

# **Semester I**

# **Data Mining**

<u>Sr.</u> No.	<u>Title</u>	<u>Sign</u>
1	Prepare the Analysis services for Adventure Works Cycles or (any other database) Build the data mining model structure and built the decision tree with proper decision nodes. And infer atleast five different types of reports	
2	Prepare the Analysis services for Adventure Works Cycles or (any other database) .Build the data mining model structure. Implement the clustering Algorithm.	
3	Prepare the Analysis services for Adventure Works Cycles or (any other database) .Build the data mining model structure and Implement Naïve Bayes Algorithm.	
4	Prepare the Analysis services for Adventure Works Cycles or (any other database) .Build the basic Time series model structure and create the predictions	
5	Prepare the Analysis services for Adventure Works Cycles or (any other database) .Build the basic data mining model and show the implementation of Association algorithm	
6	Using R-Tool, show the analysis for social networking sites.	
7	Consider the suitable data for text mining and Implement the Text Mining technique using R-Tool	
8	Consider the suitable data for Apriori Algorithm and Implement the Apriori Algorithm using R-Tool.	

### Practical 1:

<u>Aim:</u> Prepare the Analysis services for Adventure Works Cycles or (any other database) Build the data mining model structure and built the decision tree with proper decision nodes. And infer atleast five different types of reports

### **Theory:**

Decision tree induction is the learning of decision trees from class- labelled training tuples. A decision tree is a flowchart- 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. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals.



A typical decision tree is shown in Figure. It represents the concept buys computer, that is, it predicts whether a customer at AllElectronics is likely to purchase a computer.

### The benefits of having a decision tree are as follows -

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

### Tree Pruning

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

### **Tree Pruning Approaches**

There are two approaches to prune a tree -

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

### Cost Complexity

The cost complexity is measured by the following two parameters -

- Number of leaves in the tree, and
- Error rate of the tree.

Algorithm: Generate\_decision\_tree. Generate a decision tree from the training tuples of data partition D.

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 the data tuples into individual classes. This criterion consists of a splitting\_attribute and, possibly, either a split point or splitting subset.

Output: A decision tree.

### **Practical**

### Step 1: Open Weka



# **Step 2: Click on Explorer**

🜍 Weka Explorer				– 🗆 X
Preprocess Classify Cluster Associate	e Select attributes Visualize Forecast			
Open file Open URL	Open DB Gene	rate Un	do Edit	Save
Filter				
Choose None				Apply Stop
Current relation		Selected attribute		
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
All None All Remo	Invert Pattern			Visualize All
Welcome to the Weka Explorer				Log 🛷 x0

### Step 3: Click on Open File Option

🥝 Weka Explorer		- 🗆 ×
Preprocess Classify Cluster	Associate Select attributes Visualize Forecast	
Open file Op	pen URL Open DB Generate Undo Edit	Save
Choose None	Open :	X Apply Stop
Current relation Relation: None Instances: None	Look In: Documents	Type: None Unique: None
All No	Wirtual Machines     Note:       Visual Studio 2015     Some file formats offer additional options which can be customized when invoking the options dialog	
	File Name:       Files of Type:       Arff data files (*.arff)	Visualize All
	Open Cancel	
	Remove	
Status Welcome to the Weka Explorer		Log 📣 x0

Step 4: Go to C://Program Files/Weka-3-8/data. Select weather.nominal.arff database

Weka Explorer	r Associate Select attributes Visualize Forecast	- 🗆 X
Open file O	pen URL Open DB Generate Undo Edit	Save
Choose None Current relation Relation: None Instances: None Attributes All No	Copen X Look In: adata ff Reuters Grain-test aff Unbalanced.aff Unbalanced.aff Note: Corn-test aff Segment-challenge.aff Vote.aff Some file formats offer additional options which can be customized when invoking the options dialog	Apply Stop Type: None Unique: None
	File Name: weather.numeric.arff Files of Type: Arff data files (*.arff) Open Cancel	Visualize All
Status Welcome to the Weka Explorer	Remove	Log x0

# Step 5: Open the database

Preprocess       Classify       Cluster       Associate       Select attributes       Forecast         Open file       Open URL       Open DB       Generate       Undo       Edit       Save         Filter       Choose       None       Apply       Stop         Current relation       Selected attributes       5       Selected attribute         Relation: weather       Attributes: 5       Name: outlook       Type: Nominal         Instances: 14       Sum of weights: 14       Missing: 0 (0%)       Distinct: 3       Unique: 0 (0%)         No.       Name       Invert       Patern       5       5.0       2       overcast       4       4.0         No.       Name       Name       Stop       5.0       5.0       5.0       5.0	🥝 Weka Explorer		- 🗆 X
Open file       Open URL       Open DB       Generate       Undo       Edit       Save         Filter       Apply       Stop       Apply       Stop         Current relation       Selected attribute       Name: outlook       Type: Nominal         Relation: weather Instances: 14       Sum of weights: 14       Name: outlook       Type: Nominal         Attributes       Sum of weights: 14       Missing: 0 (0%)       Distinct: 3       Unique: 0 (0%)         No.       Name       Invert       Pattern       5       5.0         No.       Name       Sum of weights: 14       Au       4.0	Preprocess Classify Cluster Associate Select attributes Visualize Foreca	ist	
Selected attribute       Current relation       Relation: weather Instances: 14     Attributes: 5 Sum of weights: 14       Attributes       All     None       Invert     Pattern       No.     Name	Open file Open URL Open DB Ge	nerate Undo Edit	t Save
Choose     None     Apply     Stop       Current relation     Selected attributes: 5 Instances: 14     Sum of weights: 14       Attributes     Name: outlook     Type: Nominal Missing: 0 (0%)       All     None     Invert       No.     Name       No.     Name	Filter		
Current relation     Selected attribute       Relation: weather Instances: 14     Attributes: 5 Sum of weights: 14     Name: outlook     Type: Nominal Unique: 0 (0%)       Attributes     None     Invert     Pattern       No.     Name     1     sunny       No.     Name     5     5.0       2     overcast     4     4.0       3     rainy     5     5.0	Choose None		Apply Stop
Relation: weather Instances: 14     Attributes: 5 Sum of weights: 14     Name: outlook     Type: Nominal Distinct: 3       Attributes     None     Invert     Pattern       No.     Name	Current relation	Selected attribute	
Attributes     No.     Label     Count     Weight       All     None     Invert     Pattern       No.     Name	Relation: weather Attributes: 5 Instances: 14 Sum of weights: 14	Name: outlook Missing: 0 (0%) Distinct: 3	Type: Nominal Unique: 0 (0%)
All         None         Invert         Pattern         1         sunny         5         5.0         2         overcast         4         4.0         3         rainy         5         5.0         5.0         5.0         2         overcast         4         4.0         3         rainy         5         5.0	Attributes	No. Label Count	Weight
1 outlook 2 temperature	All None Invert Pattern No. Name 1 outlook 2 temperature	1 sunny 5 2 overcast 4 3 rainy 5	5.0 4.0 5.0
3     humidity       4     windy       5     play         Remove         Status         OK	3 humidity 4 windy 5 play Remove	Class. play (10011)	S Log

# **Step 6: Select all the attributes**

Weka Explorer			– 🗆 X
Preprocess Classify Cluster Associate Select attributes Visualize Foreca	st		
Open file Open URL Open DB Ger	nerate Undo	Edit	Save
Filter			
Choose None			Apply Stop
Current relation	Selected attribute		
Relation: weather         Attributes: 5           Instances: 14         Sum of weights: 14	Name: outlook Missing: 0 (0%)	Distinct: 3	Type: Nominal Unique: 0 (0%)
Attributes	No. Label	Count	Weight
All None Invert Pattern No. Name 1 J outlook	1 sunny 2 overcast 3 rainy	5 4 5	5.0 4.0 5.0
2 V temperature 3 V humidity 4 V windy 5 V play Remove	Class: play (Nom)	4	▼ Visualize All
Status			
ок			Log 📣 x0

# Step 7: Go to Classify tab

🚱 Weka Explorer	-		×
Preprocess Classify Cluster Associate Select attributes Visualize Forecast			
Classifier			
Choose ZeroR			
Test options Classifier output			
Use training set         Supplied test set         Supplied test set         Cross-validation         Folds         Percentage split         %         More options			
(Nom) play       Start     Stop       Result list (right-click for options)			
Status			
OK	Log	-00	». x0

Step 8: Choose algorithm to apply. For that Go to->Choose tab->Trees-> J48

🥥 Weka Explo	orer					
Preprocess	Classify	Cluster	Associate	Select attributes	Visualize	Forecast
Classifier						
Veka				-		
	assifiers					
	bayes			-		
N	functions					
	lazy					
	meta					
N 🕨 🕨	misc					
► 🗎	rules					
► 📄	timeseries					
▼ 🚍	trees					
	📄 Decisio	nStump				
2	🕒 Hoeffdii	ngTree				
0	🕒 J48					
9	🕒 LMT					
	M5P					
4	🕒 Randor	nForest				
Re	🕒 Randor	nTree				
	REPTre	e				





Step 10: Right click on the created structure and select Visualize Tree option

Preproces	Glassify	Cluster	Associal	te I	Select att	ibutes	Visualiz	e Forecas	#]							
lassifier							-	_								
	1															
Choose	J48-C 0.2	25-M 2														
Fest options			Cla	issil	ier output											
O Use to	sining set			oon	rectary o	19991	1100 1005	ances	,		\$9.6507					
0 and		(C + 1		Inco	crectly	Clas	sified In	stances	5		35.7143	1				- 61
O Subbe	ed test set	Set		Kapp	pa stati	stic			0.18	6						
Cross	validation F	olds 10		Near	h absolu	te er	ror		0.28	57						
0.0				Bela	s mean o	quare	d error		40							
O Percer	nage spik	76 00		Root	t relati	ve ac	uared err	or	97.65	16 1						
	Nore options			Tote	al Numbe	r of	Instances		14							
			- 11													
					Detaile	d Acc	uracy By	Class								
(Nom) play										B				-		
			_				o 770	fr sate	Precision	Recall 0.770	r-measure	NCC 0.149	a Teo	PRC Area	Class	
Start		Stop					0.400	0.222	0.500	0.400	0.444	0.189	0.785	0.738	pea po	
esult list (ri	ght-click for	options)		Neig	phted 3w	<b>.</b>	0.643	0.465	0.629	0.643	0.632	0.109	0.709	0.000		5
10.0040	hann 140				_											
19/23/12 0	View in ma	in window			Conduct	рп Иа	trix									
	View in se	parate win	dow			-1488	ified as									
	Save result	t buffer				res										
	Delete res	ult buffer(s	()			-0										
	Load mod	el														
	Save mod	el														٧
Tabua .	Re-evaluat	te model o	n current t	lest	set											
carus	Re-apply th	his model	s configur	ation	1											
OK	Visualize o	dessifier e	mora											Log	-10	p. xi
-	Visualize b	160														_
	Visualize n	nargin cun	49			- 7	-									
	Visualize ti	hreshold o	urve				-									
		-							_							_

Step 11: View the Decision tree



### Practical 2:

Aim: Prepare the Analysis services for Adventure Works Cycles or (any other database) .Build the data mining model structure and Implement the clustering Algorithm.

**Theory:** Clustering can be considered the most important unsupervised learning problem. So, as every other problem of this kind, it deals with finding collection a structure in а of unlabelled data. A loose definition of clustering could be "the process of organizing objects into members are similar some whose in wav". groups A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.

We can show this with a simple graphical example:



In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is distance: two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called distance-based *clustering*.

Another kind of clustering is conceptual clustering: two or more objects belong to the same cluster if this one defines a concept *common* to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.

Clustering algorithms may be classified as listed below:

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering
- Probabilistic Clustering

K-means is an *exclusive clustering* algorithm. K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem.

The algorithm is composed of the following steps:

- 1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
- 2. Assign each object to the group that has the closest centroid.
- 3. When all objects have been assigned, recalculate the positions of the *K* centroids.
- 4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

### **Practical**

### Step 1: Open Weka

🥥 Weka GUI Chooser		_		$\times$	
Program Visualization Tools Help					
	App	olicati	ons		
		E	xplorer		
WEKA The University of Waikato		Experimenter			
		KnowledgeFlow			
Waikato Environment for Knowledge Analysis		Workbench			
Version 3.8.2 (c) 1999 - 2017 The University of Waikato Hamilton, New Zealand		Simple CLI			

# **Step 2: Click on Explorer**

🜍 Weka Explorer				– 🗆 X
Preprocess Classify Cluster Associate	Select attributes Visualize Forecast	]		
Open file Open URL	Open DB Gene	rate Un	do Edit	Save
Filter				
Choose None				Apply Stop
Current relation		Selected attribute		
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
All None	Invert Pattern			Visualize All
Status				
Welcome to the Weka Explorer				Log 🛷 x0

### Step 3: Click on Open File Option

Weka Explorer					- 🗆 X
Preprocess Classify Cluster	Associate	Select attributes Visualize	Forecast		
Open file Op	pen URL	Open DB	Generate	Undo Edit	Save
Choose None	🖸 Open			×	Apply Stop
Current relation Relation: None Instances: None Attributes	Look In:	Documents s ites	•	Invoke options dialog	Type: None Unique: None
All	📄 Virtual Ma 📄 Visual Stu	chines dio 2015		Note: Some file formats offer additional options which can be customized when invoking the options dialog.	
	File <u>N</u> ame: Files of <u>T</u> ype:	Arff data files (*.arff)			Visualize All
				Open Cancel	
	Remove				
Status Welcome to the Weka Explorer					Log 💉 x 0

# Step 4: Go to C://Program Files/Weka-3-8/data. Select diabetes.arff database

Weka Explorer  Cluster Associate Select attributes Visualize Economic	×
	31
Open file Open URL Open DB Gen	ierate Undo Edit Save
Filter	
Choose None	Apply Stop
Current relation	Selected attribute
Relation: pima_diabetes     Attributes: 9       Instances: 768     Sum of weights: 768	Name: pregType: NumericMissing: 0 (0%)Distinct: 17Unique: 2 (0%)
Attributes	Statistic Value
All None Invert Pattern No. Name 1 J preg	Minimum 0 Maximum 17 Mean 3.845 StdDev 3.37
2 ✔ plas 3 ✔ pres 4 ✔ skin 5 ✔ insu	Class: class (Nom) Visualize All
6 ✔ mass 7 ✔ pedi 8 ✔ age 9 ✔ class	103 76 66
Remove	
Status	b 8.5 17
ок	Log 🛷 ×0

# Step 5: Go to Cluster tab and Choose SimpleKMeans algorithm

😮 Weka Explorer					_		×
Preprocess Classify Cluster Associate Select	t attributes Visualize	Forecast					
Clusterer		· · ·					
Choose SimpleKMeans - init 0 -max-candidates 1	100 -periodic-pruning 1	0000 -min-density 2.0 -t1 -1.25	i -t2 -1.0 -N 2 -A "weka	a.core.EuclideanDistar	nce -R first-las	st" -1 500	-nun
Cluster mode	Clusterer output						
Use training set     Supplied test set     Set     Percentage split     Classes to clusters evaluation     (Nom) class     Store clusters for visualization	plas pres skin insu mass pedi age class	120.8945 69.1055 20.5365 79.7995 31.9926 0.4719 33.2409 tested_negative te	109.98 68.184 19.664 68.792 30.3042 0.4297 31.19 sted_negative tes	141.2575 70.8246 22.1642 100.3358 35.1425 0.5505 37.0672 sted_positive			
Ignore attributes           Start         Stop           Result list (right-click for options)         10:29:25 - Simplek/Means	Time taken to	build model (full train evaluation on training	ing data) : 0.03 set ===	seconds			
19:29:39 - SimpleKMeans 19:30:16 - SimpleKMeans 19:43:10 - SimpleKMeans	0 500 ( 1 268 (	tances 65%) 35%)	J			,	
Status							
ок					Log	~	r. x 0

# Step 6: Right Click on created project and select Visualize cluster assignments

Weka Explore	y .										-		×
Preprocess	Classify 0	luster	Associate	Select	athibutes	Visualize	Forecast						
lusterer													
Choose	imple904ea	es -init	D-max-cand	idates 11	0 -periodi	>pruning 10	)000 - min- d	ensity 2.0 -t1 -1.25	12 -1.0 -N 2 -A "weks	a.core.EuclideanDista	ince -R first-las	r - 1 500	-nun
uster mode					Clustere	r output							
<ul> <li>Use trainin</li> <li>Supplied to</li> </ul>	ng set est set		Set		plas pres skin			120,8945 69,1055 20,5365	105,58 68,184 19,664	141.2575 T0.8246 22.1642			ŕ
Percentage     Classes to	e split o clusters ev	aluation	%	66	insu mass pedi			79.7995 31.9926 0.4719	68.792 30.3042 0.4297	100.3358 35.1425 0.5505			
(Nom) cla	iss Iers for visua	lization	Y		age class		test	33.2409 ed_negative tes	31.19 ted_negative ter	37.0672 sted_positive			
	Ignore at	tributes											
Start			Stop		Time	taken to	build mod	el (full traini	ng data) : 0.03	seconds			
esult list (right-	click for opt	tions)			>	odel and	evaluatio	n on training a	et				
19:29:35 - Sim 19:30:16 - Sim 19:43:10 - Sim	npleKMeans npleKMeans				Clust	ered Inst	ances 54)						
	v v s	iew in n Iew in s ave res	tain window eparate wind uit buffer	dow		268 ( 3	54)		,			,	-
		elete le	SUR DURIER(S)	2									
ок ОК	SR	ave mo e-evalu	del ate model o	n current	test set						Log	-	₽ ×0
19.45.10	- carrigana R V	e-appi) Isualize	this model's duster assi	configu gnments	ation	265	( 358)						

### Practical 3:

Aim: Prepare the Analysis services for Adventure Works Cycles or (any other database) . Build the data mining model structure and Implement Naïve Bayes Algorithm.

### Theory:

Bayesian classification is based on Bayes' theorem. Simple Bayesian classifier known as the naive Bayesian classifier. Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve."

### What is Naive Bayes algorithm?

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



 $P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$ 

Above,

- P(c/x) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- P(c) is the prior probability of *class*.

- P(x/c) is the likelihood which is the probability of *predictor* given *class*.
- P(x) is the prior probability of *predictor*.

### How Naive Bayes algorithm works?

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
nny	No
vercast	Yes
lainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
lainy	No
Rainy	No
Sunny	Yes
ainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) \* P(Yes) / P(Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 \* 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes. What are the Pros and Cons of Naive Bayes?

### Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

### Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

### 4 Applications of Naive Bayes Algorithms

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and <u>Collaborative</u> <u>Filtering</u> together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

# Step 1: Open Weka

🜍 Weka GUI Chooser	– 🗆 X
Program Visualization Tools Help	
WEKA The University of Waikato	Applications Explorer Experimenter
	KnowledgeFlow
Waikato Environment for Knowledge Analysis Version 3.8.2 (c) 1999 - 2017 The University of Waikato Hamilton, New Zealand	Simple CLI

# **Step 2: Click on Explorer**

Weka Explorer				– 🗆 ×
Preprocess Classify Cluster Assoc	iate Select attributes Visualize Forecast			
Open file Open URL	. Open DB Gener	ate Un	do Edit	Save
Choose None				Apply Stop
Current relation		Selected attribute		
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
All None All Ref Status	Invert Pattern			Visualize All
Welcome to the Weka Explorer				Log 🛷 x0

# Step 3: Click on Open File Option

♥ Weka Explorer           Preprocess         Classify         Cluster	er   Associate   Select attributes   Visualize   Forecast	- 🗆 X
Open file C	Open URL Open DB Generate Undo	Edit Save
Choose None	😮 Open	× Apply Stop
Current relation Relation: None Instances: None Attributes All No	Look In: Documents  Cook In: Documents  IISExpress  My Web Sites  Virtual Machines  Visual Studio 2015  File Name:  Files of Type: Arff data files (*.arff)  O	Type: None Unique: None Unique: None Unique: None Unique: None Visualize All
Status	Remove	
Welcome to the Weka Explorer		Log 🛷 X0

# Step 4: Go to C://Program Files/Weka-3-8/data. Select soyabean.arff database

Preprocess Classify Cluster Associate Select attributes Visualize Forecast	
Open file Open URL Open DB Generate Undo Edit	Save
Filter	
Choose None App	y Stop
Current relation Selected attribute	
Relation: soybean         Attributes: 36         Name: date         Type: Nomin           Instances: 683         Sum of weights: 683         Missing: 1 (0%)         Distinct: 7         Unique: 0 (0%)	al
Attributes No. Label Count Weight	
All         None         Invert         Pattern           No.         Name         93         93.0           1 √ date         131.0         131.0	
2 Ø plant-stand 3 Ø precip 4 Ø temp	Visualize All
5	90
Status	×0

Step 5: Go to Classify tab and Select Naïve Bayes algorithm under Bayes tab



Step 6: Right Click on created structure and select Visualize Classifier Errors

🜍 Weka Explorer												-		×
Preprocess Classi	fy Cluster Asso	ociate Select attr	ibutes	Visualize	Forec	ast								
Classifier														
Choose NaiveB	ayes													
Test options		Classifier output												
<ul> <li>Use training set</li> <li>Supplied test set</li> </ul>	Set	a b c 20 0 0 0 20 0	aer 0000	g h 0 0 0 0	1 ] 0 0 0 0	к 1 т 0 0 0 0 0 0	, n 0 0 0	q o 0 0	q r 0 0 0 0	s < 0   0	<pre>classified as a = diaporthe-stem-canker b = charcoal-rot</pre>			
Cross-validation	Folds 10	0 0 20	0 0 0	0 0	0 0	0 0 0		0 0	0 0	0	<pre>c = rhizoctonia-root-rot d = phytophthora-rot c = brown-stem-rot</pre>			
More optio	% 00 ns		0 0 20 0 0 0 0 0 0 0	0 0 20 0 77		0000	0	00		01	<pre>f = powdery-mildew g = downy-mildew h = brown-spot</pre>			
(Nom) class	•	0 0 0	0 0 0 0 0 0 0 0 0	0 0 2 0 0	200 218 002	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	, 0   0   0	0 0 0 0 0 0	0 0 0 0 0	0   0   0	<pre>i = bacterial-blight j = bacterial-pustule k = purple-seed-stain</pre>			
Start Result list (right-click f	Stop	0 0 0	000 000 000	002	0 0 0 0 0 0	0440 0017 000	0 1 91	00 00	000	0   0   0	<pre>l = anthracnose m = phyllosticta-leaf-spot n = alternarialeaf-spot</pre>			
19:49:43 - bayes.Na	veBayes	0 0 0		0 3	0 0	0 0 0	22 6	5 1 0 15	0 0	0 1	<pre>o = frog-eye-leaf-spot p = diaporthe-pod-&amp;-stem-bl </pre>	ight		
19:51:09 - bayes.Na	19:50:47 - bayes Nai 19:51:09 - bayes Nai View in separate window Save result buffer Delete result buffer(s)					0000	0	00.	0 14	0   8	<pre>q = cyst-nematode r = 2-4-d-injury s = herbicide-injury</pre>			
Status	Load model Save model Re-evaluate mo	del on current test	set											
ок	OK Re-apply this model's configuration											Log	-	р. х0
10	Size	: 23.3KB												
Type here to searc	Cost/Benefit ana Visualize cost ci	alysis urve				<b>•</b>	<del>,</del>	M	<		M 👩 🧭 😪			

**Step 7: Explore Visualize Threshold Curve** 

<b>U</b>   <del>-</del>	DM Manual - Microsoft Word (Product Activation Failed)
Weka Explorer	×
Preprocess Classify Cluster	Associate Select attributes Visualize Forecast
Classifier	
Choose NaiveBayes	
Test options	Classifier output
Use training set Supplied test set Cross-validation Folds  Percentage split More options  (Norm) class  Start Stop Result list (right-click for options)  19:49:43 - bayes NaiveBayes 19:50:47 - bayes NaiveBayes 19:51:09 - bayes NaiveBayes Status OK	a       b       c       a       b       c       a       b       c       a       b       c       a       b       c       a       b       c
	Visualize tree Ost-rientatuue Visualize margin curve 2-4-d-injury
7 Words: 462 🥸 English (U.S.)	Visualize threshold curve  herbicide-injury
Type here to search	Cost/Benefit analysis Visualize cost curve

Step 8: Explore Cost Benefit Analysis

Ŭ   <del>-</del>											DMI	Manu	ual -	Micr	osof	t W	ord (	Pro	duct	Act	tivatio	on Faile	ed)						
Weka Explorer							-																					-	×
Preprocess Cla	ssify Cli	uster	Associa	ate	Selec	t attri	butes	s 🚺	/isual	ize	Fored	cast																	
Classifier																													
Choose Naiv	veBayes																												
Test options			Cla	assifi	ier ou	tput																							
Use training s Supplied test Cross-validati Percentage sp More of (Nom) class Start Result list (right-clic 19:49:43 - bayes, 19:50:47 - bayes, 19:51:09 - bayes	set set set sion Folds plit % ptions sck for option Naive Viii Naive Viii Naive Sa	Set s 10 6 66 etop ons) es ew in ma ew in se ave result	ain wind parate v It buffer	a 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	b 0 20 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	c     d       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0       0     0	e 0 0 44 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		1     1     0	J 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	k         1           0         0	1 m 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	n 0 0 0 0 6 0 0 er	0 0 0 0 0 4 0	P 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	q 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	r 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<b>s</b> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		c a = b = c = f	lassi diapo charc rhizo phyto brown powde downy brown bacte purpl anthr phyll alter frog- diapo cyst- 2-4-d herbi	ried oal- ctor phth -ste ry-m -mil -spo rial rial e-se acno osti nari eye- rthe nema -inj cide	i as e-ste -rot nia-1 nora- em-re em-re ldew ot L-b1: L-pu: e-poo atode jury e-in;	em-c root -rot ew ight stai -lea af-s f-sp d-s- e	anke -rot :	er : Not	lght	
Status	Delete result buffer(s) Load model Save model								p p a	oacteri ourple anthra ohvilos	ial-pu -seeo cnos sticta-	istule d-sta e -leaf-	e iin -spot							_	_		_	_		_	_		 <b>V</b>
OK Re-apply this model's configuration Visualize classifier errors Visualize tree Visualize margin curve							fi c	alternarialeaf-spot frog-eye-leaf-spot diaporthe-pod-&-stem-blight cyst-nematode																		Log	ул. x0		
8   Words: 468   🧭 ) Type here to sea	En Vis Co arch Vis	sualize ti ost/Bene sualize c	nreshol fit analy cost cun	id cur ysis ve	ve			) I	h	erbic	ide-in	ijury		<b>-</b>		×				V	V.	9		Ż		-			

Step 9: Explore Cost Curve



### Practical 4:

**Aim:** Prepare the Analysis services for Adventure Works Cycles or (any other database).Build the basic Time series model structure and create the predictions

**Theory:** The Time Series mining function enables forecasting of time series values. Similar to common regression methods, time series algorithms predict a numerical value. In contrast to common regression methods, time series predictions are focused on future values of an ordered series. These predictions are commonly called forecasts.

The time series algorithms are univariate algorithms. This means that the independent variable is a time column or an order column. The forecasts are based on past values. They are not based on other independent columns.

Time series algorithms are different from common regression algorithms because they do not only predict future values but also incorporate seasonal cycles into the forecast.

### Time series algorithms

The Time Series mining function provides algorithms that are based on different underlying model assumptions with several parameters. The learning algorithms try to find the best model and the best parameter values for the given data.

If you do not specify a seasonal cycle, it is automatically determined. Also, missing values and non-equidistant time series are automatically interpolated.

The Time Series mining function provides the following algorithms to predict future trends:

- Autoregressive Integrated Moving Average (ARIMA)
- Exponential Smoothing
- Seasonal Trend Decomposition

Which of the algorithms creates the best forecast of your data depends on different model assumptions. You can calculate all forecasts at the same time. The algorithms calculate a detailed forecast including seasonal behaviour of the original time series. With the Time Series Visualizer, you can evaluate and compare the resulting curves.

### Autoregressive Integrated Moving Average (ARIMA)

The ARIMA algorithm also incorporates seasonal components. Therefore this algorithm is also referred to as Seasonal ARIMA (SARIMA).

The autoregressive part of the algorithm uses weighted previous values while the moving average part weighs the previously assumed errors of the time series.

The ARIMA algorithm assumes the error to be independent and identically distributed from a normal distribution with zero mean. The basic ARMA model works for stationary time series only. Stationary time series contain equal mean and equal variance for the whole time series. Therefore the integrated part creates stationary series by differentiation.

### **Exponential Smoothing**

Exponential Smoothing can consist of the following components:

- Basic level at a certain point in time.
- Trend.

The trend can have additive or multiplicative characteristics. Also, it can be damped or non-damped.

• The seasonal component.

Dependant on the data, trend and the seasonal component are optional. There are ARIMA models that correspond to Exponential Smoothing models and vice versa.

### **Seasonal Trend Decomposition**

Seasonal Trend Decomposition fits different seasonal trend functions to the given data and selects the best seasonal trend function according to an error measure. The following trends are used during the training run:

- Linear trend
- Quadratic trend
- Cubic trend
- Logarithmic trend
- Exponential trend
- Hyperbolic trend

The seasonality is incorporated in an additive or multiplicative way.

### Practical

Step 1: Open Weka



### **Step 2: Click on Explorer**

O Webs Fundament				
Weka Explorer				- U ×
Preprocess Classify Cluster Associate S	elect attributes Visualize			
Open file Open URL	Open DB Gene	erate Un	do Edit	Save
Filter				
Choose None				Apply Stop
Current relation		Selected attribute		
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
Attributes				
All None	Invert Pattern			Visualize All

Step 3: For this practical we need to install Forecast, as you can see last tab in Visualize tab and we need to install one more tab Forecast.

Step 4: Open Weka-> Click on Tools-> Select Package Manager

🥝 Weka GUI Chooser			- 🗆 ×	X
Program Visualization	<u>T</u> ools <u>H</u> elp			Canaarda Hinda Edi Ona
	Package manager ArffViewer SqlViewer Bayes pat editor	Ctrl+U Ctrl+A Ctrl+S Ctrl+N	Applications	Selected attribute
	The University of Waikato	Currin	Experimenter	Ars: None Name: None Weight None Type: None ghts: None Missing: None Distinct: None Unique: None
<i></i>	÷		KnowledgeFlow	
Waikato Environment for Kn Version 3.8.2	rowledge Analysis		Workbench	Visualize All
(c) 1999 - 2017 The University of Waikato Hamilton, New Zealand			Simple CLI	

Step 5: Following is the Package Manager through which we need to install

Official Refresh repository cache O Installed  Available O Al	Install Uninstall To	Install/Unins	tall/Refresh progress		Unofficial File/URL
Package	Category	Installed version	Repository version	Loaded	
AffectiveTweets	Text classification		1.0.1		
AnDE	Classification		1.2.1		
AnalogicalModeling	Classification		0.04		
ArabicStemmers_LightStemmers	Preprocessing		1.0.0		
Auto-WEKA	Classification, Regression, Attribute		2.6.1		
BANGFile	Clustering		1.0.0		
		_			
🗢 🏠 Package search	Clear				

# Step 6: Scroll Down and Select timeSeriesForecasting

Official		Install/Unins	tall/Refresh progress	Unofficial
Refresh reposito	iry cache Install Uninstal	Toggle load		File/URL
Package	Category	Installed version	Repository version	Loaded
gerjymon	Scripting Filtera Tima Sariaa		1.0.3	i i i i i i i i i i i i i i i i i i i
meseriesForecasting	Time series		1.0.1	
serClassifier	Classification/regression		1.0.3	
fdr	Classification, Online learnin	Ig	1.0.0	
ines	Density Estimation		1.0.0	
ntinnEestureIntervale	Classification		101	
누 🔥 Package sea	rch Clea	r		
timeseriesForeca	asting: Time series forecasting e http://wiki.pentaho.com/di	environment. splay/DATAMINING/Time	+Series+Analysis+and+Fore	ecasting+with+Weka
Author:	Mark Hall <mhall{[at]}pen< td=""><td>taho.com&gt;</td><td></td><td></td></mhall{[at]}pen<>	taho.com>		
Maintainer:	Mark Hall <mhall{[at]}pen< td=""><td>taho.com&gt;</td><td></td><td></td></mhall{[at]}pen<>	taho.com>		
Provides a time process of creat evaluation routin IFreeChart libra	series forecasting environmen ing lagged variables and date- nes are provided by a special e rv. Includes both command-lir	t for Weka. Includes a wra derived periodic variables evaluation module and gra ne and GUI user interfaces	pper for Weka regression s and provides the ability to o phing of predictions/forecas . Sample time series data ca	chemes that automates the do closed-loop forecasting. New sts are provided via the an be found in

# Step 7: Click on Install

Package Manager				- 0	$\times$
Official			Install/Uninstall/Refresh progress	Unofficial	
Refresh repository c	ache Install Ur le All 🔲 Ignore dependenci	ninstall Toggle load	[DefaultPackageManager] Installing: doc/weka/knov	Medgeflow/steps/package-f	٦L
Package	Category	Installed vers	ion Repository version	Loaded	
agerjymon	scripting		1.0.3		
timeSeriesFilters	Filters, Time Series		1.0.1		- 1
timeseriesForecasting	Time series		1.0.25		
userClassifier	Classification/regress	ion	1.0.3		
vfdr	Classification, Online	learning	1.0.0		
vines	Density Estimation		1.0.0		i
A T	Classification				
누 🐴 Package search		Clear			
URL: Author: Maintainer:	http://wiki.pentaho.co Mark Hall <mhall{[at]] Mark Hall <mhall{[at]]< td=""><th>m/display/DATAMINI }pentaho.com&gt; }pentaho.com&gt;</th><th>NG/Time+Series+Analysis+and+Fore</th><td>:casting+with+Weka</td><td></td></mhall{[at]]<></mhall{[at]] 	m/display/DATAMINI }pentaho.com> }pentaho.com>	NG/Time+Series+Analysis+and+Fore	:casting+with+Weka	
Provides a time se process of creating evaluation routines JFreeChart library. \${WEKA_HOME}/	ries forecasting environ I lagged variables and o are provided by a spe Includes both commar backages/timeseriesFor	ment for Weka. Inclu date-derived periodic cial evaluation modul id-line and GUI user i ecasting/sample-data	ides a wrapper for Weka regression so variables and provides the ability to o e and graphing of predictions/forecas interfaces. Sample time series data ca	chemes that automates the lo closed-loop forecasting. New its are provided via the in be found in	V

### Step 8: Now again go to Explorer and you can find new tab is been added

Weka Explorer					– 🗆 X
Preprocess Classify	Cluster Associate Selec	t attributes Visualize Forecast			
Open file	Open URL	Open DB Gene	erate Un	do Edit	Save
Filter					
Choose None					Apply Stop
Current relation			Selected attribute		
Relation: None Instances: None		Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
Attributes					
All	None	Pattern			Visualize All

### Step 9: Create the following database. Open Notepad and type the following

@relation AnkaraPopulation

@attribute year numeric

@attribute population numeric

@data

2007, 70586256

2008,71517100

2009,72561312

2010,73722988

Save the file with .arff extension

### **Step 10: Open the created database**

Weka Explorer	r Associate Select attributes Visualize Forecast		×
Open file O	pen URL Open DB Generate	Undo Edit	Save
Filter			
Choose None			Apply Stop
Current relation	Se	lected attribute	
Relation: AnkaraPopulation Instances: 4	Open	×	Type: Numeric Unique: 4 (100%)
Attributes	Look In: USB Drive (K:)		
All     No       No.     Name       1 √ year       2 √ population	AVISHKAR 2018 Core JAva MSc.IT TimeSeries.arff Turkeypopulation.arff	<ul> <li>Invoke options dialog</li> <li>Note:</li> <li>Some file formats offer additional options which can be customized when invoking the options dialog.</li> </ul>	
	File Name: turkeypopulation.arff Files of Type: Arff data files (*.arff)		Visualize All
		Open Cancel	

# **Step 11:** Go to Forecast tab and under basic configuration do the following changes:

**Step 12: Go to Advanced Configuration tab and select LinearRegression Algorithm under functions option.** 



### **Step 13: Click on start button**

**			_		
Start Stop Help	Output/Visualization				 
Result list	Output Train future	pred.			
20:45:14 - LinearRegression [-F	population:	· · · · ·			
20:45:27 - LinearRegression [-F	Linear Regression	Model			
	population =				
	+				
	72096914				
	=== Future predict	ions from end	d of training	data ===	
	inst#	year	population		
	70586256	2007	70586256		
	71631833.3333	2008	71517100		
	72677410.6667	2009	72561312		
	73722988	2010	73722988		
	74768565.3333*	2008.5	72096914		

### Step 14: Go to Train future pred option.



### Practical 5:

**Aim:** Prepare the Analysis services for Adventure Works Cycles or (any other database) .Build the basic data mining model and show the implementation of Association algorithm. And also apply the DMX queries.

### Theory:

Rule support and confidence are two measures of rule interestingness. They respectively reflect the usefulness and certainty of discovered rules. Asupport of 2% for Association Rule means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together. A confidence of 60% means that 60% of the customers who purchased a computer also bought the software. Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold.

A transaction T is said to contain A if and only if  $A \subseteq T$ . An association rule is an implication of the form  $A \Rightarrow B$ , where  $A \subset I$ ,  $B \subset I$ , and  $A \cap B = \varphi$ . The rule  $A \Rightarrow B$  holds in the transaction set D with support s, where s is the percentage of transactions in D that contain AUB (i.e., the union of sets A and B, or say, both A and B).

This is taken to be the probability,  $P(A \cup B)$ .1 The rule  $A \Rightarrow B$  has confidence c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, P(B|A). That is,

 $support(A \Rightarrow B) = P(A \cup B)$  $confidence(A \Rightarrow B) = P(B|A).$ 

 $confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}$ 

In general, association rule mining can be viewed as a two-step process:

### 1. Find all frequent itemsets:

By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, min sup.

#### 2. Generate strong association rules from the frequent itemsets:

By definition, these rules must satisfy minimum support and minimum confidence.

# Step 1: Open Weka

🥥 Weka GUI Chooser	_		×
Program Visualization Tools Help			
WEKA The University of Waikato	Applica	ations Explorer xperiment	er
	Kno	owledgeFl	ow
Waikato Environment for Knowledge Analysis Version 3.8.2	V	Vorkbenct	1
(c) 1999 - 2017 The University of Waikato Hamilton, New Zealand		Simple CL	

# Step 2: Click on Explorer

Weka Explorer				– 🗆 ×
Preprocess Classify Cluster Asso	ciate Select attributes Visualize Forecast			
Open file Open URL	Open DB Gene	rate Un	do Edit	Save
Choose None				Apply Stop
Current relation		Selected attribute		
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
All None All Re Status	Invert Pattern			Visualize All
Welcome to the Weka Explorer				Log 🛷 x0

# Step 3: Click on Open File Option

Weka Explorer	luster Associate Select attributes Visualize Forecast	- 0 X
Open file	Open URL Open DB Generate Undo	Edit Save
Choose None	🕝 Open	X Apply Stop
Current relation Relation: None Instances: None Attributes All	Look In:       Documents       Image: Constraint of the second se	Type: None Unique: None
	File Name:	Visualize All
	Files of Type: Arff data files (*.arff)	Zancel
Status	Remove	
Welcome to the Weka Explo	orer	Log 🛷 x0

Step 4: Go to C://Program Files/Weka-3-8/data. Select weather.nomial.arff database. Select all the attributes.

💙 Weka Explorer		– 🗆 X
Preprocess Classify Cluster Associate Select attributes Visualize Forecast		
Open file Open URL Open DB Gene	rate Undo Edit	Save
Filter		
Choose None		Apply Stop
Current relation	Selected attribute	
Relation: weather.symbolic         Attributes: 5           Instances: 14         Sum of weights: 14	Name: outlook Missing: 0 (0%) Distinct: 3	Type: Nominal Unique: 0 (0%)
Attributes	No. Label Count	Weight
All     None     Invert     Pattern       No.     Name       1	1 sunny 5 2 overcast 4 3 rainy 5 Class: play (Nom)	5.0 4.0 5.0 Visualize All
Remove       Status       OK		Log 🛷 x0

Step 5: Go to Associate tab and Choose Apriori Algorithm



### Step 6: Right Click on created structure and select View in separate window



### Practical 6:

Aim: Using R-Tool, show the analysis for social networking sites.

### Theory:

Social network analysis is based on an assumption of the importance of relationships among interacting units.

The social network perspective encompasses theories, models, and applications that are expressed in terms of relational concepts or processes.

Along with growing interest and increased use of network analysis has come a consensus about the central principles underlying the network perspective. In addition to the use of relational concepts, we note the following as being important:

- Actors and their actions are viewed as interdependent rather than independent, autonomous units
- Relational ties (linkages) between actors are channels for transfer or "flow" of resources (either material or nonmaterial)
- Network models focusing on individuals view the network structural environment as providing opportunities for or constraints on individual action
- Network models conceptualize structure (social, economic, political, and so forth) as lasting patterns of relations among actors

The unit of analysis in network analysis is not the individual, but an entity consisting of a collection of individuals and the linkages among them.

Network methods focus on dyads (two actors and their ties), triads (three actors and their ties), or larger systems (subgroups of individuals, or entire networks.

Social network analysis [SNA] is the mapping and measuring of relationships and flows between people, groups, organizations, computers or other information/knowledge processing entities.

The nodes in the network are the people and groups while the links show relationships or flows between the nodes. SNA provides both a visual and a mathematical analysis of complex human systems.

### **Procedure**

The data to analyze is Twitter text dataand it can be downloaded as file "termDocMatrix.rdata" at<u>http://www.rdatamining.com/data</u>.

### 1) Load Data

> # load termDocMatrix

- > load("data/termDocMatrix.rdata")
- > # inspect part of the matrix
- > termDocMatrix[5:10,1:20]

### Docs

Terms	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
data 1	1002000001211101000
examples	000000000000000000000000000000000000000
introductio	on 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
mining	000000000000110000000
network	000000000000000000000000000000000000000
package	0001100000010000000000000

### 2) Transform Data into an Adjacency Matrix

- > # change it to a Boolean matrix
- > termDocMatrix[termDocMatrix>=1] <- 1
- > # transform into a term-term adjacency matrix
- > termMatrix <- termDocMatrix %\*% t(termDocMatrix)</pre>
- > # inspect terms numbered 5 to 10
- > termMatrix[5:10,5:10]

### Terms

Terms	d	ata ex	amples	intı	roduc	ctior	n min	ing ne	etwork package	
data	53		5	2	34		0	7		
examples	5	5	17		2	5	2	2		
introduct	ion	2	2		10	2	2	0		

mining	34	5	2	47	1	5
network	0	2	2	1	17	1
package	7	2	0	5	1	21

### 3)Build a Graph

Now we have built a term-term adjacency matrix, where the rows and columns represents terms, and every entry is the number of co-occurrences of two terms. Next we can build a graph with graph.adjacency() from package igraph.

- > library(igraph)
- > # build a graph from the above matrix

```
> g <- graph.adjacency(termMatrix, weighted=T, mode = "undirected")
```

- > # remove loops
- > g <- simplify(g)
- > # set labels and degrees of vertices
- > V(g) label <- V(g) name
- > V(g)\$degree <- degree(g)

### 4) Plot a Graph

- > # set seed to make the layout reproducible
- > set.seed(3952)
- > layout1 <- layout.fruchterman.reingold(g)
- > plot(g, layout=layout1)



A different layout can be generated with the first line of code below. The second line produces an interactive plot, which allows us to manually rearrange the layout. Details about other layout options can be obtained by running ?igraph::layout in R.

- > plot(g, layout=layout.kamada.kawai)
- > tkplot(g, layout=layout.kamada.kawai)

### Practical 7:

Aim: Consider the suitable data for text mining and Implement the Text Mining technique using R-Tool

### Theory:

Text mining, which is sometimes referred to "text analytics" is one way to make qualitative or "unstructured" data usable by a computer.

Qualitative data is descriptive data that cannot be measured in numbers and often includes qualities of appearance like color, texture, and textual description. Quantitative data is numerical, structured data that can be measured. However, there is often slippage between qualitative and quantitative categories. For example, a photograph might traditionally be considered "qualitative data" but when you break it down to the level of pixels, which can be measured.

Text mining is defined as the process or practice of examining large collections of written resources in order to generate new information, typically using specialized computer software. It is a subset of the larger field of data mining.

Some applications of text-mining include:

- Enterprise Business Intelligence/Data Mining, Competitive Intelligence
- E-Discovery, Records Management
- National Security/Intelligence
- Scientific discovery, especially Life Sciences
- Search/Information Access
- Social media monitoring

"Text mining involves the application of techniques from areas such as information retrieval, natural language processing, information extraction and data mining. These various stages of a text-mining process can be combined into a single workflow".

• **Information retrieval (IR) systems** match a user's query to documents in a collection or database. The first step in the text mining process is to find the body of documents that are relevant to the research question(s).

- Natural language processing (NLP) analyzes the text in structures based on human speech. It allows the computer to perform a grammatical analysis of a sentence to "read" the text.
- Information extraction (IE) involves structuring the data that the NLP system generates.
- **Data mining (DM)** is the process of identifying patterns in large sets of data, to find that new knowledge.

### Practical

### Step 1: Open Weka



**Step 2: Click on Explorer** 

Weka Explorer				– 🗆 X
Preprocess Classify Cluster Associa	ate Select attributes Visualize Forecas	t		
Open file Open URL	Open DB Gene	erate Und	do Edit	Save
Filter				
Choose None				Apply Stop
Current relation		Selected attribute		
Relation: None Instances: None	Attributes: None Sum of weights: None	Name: None Missing: None	Weight: None Distinct: None	Type: None Unique: None
Attributes				
All Name	Laura Dattara			
All None	Fattern			
				Visualize All
Rem	love			
Status				
Welcome to the Weka Explorer				Log 📣 X0

Step 3: Click on Open File Option and browse for created file

🥝 Weka Explorer				- 🗆 ×
Preprocess Classify Cluste	r Associate S	Select attributes Visualize Forecast		
Open file 0	pen URL	Open DB Generate	Undo Edit	Save
Filter				
Choose None				Apply Stop
Current relation		Selected attribu	te	
Relation: None Instances: None	🜍 Open		×	Type: None Unique: None
Attributes	Look <u>I</u> n:	USB Drive (K:)		
All		2018	Invoke options dialog	
	Core JAva		Note:	
	text.arff		Some file formats offer additional	
	timeSerie:	s.arff	options which can be customized	
	turkeypop	ulation.arff	when invoking the options dialog.	
	File Name:	text arff		
	Files of Taxas			Visualize All
	Flies of <u>Type</u> :	Ant data files (*.am)		
			Open Cancel	

Step 4: Click on Choose button and Select Filter->Unsupervised Learning-> Attributes->StringtoVector option

Weka Explorer	Y						
reprocess Classify Cluster Associate Select at	tributes Visualize Forec	ast					
Open file Open URL Open DB Generate Undo Edit Save							
er							
/ 🚞 weka	ate -1.0 -N 0 -stemme	r weka.core.stemmers.NullStemmer - sto	pwords-handler weka.core.s	to: Apply Stop			
🔻 🚞 filters		Colored attacked					
AllFilter		Selected attribute					
MultiFilter	Attributes: 2	Name: textoDocumento	Typ	e: String			
RenameRelation	Sum of weights: 6	Missing: 0 (0%) Di	istinct: 2 Uniqu	ie: 1 (17%)			
supervised							
Insupervised							
attribute	Battern						
Add	Fallen						
AddCluster							
AddExpression							
AddValues							
CartesianProduct							
Center		Class: docClass (Nom)		<ul> <li>Visualize</li> </ul>			
ChangeDateFormat							
ClassAssigner							
ClusterMembership							
Сору							
DateToNumeric							
Discretize		Attribute	is neither numeric nor nominal.				
FirstOrder							
FixedDictionaryStringToWordVector	-						
lnterquartileRange	V						

# Step 5: Click on the StringtoVector

🕝 Weka Explorer	– 🗆 X						
Preprocess Classify Cluster Associate Select attributes Visualize Foreca	st						
Open file Open URL Open DB Generate Undo Edit Save							
Filter Choose StringToWordVector -R first-last -W 1000 - prune-rate -1.0 - N 0 - stemmer weka.core.stemmers.NullStemmer - stopwords-handler weka.core.stop							
Current relation Left-click to edit properties for this object, right-click/Alt+Shift+left-cli	attribute						
Relation: Medicina         Attributes: 2           Instances: 6         Sum of weights: 6	Name:         textoDocumento         Type:         String           Missing:         0 (0%)         Distinct:         2         Unique:         1 (17%)						
Attributes							
All None Invert Pattern No. Name 1 text0Documento 2 docClass							
Remove	Attribute is neither numeric nor nominal.						
Status							

# Step 6: ObjectEditor appears

About	
Converts string attributes into representing word occurrence contained in the strings.	a set of numeric attributes More Information from the text Capabilities
IDFTransform	False
TFTransform	False
attributeIndices	first-last
attributeNamePrefix	
debug	False
dictionaryFileToSaveTo	set me
doNotCheckCapabilities	False
doNotOperateOnPerClassBasis	False
invertSelection	False
IowerCaseTokens	False
minTermFreq	1
normalizeDocLength	No normalization
outputWordCounts	False
periodicPruning	-1.0

Step 7: Make the following changes

🥥 weka.gui.GenericObjectEditor		×
weka.filters.unsupervised.attribute.	StringToWordVector	_
IDFTransform	True	ĥ
TFTransform	False	
attributeIndices	first-last	
attributeNamePrefix		
debug	False	
dictionaryFileToSaveTo	set me	
doNotCheckCapabilities	False	
doNotOperateOnPerClassBasis	True	
invertSelection	False	
IowerCaseTokens	True	
minTermFreq	2	
normalizeDocLength	No normalization	
outputWordCounts	False	
periodicPruning	2	
saveDictionaryInBinaryForm	False	j
stemmer	Choose NullStemmer	
stopwordsHandler	Choose Null	V
Open Save	e OK Cancel	

Step 8: Click on Apply

,				
Preprocess Classify Cluster Associate Sele	ect attributes Visualize Forecast			
Open file Open URL	Open DB Generate	e Undo	Edit	Save
Choose StringToWordVector -R first-last -W 10	00 -prune-rate 2.0 -I -N 0 -L -stemmer w	eka.core.stemmers.NullStemme	r-stopwords-handler weka.o	core. Apply Stop
irrent relation	S	elected attribute		Apply the cu
Relation: Medicina Instances: 6	Attributes: 2 Sum of weights: 6	Name: textoDocumento Missing: 0 (0%)	T Distinct: 2 Uni	ype: String que: 1 (17%)
All None In	vert Pattern			
No. Name       1     textoDocumento       2     docClass		lass: docClass (Nom) Atribu	te is neither numeric nor nominal.	Visualize

# **Step 9: Following will appear**

🥝 Weka Explorer	- 0	×					
Preprocess Classify Cluster Associate Select attributes Visualize Forecas	ist						
Open file     Open URL     Open DB     Generate     Undo     Edit     Save							
Choose StringToWordVector -R first-last -W 1000 - prune-rate 2.0 -I -N 0 -L - stemmer weka.core.stemmers.NullStemmer - stopwords-handler weka.core. Apply Stop							
Current relation	Selected attribute						
Relation: sms_test-weka.filters.unsupervised.attribute Attributes: 128 Instances: 200 Sum of weights: 200	Name:         spamclass         Type:         Nominal           Missing:         0 (0%)         Distinct:         2         Unique:         0 (0%)						
Attributes	No. Label Count Weight						
All None Invert Pattern	1 spam 33 33.0 2 ham 167 167.0						
No.         Name           1         spandass           2         #>           3         &It           4         -           5         2           6         87077           7         a           8         all           9         already           10         am           111         and           12         anything	Class: ' (Num) Visu	alize All					
13 are 14 as 15 at 16 awarded 17 back	33						

# **Step 10: Select one or two attributes**

🔇 Weka Explorer	- 🗆 X					
Preprocess Classify Cluster Associate Select attributes Visualize Fore	cast					
Open file Open URL Open DB Generate Undo Edit Save						
ilter						
Choose StringToWordVector - R first-last -W 1000 - prune-rate 2.0 -I - N 0 - L - stemmer weka.core.stemmers.NullStemmer - stopwords-handler weka.core Apply Stop						
Current relation	Selected attribute					
Relation: sms_test-weka.filters.unsupervised.attribute Attributes: 12 Instances: 200 Sum of weights: 20	Name: have         Type: Numeric           Missing: 0 (0%)         Distinct: 2         Unique: 0 (0%)					
Attributes	Statistic Value					
	Minimum 0					
All None Invert Pottern	Maximum 1.966					
	Mean 0.285					
No Nomo	SidDev 0.094					
NU. Name 34 Cut						
35 delivery						
36 📃 did						
37 dogging						
38 end						
40 V entry						
41 fa	Visualize All					
42 🗹 finish						
43 🗹 for	171					
44 🔲 free						
45 📃 got						
46 gram						
47 V great						
49 V Have						
51 her	T					
	29					
Remove						
	0.98 1.97					

### Practical 8:

**Aim:** Prepare the Analysis services for Adventure Works Cycles or (any other database). Build the data mining model and implement Apriori algorithm.

### Theory:

Apriori is a seminal algorithm proposed by R.Agrawal and R.Srikant in 1994 formining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties, as we shall see following. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore(k+1)-itemsets.

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented below, is used to reduce the search space. We will first describe this property, and then show an example illustrating its use.

# Apriori property: All nonempty subsets of a frequent itemset must also be frequent.

A two-step process is followed, consisting of join and prune actions.

**The join step:** To find Lk, a set of candidate k-itemsets is generated by joining Lk–1 with itself. This set of candidates is denoted Ck.

**The prune step** : Ck is a super set of Lk , that is , its members may or may not be frequent, but all of the frequentk-itemsets are included in Ck. A scan of the database to determine the count of each candidate in Ck would result in the determination of Lk

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

#### Input:

D, a database of transactions;

*min\_sup*, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

(1)	$L_1 = \text{find}_{\text{frequent}_1-\text{itemsets}}(D);$
(2)	for $(k = 2; L_{k-1} \neq \phi; k++)$ {
(3)	$C_k = \operatorname{apriori}_{gen}(L_{k-1});$
(4)	for each transaction $t \in D \{ // \text{ scan } D \text{ for counts} \}$
(5)	$C_t$ = subset( $C_k$ , $t$ ); // get the subsets of $t$ that are candidates
(6)	for each candidate $c \in C_t$
(7)	c.count++;
(8)	}
(9)	$L_k = \{c \in C_k   c.count \ge min\_sup\}$
(10)	}
(11)	return $L = \bigcup_k L_k$ ;

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)
        for each itemset l_1 \in L_{k-1}
(2)
            for each itemset l_2 \in L_{k-1}
                 if (l_1[1] = l_2[1]) \land (l_1[2] = l_2[2]) \land ... \land (l_1[k-2] = l_2[k-2]) \land (l_1[k-1] < l_2[k-1]) then {
(3)
                     c = l_1 \bowtie l_2; // join step: generate candidates
(4)
                     if has_infrequent_subset(c, L_{k-1}) then
(5)
                          delete c; // prune step: remove unfruitful candidate
(6)
(7)
                     else add c to C_k;
(8)
(9)
        return C_k;
procedure has_infrequent_subset(c: candidate k-itemset;
```

 $L_{k-1}$ : frequent (k-1)-itemsets); // use prior knowledge

- (1) for each (k-1)-subset *s* of *c*
- (2) if  $s \notin L_{k-1}$  then
- (3) return TRUE;
- (4) return FALSE;

### **Procedure:**

The R package "arules" is used for for implementing Apriori Algorithm. After the user installed the necessaries packages, he must load them. This can be done using the function "library(*package name*)".

### **Reading the Data:**

Transactions can be read from files in the basket format, with the command read.transactions.

```
>tr<-read.transactions("c:/rpracs/testfile.txt",format="basket",sep=",")
```

The object "tr" is used to store the transactions read from the file named "testfile", where each item is separated by a ",". "testfile" could be, for example:

A,B,C B,C A,B,D A,B,C,D A B

One way to visualize the data is inspect(object). For example: >inspect(**tr**)

items

1 {	Α,			
В				
С	}			
2 {I	3,			
С	}			
3 { /	Α,			
В				
D	}			
4 {	Α,			
В				
С				
D	}			
5 { /	<b>A</b> }			
6 {I	3}			

Additionally, you can visually inspect binary incidence matrices, or plot the frequency of items sets:

```
>image(tr)
>itemFrequencyPlot(tr, support = 0.1)
```



Figure:Output of >image(tr)



Figure:Output of>itemFrequencyPlot(**tr**, support = 0.1)

To show the number of items in transactions read from the file named "teste" do:

>length(**tr**) [1] 6

### **Rules:**

The function to mine frequent itemsets, association rules or association hyperedges, using the Apriori algorithm, takes 2 parameters:

Data: the object that contains the data

parameter: a multi-dimensional parameter to set up support and confidence

For example, using the dataset gathered in the previous section:

>rules <- apriori(**tr**, parameter= list(supp=0.5, conf=0.5))

The rules can be visualized with the command inspect:

>inspect(rules)

rhs support confidence lift 1 {} => {C} 0.5000000 0.5000000 1.0 2 {} => {A} 0.6666667 0.66666667 1.0 3 {} => {B} 0.8333333 0.8333333 1.0 4 {C} => {B} 0.5000000 1.0000000 1.2 5 {B} => {C} 0.5000000 0.6000000 1.2 6 {A} => {B} 0.5000000 0.7500000 0.9 7 {B} => {A} 0.5000000 0.6000000 0.9 To get a summary of the rules' characteristics, the function "summary" can be used:

>summary(rules)

set of 7 rules

rule length distribution (lhs + rhs):sizes 1 2 3 4

Min. 1st Qu. Median Mean 3rd Qu. Max. 1.000 1.000 2.000 1.571 2.000 2.000

summary of quality measures:

support confidence lift Min. :0.5000 Min. :0.5000 Min. :0.900 1st Qu.:0.5000 1st Qu.:0.6000 1st Qu.:0.950 Median :0.5000 Median :0.6667 Median :1.000 Mean :0.5714 Mean :0.7071 Mean :1.029 3rd Qu.:0.5833 3rd Qu.:0.7917 3rd Qu.:1.100 Max. :0.8333 Max. :1.0000 Max. :1.200

mining info: data ntransactions support confidence tr 6 0.5 0.5

### Outher quality measures of the rules can be displaed with:

>interestMeasure(rules, c("support", "chiSquare", "confidence", "conviction", "cosine", "coverage", "leverage", "lift", "oddsRatio"), **tr**)

	suppo	ort chiSq	juare c	onfidence	convictio	n cosine	coverage	leverage
1	0.500	0000	NaN	0.500000	0 1.00000	00 0.7071	068 1.0000	000000000 000
2	0.666	6667	NaN	0.666666	7 1.00000	00 0.8164	966 1.0000	000000000 000
3	0.833	3333	NaN	0.8333333	3 1.00000	00 0.9128	3709 1.0000	000000000 000
4	0.500	0000	1.2 1	.0000000	Inf 0.	7745967 (	).5000000 (	).08333333
5	0.500	0000	1.2 0	.6000000	1.250000	0 0.77459	67 0.833333	33 0.08333333
6	0.500	0000	0.6 0	.7500000	0.666666	7 0.67082	04 0.666666	67 -0.05555556
7	0.500	0000	0.6 0	.6000000	0.833333	3 0.67082	04 0.833333	33 -0.05555556
]	lift ode	dsRatio						
1	1.0	NaN						
2	1.0	NaN						
3	1.0	NaN						
4	1.2	Inf						
5	1.2	Inf						
6	0.9	0						

### 7 0.9 0

### To calculate a single measure and add it to the quality slot:

>quality(rules)<- cbind(quality(rules), hyperConfidence =interestMeasure(rules, method = "hyperConfidence", a)) >inspect(head(SORT(rules, by = "hyperConfidence")))

Finally, to send the output to a file use:

```
>sink("sink-examp.txt")
```

>inspect(**head**(SORT(rules, by = "hyperConfidence")))