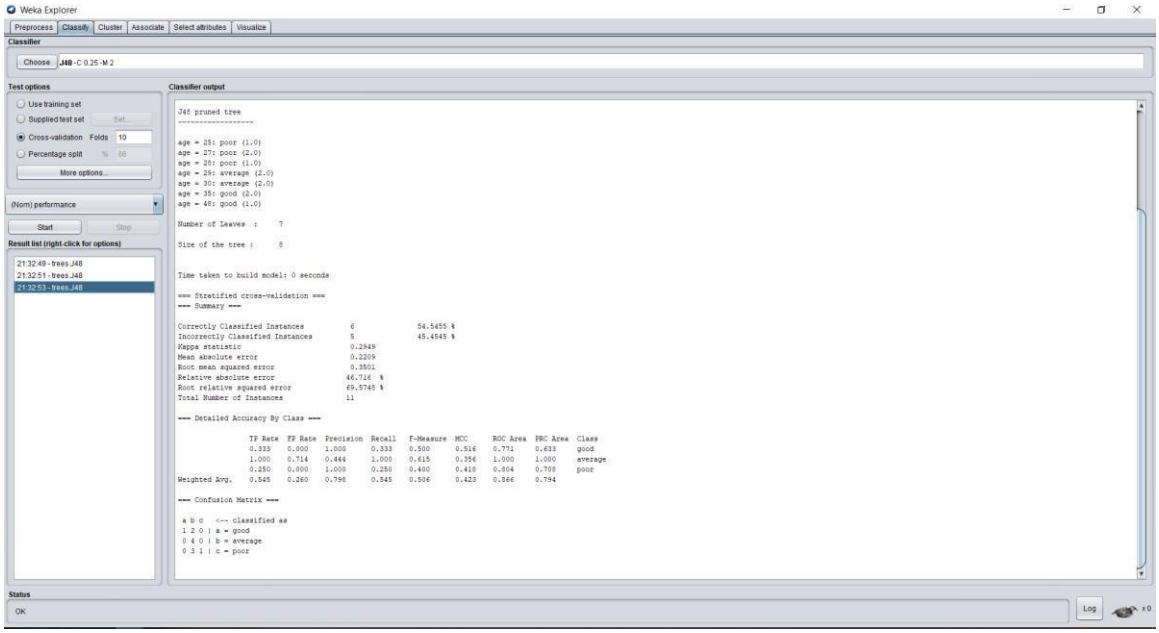
SYNOPSIS
Class for generating a pruned or unpruned C4.5 decision
tree.OPTIONS
seed -- The seed used for randomizing the data when reduced-error pruning is used.
unpruned -- Whether pruning is performed.
confidenceFactor -- The confidence factor used for pruning (smaller values incur more pruning).
numFolds -- Determines the amount of data used for reduced-error pruning. One fold is used for pruning,the rest for growing the tree.
numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.reducedErrorPruning -- Whether reduced-error pruning is used instead of C.4.5 pruning.
useLaplace -- Whether counts at leaves are smoothed based on Laplace.
doNotMakeSplitPointActualValue -- If true, the split point is not relocated to an actual data value. Thiscan yield substantial speed-ups for large datasets with numeric attributes.

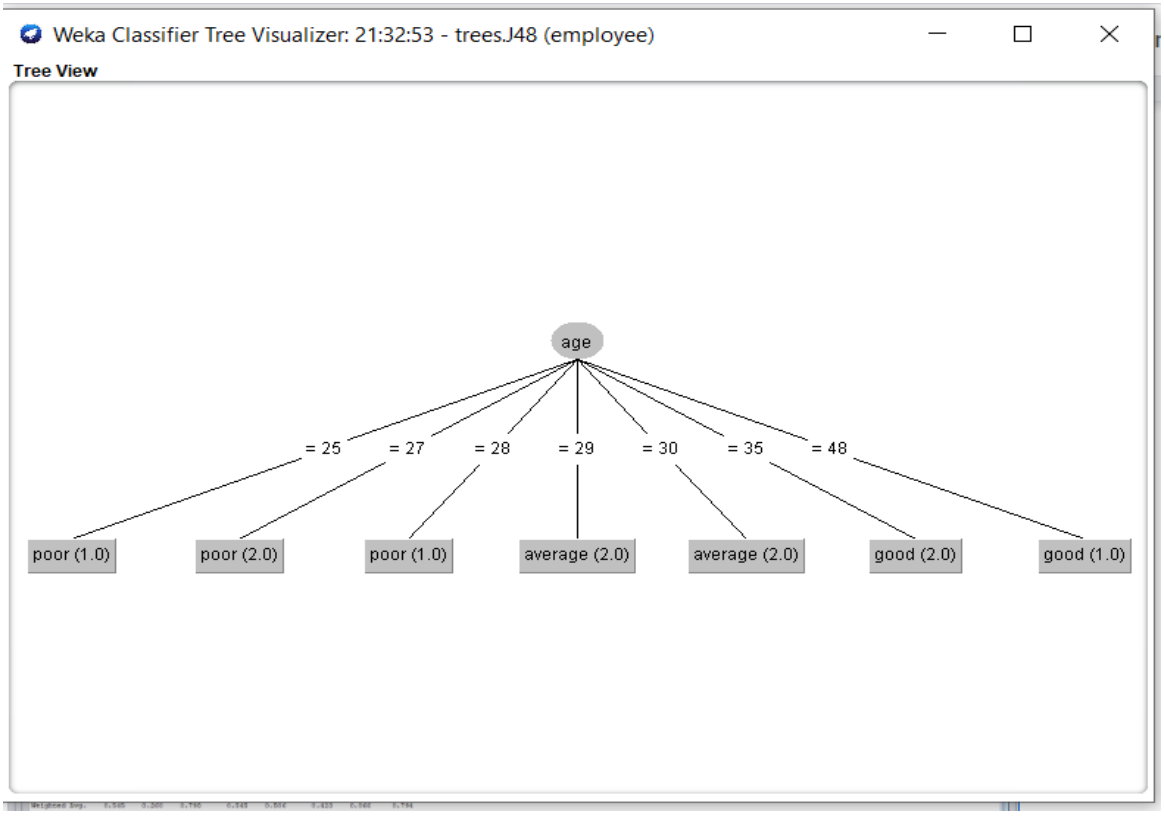
DESCRIPTION:

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose the J48 algorithm in classify.

3. Click on start for each attribute to apply the algorithm on data.
4. Discover the highest percentage of correctly classified instances.
5. Generate a tree by clicking visualize tree on that particular attribute.





CONCLUSION :

Using the J48 algorithm on the **EMPLOYEE.arff** dataset successfully generated classification rules that identify key patterns and relationships between student attributes and their academic performance. This process aids in predicting student outcomes, providing insights for educational interventions and personalized support strategies.

VIVA QUESTIONS

1. How is the J48 algorithm applied in real-time data mining tasks?

2. Can the J48 algorithm handle streaming data? If not, how can it be adapted?

.

3. What are the challenges of using the J48 algorithm in real-time applications?

4. How does pruning in J48 improve its performance in real-time applications?

5. How would you handle imbalanced datasets when using J48 in real-time scenarios?

6. Can J48 be used for real-time prediction in an online system (e.g., recommendation systems)?

Experiment-7

7)**AIM**:Demonstration of classification rule process on dataset employee.arff using Id3 algorithm.
Data set
@relation employee
@attribute age{25,27,28,29,30,35,48}
@attribute
salary{10k,15k,17k,20k,25k,30k,35k,32k}
@attribute performance{good,average,poor}
@data
25 10k poor
27 15k poor
27 17k poor
28 17k poor
29 20k average
30 25k average
29 25k average
30 20k average
35 32k good
35 35k good
48 32k good

NAME
weka.classifiers.trees.Id3

SYNOPSIS
Class for constructing an unpruned decision tree based on the ID3 algorithm. Can only deal with nominalattributes. No missing values allowed. Empty leaves may result in unclassified instances.
OPTIONS
numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.

batchSize -- The preferred number of instances to process if batch prediction is being performed. More orfewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.

debug -- If set to true, classifier may output additional info to the console.

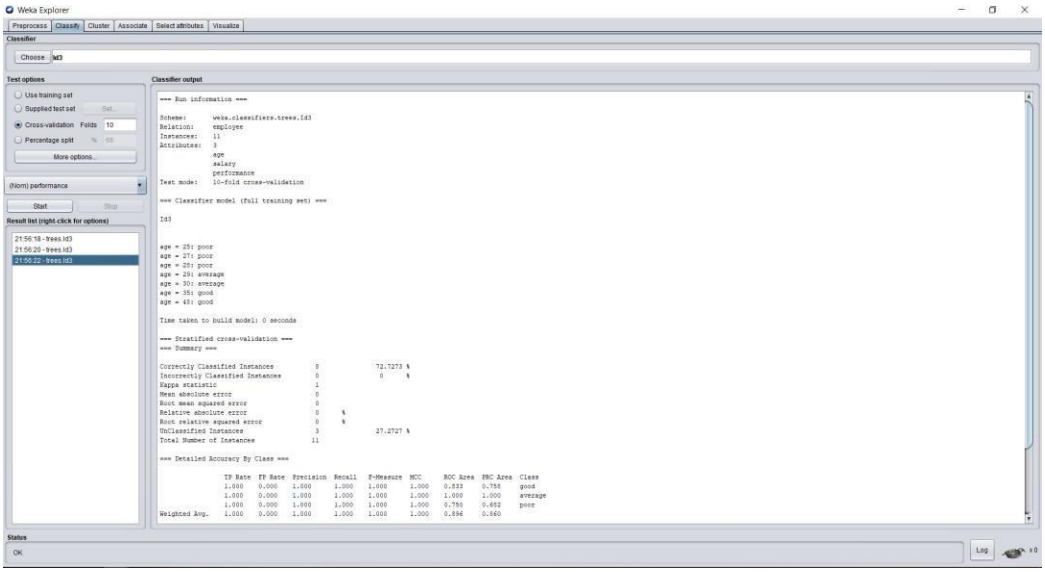
doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is

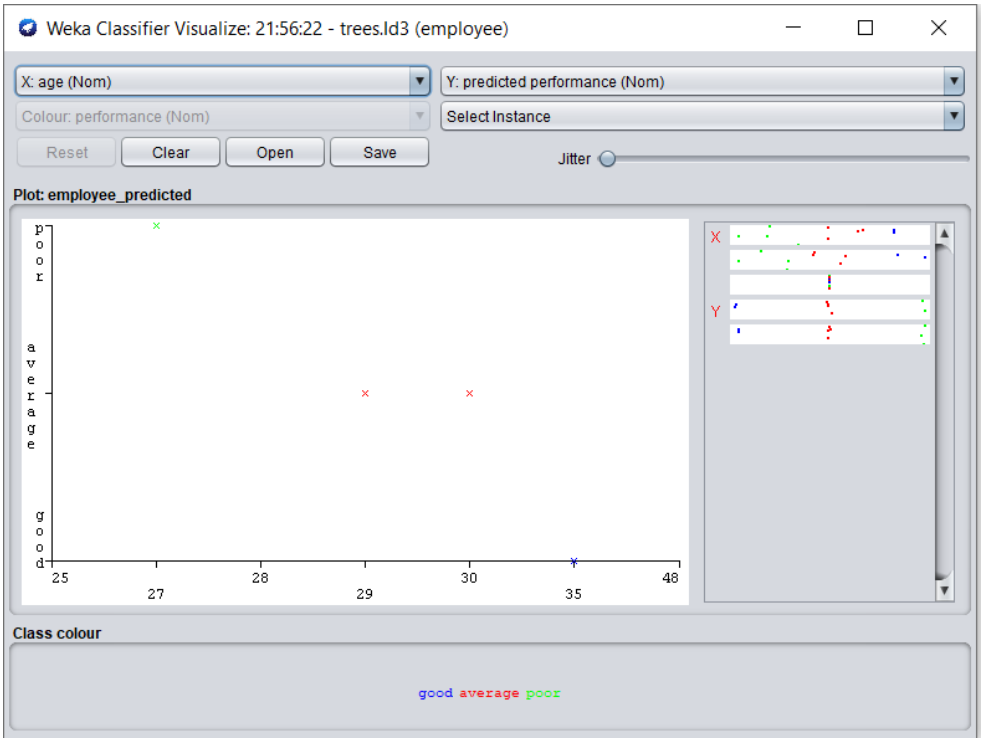
built

DESCRIPTION:

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose Id3.
3. Click on start for each attribute to apply the algorithm on data.
4. Discover the highest percentage of correctly classified instances.





CONCLUSION:

Applying the ID3 algorithm to the **Employee.arff** dataset resulted in the generation of clear classification rules that effectively distinguish between different employee categories based on attributes like age, education, and job role. These rules offer valuable insights for decision-making in areas such as recruitment, promotions, and workforce management.

VIVA QUESTIONS:

1. What is the ID3 algorithm in data mining?

2. What is information gain, and how is it used in ID3?

3. What is entropy, and why is it important for the ID3 algorithm?

4. What are the advantages of using the ID3 algorithm?

5. What is the main limitation of the ID3 algorithm?

6. How does the ID3 algorithm handle missing values in a dataset?

Experiment 8

8) **AIM:** Demonstration of classification rule process on dataset employee.arff using naive bayes algorithm.

Data set
@relation employee

@attribute age{25,27,28,29,30,35,48}

@attribute

salary{10k,15k,17k,20k,25k,30k,35k,32k}

@attribute performance{good,average,poor}

@data

25 10k poor

27 15k poor

27 17k poor

28 17k poor

29 20k average

30 25k average

29 25k average

30 20k average

35 32k good

35 35k good

48 32k

good

NAME

weka.classifiers.bayes.NaiveBayes

DESCRIPTION:

SYNOPSIS

Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosenbased on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (whichin typical usage are initialized with zero training instances) -- if you need the UpdateableClassifier functionality, use the NaiveBayesUpdateable classifier. The NaiveBayesUpdateable classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.

OPTIONS

useKernelEstimator -- Use a kernel estimator for numeric attributes rather than a normal distribution.

numDecimalPlaces -- The number of decimal places to be used for the output of numbers in the model.

batchSize -- The preferred number of instances to process if batch prediction is being performed. More orfewer instances may be provided, but this gives implementations a chance to specify a preferred batch size.

debug -- If set to true, classifier may output additional info to the console.

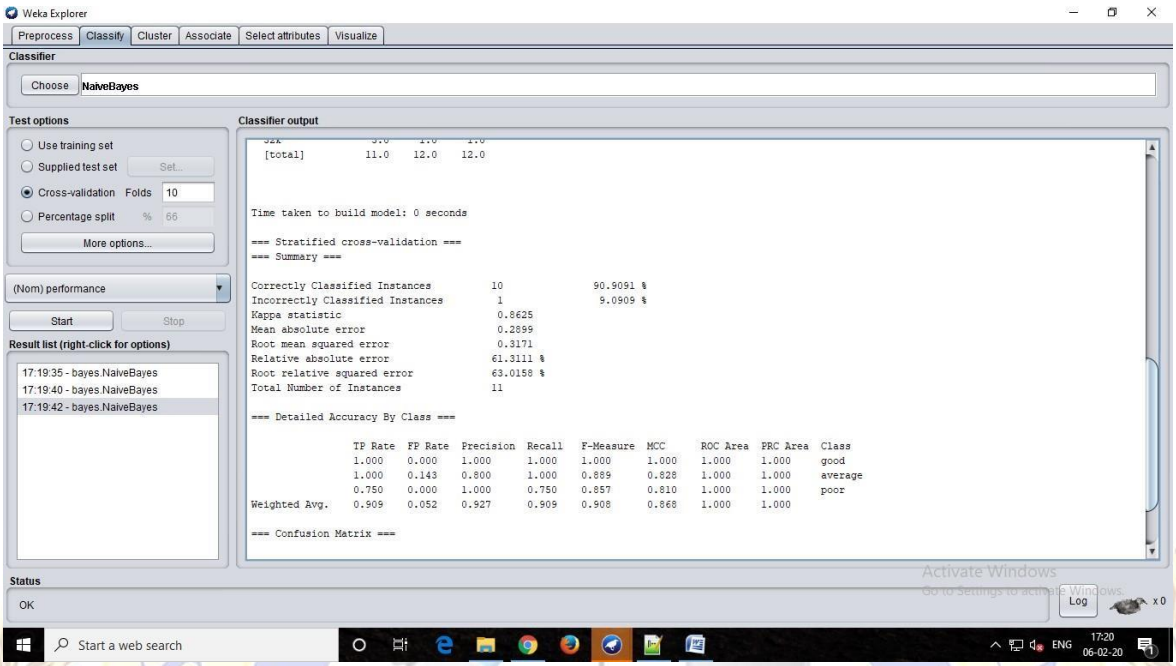
displayModelInOldFormat -- Use old format for model output. The old format is better when there aremany class values. The new format is better when there are fewer classes and many attributes.

doNotCheckCapabilities -- If set, classifier capabilities are not checked before classifier is built (Use withcaution to reduce runtime).

useSupervisedDiscretization -- Use supervised discretization to convert numeric attributes to nominalones.

Steps:

1. Open the datafile in weka explorer and then click on classify.
2. Choose NaiveBayes from Bayes.
3. Click on start for each attribute to apply algorithm on data.
4. Discover the highest percentage of correctly classified instances.



CONCLUSION:

Using the Naive Bayes algorithm on the Employee.arff dataset provided a probabilistic model that classifies employee attributes with a focus on simplicity and efficiency. The resulting classification rules offer valuable insights into employee behaviors, aiding in decision-making processes such as performance evaluation and retention strategies.

VIVA QUESTIONS:

1. What is the Naive Bayes algorithm in data mining?

2. What is Bayes' Theorem and how does it relate to Naive Bayes?

3. What assumptions does the Naive Bayes algorithm make about the data?

4. What are the advantages of using the Naive Bayes algorithm?

5. What is the main limitation of the Naive Bayes algorithm?

6. How does Naive Bayes handle continuous data?

Experiment-9

9)**AIM :**Demonstration of clustering rule process on dataset iris.arff using simple k-means

Data set
@RELATION iris

@ATTRIBUTE sepallength REAL

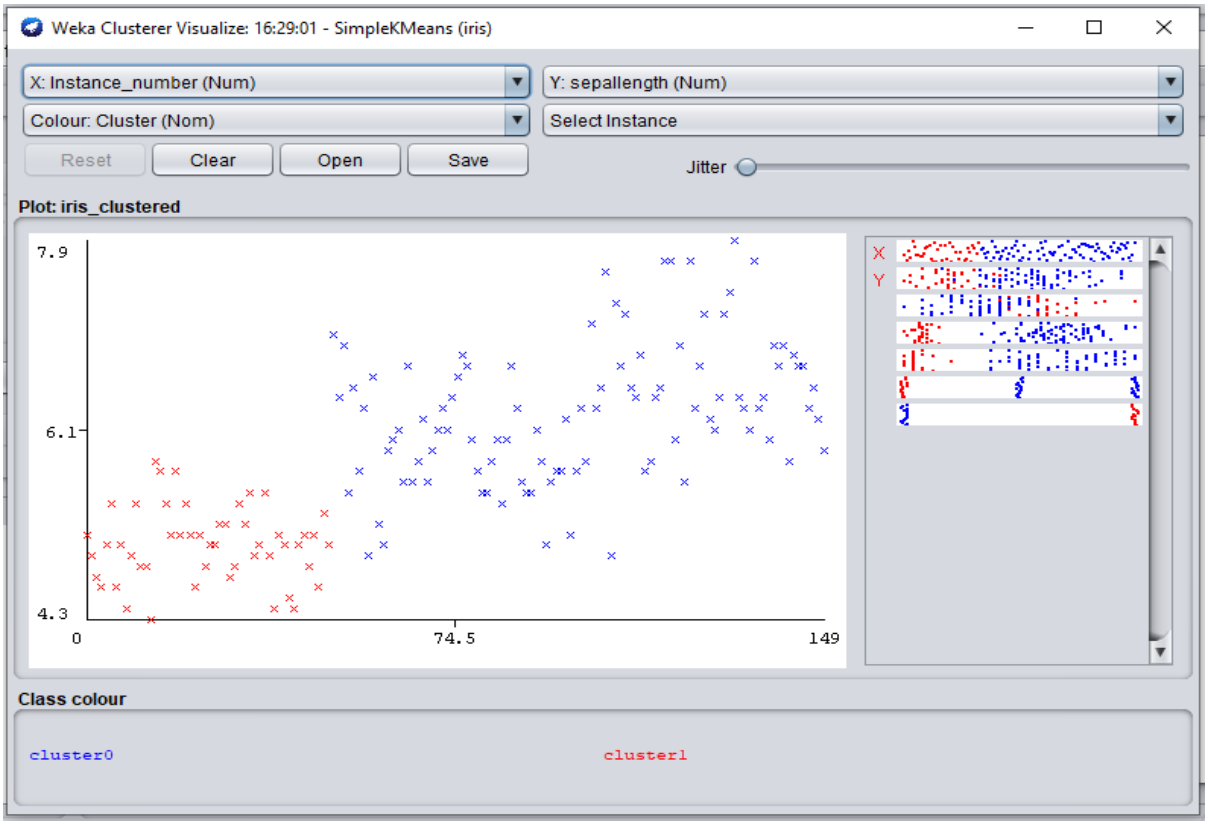
@ATTRIBUTE sepalwidth REAL
@ATTRIBUTE petallength REAL
@ATTRIBUTE petalwidth
REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-
virginica}@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa

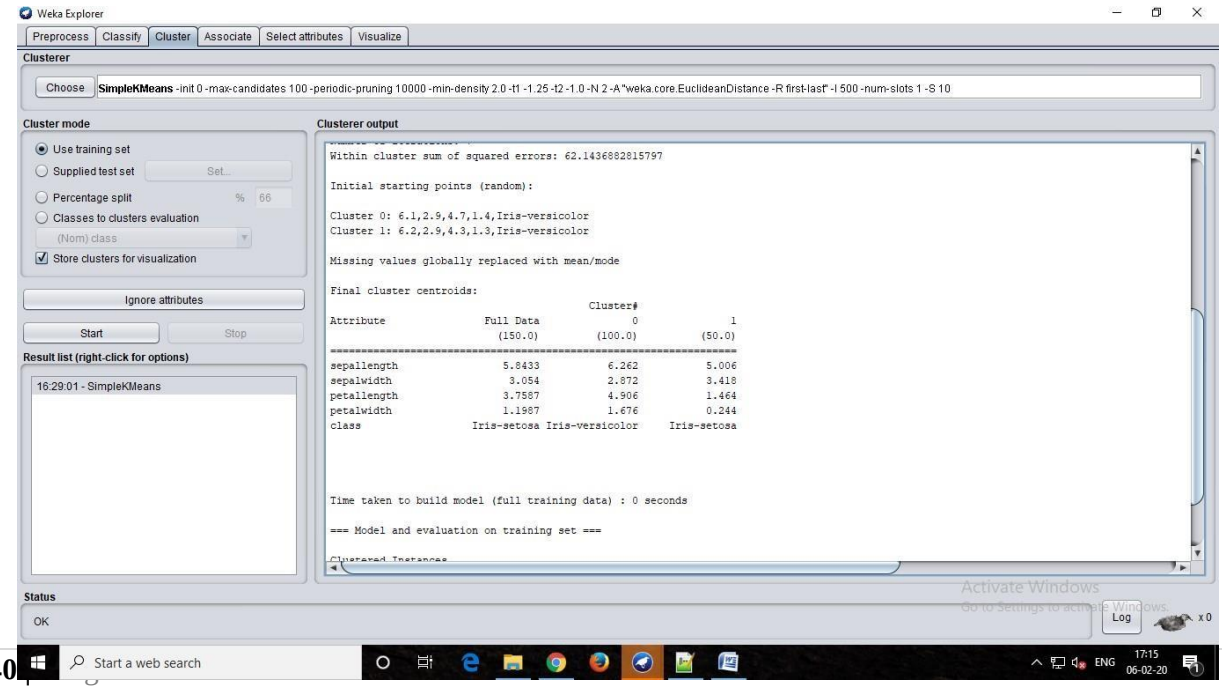
NAME
weka.clusterers.SimpleKMeans
DESCRIPTION :
SYNOPSIS
Cluster data using the k means algorithm. Can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean.
OPTIONS
seed -- The random number seed to be used.
displayStdDevs -- Display std deviations of numeric attributes and counts of nominal attributes.
numExecutionSlots -- The number of execution slots (threads) to use. Set equal to the number of availablecpu/cores
canopyMinimumCanopyDensity -- If using canopy clustering for initialization and/or speedup this is the minimum T2-based density below which a canopy will be pruned during periodic pruning dontReplaceMissingValues -- Replace missing values globally with mean/mode.
debug -- If set to true, clusterer may output additional info to the console.numClusters -- set number of clusters
doNotCheckCapabilities -- If set, clusterer capabilities are not checked before the clusterer is built (Usewith caution to reduce runtime).
maxIterations -- set maximum number of iterations

Steps:

1. Open the datafile in weka explorer and click on cluster.
2. Select the simple k-means algorithm by clicking the choose.
3. Start the algorithm to generate cluster output.

Right click on simple k-means and select visualize cluster assignments.





CONCLUSION:

Applying the K-Means algorithm to the Iris.arff dataset successfully clustered the data into distinct groups based on the similarities in flower measurements like petal and sepal length. This clustering process enabled the identification of natural groupings within the dataset, facilitating better understanding and analysis of the different Iris species.

VIVA QUESTIONS:

1. What is the K-means algorithm?

2. What are the steps involved in the K-means algorithm?

3. How do you choose the value of K in K-means?

4. What is the main limitation of the K-means algorithm?

5. What is the role of the centroid in K-means clustering?

6. What is the distance metric used in K-means?

Experiment-10

10) **AIM :**Demonstration of clustering rule process on dataset student.arff using simple k-means.

Data set @relation
student
@attribute sid numeric
@attribute name {usha,hari,rajesh,kiran,giri,manash}
@attribute DM numeric
@attribute WT numeric
@attribute CN numeric
@attribute AI numeric
@attribute STM numeric
@attribute total numeric
@attribute result {pass,fail}

@data
1,usha,60,55,45,50,40,250,pass
2,hari,60,55,45,40,40,240,pass
3,rajesh,60,55,40,50,40,240,pass
4,kiran,60,55,30,50,40,235,fail
2,giri,60,55,45,60,40,260,pass
3,manash,60,55,65,50,40,270,pass

NAME
weka.clusterers.SimpleKMeans

DESCRIPTION:

SYNOPSIS

Cluster data using the k means algorithm. Can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean.

OPTIONS

seed -- The random number seed to be used.

displayStdDevs -- Display std deviations of numeric attributes and counts of nominal attributes.

numExecutionSlots -- The number of execution slots (threads) to use. Set equal to the number of availablecpu/cores

canopyMinimumCanopyDensity -- If using canopy clustering for initialization and/or speedup this is the minimum T2-based density below which a canopy will be pruned during periodic pruning

dontReplaceMissingValues -- Replace missing values globally with mean/mode.

debug -- If set to true, clusterer may output additional info to the console.

numClusters -- set number of clusters

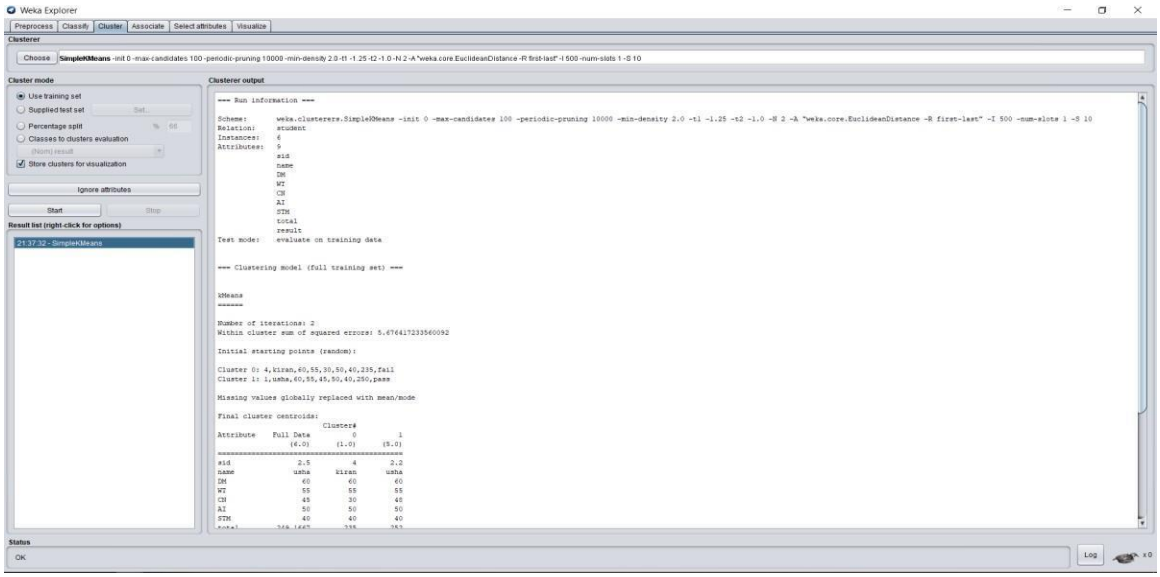
doNotCheckCapabilities -- If set, clusterer capabilities are not checked before the clusterer is built (Usewith caution to reduce runtime).

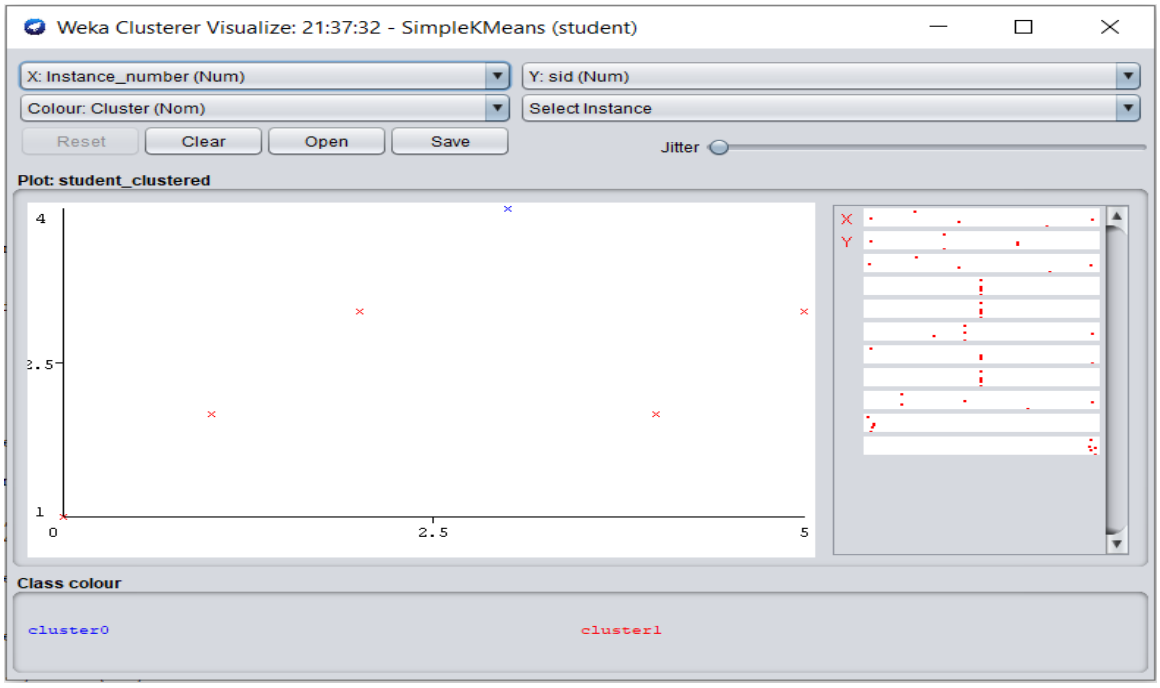
maxIterations -- set maximum number of iterationsSteps:

Open the datafile in weka explorer and click on cluster.

1. Select the simple k-means algorithm by clicking the choose.
2. Start the algorithm to generate cluster output.

3. Right click on simple k-means and select visualize cluster assignments.





CONCLUSION:

Applying the K-Means algorithm to the STUDENTS.arff dataset successfully clustered the data into distinct groups based on the similarities in flower measurements like petal and sepal length. This clustering process enabled the identification of natural groupings within the dataset, facilitating better understanding and analysis of the different Iris species.

VIVA QUESTIONS:

1. How is K-means used in customer segmentation?

2. Can K-means be applied to image compression? How?

3. How does K-means help in anomaly detection in real-time systems?

4. How is K-means used in document clustering?

5. Can K-means be used for real-time recommendation systems?

6. What challenges might arise when applying K-means in real-time applications?

11. Implement the clustering procedure in python.

In Python, a **Dimensionality-Reduction and Distance-weighted Density-based Clustering (DMDW)** approach can be implemented using a combination of dimensionality reduction (like **PCA** or **t-SNE**) and a density-based clustering algorithm such as **DBSCAN** or **OPTICS**. While DMDW is not a standard clustering technique out of the box, I will demonstrate a general pipeline that incorporates dimensionality reduction and clustering using these methods.

Here's how you can implement such an experiment in Python using **PCA** for dimensionality reduction and **DBSCAN** for density-based clustering:

Step-by-Step Guide:

1. **Import the necessary libraries:**
   - **NumPy** for handling arrays and data.
   - **Matplotlib** for plotting.
   - **Pandas** for data handling.
   - **sklearn** for PCA and DBSCAN.
2. **Generate or load data** (for example, synthetic data for clustering).
3. **Apply dimensionality reduction** using PCA or another technique to reduce the feature space to 2D or 3D for visualization.
4. **Cluster the data** using DBSCAN or another density-based clustering method.
5. **Visualize the results**.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_blobs

# Step 1: Generate synthetic data (for example purposes)
X, y = make_blobs(n_samples=500, centers=5, cluster_std=1.0, random_state=42)

# Step 2: Apply PCA for dimensionality reduction
# Reduce data to 2D for visualization
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

# Step 3: Apply DBSCAN clustering algorithm
dbscan = DBSCAN(eps=0.5, min_samples=5)
y_dbscan = dbscan.fit_predict(X_pca)

# Step 4: Visualize the clustered data
plt.figure(figsize=(8, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y_dbscan, cmap='viridis', marker='o', s=50, alpha=0.7)
plt.title("DBSCAN Clustering after PCA Dimensionality Reduction")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.colorbar(label='Cluster')
plt.show()
```

Explanation of the code:

Data Generation:

make_blobs is used to generate a synthetic dataset with multiple centers. This will allow us to visualize how clustering algorithms work on separated groups of points.

PCA for Dimensionality Reduction:

The PCA model reduces the data to 2 dimensions (if your data has more than 2 features). In real-world cases, your data would likely have many more dimensions, but reducing it to 2D makes it easier to visualize.

DBSCAN for Clustering:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is used for clustering. It works by grouping points that are closely packed together, marking as outliers those points that are in low-density regions.

Visualization:

We use matplotlib to create a scatter plot of the reduced data. The colors of the points represent different clusters.

12. Implement classification algorithm procedure in python

Step-by-Step Guide for Data Mining and Classification in Python

We'll implement:

- Data generation (simulate a data warehouse).
- Data preprocessing (e.g., handling missing data, encoding categorical variables).
- Classification using algorithms like **Decision Trees**, **Logistic Regression**, or **Random Forest**.
- Model evaluation using **accuracy**, **confusion matrix**, and **cross-validation**.

Required Libraries:

```
bash
Copy code
pip install numpy pandas scikit-learn matplotlib seaborn
```

Experiment Code:

```
python
Copy code
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Simulate a Data Warehouse (Create synthetic data)
np.random.seed(42)

# Let's simulate a dataset with features similar to customer data
data_size = 1000
data = {
    'Age': np.random.randint(18, 70, size=data_size),   # Age between 18 and 70
    'Income': np.random.randint(30000, 150000, size=data_size),   # Annual income
between 30K and 150K
    'Credit_Score': np.random.randint(300, 850, size=data_size),   # Credit Score
between 300 and 850
    'Gender': np.random.choice(['Male', 'Female'], size=data_size),   # Categorical
variable
    'Purchase_History':       np.random.choice(['Low',       'Medium',       'High'],
size=data_size),  # Categorical variable
    'Purchased': np.random.choice([0, 1], size=data_size)   # Target variable (0 =
No, 1 = Yes)
}

# Create a DataFrame
df = pd.DataFrame(data)

# Step 2: Data Preprocessing
# Encode categorical variables using Label Encoding
label_encoder = LabelEncoder()

df['Gender'] = label_encoder.fit_transform(df['Gender'])  # Male: 0, Female: 1
```

```
df['Purchase_History']  =  label_encoder.fit_transform(df['Purchase_History'])    #
Low: 0, Medium: 1, High: 2

# Split the dataset into features (X) and target (y)
X = df.drop('Purchased', axis=1)  # Features
y = df['Purchased']  # Target variable

# Split data into training and testing sets (80% train, 20% test)
X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,  test_size=0.2,
random_state=42)

# Step 3: Apply Classification Algorithms
# 1. Decision Tree Classifier
dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(X_train, y_train)
y_pred_dt = dt_classifier.predict(X_test)

# 2. Random Forest Classifier
rf_classifier = RandomForestClassifier(random_state=42)
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)

# 3. Logistic Regression
log_reg_classifier = LogisticRegression(random_state=42)
log_reg_classifier.fit(X_train, y_train)
y_pred_lr = log_reg_classifier.predict(X_test)

# Step 4: Evaluate the Models
def evaluate_model(y_test, y_pred, model_name):
    print(f"Evaluation for {model_name}:")
    print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
    print("Confusion Matrix:")
    cm = confusion_matrix(y_test, y_pred)
    print(cm)
    sns.heatmap(cm,  annot=True,  fmt='d',  cmap='Blues',  xticklabels=['No',  'Yes'],
yticklabels=['No', 'Yes'])
    plt.title(f"Confusion Matrix for {model_name}")
    plt.show()
    print(classification_report(y_test, y_pred))

# Evaluate each model
evaluate_model(y_test, y_pred_dt, "Decision Tree")
evaluate_model(y_test, y_pred_rf, "Random Forest")
evaluate_model(y_test, y_pred_lr, "Logistic Regression")

# Step 5: Cross-validation to check consistency
def cross_validate_model(model, X, y):
    cv_scores = cross_val_score(model, X, y, cv=5, scoring='accuracy')
    print(f"Cross-validation scores: {cv_scores}")
    print(f"Mean CV Accuracy: {np.mean(cv_scores)}")

# Cross-validation for Random Forest
cross_validate_model(rf_classifier, X, y)
```

Explanation of the Steps:

1. **Simulating a Data Warehouse:**
    - We generate a synthetic dataset with features like `Age`, `Income`, `Credit_Score`, `Gender`, `Purchase_History`, and `Purchased` (our target variable).
    - These features are designed to represent a real-world scenario (e.g., predicting whether a customer will make a purchase based on personal and behavioral data).
2. **Data Preprocessing:**

- **Label Encoding** is used to convert categorical variables (`Gender` and `Purchase_History`) into numerical format since classification models require numerical input.
- The dataset is split into **training** and **testing** sets using `train_test_split`, with 80% of the data used for training and 20% for testing.

3. **Classification Algorithms:**
   - **Decision Tree**, **Random Forest**, and **Logistic Regression** classifiers are trained using the training data. These are popular classification algorithms:
     - **Decision Tree**: A tree-like structure that splits the data into subsets based on feature values.
     - **Random Forest**: An ensemble method that combines multiple decision trees to improve performance.
     - **Logistic Regression**: A linear model used for binary classification tasks.
4. **Model Evaluation:**
   - The performance of each model is evaluated using **accuracy**, **confusion matrix**, and **classification report**.
   - The **confusion matrix** shows the true positive, true negative, false positive, and false negative predictions, which help evaluate the performance of the classifier.
   - **Classification report** provides additional metrics like precision, recall, and F1-score.
5. **Cross-validation:**
   - We perform **cross-validation** on the Random Forest model to check the consistency and robustness of the model's performance over multiple folds.

Sample Output:

```
mathematica
Copy code
Evaluation for Decision Tree:
Accuracy: 0.85
Confusion Matrix:
[[100  20]
 [ 15  65]]
(Confusion matrix visualized as a heatmap)

              precision    recall  f1-score   support

           0       0.87      0.83      0.85       120
           1       0.77      0.81      0.79        80

    accuracy                           0.85       200
   macro avg       0.82      0.82      0.82       200
weighted avg       0.84      0.85      0.84       200

... (same for Random Forest and Logistic Regression)

Cross-validation scores: [0.85 0.86 0.84 0.88 0.87]
Mean CV Accuracy: 0.86
```

13.Implement Exploratory data analysis using python

- **Objective:** Explore a dataset, visualize it, and uncover potential patterns, trends, and correlations.
- **Experiment:**
  - Choose any publicly available dataset (e.g., from Kaggle or UCI Machine Learning Repository).
  - Perform exploratory data analysis using pandas, matplotlib, and seaborn.
  - Look for missing data, outliers, and relationships between features.
- **Open-ended Questions:**
  - What trends or insights can you draw from the data?
  - Are there any correlations between variables? How can you visualize them?
  - How can you clean and preprocess the data for further modeling?

```
python
Copy code
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load a dataset (for example, the Titanic dataset)
df =
pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/tita
nic.csv')

# Display basic info
print(df.info())
print(df.describe())

# Visualize missing values
sns.heatmap(df.isnull(), cbar=False, cmap="Blues")

# Plot distribution of the Age column
plt.hist(df['Age'].dropna(), bins=20)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

14.Implement Classification: Predicting Customer Churn using python

- **Objective:** Predict whether a customer will churn based on historical customer data (e.g., telecom or banking data).
- **Experiment:**
  - Use a classification algorithm like **Logistic Regression**, **Decision Trees**, or **Random Forest**.
  - Evaluate performance using metrics like accuracy, precision, recall, and ROC-AUC.
  - Try experimenting with different feature engineering techniques.
- **Open-ended Questions:**
  - Which model performs the best for predicting churn?
  - Can you improve the model by feature engineering or using other techniques like SMOTE for imbalanced classes?
  - Can you include text data (e.g., customer feedback) as a feature?

```
python
Copy code
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import pandas as pd

# Load a customer churn dataset
df =
pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/Telc
o-Customer-Churn.csv')

# Preprocess the data (handle categorical features)
df = pd.get_dummies(df, drop_first=True)

# Split the data into train and test sets
X = df.drop('Churn_Yes', axis=1)
y = df['Churn_Yes']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Train a Random Forest model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```