MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

Banjara Hills, Hyderabad, Telangana



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Data Mining Laboratory Manual



Academic Year 2016-2017

Table of Contents

I Contents

1.	Vision of the Institution	i
2.	Mission of the Institution	i
3.	Department Vision	ii
4.	Department Mission	ii
5.	Programme Education Objectives	iii
6.	Programme Outcomes	iv
7.	Programme Specific Outcomes	v
8.	Introduction	vi

II Programs

1.	Perform data preprocessing using weka	1
2.	Obtain decision tree for different data sets using WEKA	28
3.	Classification algorithms using WEKA	37
4.	Obtain The Association Rules For Given Dataset Using WEKA	45
5.	Perform data transformations using an ETL tool.	49
6.	PROGRAM-6: Write a program to implement star schema.	59
7.	PROGRAM-6A: write a program to implement K-Means algorithm	64
8.	PROGRAM-6B: K-Means algorithm for two dimension data items.	67
9.	a program to implement k-mediod algorithm	70
10.	CASE STUDY 1- K-NEAREST NEIGHBOR	74
11.	CASE STUDY 2 - KDD PROCESS	77
12.	DECISION TREE INDUCTION	81
13.	Case Study On Apriori	85

Part I

Contents

1. Vision of the Institution

To be part of universal human quest for development and progress by contributing high calibre, ethical and socially responsible engineers who meet the global challenge of building modern society in harmony with nature.

2. Mission of the Institution

- To attain excellence in imparting technical education from undergraduate through doctorate levels by adopting coherent and judiciously coordinated curricular and co-curricular programs.
- To foster partnership with industry and government agencies through collaborative research and consultancy.
- To nurture and strengthen auxiliary soft skills for overall development and improved employability in a multi-cultural work space.
- To develop scientific temper and spirit of enquiry in order to harness the latent innovative talents.
- To develop constructive attitude in students towards the task of nation building and empower them to become future leaders
- To nourish the entrepreneurial instincts of the students and hone their business acumen.
- To involve the students and the faculty in solving local community problems through economical and sustainable solutions.

3. Department Vision

To contribute competent computer science professionals to the global talent pool to meet the constantly evolving societal needs.

4. Department Mission

Mentoring students towards a successful professional career in a global environment through quality education and soft skills in order to meet the evolving societal needs.

5. Programme Education Objectives

- 1. Graduates will demonstrate technical skills and leadership in their chosen fields of employment by solving real time problems using current techniques and tools.
- 2. Graduates will communicate effectively as individuals or team members and be successful in the local and global cross cultural working environment.
- 3. Graduates will demonstrate lifelong learning through continuing education and professional development.
- 4. Graduates will be successful in providing viable and sustainable solutions within societal, professional, environmental and ethical contexts

6. Programme Outcomes

- 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. Lifelong learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

7. Programme Specific Outcomes

The graduates will be able to:

- **PSO1:** Demonstrate understanding of the principles and working of the hardware and software aspects of computer systems.
- **PSO2:** Use professional engineering practices, strategies and tactics for the development, operation and maintenance of software
- **PSO3:** Provide effective and efficient real time solutions using acquired knowledge in various domains.

8. Introduction

Data Mining is defined as the procedure of extracting information from huge sets of data. In other words, we can say that data mining is mining knowledge from data. The tutorial starts off with a basic overview and the terminologies involved in data mining and then gradually moves on to cover topics such as knowledge discovery, query language, classification and prediction, decision tree induction, cluster analysis, and how to mine the Web.

<u>Audience</u>

This tutorial has been prepared for computer science graduates to help them understand the basic-to-advanced concepts related to data mining.

Prerequisites

Before proceeding with this tutorial, you should have an understanding of the basic database concepts such as schema, ER model, Structured Query language and a basic knowledge of Data Warehousing concepts.

Applications

Data mining is highly useful in the following domains

- Market Analysis and Management
- Corporate Analysis & Risk Management
- Fraud Detection
- Production Control
- Science Exploration

Tools

Audience

This tutorial has been prepared for computer science graduates to help them understand the basic-to-advanced concepts related to data mining.

Prerequisites

Before proceeding with this tutorial, you should have an understanding of the basic database concepts such as schema, ER model, Structured Query language and a basic knowledge of Data Warehousing concepts.

Applications

Data mining is highly useful in the following domains

- Market Analysis and Management
- Corporate Analysis & Risk Management

- Fraud Detection
- Production Control
- Science Exploration

Tools

Large variety of DM software is now available. Some more widely used software is:

- IBM Intelligent Miner
- Weka(Waikato Environment of Knowledge Analysis)
- Cognos
- Informatica
- SAS Enterprise Miner
- Silicon Graphics Mine Set
- Oracle Thinking Machines Darwin
- Angoss knowledge SEEKER

Software and Hardware Requirements

HARDWARE SPECIFICATIONS:

Processor : Pentium Dual Core

RAM : 1GB

Hard Disk : 160GB

SOFTWARE SPECIFICATIONS:

- Operating System : Windows XP
- Utility Software : Java 5.0 and above
- Data Mining Software : weka 3.6.13

What Motivated Data Mining

- Data collection and data availability
 - Automated data collection tools, Database systems, Web, computerized society
- Major sources of abundant data
 - Business: Web, e-commerce, transactions, stocks,
 - Science: Remote sensing, bioinformatics, scientific simulation,
 - Society and everyone: news, digital cameras, YouTube

Architecture of Data Mining Process



Part II Programs

Program 1

DATA PROCESSING USING WEKA

Problem Definition

Perform data preprocessing using weka

Problem Description

Data Preprocessing in Data mining is a technique that involves transforming raw data into an understandable form. It is used to filter out the data in the desired form by using techniques like cleaning, normalization, transformation, feature extraction and selection, etc .

- Data in the real world is dirty
 - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 e.g., occupation="""
 - noisy: containing errors or outliers e.g., Salary=-10
 - inconsistent: containing discrepancies in codes or names e.g., Age=42 Birth-day=03/07/1997
 - e.g., Was rating 1,2,3, now rating A, B, C
 - e.g., discrepancy between duplicate records

Following is the demonstration with respect to preprocessing commands:

1. <u>ADD</u>:

DESCRIPTION: An instance filter that adds a new attribute to the dataset. The new attribute will contain all missing values.

OPTIONS

Attribute Index – The position (starting from 1) where the attribute will be inserted (first and last are valid indices).

Attribute Name – Set the new attribute's name.

Attribute Type – Defines the type of the attribute to generate.

STEPS

Figure 1: Start Weka you get the Weka GUI chooser window.



Figure 2: Click on the Explorer button and you get the Weka Knowledge Explorer window.

) Weka Explorer				
reprocess Classify Cluster Associ	iate Select attributes Visualize			
Open file Open URL	Open DB Gen	erate Und	do) [Edit	Save
filter Choose None				Apply
Current relation Relation: None Instances: None	Attributes: None Sum of weights: None	Selected attribute Name: None Missing: None	Distinct: None	Type: None Unique: None
ttributes				
All None	Invert Pattern			
	<u></u>			
		-		Visualize a
Rei	move			
rahue.				
Nelson - to the Utalia Produces				100

You see here the default parameters of this filter. Click on More to get more

Figure 3: Click on the Open File. button and open an ARFF file (try it first with an example supplied in Weka-3-6/data, e.g. Weather.arff). You get the following:

reprocess	Classify Cluster Associate Select attribute	s Visualize			
Open fi	Open Open Ger	iera	Undo	Edit	Save
Filter					
Choose	None				Apply
Current relati Relation: Instances:	on weather Attributes: 5 14 Sum of weights: 14	Selected at Name: o Missing: 0	tribute utlook (0%) Di:	T stinct: 3 Uni	ype: Nominal que: 0 (0%)
Attributes		No.	Label	Count	Weight
		1	sunny	5	5.0
All	None Invert Patt	2	overcast	4	4.0
No.	Name	3	rainy	5	5.0
11	outlook	Class: play ()	Nom)	+	Visualize All
2	temperature		1999 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		
3	humidity				
4	windy	5	18	5	
5	play		4		
[Remove				

Figure 4: VALIDATION : Click on Choose and select filters/unsupervised/attribute/Add.

Open fi Open Open	Gener	a	Undo	Edit	Save
ilter				7	
📗 weka	*				Apply
🖹 퉲 filters		elected at	tribute		
- AllFilter		Name: 0	utlook	1	vne: Nominal
MultiFilter		Missing: 0	(0%) Di	stinct: 3 Un	ique: 0 (0%)
		No	Label	Count	Weight
attribute	E	1	Laber	le	l co
Add	11	1	sunny	3	3.0
- AddCluster		2	rainy	5	5.0
AddExpression			Trainty	12	10.0
- AddID		ass: play (Nom)	*	Visualize Al
AddNoise	1	1	6 8		
 AddValues 					
Center		5	-	6	
ChangeDateFormat			4		
ClassAssigner					
ClusterMembership					
Сору					
- Discretize					
FirstOrder				Log	Anton
1 Mit Transformer and Annual Market Physics and a 1		1		-	A

information about these parameters.

Try other parameters for the filter and see how the Addition changes. Dont forget to reload the original (numeric) relation or Undo the Addition before applying another one.

eka.niters.unsu	pervised.attribute.Add	
About		
An instance	filter that adds a new attribute to the dataset.	More
		Capabilities
attributeIndex	last	
attributeName	unnamed	
attributeType	Numeric attribute	,
dateFormat	yyyy-MM-dd'T'HH:mm:ss	
nominalLabels		

Figure 5: Then click on the area right of the Choose button. You get the following:

Figure 6: Click on the Apply button to do the Addition and see how it is Added in the Selected attribute window.

Preprocess	Classify Cluster Associate Se	elect attributes Visua	lize	
Ope	Ope	Gen Undo	Edit	Sav
Filter				
Choose	Add -N unnamed -C last			Apply
Current relati Relation: Instances:	on Attributes: 6 14 Sum of weights: 1	Selected attribu Name: unna 4 Missing: 14(1	ite med Distinct: U	Type <mark>:</mark> N Inique: 0
Attributes		Statistic	Value	
		Minimum	NaN	
		Maximum	NaN	0
No.	Name	Mean	NaN	
2	temperature	 Class: unnamed 	(Num) 👻 🗍	Visualize All
3	humidity		<u> </u>	
4	windy	III		
5	play			
6	unnamed	<u> </u>		
	Remove	<u> </u>		
V°.		NaN	NaN	Na
Status				
01/			100	and the V

2. Discretization :

DESCRIPTION: An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes. Discretization is by simple binning. Skips the class attributes if set.

OPTIONS

Attribute Indices – 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".

bins – Number of bins.

Desired Weight Of Instances Per Interval – Sets the desired weight of instances per interval for equal-frequency binning.

Find Num Bins – Optimize number of equal-width bins using leave-one-out. Doesn't work for equal-frequency binning

ignore Class – The class index will be unset temporarily before the filter is applied.

Invert Selection – Set attribute selection mode. If false, only selected (numeric) attributes in the range will be discretized; if true, only non-selected attributes will be discretized.

Make Binary – Make resulting attributes binary.

Use Equal Frequency - If set to true, equal-frequency binning will be used instead of equal-width binning.

STEPS

a. Start Weka you get the Weka GUI chooser window.



b. Click on the Explorer button and you get the Weka Knowledge Explorer window.

🖉 Weka Explorer	
Preprocess Classify Cluster Associate Select attributes Visualize	
Open file Open URL Open DB Gener	rate Undo Edit Save
Filter	
Choose None	Apply
Current relation Relation: None Attributes: None Instances: None Sum of weights: None	Selected attribute Name: None Type: None Missing: None Distinct: None Unique: None
Attributes	
All None Invert Pattern	Visualize All
Remove Status	
Welcome to the Weka Explorer	Log ×0

c. **VALIDATION**: Click on the Open File. button and open an ARFF file (try it first with an example supplied in Weka-3-6/data, e.g. Weather.arff). You get the following:

🖉 Weka Explorer	
Preprocess Classify Cluster Associate Select attributes Visualize	
Open file Open URL Open DB Gener	ate Undo Edit Save
Choose None	Apply
Current relation Relation: labor-neg-data Attributes: 17 Instances: 57 Sum of weights: 57	Selected attribute Name: duration Type: Numeric Missing: 1 (2%) Distinct: 3 Unique: 0 (0%)
Attributes	Statistic Value
	Minimum 1
All None Invert Pattern	Maximum 3
	Mean 2.161
No. Name	StdDev 0.708
2 wage-increase-first-year 3 wage-increase-second-year 4 wage-increase-third-year	
6 Working-bours	Class: class (Nom) Visualize All
7 Dension	
8 standby-pay	
9 shift-differential	
10 education-allowance	
11 statutory-holidays	19
12 vacation	
13 longterm-disability-assistance	10
14 contribution-to-dental-plan	
Remove	
	1 2
Status OK	Log

 ${\rm d.}\ {\bf VALIDATION: Click \ on \ Choose \ and \ select \ filters/unsupervised/ \ attribute/Discretize.}$



e. Then click on the area right of the Choose button. You get the following:

veka.filters.unsupervised.attribute.Disc	retize	
About An instance filter that discretizes attributes in the dataset into nom	a range of numeric inal attributes.	More Capabilities
attributeIndices	first-last	
bins	10	
desiredWeightOfInstancesPerInterval	-1.0	
findNumBins	False	•
ignoreClass	False	•
invertSelection	False	
makeBinary	False	¥
useEqualFrequency	False	-

You see here the default parameters of this filter. Click on More to get more information about these parameters.

f. **VALIDATION**: Click on the Apply button to do the discretization. Then select one of the original numeric attributes (e.g. temperature) and see how it is discretized in the Selected attribute window..

eprocess C	lassify Cluster Associate Select attributes Visualize						
Open file	Open URL Open DE Determine r	elevano	e of attribute	s Undo	Edit	. Save	
ter							
Choose	Discretize -B 10 -M -1.0 -R first-last						Apply
urrent relati	20		Selected at	tribute			
Relation: I Instances: S	abor-neg-data-weka.filters.unsuper Attributes 57 Sum of weights	: 17 : 57	Name: o Missing:	duration 1 (2%)	Distinct: 3	Type: Nominal Unique: 0 (0%)	
ttributes			No.	Label	Count	Weight	
and the second				(//-inf-1.21	10	10.0	
All	None Invert Pattern			2 '(1.2-1.4]'	0	0.0	-
				3 '(1.4-1.6]'	0	0.0	1
Vo.	Name		4	+ '(1.6-1.8]'	0	0.0	1
1	duration	~	9	5 '(1.8-2]'	27	27.0	
2	wage-increase-first-year		6	5 (2-2.2)	0	0.0	
3	wage-increase-second-year	-	7	7 '(2.2-2.4]'	0	0.0	1
4	wage-increase-third-year			3 12.4-2.61	0	0.0	
5	cost-of-living-adjustment		Class: class ((Nom)		Vieus	aliza
6	working-hours	- 11	Cid331 Cid331	(wom)		*13dc	11120
7	pension						
8	standby-pay				27		
9	shift-differential						
10	education-allowance						
11	statutory-holidays	_					19
12	vacation						
13	longterm-disability-assistance	-					
14	[contribution-to-dental-plan		10				
	Domouro						
	INGINE Y G						

g. Try other parameters for the filter and see how the discretization changes. Dont forget to reload the original (numeric) relation or Undo the discretization before applying another one.

3. Normalize:

DESCRIPTION:

Normalizes all numeric values in the given data set (apart from the class attribute, if set). The resulting values are by default 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 values in the range [-1,+1].

OPTIONS

ignoreClass – The class index will be unset temporarily before the filter is applied.

scale – The factor for scaling the output range (default: 1).

translation – The translation of the output range (default: 0).

STEPS

a. Start Weka you get the Weka GUI chooser window.



b. Click on the Explorer button and you get the Weka Knowledge Explorer window.

reprocess Classify Cluster	Associate Select attributes Visualize			
Open file Op Filter Choose None	pen URL Open DB Gene	rate Und	o) [Edit	Save
Current relation Relation: None Instances: None	Attributes: None Sum of weights: None	Selected attribute Name: None Missing: None	Distinct: None	Type: None Unique: None
All No	ne Invert Pattern			✓ Visualze All
	Remove			
tatus Welcome to the Weka Explor	er			Log

c. Click on the Open File. button and open an ARFF file (try it first with an example supplied in Weka-3-6/data, e.g. whether.arff). You get the following:

reprocess	Classify Cluster Associate Select attribu	tes Visualize			
Open fi	Open Open G	enera	Undo	Edit	Save
Filter					
Choose	None				Apply
Current rel Relatior Instances	ation : weather Attributes: 5 : 14 Sum of weights: 14	Selected at Name: o Missing: 0	tribute utlook (0%) Dis	T stinct: 3 Uni	ype: Nominal que: 0 (0%)
Attributes		No.	No. Label		Weight
All None Invert Patt		1	sunny	5	5.0
L		2	overcast	4	4.0
No.	Name	3	rainy	5	5.0
	outlook.	Class: play (Nom)	*	Visualize All
1	temperature				(
	humidity			50	
13	windy	5		5	
	E play	4	4		
	Remove				
Status					
10000				100	and the second

d. Click on Choose and select filters/unsupervised/attribute/Normalize.



Relati	on: weather				
No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

I

e. Then click on the area right of the Choose button. You get the following:

eka.filters.u	insupervised.attribute.Normalize
About	
Normalize	es all numeric values in the given dataset (apart from More
the class	attribute, if set). Capabilities
e 704	reave-
ignoreClass	False
scale	1.0
scale	1.0

You see here the default parameters of this filter. Click on More to get more information about these parameters.

f. **VALIDATION**: Click on the Apply button to do the normalization. Then select edit tab to view data and see how it is normalized in the data window.

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	1.0	0.645161	FALSE	no
2	sunny	0.761904761	0.806451	TRUE	no
3	overcast	0.904761904	0.677419	FALSE	yes
4	rainy	0.285714285	1.0	FALSE	yes
5	rainy	0.190476190	0.483870	FALSE	yes
6	rainy	0.047619047	0.161290	TRUE	no
7	overcast	0.0	0.0	TRUE	yes
8	sunny	0.380952380	0.967741	FALSE	no
9	sunny	0.238095238	0.161290	FALSE	yes
10	rainy	0.523809523	0.483870	FALSE	yes
11	sunny	0.523809523	0.161290	TRUE	yes
12	overcast	0.380952380	0.806451	TRUE	yes
13	overcast	0.809523809	0.322580	FALSE	yes
14	rainy	0.333333333	0.838709	TRUE	no

g. Try other parameters for the filter and see how the normalization changes. Dont forget to reload the original (numeric) relation or Undo the normalization before applying another one.

4. <u>**Remove**</u>:

DESCRIPTION:

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

Attribute Indices – 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".

Invert Selection – 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.

STEPS

a. Start Weka you get the Weka GUI chooser window.



b. Click on the Explorer button and you get the Weka Knowledge Explorer window.

Preprocess Classify Cluster Associate Select attributes Visualize Copen file Copen URL Copen DB Generate Undo Filter Choose None Current relation Instances: None Sum of weights: None Attributes All None Invert Pattern	
Open Ile Open DB Generate Undo Fiker Choose None Relation: None None Current relation Attributes: None Selected attribute Missing: None Distinct: Attributes All None Invert Pattern Invert Invert	
Filer Choose None Current relation Attributes: None Name: None Instance: None Sum of weights: None Name: None All None Invent	Edit Save
Current reliation Current reliation None Instance: None Attributes All None Invent Pattern	1 martin
All None Invert Pattern	Poppy
All None Invert Pattern	Type: None one Unique: None
All None Invert Pattern	
	Visualize A
REDEVE	

c. Click on the Open File. button and open an ARFF file (try it first with an example supplied in Weka-3-6/data, e.g. Weather.arff). You get the following:

reprocess	Classify Cluster Associate Select attribute	visualize			
Open fi.	Open Open Ger	nera	Undo	Edit	Save
Filter					<u> </u>
Choose	None				Apply
Current rel Relation Instances	ation : weather Attributes: 5 : 14 Sum of weights: 14	Selected at Name: o Missing: 0	tribute utlook (0%) Dis	T stinct: 3 Uni	ype: Nominal que: 0 (0%)
Attributes		No.	Label	Count	Weight
	None Invert Patt	1	sunny	5	5.0
		2	overcast	4	4.0
No.	Name	3	rainy	5	5.0
	n outlook	Class: play (Nom)	•	Visualize All
1	temperature				
	humidity	59			
in the second se	windy	5	10	5	
	play		4		
[Remove .				

d. Click on Choose and select filters/unsupervised/attribute/Remove.

Data	Mining	Lab	Manual	l
Dava		Luo	manaa	-

eproces:	Classify Cluster Associate Select a	ttributes	Visua	lize				
Open f	ile Open UR Open DB.		Gener	ate		Undo	Edit	Save
lter								
	FirstOrder	~	1					Apply
	InterquartileRange KernelFiter MakeIndicator		4	Selected Name Missing	d attr e: ou g: 0 (ribute itlook (0%) [Distinct: 3	Type: Nominal Unique: 0 (0%)
	Mathexpression MergeMapyValues			No.		Label	Count	Weight
	MergeTwoValues		h	0000000	1	sunny	5	5.0
	NominalToBinary				2	overcast	4	4.0
	NominalToString				3	rainy	5	5.0
	Normalize NumericCleaner NumericToBinary NumericToNominal NumericTransform							
	Obfuscate PartitionedMultiFilter	E		Class: pla	ay (N	lom)		▼ Visualize A
	PKIDiscretize			5				5
	PrincipalComponents BandemBreigetien			ř				Ĭ.
	RandomSubset					4		
	- Remove							
	RemoveByName							
	and the second second second part of the second sec							

e. Then click on the area right of the Choose button. You get the following:

veka.filters.unsup	ervised.attribute.Remove	
About		
A filter that rer	noves a range of attributes from the dataset.	More
		Capabilities
attributaIndicas	le .	Capabilities
attributeIndices	5	Capabilities

f. VALIDATION : Click on the Apply button to do the remove.

You see here the default parameters of this filter. Enter the Indices of attribute to be remove Click on more to get more information about these parameters.

Open file Open UR Open DB Gen	erate] [Undo	Edit	Save
Choose Remove -R 5				Apply
rrent relation Relation: weather-weka.filters.un Attributes: 4 Instances: 14 Sum of weights: 14	Selected at Name: 0 Missing: 0	ttribute outlook 0 (0%) D	listinct: 3	Type: Nominal Unique: 0 (0%)
tributes	No.	Label	Count	Weight
All None Invert Pattern		1 sunny 2 overcast	5	5.0 4.0
o. Name		3 rainy	5	5.0
1 📰 outlook				
2 temperature	2			
3 humidity 4 windy	ŝ			
	Class: windy	y (Nom)		▼ Visualize A

g. Try other parameters for the filter and see how the remove changes. Dont forget to reload the original (numeric) relation or Undo the remove before applying another one.

h. Replace Missing Values

Description: Replaces all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data.

OPTIONS

Ignore Class – The class index will be unset temporarily before the filter is applied.

STEPS

i. Start Weka you get the Weka GUI chooser window.



ii. Click on the Explorer button and you get the Weka Knowledge Explorer window.

) Weka Explorer	
Open file Open URL Open DB C	enerate Undo Edit Save
Choose None	Apply
urrent relation Relation: None Attributes: Non Instances: None Sum of weights: Non	Selected attribute Name: None Type: None Missing: None Distinct: None Unique: None
All None Invert Pattern	Visualize A
Remove	

iii. Click on the Open File. button and open an ARFF file (try it first with an example supplied in Weka-3-6/data, e.g. weather.arff). You get the following:

Open file Open URL Open DB Gener	ate	Undo	Edit		Save
Filter) <u>(</u>			
Choose ReplaceMissingValues -unset-class-temporarily					Apply
Current relation Relation: marks-weka.filters.unsupervised.attri Attributes: 2 Instances: 7 Sum of weights: 7	Selected Name Missing	attribute : Name : 0 (0%)	Distinct: 7	Type: Nom Unique: 7 (1)	inal 00%)
Attributes	No.	Label	Count	Weight	
All None Invert Dattern		1 rahul	1	1.0	1.
All None Linen Patient		2 roy	1	1.0	C.
No. Name	-	3 rohit	1	1.0	
1 Name 2 marks	Class: ma	rks (Num)		•]	Visualize All
	1	1 1	1		1
Demove					

No.	1: Name Nominal	2: marks Numeric
1	rahul	
2	roy	55.0
3	rohit	22.0
4	ajay	87.0
5	asad	İ. İ
6	asha	65.0
7	seema	1

 $iv. \ \ Click \ on \ Choose \ and \ select \ filters/\ unsupervised/attribute/ReplaceMissingValues.$



v. Then click on the area right of the Choose button. You get the following:

About Replaces all missing values for nominal and numeric attributes More in a dataset with the modes and means from the training data. Capabilities gnoreClass True	eka.filters.u	nsupervised.at	tribute.ReplaceN	1issingValues		
Replaces all missing values for nominal and numeric attributes More in a dataset with the modes and means from the training data. Capabilities gnoreClass True	About					
in a dataset with the modes and means from the training data. Capabilities	Replaces	all missing v	alues for nom	nal and numeric	attributes Mo	re
gnoreClass True	in a datas	ot with the m	adea and maa	no from the trainin	a data	
	in a datac	set with the m	odes and mea	its notif the trainin	Capab	oilities
	ignoreClass	True	odes and mea		Ig Gata. Capab	oilities

You see here the default parameters of this filter. Click on more to get more information about these parameters.

vi. **VALIDATION :** Click on the Apply button to do the Replace Missing Values. Then select edit tab to view data and see how it Replaced missing values in the data window.

No.	1: Name Nominal	2: marks Numeric	
1	rahul	57.25	
2	roy	55.0	
3	rohit	22.0	
4	ajay	87.0	
5	asad	57.25	
6	asha	65.0	
7	seema	57.25	

vii. Try other parameters for the filter and see how the replace values changes. Dont forget to reload the original (numeric) relation or Undo the replaced before applying another one.

i. <u>Standardize</u>:

DESCRIPTION:

Standardizes all numeric attributes in the given dataset to have zero mean and unit variance (apart from the class attribute, if set).

OPTIONS

Ignore Class – The class index will be unset temporarily before the filter is applied.

STEPS

i. Start Weka you get the Weka GUI chooser window.

🥥 Weka GUI Chooser	
Program Visualization Tools Help	
	Applications
WEKA	Explorer
The University of Waikato	Experimenter
Waikato Environment for Knowledge Analysis Version 3.7.4	KnowledgeFlow
(c) 1999 - 2011 The University of Waikato Hamilton, New Zealand	Simple CLI

- ii. Click on the Explorer button and you get the Weka Knowledge Explorer window.
- iii. Click on the Open File. button and open an ARFF file (try it first with an example supplied in Weka-3-6/data, e.g. Weather.arff). You get the following:
- iv. Click on Choose and select filters/unsupervised/attribute/Standardize.
- v. Then click on the area right of the Choose button. You get the following:

You see here the default parameters of this filter. Enter the Indices of attribute to be remove Click on more to get more information about these parameters.
anness Classify Cluster Associate Select attributes (Visitaire)				
and a second and a				
Open file Open URL Open DB Ger	nerate	Undo	Edit	Save
Choose None				den.
				- MP
Pelation Attributes: None	Selected attribute			Type: None
Instances: None Sum of weights: None	Missing: None	Disting	t: None	Unique: None
tributes				
All Store Toward Datase				
Mil Noris Liver, Pacen				
				Visualiz
	4			
Remove				
	1			
itus deema ta tha Waka Evaluera				
Weka Explorer				
reprocess Classify Cluster Associate Select attribut	as Visualize			
Open fi Open Open Ge	nera] U	Indo	Edit	Save
Open fi Open Open Ge	nera U	Indo] [Edit	Save
Open fi Open Open Ge	nera) 🗌 U	Indo] [Edit	Save
Open fi Open Open Ge Filter Choose None	nera) [U	indo] [Edit	Save
Open fi Open Open Ge Filter Choose None	nera)	Indo] [Edit	Save
Open fi Open Open Ge Filter Choose None Current relation	Selected attrib	Indo) (Edit	Save
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Attributes: 5 Current relation	Selected attrib	indo (Edit	Save Apply Type: Nominal
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14	Selected attrib Name: outi Missing: 0 (0	indo) [nute ook %) Dis	Edit tinct: 3	Save Apply Type: Nominal Unique: 0 (0%)
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes	Selected attrib Name: outl Missing: 0 (0 No., L:	indo) [nute ook %) Dis abel	Edit tinct: 3 Count	Save Apply Type: Nominal Unique: 0 (0%) Weight
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes XIII Tributes Tributes Tributes	Selected attrib Name: outl Missing: 0 (0 No. L 1 st	indo) [nute ook %) Dis abel unny	Edit tinct: 3 Count	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Instances: 14 Attributes All None Invert Patt	Selected attrib Name: outh Missing: 0 (0 No. L 1 st 2 oth	indo) (oute ook %) Dis abel unny vercast	Edit tinct: 3 Count 5 4	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes All None	Selected attrib Name: outi Missing: 0 (0 No. L 1 [st 2 o'	indo) (ook %) Dis abel unny vercast sinv	Edit tinct: 3 Count 5 4 5	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes All None All None Invert No. Name	Selected attrib Name: outh Missing: 0 (0 No. L 2 or 3 ra	indo (ook %) Dis abel unny vercast ainy	Edit tinct: 3 Count 5 4 5	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0
Open fi Open Open Ge Filter Choose None Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes All None All None Invert No. Name 1 Joutbook	Selected attrib Name: outh Missing: 0 (0 No. L 1 st 2 or 3 re Class: play (Nor	indo (uute ook %) Dis abel unny vercast ainy m)	Edit tinct: 3 Count 5 4 5	Save Save Save Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes Attributes All None Invert No. Name 1 outlook 2 temperature	Selected attrib Name: outi Missing: 0 (0 No. L 1 st 2 or 3 rz Class: play (Nor	indo (ook %) Dis abel unny vercast ainy m)	Edit tinct: 3 Count 5 4 5	Save Save Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes Ali None Ali None Invert No. Name 1 Guttools 2 temperature 3 humidity	Selected attrib Name: outi Missing: 0 (0 No. L 1 st 2 o 3 rz Class: play (Nor	indo (ook ook abel unny vercast iiny m)	Edit	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 All None Invert All None Invert No. Name 1 remerature 3 humidity 4 windy	Selected attrib Name: outh Missing: 0 (0 No. L 1 s. 2 or 3 re Class: play (Nor 5	indo (ook %) Dis abel unny vercast iniy m)	Edit tinct: 3 Count 5 4 5	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 Visualize A 5
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes All None All None Invert No. Name 1 outlook: 2 temperature 3 humidity 4 windy 5 play	Selected attrib Name: outi Missing: 0 (0 No. L 1 st 2 or 3 rz Class: play (Nor 5	indo (ook %) Dis abel unny vercast ainy m)	Edit tinct: 3 Count 5 4 5	Save Save Apply Type: Nominal Unique: 0 (0%) Weight S.0 4.0 5.0 Visualize A 5
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes All None All None Invert No. Name 1 Goutools 2 temperature 3 humidity 4 windy 5 play	Selected attrib Name: outi Missing: 0 (0 No, L 1 st 2 o 3 re Class: play (Nor 5	indo (nute ook %) Dis abel unny vercast ainy m)	Edit tinct: 3 Count 5 4 5	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes All None All None Invert No. Name 1 outlook 2 temperature 3 humidity 4 windy 5 play	Selected attrib Name: outh Missing: 0 (0 No. L 2 or 3 ra Class: play (Nor 5	indo (ook 9%) Dis abel unny vercast ainy m)	Edit tinct: 3 Count 5 4 5	Save Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes Attributes All None Invert No. Name 1 outlook: 2 temperature 3 humidity 4 windy 5 play	Selected attrib Name: outh Missing: 0 (0 No, L 1 (si 2 or 3 rr Class: play (Nor 5	indo (ook %) Dis abel unny vercast iny m)	Edit	Save Apply Type: Nominal Unique: 0 (0%) Weight S.0 4.0 5.0 Visualize A 5.
Open fi Open Open Ge Filter Choose None Concent relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: All None Invert All None Invert All None Invert 2 temperature 3 humidity 4 windy 5 play	Selected attrib Name: outi Missing: 0 (0 No. L 1 st 2 o' 3 rz Class: play (Nor 5	indo (ook ook wercast abel m) ()	Edit tinct: 3 Count 5 4 5	Save Save Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A
Open fi Open Open Ge Filter Choose None Current relation Current relation Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14 Attributes Ali None Ali None Invert No. Name 1 outlook 2 temperature 3 humidity 4 windy 5 play	Selected attrib Name: outi Missing: 0 (0 No. L 1 s. 2 or 3 ra Class: play (Nor 5	indo (ook %) Dis abel unny vercast iny m)	Edit tinct: 3 Count 5 4 5	Save Apply Type: Nominal Unique: 0 (0%) Weight 5.0 4.0 5.0 Visualize A

- vi. VALIDATION : Click on the Apply button to do the Standardization. Then select edit tab to view data and see how it standard the values in the data window.
- vii. Try other parameters for the filter and see how the standardize changes. Dont forget to reload the original (numeric) relation or Undo the standardize before applying another one.

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no



veka, filters.u	nsupervised.attribut	te.Standardize		
About				
Standard	zes all numeric at	ttributes in the gi	ven dataset to have	More
zero mea	n and unit varianc	e (apart from the	class attribute. if	1
set)				Capabilities
set).	127			Capabilities
set). ignoreClass	False			Capabilities

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	1.739067215	0.326404	FALSE	no
2	sunny	0.978225308	0.812539	TRUE	no
3	overcast	1.434730452	0.423631	FALSE	yes
9	rainy	-0.543458504	1.395900	FALSE	yes
5	rainy	-0.847795267	-0.15972	FALSE	yes
5	rainy	-1.304300411	-1.13199	TRUE	no
7	overcast	-1.456468792	-1.61813	TRUE	yes
1	sunny	-0.239121742	1.298673	FALSE	no
)	sunny	-0.695626886	-1.13199	FALSE	yes
0	rainy	0.217383401	-0.15972	FALSE	yes
1	sunny	0.217383401	-1.13199	TRUE	yes
.2	overcast	-0.239121742	0.812539	TRUE	yes
13	overcast	1.130393690	-0.64586	FALSE	yes
14	rainy	-0.391290123	0.909766	TRUE	no

Program 2

OBTAIN DECISION TREE FOR DIFFERENT DATA SETS USING WEKA

Problem Definition

Obtain decision tree for different data sets using WEKA

Problem Description

Class for generating a pruned or unpruned C4.5 decision tree. Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

OPTIONS

Binary Splits – Whether to use binary splits on nominal attributes when building the trees.

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

Min Num Obj – The minimum number of instances per leaf.

Num Folds – Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.

Reduced Error Pruning – Whether reduced-error pruning is used instead of C.4.5 pruning.

Save Instance Data – Whether to save the training data for visualization.

seed – The seed used for randomizing the data when reduced-error pruning is used.

Sub tree Raising – Whether to consider the subtree raising operation when pruning.

Un pruned – Whether pruning is performed.

Use Laplace – Whether counts at leaves are smoothed based on Laplace.

Click on classify, then choose

Click on start

🏁 Weka Explorer		
Preprocess Classify Cluster Associate Select attributes Visualize		
Open file	nerate Undo Edit.	
Choose None		Apply
Current relation Relation: weather Instances: 14 Attributes: 5	Selected attribute Name: outlook Missing: 0 (0%) Distinct: 3	Type: Nominal Unique: 0(0%)
Attributes	No. Label	Count
	1 sunny	5
All None Invert Pattern	2 overcast	4 E
2 Geoperature 3 Furnidy 4 windy 5 play	Class: play (Nom)	Visuaize All
Status		
OK		Log 🛷 ×

Weka Explorer Preproces: Classify Classify Cluster Associate Select attributes Visualize Classifier Weka Image: Classifier

Click on percentage split

Click on 102415-treesJ48

Classifier		78			
ALC: NOT STATE		••			
Choose J48 -C 0.25 -M 2	2				1
Test entires	Caseline	a desid			
Contractions	Classifier of	output			
O Compliant last ant	Set				
O Grass usfidation - Falds	10				
Cross-validation Poids Percentage split	66				
Mara antiany					
More options	<u></u>				
(Nom) play	~				
Start	Stop				
Hesuit list (right-click for options)					
Status					
ОК				Log	
🏁 Weka Explorer					
Preprocess Classify Cluster A	Associate Select attributes Visual	ze			
Classifier					
Choose J48 -C 0.25 -M 2	2				
Test options	Classifier output				
O Use training set	Pun infor	metion			
Supplied test set	Set	macron			
Cross-validation Folds	Scheme:	weka.classifiers.trees.J	48 -C 0.25 -M 2		
Percentage split	66 Instances:	weather 14			
More options	Attributes:	5			
		outlook			
(Nom) play	~	humidity			
Start	Stop	windy			
- Result list fright click for options'	Test mode:	10-fold cross-validation			
10:24 15 - trees J48		r model (full training go			
Photo State State State State State	Classifie	r moder (rurr craining se	c)		
	J48 pruned tr	ee			
	J48 pruned tr	ee 			
	J48 pruned tr outlook = sun	ee ny			
	J48 pruned tr outlook = sun humidity humidity	ee ny <= 75: yes (2.0) > 75: no (3.0)			
	J48 pruned tr 	ee <<= 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)			~
	J48 pruned tr outlook = sum humidity humidity outlook = ove	ee ny <= 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)			×
Status	J48 prumed tr outlook = sum I humidity Utlook = ove outlook = ove	ee e my < > 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)			×
Status OK	J48 prumed tr outlook = sum humidity outlook = ove	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	×1
Status DK	J48 prumed tr outlook = sum humidity outlook = ove	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	×(
Status DK	J48 prumed tr outlook = sum humidity outlook = ove	re ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	× (
Status DK	J48 prumed tr outlook = sum humidity outlook = ove	ee my <-75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	× (
Status OK Weka Explorer Preprocess Classify Cluster	Associate Select attributes Visual	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	× 1
Status OK Weka Explorer Preprocess Clessify Cluster / Clessifier	Associate Select attributes Visual	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	× (
Status OK Weka Explorer Preprocess Classify Cluster Classifer Choose J48-C 0.25-M	Associate Select attributes Visual	ee * 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)		Log	× (
Status OK Weka Explorer Preprocess Classify Chose J48-C 0.25-M Test options	Associate Select attributes Visual 2 Classifier output	ree ny < 75: yes (2.0) > 75: no (3.0) reast: yes (4.0) generations are			×(
Veka Explorer Preprocess Classify Chose J48-C 0.25-M Test options Use training set	Associate Select attributes Visual 2 Classifier output Currency Ture	ee ny <-75: yes (2.0) >75: no (3.0) reast: yes (4.0) 	5	Log ,	× (
Status DK Weka Explorer Preprocess Classify Chose J48-C 0.25-M: Test options Use training set Supplied test set	Associate Select attributes Visual 2 Classifier output Classifier output Classifier output Set	ee my <pre><pre>c 75: yes (2.0) > 75: no (3.0) reast: yes (4.0) </pre></pre>	5 0.186	Log . 94.2007 % 35.7143 %	
Status DK Weke Explorer Preprocess Classify Classifie Choose J48-C 0.25-M Test options O Use training set Supplied test set Cross-validation Folds	Associate Select attributes Visual 2 Classifier output Classifier	ee ny < 75: yes (2.0) > 75: no (3.0) reast: yes (4.0) se 	5 5 0.186 0.2857 0.419	0472007 % 35.7143 %	× (
Status OK Welka Explorer Preprocess Classify Classifier Choose J48-C 0.25-M: Test options Use training set Supplied test set Cross-validation Folds OPercentage split	Associate Select attributes Visual 2 Classifier output Classifier	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0)	5 5 0.186 0.2857 0.4818 60 *	U47.2007 % 35.7143 %	× (
Status OK Welka Explorer Preprocess Classify Choose J48-C 0.25-M Test options Use training set Supplied test set Cross-validation Folds O Percentage split % More options.	Associate Select attributes Visual Cassifier output Cassifier ou	ee my <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	5 5 0.186 0.2857 0.4818 97.6586 \$	U. 2007 4 35.7143 %	
Status OK Weka Explorer Preprocess Classify Cluster Choose J48-C 0.25-M Test options Ouse training set Cross-validation Folds Orces-validation Folds Precentage spR % More options	Associate Select attributes Visual Classifier output Classifier outp	ee ny < 75: yos (2.0) > 75: no (3.0) rcast: yes (4.0)	5 5 0.186 0.2857 0.4818 50 97.6596 \$ 14	U47.2007 % 35.7143 %	× (
Status OK Weka Explorer Preprocess Classify Chose J48-C 0.25 -M. Test options Use training set Supplied test set Cross-validation Folds Percentage spik X More options (Nom) play	Associate Select attributes Visual Classifier output Classifier outp	ce ny <-75: ycs (2.0) >75: no (3.0) rcast: yes (4.0) cast: yes (4.0) ce 	5 5 0.186 0.2857 0.4818 60 % 97.6586 % 14	U4.2037 ¥ 35.7143 ¥	× (
Status OK Weka Explorer Preprocess Clessify Chose J48-C 0.25 M Test options Use training set Supplied test set Cross-validation Folds Percentage spit More options (Nom) play Start	Associate Select attributes Visual Classifier output Classifier ou	ee ny < 75: yos (2.0) > 75: no (3.0) reast: yes (4.0)	5 0.186 0.2857 0.4818 60 % 97.5586 % 14	UNI-2007 % 35.7143 %	
Status OK Weka Explorer Preprocess Classify Custer Choose J48-C0.25-M: Coose Supplied test set Coose Supplied test set Coose Supplied test set Supplied test s	Associate Select attributes Visual Classifier output Classifier ou	ee ny < 75: yes (2.0) > 75: no (3.0) reast: yes (4.0) set 	5 5 0.186 0.2857 0.4818 60 % 97.6586 % 14 cision Recall F 0.7 0.778	U4.2027 4 35.7143 8	× 10
Status OK Welka Explorer Preprocess Classify Cluster / Classifier Choose J48-C 0.25-M: Classifier Choose J48-C 0.25-M: Use training set Use training set Cross-validation Folds O Percentage split % More options (Nom) play Start Start Classific fright-click for options) Cla2415-UneesJ48	Associate Select attributes Visual Classifier output Classifier output Classifier output Currecuty Cre Reprint attributes Visual Currecuty Cre Reprint attributes Cont Rean app Reprint attributes Root relative Roo	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0) see 	5 5 0.186 0.2857 0.4018 640 % 97.6586 % 14 cision Recall J 0.7 0.778 0.5 0.4 0.629 0.647	U4:2007 4 35.7143 4 0.737 0.444	0C Are 0.789 0.789
Status OK Velka Explorer Preprocess Classify Custer Choose J48-C 0.25-M Test options Use training set Supplied test set Gross-validation Folds Percentage spik % More options (Nom) play Fesult list (right-click for options) 10.24115-treesJ48	Associate Select attributes Visual Casside output C	ee ny < 75: yos (2.0) > 75: no (3.0) rcast: yes (4.0)	5 5 0.186 0.2857 0.4018 60 % 97.6586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.5 0.4 0.629 0.643	Log . Joy-2007 % 35.7143 %	OC Are 0.789 0.789 0.789
Status OK Status OK Status Classify Cluster Choose J48-C 0.25-M Classifier Choose J48-C 0.25-M Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Choose Cluster Cluster Choose Cluster Cluster Choose Cluster Choose Cluster Choose Cluster Cluster Choose Cluster Choose Cluster Cluster Choose Cluster Cluste	Associate Select attributes Visual Classifier output Classifier ou	ee ny < 75: yos (2.0) > 75: no (3.0) reast: yes (4.0)	5 5 0.186 0.2857 0.4818 60 % 97.6586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.629 0.643	Ung . Ung OC Are 0.789 0.789	
Status OK Woka Explorer Preprocess Classify Cluster Choose J48 - 0.05-M Octoss-validation Folds Octoss-validation Folds Octoss-validation Folds Percentage spit % More options (Nom) play Start f Result list (right-click for options; 1024:15-trees.J48	Associate Select attributes Visual Classifier output Classifier outp	ee ny < 75: yes (2.0) > 75: no (3.0) reast: yes (4.0)	5 5 0.186 0.2857 0.4818 60 % 97.6586 % 14 cision Recall F 0.7 0.778 14 0.7 0.43	Log . 04.2007 % 35.7143 %	OC Are 0.789 0.789
Status OK Weka Explorer Preprocess Classify Cluster Choses J48 - C 0.25 - M. Test options Use training set Supplied test set Cross-validation Folds Percentage spik % More options (Nom) play Start 10:24:15 - trees.I48	Associate Select attributes Visual Classifier output Associate Select attributes Visual Classifier output Classifier out	ee ny < 75: po (2.0) > 75: no (3.0) reast: yes (4.0) sourcest: yes (4.0) sourcest lassified Instances ic error lassified Instances ic error ared error is quared error of Instances Accuracy By Class === TP Rate FP Rate Pre 0.74 0.222 60 0.465 x === ed as	5 0.186 0.2857 0.4818 60 % 97.6586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.629 0.643	-Heasure R 0.737 0.444 0.632	00C Are 0.789 0.789
Status DK Status DK Preprocess Clossifie Choose J48-C 0.25-M: Choose J48-C 0.25-M: Choose Choose J48-C 0.25-M: Choose Choose Choose J48-C 0.25-M: Choose	Associate Select attributes Visual Classifier output Classifier ou	ee ny < 75: yes (2.0) > 75: no (3.0) reast: yes (4.0) sourcest: yes (4.0) sourcest: yes (4.0) sourcest leasified Instances de error ared error liste error squared error of Instances Accuracy By Class TP Rate FP Rate Fre 0.778 0.6 0.4 0.222 643 0.465	5 5 0.186 0.2857 0.4818 60 % 97.5586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.5 0.4 0.629 0.643	Log 94.2007 % 35.7143 %	× (00 Are 0, 789 0, 789 0, 789
Status DK Weka Explorer Preprocess Classify Cluster Classifier Choose J48-C 0.25-M: Test options Use training set Statyplied test set Cross-validation Folds O Percentage split # More options (Non) play Start 10.24:15 - trees.J48	Associate Select attributes Visual Classifier output Resolution is a sub- sub- sub- sub- sub- sub- sub- sub-	ee ny < 75: yes (2.0) > 75: no (3.0) rcast: yes (4.0) se 	5 0.186 0.2857 0.4818 640 % 97.6586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.629 0.643	Log , 04.2007 4 35.7143 4 -Measure R 0.737 0.444 0.632	0C Are 0.789 0.789
Status OK Preprocess Classify Custer Classifier Choose J48-C 0.25-M Use training set Supplied test set Oross-validation Folds Percentage spik % More options (Nom) play Stati 10:2415-treesJ48	Associate Select attributes Visual Associate Select attributes Visual Control of the set	ee ny < (5: yes (2.0) > 75: no (3.0) rcast: yes (4.0) set issified Instances is error sayared error of Instances Accuracy By Class TF Rate FP Rate Fr 0.778 0.6 0.4 0.222 643 0.465 x ed as	5 5 0.186 0.2857 0.4018 97.6586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.629 0.643	Ung . Ung . 0472007 % 35.7143 % 0.434 0.444 0.632	OC Are 0.789 0.789 0.789
Status DK Status DK Preprocess Classify	Associate Select attributes Visual Control of the second	ee ny < 75: yos (2.0) > 75: no (3.0) rcast: yes (4.0)	5 5 0.186 0.2857 0.4818 60 % 97.6586 % 14 cision Recall F 0.7 0.778 0.5 0.4 0.629 0.643	U47.2007 % 35.7143 %	0C Are 0.789 0.789 0.789
Status OK Weka Explorer Preprocess Classify Choose J48 - 0.02- M Supplied test set Cross-validation Percentage spit More options Nom) play Statt Classifie Status OK	Associate Select altibutes Visual Classifier output Classifier output Classifier output Classifier output Classifier output Classifier output Classifier output Concrectly Creating Real active absoc Root is eas any Real active absoc Real active absoc Real active absoc Real active absoc Real active absoc Real active absoc Real active abs	ee ny < 75: pes (2.0) > 75: no (3.0) reast: yes (4.0)	5 0.186 0.2857 0.4818 60 % 14 cision Recall F 0.7 0.778 14 cision Recall F 0.7 0.43	Log , 94.2007 % 35.7143 %	0C Are 0.789 0.789 0.789

CSE Department, MJCET



Second data set:

Bankprediction data set

* Weka Explorer		
Preprocess Classify Cluster Associate Select attributes Visualize		
Open file	enerate Undo	Edit Save
Choose None		Apply
Current relation Relation: bank_predicted Instances: 300 Attributes: 11	Selected attribute Name: Instance_number Missing: 0 (0%) Distir	Type: Numeric nct: 300 Unique: 300 (100%)
Attributes	Statistic	Value
	Minimum	0
All None Invert Pattern	Maximum	299
	Mean	149.5
No. Name	StdDev	86.747
1 Instance number		
2 age		
3 sex		
4 region	Classy page (Nom)	1/involto A
5 income	Class. pep (wont)	Visualize A
6 married		
7 🗖 children	43 43 43	42 43 43 43
8 car		
9 mortgage		
10 predictedpep		
11pep		
Remove		
	0	149.5
Status		
OK		

Click on classify

Then click on choose and in trees choose j48

🎌 Weka Explorer	
Preprocess Classify Cluster Associate Select attributes	Visualize
Classifier	
veka classifiers bayes functions JuhonClassifier Juh	ier output
Status OK	Log 💉 x

Click on start

^{e.} Weka Explorer			
Preprocess Classify Cluster Associate S	elect attributes Visua	lize	
Classifier			
Choose J48 -C 0.25 -M 2			
Test options	Classifier output		
			~
	=== Run info	rmation ===	
O Supplied test set	Scheme:	weka.classifiers.trees.J48 -C 0.25 -M 2	
Cross-validation Folds 10	Relation:	bank_predicted	
O Percentage split % 66	Instances:	300	
More options	Attributes:	11	
		Instance_number	
	er (age	
(Nom) pep 🞽		sex	
Start		income	
		narried	
Result list (right-click for options)	Y	children	
11:23:34 - trees.J48		car	
		mortgage	
		predictedpep	
		pep	
	Test mode:	10-fold cross-validation	
	=== Classifi	er model (full training set) ===	
	J48 pruned t	ree	~
	<		
Status	/5		
ОК			Log XI

Click on percentage split

Choose J48 -C 0.25 -M 2							
Test options Use training set Supplied test set Cross-validation Polds 10 Percentage split % 66	Classifier output Correctry Classifier Kappa statistic Mean absolute error Root mean squared en Relative absolute er Robit relative square	r in ied rror rror ror	scances Instances	62 0.58 0.31 0.40 62.73 82.18	04 17 96 94 % 27 %	7 9:000 20.6667	÷ _
Nom) pep Start Stop Result list fright-click for options)	Total Number of Inst === Detailed Accurate TP Re 0.7	tanco cy B ate 717	es y Class === FP Rate 0.142	300 Precision 0.811	Recall 0.717	F-Measure 0.762	ROC Are 0.76
11:23:34 - trees.J48 Wi S4 D2 C2 S2 R6	0.6 win in main window win is separate window ve result buffer leter esult buffer ad model ve model -evaluate model on current test set	358 3 11 11	0.283 0.218 • ed as	0.781 0.795	0.858	0.818 0.792	0.76
Vie Vie	ualize classifier errors ualize tree					j	>
Status Vie OK Co	ualize margin curve ualize threshold curve) st/Benefit analysis) uelize cost curve					Log] 🛷



Third data set:

Sleep data set

🍄 Weka Explorer		
Preprocess Classify Cluster Associate Select attributes Visualize		
Open file Open URL Open DB Gen	erate Undo	Edit Save
Choose None		Apply
	Para and and and	(1497)
Relation ins Instances: 150 Attributes: 5	Name: sepallength Missing: 0 (0%) Distinct:	Type: Numeric 35 Unique: 9 (6%)
Attributes	Statistic	Value
	Minimum	4.3
All None Invert Pattern	Maximum	7.9
	Mean	5.843
No. Name	StaDev	0.828
i sepalength		
3 netallength		
4 petalwidth		
5 class	Llass: class (Nom)	Visualize All
Remove	30 10 10	25
	4.3	6.1 7.9
Status OK		Log 💉 ×0

Click on classify and choose from trees choose j48 algorithm

Prennoness Classify Cluster Associate Sele	ct attributes Visualize	(all) (all)
Classifier		
Classifiers Classifiers Classifiers Classifier Cl	ier oukput	
orenus DK		

Click on start and then click on percentage split

VALIDATION :

🐃 Weka Explorer							
Preprocess Classify Cluster Associa	te Select attributes Visualize						
Classifier							
Choose J48 ·C 0.25 ·M 2							
Test options	Classifier output						
🔘 Use training set	Mean absolute	error		0.03	5		~
O Supplied test set Set	Root mean squar	red error		0.15	86		
	Relative absolu	ite error		7.87	05 %		
	Root relative :	squared e f Tratora	rror	33.63	53 %		
Percentage split % 66	Tocar Number of	t instant	-3	150			
More options	Detailed A	ccuracy B	y Class ==	2			
(Nom) class	~	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Are
		0.98	0	1	0.98	0.99	0.99
Start Stop		0.94	0.03	0.94	0.94	0.94	0.952
Result list (right-click for options)		0.96	0.03	0.941	0.96	0.95	0.961
10-20-57 - trees J48	weighted Avg.	0.96	0.02	0.96	0.96	0.90	0.960
View in r View in s Save re: Delete r	nain window separate window sult buffer esult buffer	rix ==	= ied as				
Load mo Save mo Re-avail	idel idel jate model on current test set	Iris-s Iris-v Iris-v	etosa ersicolor irginica				
Visualize	classifier errors						
Visualize	Itree					1	>
Status Visualize	: margin curve : threshold curve nefit analysis					Log)
Visualize	cost curve						J 🥪



Data Mining Lab Manual Program 3 CLASSIFICATION ALGORITHMS USING WEKA

Problem Definition

Classification algorithms using WEKA

Bank prediction data set using nb tree:

[©] Weka Explorer Preprocess: Classify Cluster Associate Select attributes Visualize		
Open file Open URL Open DB Ger Filter	nerate Undo	Edit Save
Current relation Relation: bank_predicted Instances: 300 Attributes: 11	Selected attribute Name: Instance_number Missing: 0 (0%) Distinct:	Type: Numeric 300 Unique: 300 (100%)
Attributes	Statistic	Value
	Minimum	0
All None Invert Pattern	Maximum	299
	Mean	149.5
No. Name Inverts the current attribute	selection _{ev}	86.747
2 age 3 sex 4 region 5 income	Class: pep (Nom)	Visualize A
6 married		
7 children	43 43 43 42	43 43 43
8 car		
9 mortgage		
10 predictedpep		
Прер		
Remove		
	0	149.5
itatus DK		Log

Bank prediction data set using j48graft :

😤 Weka Explorer			
Preprocess Classify Cluster Associate Sele	ect attributes Visualize		
Classifier			
💼 weka)
🖨 🚞 classifiers			
tunctions			
JythonClassifier	v Classified Instances	62	20.6667 %
🖅 🚞 lazy	istic	0.5804	
⊞ <u>e</u> meta	ate error	0.3117	
terness mise	squared error	0.4096	
🖃 💼 rules	psolute error	62.7394 %	
E- irees	er of Instances	300	
ADTree			
0 DecisionStump	ed Accuracy By Class ==		
• FT			
L 1d3	TP Rate FP Rate	Precision Recall	F-Measure RUC Are
J48	0.858 0.283	0.781 0.858	0.818 0.76
ADTree	7g. 0.793 0.218	0.795 0.793	0.792 0.76
• LMT			
M5P	ion Matrix ===		
RandomForest	- plassified ac		
RandomTree	a = YES		
REPTree	b = N0		
Filter Remove filter	Close		
			×
J.	<u> </u>		
🗣 Weka Explorer			
Preprocess Classify Cluster Associate Sel	ect attributes Visualize		
Classifier			
Choose NBTree			
Test options	Classifier output		
O Use training set	Incorrectly Classified Instances	22	21.5686 %
O Supplied test set Set	Kappa statistic	0.5584	
Cross-validation Folds 10	Mean absolute error Root mean squared error	0.3211	
Percentage split % 66	Relative absolute error	64.6668 %	
	Root relative squared error	82.528 %	
More options	Total Number of Instances	102	
(Nom) pep	=== Detailed Accuracy By Class ==	=	
Start Stop	TP Rate FP Rate	Precision Recall	F-Measure ROC Are
the second second second second second second second second second second second second second second second se			the second second second second second second second second second second second second second second second se
Result list (right-click for options)	0.711 0.158	0.78 0.711	0.744 0.816
Result list (right-click for options)	0.711 0.158 0.842 0.289	0.78 0.711 0.787 0.842	0.744 0.816 0.814 0.816
Result list (right-click for options) 11:23:34 - trees.J48 11:30:11 - trees.NBTree	0.711 0.158 0.842 0.289 Weighted Avg. 0.784 0.231	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.J48 11:30:11 - trees.NBTree	0.711 0.158 0.842 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix ===	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.J48 11:30:11 - trees.NBTree	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix ===	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.J48 11:30:11 - trees.NBTree	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix === a b < classified as 22 10 4 - 9 WC	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.J48 11:30:11 - trees.NBTree	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix === a b < classified as 32 13 a = YES 9 48 b = NO	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.448 11:30:11 - trees.NBTree	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix === a b < classified as 32 13 a = YES 9 48 b = N0	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.48 11:30:11 - trees.NBTree	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix === a b < classified as 32 13 a = YES 9 48 b = N0	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.448 11:30:11 - trees.NBTree	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix === a b < classified as 32 13 a = YES 9 48 b = N0	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816
Result list (right-click for options) 11:23:34 - trees.448 11:30:11 - trees.NBTrees C Status	0.711 0.158 0.642 0.289 Weighted Avg. 0.784 0.231 === Confusion Matrix === a b < classified as 32 13 a = YES 9 48 b = N0	0.78 0.711 0.787 0.842 0.784 0.784	0.744 0.816 0.814 0.816 0.783 0.816

Bank prediction data set using NBtree : Bank prediction data set using nb tree:

🐑 Weka Explo	orer							
Preprocess Cla	ssify Cluster Associate S	elect attributes V	isualize					
Classifier	A. 10. 1101							
Choose	NBTree							
Test options		Classifier output	ıt					
🚫 Use trainin	g set	Incorrect	Ly Classified]	Instances	22		21.5686	¥
O Supplied te	est set Set	Kappa stat	tistic		0.55	84		
~ ~ ~		Mean absol	lute error		0.32	11		
O Cross-valid	lation Folds 10	Root mean	squared error		0.41	05		
Percentage	e split % 66	Relative a	absolute error		64.66	68 %		
<u> </u>	fore options	Root relat	cive squared en	cror	82.52	8 %		
W	noie options	Total Numb	per of Instance	23	102			
Start Result list (right- 11:23:34 - trees	Stop click for options)	Weighted 4	TP Rate 0.711 0.842	FP Rate 0.158 0.289 0.231	Precision 0.78 0.787 0.784	Recall 0.711 0.842 0.784	F-Measure 0.744 0.814 0.783	ROC Are 0.816 0.816 0.816
11:30:11 - trees	View in main window View in separate window Save result buffer Delete result buffer	v	ion Matrix === classified	as				
	Load model Save model Re-evaluate model on d	urrent test set.	p = NO					
	Visualize classifier errors							>
<u></u>	Visualize tree		5					
OK OK	Visualize margin curve Visualize threshold curve						Log	

🖗 Weka Classifier Tree Visualizer: 11:30:11	trees.NBTree (ba	
Tree View		
NB model		

eprocess Classify Cluster Associate Select attributes Visualize		
Open/Edit/Save instances Open Itie Upen URL Open DB Ge ilter	nerate Undo	Edit Save
Choose None		Арр
urrent relation Relation: bank_predicted Instances: 300 Attributes: 11	Selected attribute Name: Instance_number Missing: 0 (0%) Distir	Type: Numeric ict: 300 Unique: 300 (100%)
ttributes	Statistic	Value
	Minimum	0
All None Invert Pattern	Maximum	299
	Mean	149.5
No. Name	StdDev	86.747
2 age 3 sex 4 region 5 income	Class: pep (Nom)	Visualize
6 married		
7 children	43 43 43	42 43 43 43
8 car		
9 mortgage		
11 Tese		
i i Theh		
Remove		
	T.	
	0	149.0

reprocess Classify Cluster Associate Select att	ibutes Visu	alize					
Classifier							
📴 weka	~						
🖻 🚊 classifiers							
IuthonClassifier	7	Classified]	Instances	21		20.5882	ş 🖉
	is	tic		0.57	55		
🗐 🦲 meta	lat	e error		0.31	28		
🗄 👘 mi	30	uared error		0.40	5		
🗄 🤖 misc	05	olute error		63.00	32 %		
🗄 🚞 rules	iv	e squared en	ror	81.42	72 %		
😑 🚞 trees	er	of Instance	25	102			
ADTree							
BFTree	ed	Accuracy By	7 Class ===				
DecisionStump							
• FT		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Are
- U0		0.689	0.123	0.816	0.689	0.747	0.783
140 J46		0.877	0.311	0.781	0.877	0.826	0.783
	79	r. 0.794	0.228	0.796	0.794	0.791	0.783
• LAD HEE							
MSP	io	n Matrix ===					
NBTree							
RandomForest		classified	as				
RandomTree	а	= YES					
REPTree	 o	= NO					
Filter Remove filter	Close						
							>
2 1 Control 1						and the second se	
latus							1

Preprocess C	lassify Cluster Associate	Select attributes	Visualize						
Classifier									
Choose	J48graft -C 0.25 -M 2								
Test options		Classifier out	out						
🔿 Use traini	ing set	Incorrect	ly Classif	ied :	Instances	21	ow.	20.5882 ^s	
O Supplied	test set Set	Kappa st	atistic			0.57	55		
O Consul	raucu estacilian	Mean abs	lute error	10		0.31	28		
O Closs-val	iudiun rolos 10	Root mean	n squared e	error		0.40	5		
Percenta	age split % 66	Relative Dest volu	absolute e	error		63.00	32 % 72 %		
	More options	Total Nu	iber of Ins	stance	23	102	12 3		
Nom) pep		=== Deta:	led Accura	асу Ву	y Class ==:				
Start	Stop	Y	TP F	late	FP Rate	Precision	Recall	F-Measure	ROC Are
Recult list fride	at aliak for options)		00	689	0.123	0.816	0.689	0.747	0.783
rresolic list (rigi	Inclick for options)	1	0.	877	0.311	0.781	0.877	0.826	0.783
11:23:34 - tree 11:30:11 - tree	es.J48 es NRTree	Weighted	Avg. 0.	794	0.228	0.796	0.794	0.791	0.783
11:31:30 - tree	es.J48graft	=== Confi	usion Matri	X ==:					
11:32:49 - tree	Alternation View in main window View in separate wind Save result buffer Delete result buffer	ow	classi a = YES b = NO	lfied	as				
	Load model Save model				jji.	19			>
Status	Re-evaluate model on	current test set		_					
OK	Visualize classifier erro	ors						Log	
UIX	Visualize tree								



eprocess Classify Cluster Associate ?	elect attributes Visualizi	•					
lassifier							
Choose NBTree							
	12.85 20 12 19						
est options	Classifier output						
🔿 Use training set	Mean absolute	error		0.04	78		
O Supplied test set	Root mean squa	red error		0.163	21		
Cross uslidation Folds 10	Relative absol	ute error		10.74	74 %		
	Root relative	squared en f Tratana	ror	34.38	J7 %		
O Percentage split % 66	local Number o	r instance	13	150			
More options	=== Detailed A	ccuracy By	7 Class ===				
om) pep	2	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Ar
onių pep		1	0	1	1	1	1
Start Stop		0.92	0.05	0.902	0.92	0.911	0.99
esult list (right-click for options)		0.9	0.04	0.918	0.9	0.909	0.98
	Weighted Avg.	0.94	0.03	0.94	0.94	0.94	0.99
un 45 destrictue	=== Confusion	Matrix ==:	-				
	abc <-	- classifi	ied as				
	50001a	= Iris-se	etosa				
	0464 b	= Iris-ve	ersicolor				
	U 5 45 C	= Iris-Vi	rginica				
	<						>
	1 town						

🌣 Weka Explorer							
Preprocess Classify Cluster Associate Sel	ect attributes Visualiz	e					
Classifier							
Choose NBTree							
Test options	Classifier output						
🔿 Use training set	Mean absolute	error		0.04	78		
O Supplied test set Set	Root mean squa	red error		0.16	21		
	Relative absol	ute error		10.74	74 %		
Cross-validation Folds	Root relative	squared en	ror	34.38	07 %		
O Percentage split % 66	Total Number o	of Instance	23	150			
More options	=== Detailed A	ccuracy By	y Class ==:	-			
(Nom) pep 😽		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Are
		1	0	1	1	1	1
Start Stop		0.92	0.05	0.902	0.92	0.911	0.991
Result list Provide the elevel first for	77.7.7.7.4.4	0.9	0.04	0.918	0.9	0.909	0.983
11:17:49 - trees NRTree	weighted Avg.	0.94	0.03	0.94	0.94	0.94	0.991
	=== Confusion	Matrix ===	-				
	ahc c-	- classifi	se hei				
	50 0 0 1 8	= Iris-se	etosa				
	046411	= Iris-ve	ersicolor				1000
	0 5 45 1 0	= Iris-vi	irginica				
	1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.						
							~
	<						>
Status							
OK						Log) × × 0
U.S.							1

Weka Explorer							
Preparcent Clenty Cluster Associate S	electrathibutes [Visuala	ie l					
Cattlet							
Choose NUTree							
Use training set Use training set Supplied text set End Dose-validation Failt 10	Choolee output Arean abrolute Boot arean spo- Bolarive absol Boot selative	ettox szed erros lute ettos oguaced e	1101	0,64 0,16 10,74 34,30	28 21 24 5 07 8		
O Percentage spik	Tetal Number (of Instance Accuracy B	ez 9 Ciwa ++	250			
Nonipep		TP Bete 0,92 0,92 0,92	9 0.05 0.04	Previation 1 0.910 0.928	Recall 1 0.92 0.9	T-Beanize 1 0.911 0.929	900 Aze 1 1 80,0 1 80,0 0.90
UN TRAF Union NET Dawn	Program Ang. In Han whole In Han whole In separate whole In result Suffer In result Suffer In result In the International Internation	p,94	et er tosa tascolos rginica	0.94	0.94	0.94	0.991
794	aloe dassher errors Nov bro	PERSON SHE	-				ž
Stear We We Co	alor megn carre alor threitold carve Liberefit andysis alor cost carve					Log	· •



Program 4

OBTAIN THE ASSOCIATION RULES FOR GIVEN DATASET USING WEKA

Problem Definition

Obtain The Association Rules For Given Dataset Using WEKA.

Problem Description

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

The algorithm has an option to mine class association rules.

R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules in Large Databases. In: 20th International Conference on Very Large Data Bases, 1994.

Bing Liu, Wynne Hsu, Yiming Ma: Integrating Classification and Association Rule Mining. In: Fourth International Conference on Knowledge Discovery and Data Mining, 1998.

OPTIONS

car – If enabled class association rules are mined instead of (general) association rules.

Class Index – Index of the class attribute. If set to -1, the last attribute is taken as class attribute.

delta – Iteratively decrease support by this factor. Reduces support until min support is reached or required number of rules has been generated.

Lower Bound Min Support – Lower bound for minimum support.

Metric Type – Set the type of metric by which to rank rules. Confidence is the proportion of the examples covered by the premise that are also covered by the consequence(Class association rules can only be mined using confidence). Lift is confidence divided by the proportion of all examples that are covered by the consequence. This is a measure of the importance of the association that is independent of support. Leverage is the proportion of additional examples covered by both the premise and consequence above those expected if the premise and consequence were independent of each other. The total number of examples that this represents is

presented in brackets following the leverage. Conviction is another measure of departure from independence. Conviction is given by P(premise)P(!consequence) / P(premise, !consequence).

 \min Metric – Minimum metric score. Consider only rules with scores higher than this value.

Num Rules – Number of rules to find.

Output Item Sets – If enabled the item sets are output as well.

Remove All Missing Cols – Remove columns with all missing values.

Significance Level – Significance level. Significance test (confidence metric only).

Upper Bound Min Support – Upper bound for minimum support. Start iteratively decreasing minimum support from this value.

verbose – If enabled the algorithm will be run in verbose mode.

STEPS

Choose a data set

Iris data set

🕅 Weka Explorer		
Preprocess Classify Cluster Associate Select attributes Visualize		10.000
Open file	nerate	Edit Save
Choose None		Apply
Current relation Relation: iris Instances: 150 Attributes: 5	Selected attribute Name: sepallength Missing: 0 (0%) Distinct: 3	Type: Numeric 35 Unique: 9 (6%)
Altributes	Statistic	Value
	Minimum	4.3
All None Invert Pattern	Maximum	7.9
	Mean	5.843
No. Name	StdDev	0.828
2 epaWoldh 3 petaWoldh 4 petaWoldh 5 class	Class: class (Nom)	Visualize Al
	30 34 28	28 (5.843, 6.357)
Remove	43	
Status OK		Log 🛷 ×

Click on classify and choose ${\bf ZeroR}$

😤 Weka Explorer	
Preprocess Classify Cluster Associate Select attributes	lisualize
Classifier	
weka Classifiers Javes Souther Classifier OriginativeRule Originative	ier output
OK	Log 💉 x0

🏾 Weka Explorer		×
Preprocess Dassily Cluster Associate Sel	ect at blocks	
Classifier		
Choose ZeroB		
Test options	Destiler output	2
O Use training set	and Dep Information and	8
O Surrelied text set	The Analysis of the State of th	
O Compatibility Edge 10	Scheme: weka.classifiets.rules.ZeroR	
C Dott-Valcatori Poldi 10	Relation: into	
O Percentage split 32 65-	Instances: 1.30	
More options	seallenth	
	sepalwidth	
(Non) class 🗸 🗸	pstallength	
	petalwidth	
Star	Class	
Result list (right-click for options)		
10.40.01 - ruleo ZeroR	Classifier model (full training set)	
	ZeroR predicts class value: Iris-setosa	
	Time taken to build model: 0 peconds	
	and Stratified cross-validation and	
	Summary	
	Correctly Classified Instances 50 33.3333 %	
	Incorrectly Classified Instances 100 66.6667 4	
	Mean absolute error 0.4444	
	Root mean squared error 0.4714	
	Relative absolute error 100 %	
	Root relative squared error 100 6	
	Total Mumher of Instances 1.0	
	Detailed Accuracy By Class	đ
Status		
DK		.0

VALIDATION :

🎌 Weka Explorer		
Preprocess Classify Cluster Associate Se	ect atributes Visualize	
Classifier		
Choose ZeroR		
Test options	Classifier output	
O Use training set	ZeroS predicts class value: Iris-setosa	^
Supplied test set Sitt	Time taken to build model: 0 seconds	
Cross-validation Folds 10		
Percentana mit	Stratified cross-validation	
	Sumary	
More options	Correctly Classified Instances 50 33.3333 b	
West store	Incorrectly Classified Instances 100 66.6667 %	
Inoni Gass	Kappa statistic 0	
Start Stop	Reen absolute error 0.4944	
Result list (right-click for options)	Relative abolute error 100 %	
10400Lotes 2608	Root relative squared error 100 %	
	Total Number of Instances 150	
	and Datallal Lower Br Class and	
	- correction accorded by cross	
	TP Rate FP Rate Precision Recall F-Measure ROC Area Class	
	1 1 0.333 1 0.5 0.5 Iris-setosa	
	Weighted Avg. 0.333 0.333 0.111 0.333 0.167 0.5	
	Confusion Matzix	
	a h a c- classified as	
	50 0 0 a = Iris-setosa	
	50 0 0 i b = Iris-versicolor	
	50 0 0 c = Iris-virginica	
		-
		M
Status		
OK.		-00 10

Program 5

PERFORM DATA TRANSFORMATIONS USING ETL TOOL

Problem Definition

Perform data transformations using an ETL tool.

Problem Description

Extraction, Transformation, and Loading (ETL) Processes

- The plumbing work of data warehousing
- Data are moved from source to target data bases
- A very costly, time consuming part of data warehousing

Sample ETL Tools

- Tera data Warehouse Builder from Tera data
- Data Stage from Ascential Software
- SAS System from SAS Institute
- Power Mart/Power Center from Informatica
- Sagent Solution from Sagent Software
- Hummingbird Genio Suite from Humming bird Communications

Introduction to MySQL:

- MySQL is a very popular, open source database.
- Officially pronounced my Ess Que Ell (not my sequel).
- Handles very large databases; very fast performance.
- Why are we using MySQL?
 - Free (much cheaper than Oracle!)
 - Each student can install MySQL locally.
 - Easy to use Shell for creating tables, querying tables, etc.
 - Easy to use with Java JDBC

A) For products and orders

```
mysql> use nida;
Database changed
mysql> create table orders(orderid varchar(5),
               -> productid varchar(5),quantity int(5),
               -> unitsaleprice int(5),discountprice int(5),
                    -> numoffreeservice int(3));
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> desc orders;
```

Field	Type	Null	Key	Default	Extra
orderid	$\operatorname{varchar}(5)$	YES		NULL	
productid	$\operatorname{varchar}(5)$	YES		NULL	
quantity	int(5)	YES		NULL	
unitsaleprice	int(5)	YES		NULL	
discountprice	int(5)	YES		NULL	
numoffreeservice	int(3)	YES		NULL	

```
6 rows in set (0.01 sec)
```

mysql> insert into orders values('0101','p121',10,5000,10,2); Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values('0101', 'p121',10,5000,10,2); Query OK, 1 row affected (0.02 sec)

mysql> insert into orders values('0101','p01',1,234,5,3); Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values('0101','p180',2,4000,12,3); Query OK, 1 row affected (0.03 sec)

mysql> insert into orders values('0102','p02',5,2500,3,2); Query OK, 1 row affected (0.02 sec)

```
mysql> insert into orders values('0102','p122',2,2800,3,2);
Query OK, 1 row affected (0.03 sec)
```

orderid	productid	quantity	unitsaleprice	discountprice	numoffreeservice
0101	p121	10	5000	10	2
0101	p121	10	5000	10	2
0101	p01	1	34	5	3
0101	p180	2	4000	12	3
0102	p02	5	2500	3	2
0102	p122	2	2800	3	2

mysql> select * from orders;

6 rows in set (0.00 sec)

mysql> create table products(productid varchar(5) primary key,

```
-> companyid varchar(5),
```

- -> productname varchar(10),
- -> producttype varchar(10),
- -> productprice int(5),
- -> productdom date,
- -> productinstock int(5));

Query OK, 0 rows affected (0.06 sec)

Field	Type	Null	Key	Default	Extra
productid	$\operatorname{varchar}(5)$	NO	PRI	NULL	
companyid	$\operatorname{varchar}(5)$	YES		NULL	
productname	$\operatorname{varchar}(10)$	YES		NULL	
producttype	$\operatorname{varchar}(10)$	YES		NULL	
productprice	int(5)	YES		NULL	
productdom	date	YES		NULL	
productinstock	int(5)	YES		NULL	

7 rows in set (0.02 sec)

```
'2003-08-15',15);

Query OK, 1 row affected (0.03 sec)

mysql> insert into products values('P180','2','FAX',168,2100,

'2003-03-12',12);

Query OK, 1 row affected (0.03 sec)
```

mysql> select * from products;

product	company	product	product	product	product	product
id	id	name	type	price	dom	instock
P01	30	CABLE	121	234	1991-01-09	25
P02	28	OPCABLE	122	500	1998-08-04	35
P121	12	MONITOR	147	5000	2001-09-25	19
P122	11	BATTERY	124	1400	2003-08-15	15
P180	2	FAX	168	2100	2003-03-12	12

5 rows in set (0.00 sec)

B) For birth certificate

```
mysql> use nida;
Database changed
```

mysql> select * from birthcertificate;

Name	Fathername	Birth	Location	SSN
GHI	MNO	1990-11-17	BANGLORE	645
KLM	STR	1988-03-27	Delhi	665
ABC	XYZ	1989-05-03	HYDERABAD	667

3 rows in set (0.00 sec)

```
Data Mining Lab Manual
mysql> insert into birthcertificate values('DEF', 'PQR', '1987-06-07',
     'HYDERABAD', '785');
Query OK, 1 row affected (0.02 sec)
mysql> create table college(Name varchar(30), Fathername varchar(20),
Birth date, SSN varchar(3),Rollno int(3));
Query OK, 0 rows affected (0.06 sec)
mysql> insert into college values('ABC','XYZ','1989-05-03','667',10);
Query OK, 1 row affected (0.03 sec)
mysql> insert into college values('KLM', 'STR', '1988-03-27', '665',11);
Query OK, 1 row affected (0.01 sec)
mysql> insert into college values('GHI', 'MNO', '1990-11-17', '645', 15);
Query OK, 1 row affected (0.02 sec)
mysql> insert into college values('DEF', 'PQR', '1987-06-07', '785', 14);
Query OK, 1 row affected (0.03 sec)
mysql> insert into college values('GTH', 'THR', '1989-06-01', '765', 12);
Query OK, 1 row affected (0.02 sec)
mysql> insert into college values('RTU', 'YHN', '1994-07-24', '378',19);
Query OK, 1 row affected (0.03 sec)
mysql> create table addrproof(Name1 varchar(30), Name2 varchar(30),
           Address varchar(30),SSN varchar(30) references
birthcertificate(SSN)):
Query OK, 0 rows affected (0.08 sec)
mysql> insert into addrproof values
('ABC', 'XYZ', 'HYDERABAD', '667');
Query OK, 1 row affected (0.02 sec)
mysql> insert into addrproof values
('RTU', 'YHN', 'HYDERABAD', '378');
Query OK, 1 row affected (0.02 sec)
mysql> insert into addrproof values
('GHI', 'MNO', 'DELHI', '645');
Query OK, 1 row affected (0.02 sec)
mysql> insert into addrproof values
     ('KLM', 'STR', 'BANGLORE', '665');
Query OK, 1 row affected (0.02 sec)
```

mysql> SELECT * FROM ADDRPROOF;

Name1	Name2	Address	SSN
ABC	XYZ	HYDERABAD	667
RTU	YHN	HYDERABAD	378
GHI	MNO	DELHI	645
KLM	STR	BANGLORE	665
GTH	THR	BANGLORE	765
DEF	PQR	CHENNAI	785

```
6 rows in set (0.02 \text{ sec})
```

```
mysql> create table citizeninfosystem as
        (select birthcertificate.Name,birthcertificate.Fathername,
        birthcertificate.Birth, birthcertificate.SSN,
        (birthcertificate.Location) as birthloc,addrproof.Address,
            college.Rollno from birthcertificate inner join addrproof on
            birthcertificate.Name=addrproof.Name1 inner join college on
            birthcertificate.Name=college.Name);
Query OK, 4 rows affected (0.08 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

mysql> select * from citizeninfosystem;

Name	Fathername	Birth	SSN	BirthLocation	Address	RollNo
ABC	XYZ	1989-05-03	667	HYDERABAD	HYDERABAD	10
KLM	STR	1988-03-27	665	DELHI	BANGLORE	11
GHI	MNO	1990-11-17	645	BANGLORE	DELHI	15
DEF	PQR	1987-06-07	785	HYDERABAD	CHENNAI	14

4 rows in set (0.00 sec)

mysql> select * from birthcertificate;

Data Mining Lab Manual

				0
Name	Fathername	Birth	Location	SSN
GHI	MNO	1990-11-17	BANGLORE	645
KLM	STR	1988-03-27	Delhi	665
ABC	XYZ	1989-05-03	HYDERABAD	667
DEF	PQR	1987-06-07	HYDERABAD	785

4 rows in set (0.00 sec)

mysql> select * from college;

Name	Fathername	Birth	SSN	RollNo
GHI	MNO	1990-11-17	645	10
KLM	STR	1988-03-27	665	11
ABC	XYZ	1989-05-03	667	15
DEF	PQR	1987-06-07	785	14
GTH	THR	1989-06-01	667	12
RTU	YHN	1994-07-24	785	19

6 rows in set (0.00 sec)

mysq> select * from birthcertificate;

Name	Fathername	Birth	Location	SSN
GHI	MNO	1990-11-17	BANGLORE	645
KLM	STR	1988-03-27	Delhi	665
ABC	XYZ	1989-05-03	HYDERABAD	667
DEF	PQR	1987-06-07	HYDERABAD	785

4 rows in set (0.00 sec)

mysql> select * from addrproof;

Name1	Name2	Address	SSN
ABC	XYZ	HYDERABAD	667
RTU	YHN	HYDERABAD	378
GHI	MNO	DELHI	645
KLM	STR	BANGLORE	665
GTH	THR	BANGLORE	765
DEF	PQR	CHENNAI	785

6 rows in set (0.00 sec)

c) For bank table

mysql> create table bank (Custname varchar(20), Bankname varchar(20), Accnoint(10)); Query OK, 0 rows affected (0.13 sec)

mysql> desc bank;

Data Mining Lab Manual

Field	Type	Null	Key	Default	Extra
Custname	$\operatorname{varchar}(20)$	YES		NULL	
Bankname	$\operatorname{varchar}(20)$	YES		NULL	
Accno	int(10)	NO	PRI	0	

3 rows in set (0.00 sec)

mysql> insert into bank values('JOHN','SBH',123); Query OK, 1 row affected (0.02 sec)

mysql> insert into bank values('MARRY','SBI',346); Query OK, 1 row affected (0.02 sec)

mysql> insert into bank values('DANY','SBH',768); Query OK, 1 row affected (0.02 sec)

mysql> insert into bank values('NEHA','SBH',512); Query OK, 1 row affected (0.02 sec)

mysql> select * from bank;

Custname	Bankname	Accno
JOHN	SBH	123
MARRY	SBH	346
NEHA	SBH	512
DANY	SBH	768

4 rows in set (0.00 sec)

mysql> desc Customer;

	Field	Туре	Null	Key	Default	Extra
Γ	Custname	varchar(20)	YES		NULL	
	Address	varchar(20)	YES		NULL	
	Accno	int(10)	YES		NULL	

3 rows in set (0.02 sec)

```
mysql> alter table customer add foreign key(Accno) references
bank(Accno);
```

Query OK, 0 rows affected (0.16 sec) Records: 0 Duplicates: 0 Warnings: 0

mysql> desc customer;

Field	Туре	Null	Key	Default	Extra
Custname	$\operatorname{varchar}(20)$	YES		NULL	
Address	$\operatorname{varchar}(20)$	YES		NULL	
Accno	int(10)	YES	MUL	NULL	

3 rows in set (0.00 sec)

mysql> alter table customer add Balance int(30); Query OK, 0 rows affected (0.16 sec) Records: 0 Duplicates: 0 Warnings: 0

mysql> desc customer;

Field	Type	Null	Key	Default	Extra
Custname	$\operatorname{varchar}(20)$	YES		NULL	
Address	$\operatorname{varchar}(20)$	YES		NULL	
Accno	int(10)	YES	MUL	NULL	
Acctype	$\operatorname{varchar}(20)$	YES		NULL	
Balance	int(30)	YES		NULL	

```
5 rows in set (0.00 sec)
```

Data Mining Lab Manual

				0
Custname	Bankname	Accno	Acctype	Balance
JOHN	SBH	123	CURRENT	8000
MARRY	SBH	346	SAVINGS	10000
NEHA	SBH	512	SAVINGS	100000
DANY	SBH	768	CURRENT	15000

4 rows in set (0.00 sec)

Custname	Accno	Balance
JOHN	123	8000
MARRY	346	10000
NEHA	512	100000
DANY	768	15000

4 rows in set (0.02 sec)

PROGRAM-6:

Write a program to implement star schema.

DESCRIPTION:

The star schema (sometimes referenced as star join schema) is the simplest style of data warehouse schema. The star schema consists of a few fact tables (possibly only one, justifying the name) referencing any number of dimension tables. The star schema is considered an important special case of the snowflake schema.



SOURCE CODE:

```
#include<stdio.h>
//#include<conio.h>
struct location
{
int l_key;
char street[30];
char city[30];
char p_st[100];
char country[30];
}1[10];
struct sales_fact
{
int l_key;
int p_key;
int unitsold;
int amount;
}s[20];
struct product
{
int p_key;
char p_name[30];
char pur_date[30];
int cost_price;
int sell_price;
}p[10];
void main()
{
int i,j,n2,n3,k,n1;
//clrscr();
printf("*****location table****");
printf("\n enter the number of entries\n");
scanf("%d",&n1);
for(i=0;i<n1;)</pre>
{
printf("enter the values for l[%d]\n",i+1);
printf("enter the location key");
scanf("%d",&l[i].l_key);
if(i>0)
{
for(j=0;j<i;j++)</pre>
{
if(l[i].l_key==l[j].l_key)
{
printf("\n dups are not allowed");
```
```
break;
}
}
}
if(i==j||i==0)
{
printf("enter the street:");
scanf("%s",1[i].street);
printf("enter the city");
scanf("%s",1[i].city);
fflush(stdin);
printf("enter the country");
scanf("%s",l[i].country);
i++;
}
}
printf("*****production table*****");
printf("\n enter the number of entries\n");
scanf("%d",&n2);
for(i=0;i<n2;)</pre>
{
printf("enter the values for p[%d]\n",i+1);
printf("enter the product key");
scanf("%d",&p[i].p_key);
if(i>0)
{
for(j=0;j<i;j++)</pre>
{
if(p[i].p_key==p[j].p_key)
{
printf("\n dups are not allowed");
break;
}
}
}
if(i==j||i==0)
ſ
printf("enter the p_name");
fflush(stdin);
scanf("%s",p[i].p_name);
printf("enter the pur_data");
scanf("%s",p[i].pur_date);
fflush(stdin);
printf("enter the costprice");
scanf("%d",&p[i].cost_price);
printf("enter the sellprice");
```

```
scanf("%d",&p[i].sell_price);
i++;
}
}
printf("\n enter the no fact table entries:");
scanf("%d",&n3);
for(i=0;i<n3;)</pre>
{
printf("enter the values for s[%d]\n",i+1);
printf("enter the location key");
scanf("%d",&s[i].l_key);
printf("enter the product key");
scanf("%d",&s[i].p_key);
for(j=0;j<n1;j++)</pre>
{
if(s[i].l_key==1[j].l_key)
break;
}
for(k=0;k<n2;k++)</pre>
{
if(s[i].p_key==p[k].p_key)
break;
}
if(!(j<n1&&k<n2))
printf("\n he daa must be there in the d tables:");
else
{
printf("\n accept nth units sold:");
scanf("%d",&s[i].unitsold);
s[i].amount=s[i].unitsold*p[k].sell_price;
i++;
}
}
printf("\n \n the entries of location table is: \n");
for(i=0;i<n2;i++)</pre>
ł
printf("%d\t %s\t %s\n",1[i].1_key, 1[i].street,
     l[i].city, l[i].p_st, l[i].country);
}
printf("\n entries of product table is: \n");
for(i=0;i<n2;i++)</pre>
Ł
printf("%d\t %s\t %s\t %d\t %d\n",p[i].p_key,p[i].p_name,
       p[i].pur_date, p[i].cost_price, p[i].sell_price);
}
printf("the entries of fact_sales table:");
```

Output:



PROGRAM-6A:

write a program to implement K-Means algorithm.

PROBLEM DEFINITION: write a program to implement K-Means algorithm.

SOURCE CODE:

```
#include<stdio.h>
int mod(int k)
{
if(k>0) return k;
else return -k;
}
int small(int b[],int n)
{
 int m,pos,r=0; m=b[0];
for(pos=0;pos<n;pos++)</pre>
{
if(m>b[pos]) { m=b[pos];
 r=pos;
}
}
 return r;
}
void main()
{
int n,j,s=0;
int x=0,y=0,z=0;
int obj[20],c[20][20],mean[20],a[20];
int i,nc,k,m,min,count;
printf("\n\n Enter no. of items");
scanf("%d",&n);
printf("\n Enter n items");
for(i=0;i<n;i++)</pre>
scanf("%d",&obj[i]);
printf("\n Enter no of clusters");
scanf("%d",&nc);
for(i=0;i<nc;i++)</pre>
for(j=0;j<n;j++)</pre>
{
c[i][j]=0; a[i]=0; }
for(i=0;i<nc;i++)</pre>
{
```

```
c[i][0]=obj[i];
mean[i]=obj[i];
}
for(i=0;i<nc;i++)</pre>
for(j=0;j<n;j++)</pre>
 if(c[i][j]>0)
printf(" I:%d",c[i][j]);
j=nc;
for(i=1;i<n;i++)</pre>
{
if(j<n)
{
for(k=0;k<nc;k++)
a[k]=mod(obj[j]-mean[k]);
min=small(a,nc);
c[min][i]=obj[j];
for(k=0;k<nc;k++)
{
s=0;count=0;
for(m=0;m<n;m++)
{
if(c[k][m]>0)
ſ
 s=s+c[k][m];
count++;
}
}
mean[k]=s/count;
}
for(k=0;k<nc;k++)
printf("\n mean values..%d\t",mean[k]);
printf("\n");
 j++;
}
}
for(i=0;i<nc;i++)</pre>
ſ
 printf("\n");
for(j=0;j<n;j++)</pre>
{
if(c[i][j]>0)
printf("%d\t",c[i][j]);
}
}
```

Output:

"kmeans.c" [dos] 78L, 1390C written [cse1170@cse programs]\$ cc kmeans.c kmeans.c: In function `main': kmeans.c:17: warning: return type of 'main' is not `int' [cse1170@cse programs]\$./a.out Enter no. of items 2 Enter n items 3 4 Enter no of clusters 5 I:3 I:4 3

PROGRAM-6B:

K-Means algorithm for two dimension data items.

PROBLEM DEFINITION: K-Means algorithm for two dimension data items.

```
SOURCE CODE:
```

```
#include<stdio.h>
#include<math.h>
double distance(int a[][2],double b[][2],int j,int k)
{
double n=0,x1,y1,total;
int x,y;
x=a[j][0]-b[k][0];
y=a[j][1]-b[k][1];
x1=x*x;
y1=y*y;
total=x1+y1;
n=sqrt(total);
return n;
}
int small(double b[],int n)
{
int pos,r=0;double m=b[0];
for(pos=0;pos<n;pos++)</pre>
{
if(m>b[pos])
{
m=b[pos];
r=pos;
}
}
return r;
}
void main()
{
int n,j,s=0;
int x=0,y=0,z=0;
int x1,y1;
int obj[20][2],c[20][20][2];
double mean[20][2];
double a[20];
int i,nc,k,m,min,count;
printf("\n\n Enter no. of items");
```

```
scanf("%d",&n);
printf("\n Enter n items");
for(i=0;i<n;i++)</pre>
for(k=0;k<2;k++)</pre>
scanf("%d",&obj[i][k]);
printf("\n Enter no of clusters");
scanf("%d",&nc);
for(i=0;i<nc;i++)</pre>
for(j=0;j<n;j++)</pre>
{
 for(k=0;k<2;k++)
c[i][j][k]=0; a[i]=0;
}
for(i=0;i<nc;i++)</pre>
{
j=0;
for(k=0;k<2;k++)</pre>
{
c[i][j][k]=obj[i][k];
mean[i][k]=obj[i][k];
}
}
for(i=0;i<nc;i++)</pre>
{
printf("\nI%d:",i);
for(j=0;j<n;j++)</pre>
for(k=0;k<2;k++)
if(c[i][j][k]>0)
printf("%d ",c[i][j][k]);
printf("\n");
}
for(i=0;i<nc;i++)</pre>
{
for(k=0;k<2;k++)</pre>
printf("\n mean values...%lf ",mean[i][k]);
printf("\n");
}
j=nc;
for(i=1;i<n;i++)</pre>
{
if(j<n)
{
for(k=0;k<nc;k++)
a[k]=distance(obj,mean,j,k);
min=small(a,nc);
c[min][i][0]=obj[j][0];
```

```
c[min][i][1]=obj[j][1];
for(m=0;m<n;m++)</pre>
{
x1=0;y1=0;count=0;
for(k=0;k<nc;k++)
{
if(c[m][k][0]>0||c[m][k][1]>0)
{
x1=x1+c[m][k][0];
y1=y1+c[m][k][1];
count++;
}
}
if(count>0)
{
mean[k][0]=x1/count;
mean[k][1]=y1/count;
}
}
j++;
}
}
for(i=0;i<nc;i++)</pre>
{
for(j=0;j<n;j++)</pre>
for(k=0;k<2;k++)</pre>
printf("%d ",c[i][j][k]);
printf("\n");
}
printf("final kmean values are....\n");
for(i=0;i<nc;i++)</pre>
printf("%lf....%lf\n",mean[i][0],mean[i][1]);
}
```

Output:



PROGRAM-7:

a program to implement k-mediod algorithm

PROBLEM DEFINITION: Write a program to implement k-mediod algorithm

SOURCE CODE:

```
#include<stdio.h>
#include<math.h>
int distance(int [],int []);
int i,j,n,nc=3;
void main()
{
int j,count,t;
int obj[10][2],c[10][10][2],mean[10][2],c1[10][10][2];
int i,k,m,cost=0,cost1;
printf("\n enter the no. of items:");
scanf("%d",&n);
printf("\n enter the items(%d)",n);
for(i=0;i<n;i++)</pre>
for(j=0;j<2;j++)</pre>
scanf("%d",&obj[i][j]);
for(i=0;i<nc;i++)</pre>
for(j=0;j<n;j++)</pre>
for(k=0;k<2;k++)</pre>
{
c[i][j][k]=0;
c1[i][j][k]=0;
}
```

```
printf("\n enter center points");
for(i=0;i<nc;i++)</pre>
for(j=0;j<2;j++)</pre>
ſ
scanf("%d",&mean[i][j]);
c[i][0][j]=mean[i][j];
}
j=0;
for(i=1;i<=n;i++)</pre>
{
if(j<n)
{
if(distance(obj[j],mean[0])<distance(obj[j],mean[1]))</pre>
if(distance(obj[j],mean[0])<distance(obj[j],mean[2]))</pre>
for(k=0;k<2;k++)
{
 c[0][i][k]=obj[j][k];
 cost=cost+distance(obj[j],mean[0]);
}
if(distance(obj[j],mean[1])<distance(obj[j],mean[0]))</pre>
if(distance(obj[j],mean[1])<distance(obj[j],mean[2]))</pre>
for(k=0;k<2;k++)
ſ
c[1][i][k]=obj[j][k];
 cost=cost+distance(obj[j],mean[1]);
}
if(distance(obj[j],mean[2])<distance(obj[j],mean[0]))</pre>
if(distance(obj[j],mean[2])<distance(obj[j],mean[1]))</pre>
for(k=0;k<2;k++)</pre>
ſ
 c[2][i][k]=obj[j][k];
 cost=cost+distance(obj[j],mean[2]);
}
j++;
}
}
printf("\n enter the next center points:");
for(i=0;i<nc;i++)</pre>
for(j=0;j<2;j++)</pre>
{
scanf("%d",&mean[i][j]);
c1[i][0][j]=mean[i][j];
}
j=0;
for(i=1;i<=n;i++)</pre>
{
```

```
if(j<n)
{
 if(distance(obj[j],mean[0])<distance(obj[j],mean[1]))</pre>
 if(distance(obj[j],mean[0])<distance(obj[j],mean[2]))</pre>
 for(k=0;k<2;k++)</pre>
{
c1[0][i][k]=obj[j][k];
cost1=cost1+distance(obj[j],mean[0]);
}
if(distance(obj[j],mean[1])<distance(obj[j],mean[0]))</pre>
if(distance(obj[j],mean[1])<distance(obj[j],mean[2]))</pre>
for(k=0;k<2;k++)</pre>
{
c1[1][i][k]=obj[j][k];
cost1=cost1+distance(obj[j],mean[1]);
}
if(distance(obj[j],mean[2])<distance(obj[j],mean[0]))</pre>
if(distance(obj[j],mean[2])<distance(obj[j],mean[1]))</pre>
for(k=0;k<2;k++)</pre>
{
c[2][i][k]=obj[j][k];
cost1=cost1+distance(obj[j],mean[2]);
}
 j++;
}
}
if(cost<cost1)
{
for(i=0;i<nc;i++)</pre>
ſ
printf("\n");
for(j=0;j<n;j++)</pre>
for(k=0;k<2;k++)</pre>
{
 if(c[i][j][k]>0)
printf("%d\t",c[i][j][k]);
}
}
}
else
{
for(i=0;i<nc;i++)</pre>
{
printf("\n");
for(j=0;j<n;j++)</pre>
for(k=0;k<2;k++)</pre>
```

```
{
if(c1[i][j][k]>0)
printf("%d\t",c1[i][j][k]);
}
}
}
}
 int distance(int obj[],int mean[])
{
 int x1,x2,y1,y2,dist;
 x1=obj[0];
 x2=mean[0];
 y1=obj[1];
 y2=mean[1];
dist=(sqrt(pow((x1-x2),2)+pow((y1-y2),2)));
 return dist;
 }
```

Output:



Data Mining Lab Manual CASE STUDY 1- K-NEAREST NEIGHBOR

DEFINITION:

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n dimensional space. In this way, all of the training tuples are stored in an n dimensional pattern space. When given an unknown tuple, a k nearest neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k "nearest neighbors" of the unknown tuple.

WORKING:

The K nearest-neighbor method was first described in the early 1950s. The method is labor intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition.

"Closeness" is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples

XQ=(x11, x12 x1n) and X2 = (x21, x22... x2n), is

$$d(i,j) = \sqrt{(|x_{j_1} - x_{j_1}|^2 + |x_{j_2} - x_{j_2}|^2 + \dots + |x_{j_p} - x_{j_p}|^2)}$$

In other words, for each numeric attribute, we take the difference between the corresponding values of that attribute in tuple Xi and in tuple X2, square this difference, and accumulate it. The square root is taken of the total accumulated distance count.

Typically, we normalize the values of each attribute before using Equation This helps prevent attributes with initially large ranges (such as income) from outweighing attributes with initially smaller ranges (such as binary attributes). Min-max normalization, for example, can be used to transform a value v of a numeric attribute A to v' in the range [0, 1] by computing

$$v' = \frac{v - \min_A}{\max_A - \min_A},$$

where minA and maxA are the minimum and maximum values of attribute A. For k-nearest neighbor classification, the unknown tuple is assigned the most common class among its k nearest neighbors. When k = 1, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space. Nearest-neighbor classifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple. In this case, the classifier returns the average value of the real-valued labels associated with the k nearest neighbors of the unknown tuple.

To compute for attributes that not numeric, but categorical, such as color The above discussion assumes that the attributes used to describe the tuples are all numeric. For categorical attributes, a simple method is to compare the corresponding value of the attribute in tuple X1 with that in tuple X2. If the two are identical (e.g., tuples X1 and X2 both have the color blue), then the difference between the two is taken as 0. If the two are different (e.g., tuple X1 is blue but tuple X2 is red), then the difference is considered to be 1. Other methods may incorporate more sophisticated schemes for differential grading (e.g., where a larger difference score is assigned, say, for blue and white than for blue and black).

The procedure to compute if the there are missing values In general, if the value of a given attribute A is missing in tuple X1 and/or in tuple X2, we assume the maximum possible difference. Suppose that each of the attributes have been mapped to the range [0, 1]. For categorical attributes, we take the difference value to be 1 if either one or both of the corresponding values of A are missing. If A is numeric and missing from both tuples X1 and X2, then the difference is also taken to be 1. If only one value is missing and the other (which we'll call v') is present and normalized, then we can take the difference to be either

 $|1 \ v'|$ or $|0 \ v'|$ (i.e., $1 \ v'$ or v'), whichever is greater.

To determine a good value for k, the number of neighbors

This can be determined experimentally. Starting with k = 1, we use a test set to estimate the error rate of the classifier. This process can be repeated each time by incrementing k to allow for one more neighbors. The k value that gives the

minimum error rate may be selected. In general, the larger the number of training tuples is, the larger the value of k will be (so that classification and prediction decisions can be based on a larger portion of the stored tuples). As the number of training tuples approaches infinity and k = 1, the error rate can be no worse then twice the Bayes error rate (the latter being the theoretical minimum). If k also approaches infinity, the error rate approaches the Bayes error rate.

DISADVANTAGES OF KNN AND TECHINIQUES TO OVERCOME THEM:

Nearest-neighbor classifiers use distance-based comparisons that intrinsically assign equal weight to each attribute. They therefore can suffer from poor accuracy when given noisy or irrelevant attributes. The method, however, has been modified to incorporate attribute weighting and the pruning of noisy data tuples. The choice of a distance metric can be critical. The Manhattan (city block) distance or other distance measurements may also be used.

Nearest neighbor classifiers can be extremely slow when classifying test tuples. If D is a training database of ||D|| tuples and k = 1, then O(||D||) comparisons are required in order to classifier.

Parallel implementation can reduce the running time to a constant that is 0(1), which is independent of ||D||.

Other techniques to speed up classification time include the use of partial distance calculations and editing the stored tuples. In the partial distance method, we compute the distance based on a subset of the n attributes. If this distance exceeds a threshold, then further computation for the given stored tuple is halted, and the process moves on to the next stored tuple. The editing method removes training tuples that prove useless. This method is also referred to as pruning or condensing because it reduces the total number of tuples stored.

Data Mining Lab Manual CASE STUDY 2 - KDD PROCESS

DEFINITION:

KDD Process is the process of using data mining methods (algorithms) to extract (identify) what is deemed knowledge according to the specifications of measures and thresholds, using database F along with any required preprocessing, subsampling, and transformation of F.

KDD:

- In a multistep process many decisions are made by the user (domain expert):
- Iterative and interactive loops between any two steps are possible
- Usually the most focus is on the DM step, but other steps are of considerable importance for the successful application of KDD in practice

GOALS:

- Verification of users hypothesis (this against the EDA principle)
- Autonomous discovery of new patterns and models
- Prediction of future behavior of some entities
- Description of interesting patterns and models

STEPS OF DM:

- a. Domain understanding and goal setting
- b. Creating a target data set
- c. Data cleaning and preprocessing
- d. Data reduction and projection
- e. Data mining
 - Choosing the data mining task
 - Choosing the data mining algorithm(s)
 - Use of data mining algorithms
- f. Interpretation of mined patterns
- g. Utilization of discovered knowledge



a. Domain analysis

- Development of domain understanding
- Discovery of relevant prior knowledge
- Definition of the goal of the knowledge discovery

b. Data selection

- Selection and integration of the target data from possibly many different and heterogeneous sources
- Interesting data may exist, e.g., in relational databases, document collections, e mails, photographs, video clips, process database, customer transaction database, web logs etc.
- Focus on the correct subset of variables and data samples
- E.g., customer behavior in a certain country, relationship between items purchased and customer income and age

c. Data cleaning and preprocessing

- Dirty data can confuse the mining procedures and lead to unreliable and invalid outputs
- Complex analysis and mining on a huge amount of data may take a very long time
- Preprocessing and cleaning should improve the quality of data and mining results by enhancing the actual mining process
- The actions to be taken includes
- Removal of noise or outliers
- Collecting necessary information to model or account for noise

- Using prior domain knowledge to remove the inconsistencies and duplicates from the data
- Choice or usage of strategies for handling missing data fields

d. Data reduction and projection

- Data transformation techniques
- Smoothing (binning, clustering, regression etc.)
- Aggregation (use of summary operations (e.g., averaging) on data)
- Generalization (primitive data objects can be replaced by higher level concepts)
- Normalization (min max-scaling, z score)
- Feature construction from the existing attributes (PCA, MDS)
- Data reduction techniques are applied to produce reduced representation of the data (smaller volume that closely maintains the integrity of the original data)
- Aggregation
- Dimension reduction (Attribute subset selection, PCA, MDS,)
- Compression (e.g., wavelets, PCA, clustering,)
- Numerosity reduction
- parametric models: regression and log linear models
- non parametric models: histograms, clustering, sampling
- Discretization (e.g., binning, histograms, cluster analysis,)
- Concept hierarchy generation (numeric value of age to a higher level concept young, middle aged, senior)

e. Choice of data mining task

- Define the task for data mining
- Exploration/summarization
- Summarizing statistics (mean, median, mode, std,..)
- Class/concept description
- Explorative data analysis
- Graphical techniques, low dimensional plots,
- Predictive
- Classification or regression
- Descriptive
- Cluster analysis, dependency modelling, change and outlier detection
- f. Choosing the DM algorithm(s)
 - Select the most appropriate methods to be used for the model and pattern search
 - Matching the chosen method with the overall goal of the KDD process (necessites communication between the end user and method specialists)
 - Note that this step requires understanding in many fields, such as computer science, statistics, machine learning, optimization, etc.

g. Use of data mining algorithms

- Application of the chosen DM algorithms to the target data set
- Search for the patterns and models of interest in a particular representational form or a set of such representations
- Classification rules or trees, regression models, clusters, mixture models
- Should be relatively automatic
- Generally DM involves:
- Establish the structural form (model/pattern) one is interested
- Estimate the parameters from the available data
- Interprete the fitted models

h. Interpretation/evaluation

- The mined patterns and models are interpreted
- The results should be presented in understandable form
- Visualization techniques are important for making the results useful mathematical models or text type descriptions may be difficult for domain experts
- Possible return to any of the previous step

Data Mining Lab Manual CASE STUDY 3 DECISION TREE INDUCTION

DEFINITION:

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flow chart like tree structure, where each internal node(non leaf node) denotes a test on an attribute , each branch represents an outcome of the test and each leaf node(or terminal node) holds a class label. The topmost node in a tree is the root node.

EXAMPLE:

A decision tree for buys _ computer , indicating whether a customer at AllElectronics is likely to purchase a computer. Each internal node represents a test on an attribute. Each leaf node represents a class (either buys _ computer=yes or buys _ computer=no)



ALGORITHM:

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm. This Decision Tree Algorithm is known as ID3(Iterative Dichotomiser). Later he gave C4.5 which was successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm there is no backtracking, the trees are constructed in a top down recursive divide-and-conquer manner.

Generating a decision tree form training tuples of data partition D

Algorithm : Generate_decision_tree

Input:

Data partition, D, which is a set of training tuples and their associated class labels. attribute_list, the set of candidate attributes. Attribute selection method, a procedure to determine the splitting criterion that best partitions that the data tuples into individual classes. This criterion includes a splitting_attribute and either a splitting point or splitting subset.

Output:

A Decision Tree

Method:

```
create a node N;
if tuples in D are all of the same class, C then
  return N as leaf node labeled with class C;
if attribute_list is empty then
  return N as leaf node with labeled
  with majority class in D; || majority voting
apply attribute_selection_method(D, attribute_list)
to find the best splitting_criterion;
label node N with splitting_criterion;
if splitting_attribute is discrete-valued and
  multiway splits allowed then
// no restricted to binary trees
attribute_list = splitting attribute;
// remove splitting attribute
for each outcome j of splitting criterion
  // partition the tuples and grow subtrees
for each partition
  let Dj be the set of data tuples in D
satisfying outcome j;
// a partition
  if Dj is empty then
     attach a leaf labeled with the majority
      class in D to node N;
  else
      attach the node returned by Generate
     decision tree(Dj, attribute list) to node N;
```

end for return N;

TREE PRUNING

Tree Pruning is performed in order to remove anomalies in training data due to noise or outliers. The pruned trees are smaller and less complex.

Tree Pruning Approaches

Here is the Tree Pruning Approaches listed below:

- Pre pruning The tree is pruned by halting its construction early.
- Post pruning This approach removes sub-tree form fully grown tree.

COST COMPLEXITY

The cost complexity is measured by following two parameters:

- Number of leaves in the tree
- Error rate of the tree

ADVANTAGES OF DECISION TREE

- It does not require any domain knowledge.
- It is easy to assimilate by human.
- Learning and classification steps of decision tree are simple and fast.

Data Mining Lab Manual CASE STUDY 4 APRIORI

Problem Definition

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

WORKING:

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold, the Apriori algorithm identifies the item sets which are subsets of at least transactions in the database.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length from item sets of length . Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent -length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.



EXAMPLE:

Consider the following database, where each row is a transaction and each cell is an individual item of the transaction:

alpha beta epsilon alpha beta Theta alpha beta epsilon alpha beta Theta

The association rules that can be determined from this database are the following:

- a. 100% of sets with alpha also contain beta
- b. 50% of sets with alpha, beta also have epsilon
- c. 50% of sets with alpha, beta also have theta we can also illustrate this through variety of examples

LIMITATIONS:

Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset $2\sup|S| - 1onlya fterallo fits proper subsets$.

Later algorithms such as Max-Miner try to identify the maximal frequent item sets without enumerating their subsets, and perform "jumps" in the search space rather than a purely bottom-up approach.

ANNEXURE I List of programs according to O.U. Curriculum WITH EFFECT FROM THE ACADEMIC YEAR 2013-2014 DATA MINING LAB CS 481

Instruction2Periods per weekDuration of University Examination3HoursUniversity Examination50MarksSessional25Marks

- Implement the following Multidimensional Data Models
 Star Schema
 Snowflake Schema
- iii. Fact Constellation

2. Implement Apriori algorithm to generate frequent Item Sets

3. Implement the following clustering algorithms

- i. K-means
- ii. K-mediods

4. Implement the following classification algorithmsi. Decision Tree Inductionii. KNN

- 5. Perform data Pre processing using WEKA
- 6. Perform Discretization of data using WEKA
- 7. Classification algorithms using WEKA
- 8. Apriori algorithm using WEKA
- 9. Perform data transformations using an ETL Tool
- 10. A small case study involving all stages of KDD. (Datasets are available online like UCI Repository etc.)