SEMESTER: 7

LAB MANUAL

Data Mining and Business Intelligence
2170715

Name: _____

Enrollment No:_____

Batch:_____

GANDHINAGAR INSTITUE OF TECHNOLOGY

INFORMATION TECHNOLOGY & COMPUTER ENGINEERING DEPARTMENT

INDEX

SR. NO.	AIM	PAGE. NO.	DATE	REMARK	SIGN
1	Analytical Exercise-I.				
2	Study of Weka (Waikato Environment for Knowledge Analysis) Data mining Tool.				
3	Perform classification in weka.				
4	Perform clustering in weka.				
5	Learn to use Pivot Tables in Excel 2007 to Organize Data.				
6	Study of the Apriori Algorithm (Finding Frequent Item sets Using Candidate Generation).				
7	Survey of Different Data Mining Tools.				
8	Decision Tree Learning.				
9	Design & create cube by identifying measures & dimensions for star schema.				
10	Design & create cube by identifying measures & dimensions for snowflake schema.				

AIM: Analytical Exercise-I

OBJECTIVE: On completion of this exercise student will able to know about…

- Know about difference between Data Query and Data Mining.
- Understand the depth and importance of data mining.
- Categories the various techniques of data mining.

THEORY:

About Data mining

Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration.

Data mining refers to extracting or ―mining‖ knowledge from large amounts of data. The term is actually a misnomer. Remember that the mining of gold from rocks or sand is referred to as gold mining rather than rock or sand mining. Thus, data mining should have been more appropriately named ―knowledge mining from data,‖

Many other terms carry a similar or slightly different meaning to data mining, such as *knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.*

Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery from Data, or KDD.

Data mining as simply an essential step in the process of knowledge discovery

1. Data cleaning (to remove noise and inconsistent data)
2. Data integration (where multiple data sources may be combined)
3. Data selection (where data relevant to the analysis task are retrieved from the database)
4. Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
5. Data mining (an essential process where intelligent methods are applied in order to extract data patterns)
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures)
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

Database Queries vs. Data Mining

Virtually all modern, commercial database systems are based on the relational model formalized by Codd in the 60s and 70s (Codd, 1970) and the SQL language (Date,2000) which allows the user to efficiently and effectively manipulate a database. In this model a database table is a representation of a mathematical relation, that is, a set of items that share certain characteristics or attributes. Here, each table column represents an attribute of the relation and each record in the table represents a member of this relation.

In relational databases the tables are usually named after the kind of relation they represent. Figure 1 is an example of a table that represents the set or relation of all the customers of a particular store. In this case the store tracks the total amount of money spent by its customers.

ID	Name	Zip	Sex	Age	Income	Children	Car	Total Spent
5	Peter	05566	M	35	$40,000	2	Mini Van	$250.00
…	…	…		…	…	…	…	…
…	…	…		…	…	…	…	…
22	Maureen	04477	F	26	$55,000	0	Coupe	$50.00

Figure1 A relational database table representing customers of a store.

Relational databases do allow for the manipulation of the tables and the data within them. The most fundamental operation on a database is the ***query***. This operation enables the user to retrieve data from database tables by asserting that the retrieved data needs to fulfill certain criteria. As an example, consider the fact that the store owner might be interested in finding out which customers spent more than $100 at the store. The following query returns all the customers from the above customer table that spent more than $100:

SELECT * FROM CUSTOMER_TABLE WHERE TOTAL_SPENT > $100;

This query returns a list of all instances in the table where the value of the attribute Total Spent is larger than $100.

As this example highlights, queries act as filters that allow the user to select instances from a table based on certain attribute values. It does not matter how large or small the database table is, a query will simply return all the instances from a table that satisfy the attribute value constraints given in the query. This straightforward approach to retrieving data from a database has also a drawback. Assume for a moment that our example store is a large store with tens of thousands of customers(perhaps an online store). Firing the above query against the customer table in the database will most likely produce a result set containing a very large number of customers and not much can be learned from this query except for the fact that a large number of customers spent more than $100 at the store. Our innate analytical capabilities are quickly overwhelmed by large volumes of data.

This is where differences between querying a database and mining a database surface. In contrast to ***a query which simply returns the data that fulfills certain constraints, data mining***

constructs models of the data in question. The models can be viewed as high level summaries of the underlying data and are in most cases more useful than the raw data, since in a business sense they usually represent understandable and actionable items (Berry &Linoff, 2004). Depending on the questions of interest, data mining models can take on very different forms. They include decision trees and decision rules for classification tasks, association rules for market basket analysis, as well as clustering for market segmentation among many other possible models.

To continue our store example, in contrast to a query, a data mining algorithm that constructs decision rules might return the following set of rules for customers that spent more than $100 from the store database:

IF AGE > 35 AND CAR = MINIVAN THEN TOTAL SPENT > $100
OR
IF SEX = M AND ZIP = 05566 THEN TOTAL SPENT > $100

These rules are understandable because they summarize hundreds, possibly thousands, of records in the customer database and it would be difficult to glean this information off the query result. The rules are also actionable. Consider that the first rule tells the store owner that adults over the age of 35 that own a mini van are likely to spend more than $100. Having access to this information allows the store owner to adjust the inventory to cater to this segment of the population, assuming that this represents a desirable cross-section of the customer base. Similar with the second rule, male customers that reside in a certain ZIP code are likely to spend more than $100. Looking at census information for this particular ZIP code the store owner could again adjust the store inventory to also cater to this population segment presumably increasing the attractiveness of the store and thereby increasing sales.

As we have shown, the fundamental difference between database queries and data mining is the fact that in contrast to queries data mining does not return raw data that satisfies certain constraints, but returns models of the data in question. These models are attractive because in general they represent understandable and actionable items. Since no such modeling ever occurs in database queries we do not consider running queries against database tables as data mining, it does not matter how large the tables are.

Data querying is the process of asking questions of data in search of a specific answer. Unlike many forms of search (i.e. Google), queries are normally structured and require specific parameters or code, known as SQL (Structured Query Language). A query could be written to answer questions like, "How many items were sold in Region 2 last month?"

Data mining is the process of sorting through large amounts of data to identify patterns and relationships using statistical algorithms. These relationships may help us to understand which factors affected the outcome of something, or they may be used to predict future outcomes. Data mining might be used to answer questions like, "What factors affected sales in Region 2 last month?" Knowing which factors drive sales in the past could help to predict or make estimates about sales in the future.

Data Mining Task

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: descriptive and predictive.

1. **Descriptive** mining tasks characterize the general properties of the data in the database.
2. **Predictive** mining tasks perform inference on the current data in order to make predictions.

Common data mining tasks

1. Classification [Predictive]
2. Clustering [Descriptive]
3. Association Rule Discovery [Descriptive]
4. Sequential Pattern Discovery [Descriptive]
5. Regression [Predictive]

DataMining Application

1. Health Care
    a. Drug development – to help uncover less expensive but equally effective drug treatments.
    b. Medical diagnostics – imaging, real-time monitoring (e.g. Predicting women at high risk for emergency C-section).
    c. Insurance claims analysis – identify customers likely to buy new policies; define behavior patterns of risky customers

2. Business and Finance
    a. Banks - to detect which customers are using which products so they can offer the right mix of products and services to better meet
    b. Customer needs – cross sell and up sell.
    c. Credit card companies - to assist in mailing promotional materials to people who are most likely to respond.
    d. Lenders - to determine which applicants are most likely to default on a loan.

3. Sports and Gambling
    a. Sports teams – to analyze data to determine favorable player matchups and call the best plays
    b. Gaming industry - to analyze customer gambling trends at casinos.
    c. Sports Fanatics – to predict which teams will be chosen for tournament berths as well as to predict game winners.

4. Education
    a. Enrollment Management – which students are likely to attend
    b. Retention/Graduation Analysis – which students will remain enrolled after the first year and/or through graduation
    c. Donation Prediction – who is likely to donate and how much might they donate.

5. Other Application Areas
    a. Insurance – pricing, fraud detection, risk analysis
    b. Stock Market – market timing, stock selection, risk analysis
    c. Transportation – performance & network optimization to predict life-cycle costs of road pavement
    d. Telecommunications – churn reduction
    e. Retail – market basket analysis to help determine marketing strategies

EXCERSICE:

1) Define data mining?
2) For each of the following, identify if it is data query or data mining task(s).
    A. To find characteristics that differentiate people who had repeat knee construction from those who needed surgery only once
    B. A stock market analyst has been asked by his client to predict the future prices of 10 stocks 3 months in advance.
    C. When customers visit my website, what products together are they most likely to order?
    D. Develop a profile of credit card customers who are likely to carry an average monthly balance of more than $1,000.00
    E. Determine the number of customers who spent more than $100 last month in a local video store.
    F. In a locality, which group is greater in number – customers holding policies in one or several insurance companies?
    G. Do meaningful attribute relationships exist in a database containing information about credit card customers?
    H. Do single men play more golf than married men?
    I. Determine whether any credit card transaction was made by a customer last week.
3) Visit website www.kdnuggets.com and describe any five software of data mining.
4) Visit http://archive.ics.uci.edu/ml/datasets.html and describe any five datasets.

Type your text

EXPERIMENT NO: 2

TITLE: Study of Weka (Waikato Environment for Knowledge Analysis) Data mining Tool.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is weka?
- Why we use weka?
- Basic introduction about the weka tool.

THEORY:

Weka was developed at the University of Waikato in New Zealand, and the name stands for Waikato Environment for Knowledge Analysis. Outside the university the weka, pronounced to rhyme with Mecca, is a flightless bird with an inquisitive nature found only on the islands of New Zealand. The system is written in Java and distributed under the terms of the GNU General Public License. It runs on almost any platform and has been tested under Linux, Windows, and Macintosh operating systems—and even on a personal digital assistant. It provides a uniform interface to many different learning algorithms, along with methods for pre- and postprocessing and for evaluating the result of learning schemes on any given dataset.

What is in Weka?

Weka provides implementations of learning algorithms that you can easily apply to your dataset.

You can preprocess a dataset, feed it into a learning scheme, and analyze the resulting classifier and its performance—all without writing any program code at all. The workbench includes methods for all the standard data mining problems: regression, classification, clustering, association rule mining, and attribute selection. Getting to know the data is an integral part of the work, and many data visualization facilities and data preprocessing tools are provided. All algorithms take their input in the form of a single relational table in the ARFF format.

One way of using Weka is to apply a learning method to a dataset and analyze its output to learn more about the data. Another is to use learned models to generate predictions on new instances. A third is to apply several different learners and compare their performance in order to choose one for prediction. The learning methods are called classifiers, and in the interactive Weka interface you select the one you want from a menu. Many classifiers have tunable parameters, which you access through a property sheet or object editor. A common evaluation module is used to measure the performance of all classifiers.

Implementations of actual learning schemes are the most valuable resource that Weka provides. But tools for preprocessing the data, called filters, come a close second. Like classifiers, you select filters from a menu and tailor them to your requirements. We will show how different filters can be used, list the filtering algorithms, and describe their parameters. Weka also includes

implementations of algorithms for learning association rules, clustering data for which no class value is specified, and selecting relevant attributes in the data, which we describe briefly.

Installation of Weka

The program information can be found by conducting a search on the Web for WEKA Data Mining or going directly to the site at www.cs.waikato.ac.nz/~ml/WEKA .

Download weka3-7-5. From
Exe file: http://prdownloads.sourceforge.net/weka/weka-3-7-5jre.exe
Also open source file available of this exe from Source file:http://en.sourceforge.jp/projects/sfnet_weka/downloads/weka-3-7/3.7.5/weka-3-7-5.zip/

Figure 1 Weka GUI Chooser

The easiest way to use Weka is through a graphical user interface called the *Explorer*. This gives access to all of its facilities using menu selection and form filling. For example, you can quickly read in a dataset from an ARFF file (or spreadsheet) and build a decision tree from it. But learning decision trees is just the beginning: there are many other algorithms to explore. The Explorer interface helps you do just that. It guides you by presenting choices as menus, by forcing you to work in an appropriate order by graying out options until they are applicable, and by presenting options as forms to be filled out. Helpful *tooltips* pop up as the mouse passes over items on the screen to explain what they do. Sensible default values ensure that you can obtain results with a minimum of effort—but you will have to think about what you are doing to understand what the results mean.

There are two other graphical user interfaces to Weka. The *Knowledge Flow* interface allows you to design configurations for streamed data processing. A fundamental disadvantage of the Explorer is that it holds everything in main memory—when you open a dataset, it immediately loads it all in. This means that it can only be applied to small to medium-sized problems. However, Weka contains some incremental algorithms that can be used to process very large datasets. The Knowledge Flow interface lets you drag boxes representing learning algorithms and data sources around the screen and join them together into the configuration you want. It enables you to specify a data stream by connecting components representing data sources, preprocessing tools, learning algorithms, evaluation methods, and

visualization modules. If the filters and learning algorithms are capable of incremental learning, data will be loaded and processed incrementally.

Weka's third interface, the *Experimenter*, is designed to help you answer a basic practical question when applying classification and regression techniques, which methods and parameter values work best for the given problem? There is usually no way to answer this question a priori, and one reason we developed the workbench was to provide an environment that enables Weka users to compare a variety of learning techniques. This can be done interactively using the Explorer. However, the Experimenter allows you to automate the process by making it easy to run classifiers and filters with different parameter settings on a corpus of datasets, collect performance statistics, and perform significance tests. Advanced users can employ the Experimenter to distribute the computing load across multiple machines using Java remote method invocation (RMI).In this way you can set up large-scale statistical experiments and leave them to run.

Weka explorer

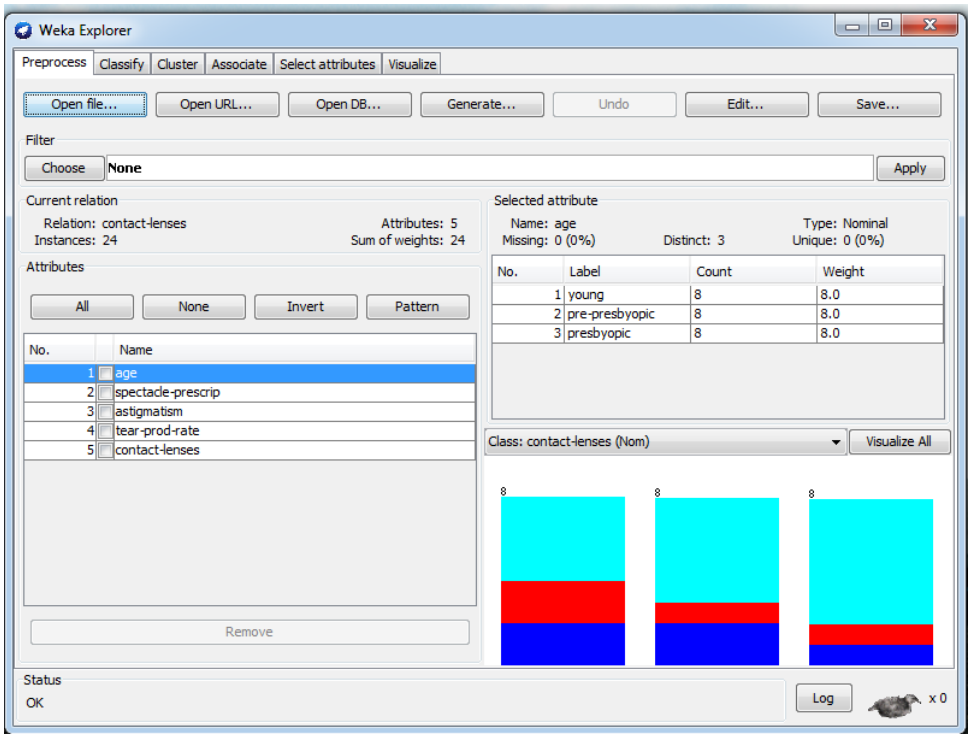


Figure 2 Weka explorer

At the very top of the window, just below the title bar, is a row of tabs. When the Explorer is first started only the first tab is active; the others are greyed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.

The tabs are as follows:

1. **Preprocess**- Choose and modify the data being acted on.
2. **Classify**- Train and test learning schemes that classify or perform regression.
3. **Cluster** -Learn clusters for the data.

4. **Associate-** Learn association rules for the data.
5. **Select attributes-** Select the most relevant attributes in the data.
6. **Visualize-** View an interactive 2D plot of the data.

Preparing the data

The data is often presented in a spreadsheet or database. However, Weka's native data storage method is ARFF format. You can easily convert from a spreadsheet to ARFF. The bulk of an ARFF file consists of a list of the instances, and the attribute values for each instance are separated by commas. Most spreadsheet and database programs allow you to export data into a file in comma-separated value (CSV) format as a list of records with commas between items.

EXCERSICE:

1) What is weka?
2) What kind of data format supported by WEKA? Explain with example.
3) Which are the modules of WEKA? Explain each of them in details.
4) Explain *preprocessing* tab in weka explorer with snap shot.
5) What is filter? Explain it in detail.

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 3

TITLE: Perform classification in weka.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is classification?
- What is the use of classification?
- How to perform classification in weka tool?

THEORY:

Classification (also known as classification trees or decision trees) is a data mining algorithm that creates a step-by-step guide for how to determine the output of a new data instance. The tree it creates is exactly that: a tree whereby each node in the tree represents a spot where a decision must be made based on the input, and you move to the next node and the next until you reach a leaf that tells you the predicted output. Sounds confusing, but it's really quite straightforward.

What Is Classification?

A bank loans officer needs analysis of her data in order to learn which loan applicants are―safe‖ and which are ―risky‖ for the bank. A marketing manager at *All Electronics* needs data
analysis to help guess whether a customer with a given profile will buy a new computer.

A medical researcher wants to analyze breast cancer data in order to predict which one of three specific treatments a patient should receive. In each of these examples, the data analysis task is classification, where a model or classifier is constructed to predict *categoric all labels*, such as
―safe‖ or ―risky‖ for the loan application data; ―yes‖ or ―no‖ for the marketing data; or ―treatment A,‖ ―treatment B,‖ or ―treatment C‖ for the medical data. These categories can be represented by discrete values, where the ordering among values has no meaning. For example, the values 1, 2, and 3 may be used to represent treatments A, B,and C, where there is no ordering implied among this group of treatment regimes.

Step for performing the classification in weka

1. Take one dataset.
2. Load this data into weka explorer.
3. Choose particular classification method.(show in figure 1, in that selected algorithm for classification is j48)
4. Select any one test option.
5. Click on start.

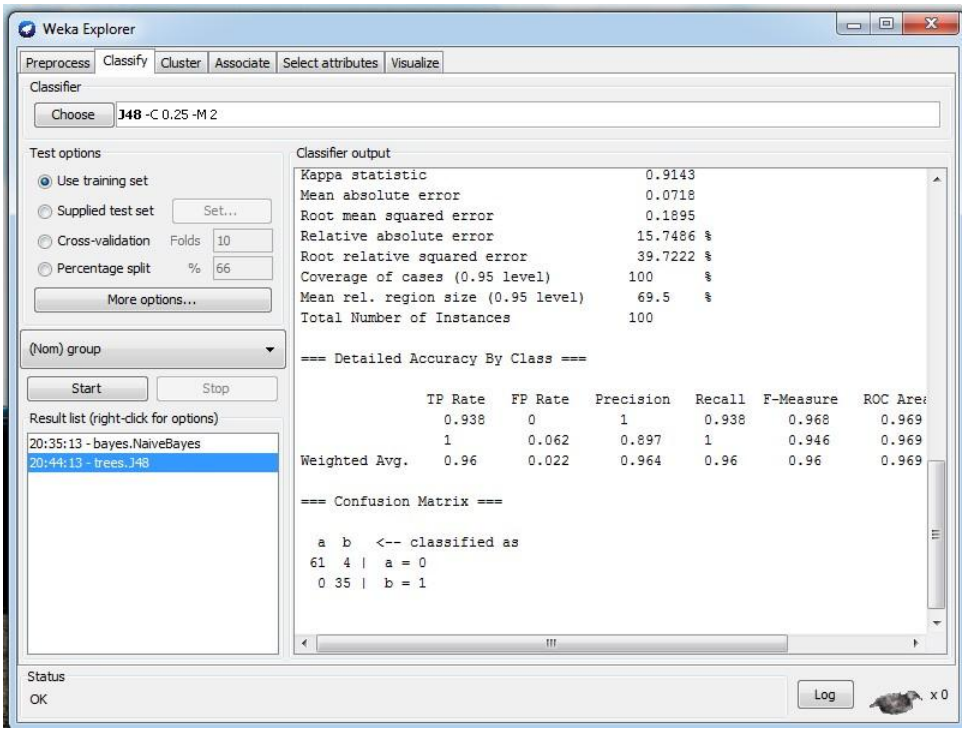


Figure 1 weka explorer (Classify tab)

At this point, we are ready to create our model in WEKA. Ensure that **Use training set** is selected so we use the data set we just loaded to create our model. Click **Start** and let WEKA run. The output from this model should look like the results in following

Output from WEKA's classification model

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weka.datagenerators.classifiers.classification.Agrawal-S_1_-n_100_-F_1_-P_0.05
Instances:    100
Attributes:   10
salary
commission
age
elevel
car
zipcode
hvalue
hyears
loan
group
Test mode:    evaluate on training data

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

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

age<= 37: 0 (31.0)
age> 37
|   age<= 62: 1 (39.0/4.0)
|   age> 62: 0 (30.0)

Number of Leaves  :   3

Size of the tree :        5


Time taken to build model: 0.03 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          96               96      %
Incorrectly Classified Instances         4                4      %
Kappa statistic                          0.9143
Mean absolute error                      0.0718
Root mean squared error                  0.1895
Relative absolute error                 15.7486 %
Root relative squared error             39.7222 %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      69.5    %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.938       0          1        0.938     0.968      0.969    0
                 1          0.062      0.897     1         0.946      0.969    1
Weighted Avg.    0.96       0.022      0.964     0.96      0.96       0.969

=== Confusion Matrix ===

a  b<-- classified as
61  4 |  a = 0
 0 35 |  b = 1
```

Classification tree visualize:

In the *result list* right click and select on the *visualize tree* option. So it generates the classification tree. See following.

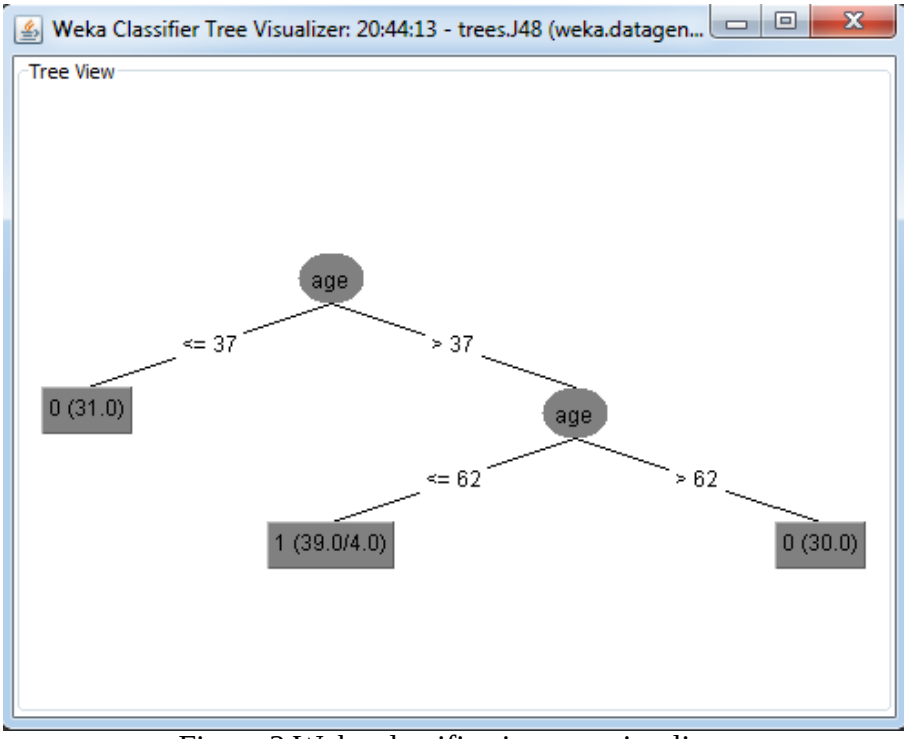


Figure 2.Weka classification tree visualize

EXCERSICE:

1) What are test options? Explain in detail.
2) Which are the different algorithms for classification available in Weka tool?
3) Take one data base from the http://archive.ics.uci.edu/ml/.And perform J48 classification algorithm on that data. Take screen shot.
4) Explain terms that is in classifier output (run information) in weka tool.
5) What is correctly classified instance and incorrectly classified instance?
6) What is confusion matrix?

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 4

TITLE: Perform clustering in weka.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is clustering?
- How to perform clustering in weka tool?

THEORY:

Clustering allows a user to make groups of data to determine patterns from the data. Clustering has its advantages when the data set is defined and a general pattern needs to be determined from the data. You can create a specific number of groups, depending on your business needs. One defining benefit of clustering over classification is that every attribute in the data set will be used to analyze the data. (If you remember from the classification method, only a subset of the attributes is used in the model.)

A major disadvantage of using clustering is that the user is required to know ahead of time how many groups he wants to create. For a user without any real knowledge of his data, this might be difficult. Should you create three groups? Five groups? Ten groups? It might take several steps of trial and error to determine the ideal number of groups to create.

However, for the average user, clustering can be the most useful data mining method you can use. It can quickly take your entire set of data and turn it in to groups, from which you can quickly make some conclusions. The math behind the method is somewhat complex and involved, which is why we take full advantage of the WEKA.

Overview of the math

This should be considered a quick and non-detailed overview of the math and algorithm used in the clustering method:

1. Every attribute in the data set should be normalized, whereby each value is divided by the difference between the high value and the low value in the data set for that attribute. For example, if the attribute is age, and the highest value is 72, and the lowest value is 16, then an age of 32 would be normalized to 0.5714.
2. Given the number of desired clusters, randomly select that number of samples from the data set to serve as our initial test cluster centers. For example, if you want to have three clusters, you would randomly select three rows of data from the data set.
3. Compute the distance from each data sample to the cluster center (our and only selected data row), using the least-squares method of distance calculation.
4. Assign each data row into a cluster, based on the minimum distance to each cluster center.
5. Compute the centroid, which is the average of each column of data using only the members of each cluster.

6. Calculate the distance from each data sample to the centroids you just created. If the clusters and cluster members don't change, you are complete and your clusters are created. If they change, you need to start over by going back to step 3, and continuing again and again until they don't change clusters.

Step for performing the clustering in weka

1. Load the data file into WEKA using the same Preprocess tab. Take a few minutes to look around the data in this tab. Look at the columns, the attribute data, the distribution of the columns, etc.
2. Choose particular clustering method.(show in figure 1, in that selected algorithm for clustering is Simple K means)
3. Select any one test option.
4. Click on start.

With this data set, we are looking to create clusters, so instead of clicking on the Classify tab, click on the Cluster tab. Click Choose and select Simple K Means from the choices that appear (this will be our preferred method of clustering for this article). Your WEKA Explorer window should look like Figure 1 at this point.

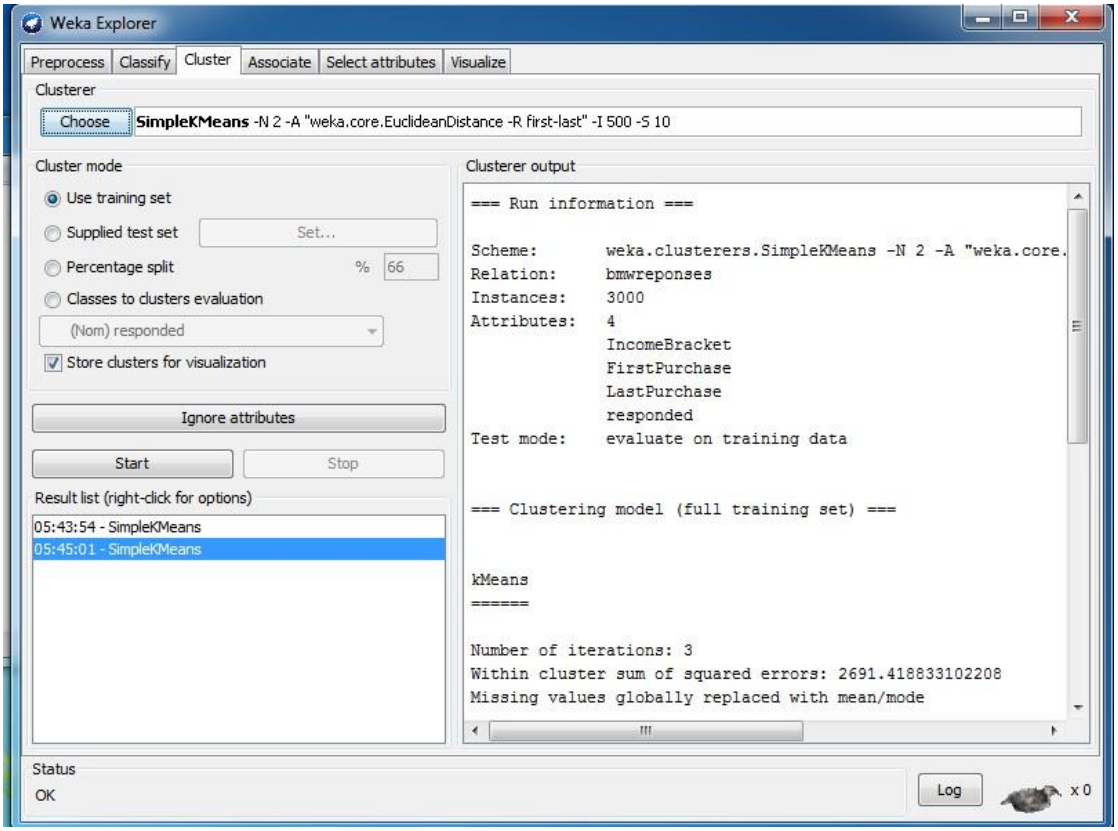


Figure 1 weka explorer (Clustering tab)

Finally, we want to adjust the attributes of our cluster algorithm by clicking SimpleKMeans (not the best UI design here, but go with it). The only attribute of the algorithm we are interested in adjusting here is the numClusters field, which tells us how many clusters we want to create. Let's change the default value of 2 to 5 for now, but keep these steps in mind later if you want to adjust the number of clusters created. Your WEKA Explorer should look like Figure 2 at this point. Click OK to accept these values.

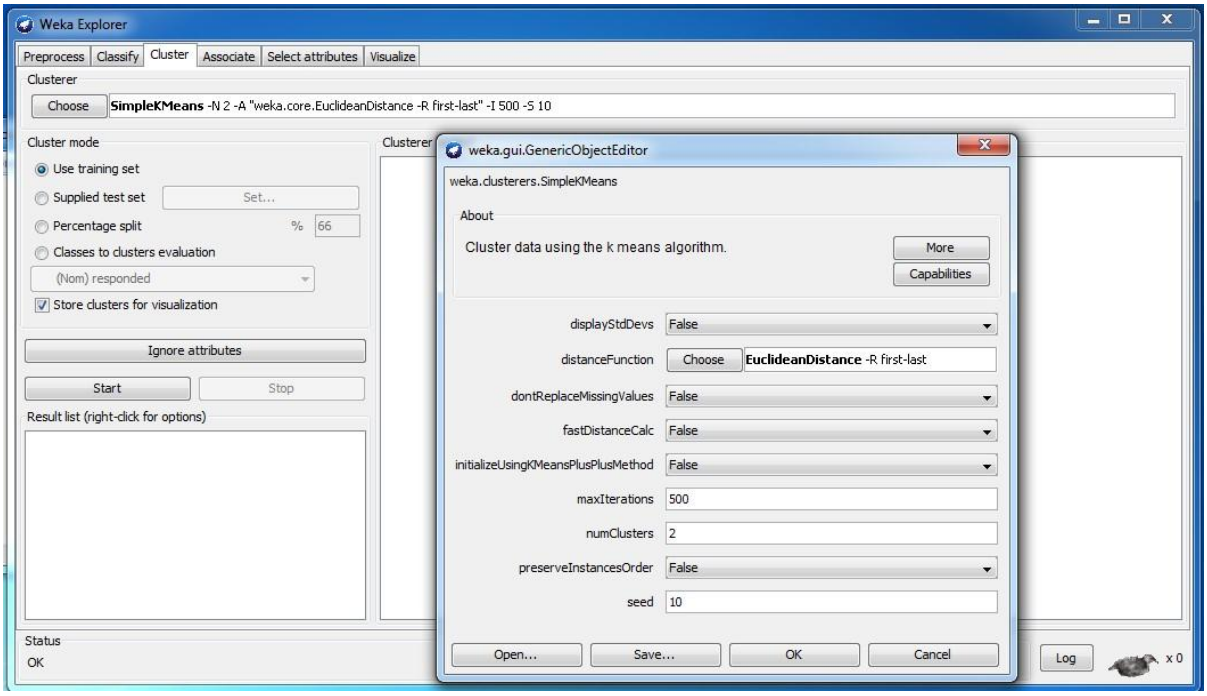


Figure 2

Output from WEKA's clustering model

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last"
-I 500 -S 10
Relation:     bmwreponses
Instances:    3000
Attributes:   4
IncomeBracket
FirstPurchase
LastPurchase
responded
Test mode:    evaluate on training data

=== Clustering model (full training set) ===

kMeans
Number of iterations: 3
Within cluster sum of squared errors: 2691.418833102208
```

Missing values globally replaced with mean/mode

Cluster centroids:

Attribute	Cluster# Full Data (3000)	0 (1525)	1 (1475)
IncomeBracket	0	0	0
FirstPurchase	200103.3653	200077.002	200130.6224
LastPurchase	200552.169	200556.6636	200547.522
responded	1	1	0

Time taken to build model (full training data) : 0.03 seconds

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

Clustered Instances

0 1525 (51%)
1 1475 (49%)

EXCERSICE:

1) Which are the different algorithms for clustering available in Weka tool?
2) Take one data base from the http://archive.ics.uci.edu/ml/. And perform Simple K-Means algorithm on that data. Take screen shot.
3) How do we interpret the clustering output?

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 5

TITLE: Learn to use Pivot Tables in Excel 2007 to Organize Data.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is pivot table?
- How to create pivot table in Excel 2007?
- How to organize the data?

THEORY:

What is a Pivot Table?

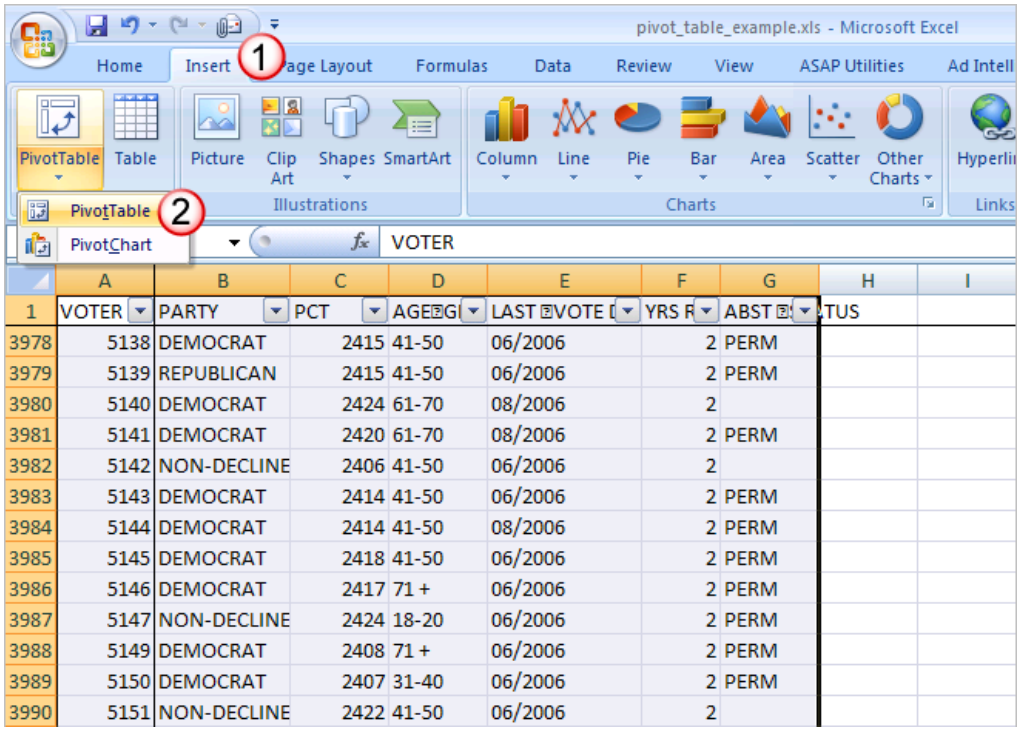
Excel‘s Pivot Table is probably the most useful and time-saving tool for analyzing data that‘s in table format. In the simplest Pivot Table, one identifies a row value, a column value, and a data value. The data value (usually a numeric value) in this simple PivotTable is automatically summarized at each row and column intersection.

How to Create a Pivot Table in Excel 2007

If you have a large spreadsheet with tons of data, it‘s a good idea to create a Pivot Table to easily analyze data more easily. Today we take a look at creating a basic Pivot Table to better organize large amounts of data to identify specific areas.

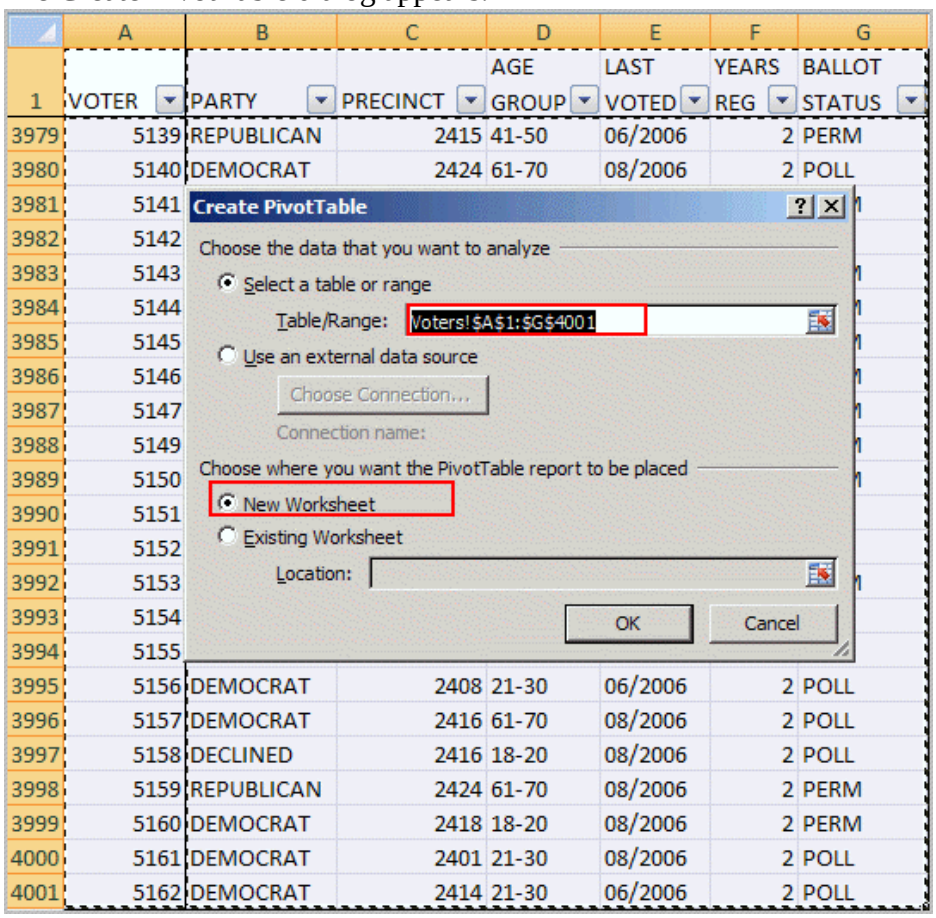
Create a Pivot Table

1. Open your original spreadsheet and remove any blank rows or columns.
2. Make sure each column has a heading, as it will be carried over to the Field List.
3. Make sure your cells are properly formatted for their data type.
4. Highlight your data range
5. Click the **Insert** tab.
6. Select the **PivotTable** button from the **Tables** group.
7. Select **PivotTable** from the list.



	A	B	C	D	E	F	G	H	I
1	VOTER	PARTY	PCT	AGE	LAST VOTE	YRS R	ABST	TUS	
3978	5138	DEMOCRAT	2415	41-50	06/2006	2	PERM		
3979	5139	REPUBLICAN	2415	41-50	06/2006	2	PERM		
3980	5140	DEMOCRAT	2424	61-70	08/2006	2			
3981	5141	DEMOCRAT	2420	61-70	08/2006	2	PERM		
3982	5142	NON-DECLINE	2406	41-50	06/2006	2			
3983	5143	DEMOCRAT	2414	41-50	06/2006	2	PERM		
3984	5144	DEMOCRAT	2414	41-50	08/2006	2	PERM		
3985	5145	DEMOCRAT	2418	41-50	06/2006	2	PERM		
3986	5146	DEMOCRAT	2417	71 +	06/2006	2	PERM		
3987	5147	NON-DECLINE	2424	18-20	06/2006	2	PERM		
3988	5149	DEMOCRAT	2408	71 +	06/2006	2	PERM		
3989	5150	DEMOCRAT	2407	31-40	06/2006	2	PERM		
3990	5151	NON-DECLINE	2422	41-50	06/2006	2			

The Create PivotTable dialog appears.



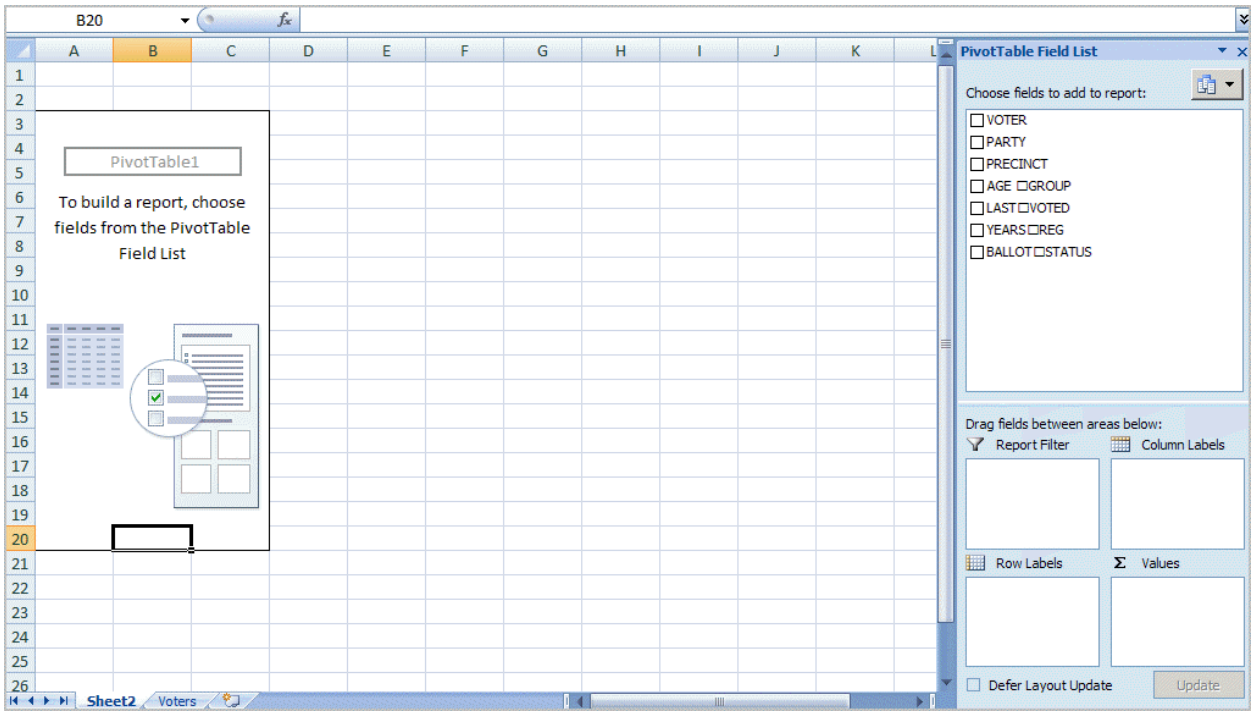
	A	B	C	D	E	F	G
1	VOTER	PARTY	PRECINCT	AGE GROUP	LAST VOTED	YEARS REG	BALLOT STATUS
3979	5139	REPUBLICAN	2415	41-50	06/2006	2	PERM
3980	5140	DEMOCRAT	2424	61-70	08/2006	2	POLL
3981	5141						
3982	5142						
3983	5143						
3984	5144						
3985	5145						
3986	5146						
3987	5147						
3988	5149						
3989	5150						
3990	5151						
3991	5152						
3992	5153						
3993	5154						
3994	5155						
3995	5156	DEMOCRAT	2408	21-30	06/2006	2	POLL
3996	5157	DEMOCRAT	2416	61-70	08/2006	2	POLL
3997	5158	DECLINED	2416	18-20	08/2006	2	POLL
3998	5159	REPUBLICAN	2424	61-70	08/2006	2	PERM
3999	5160	DEMOCRAT	2418	18-20	08/2006	2	PERM
4000	5161	DEMOCRAT	2401	21-30	08/2006	2	POLL
4001	5162	DEMOCRAT	2414	21-30	06/2006	2	POLL

8. Double-check your **Table/Range:** value.

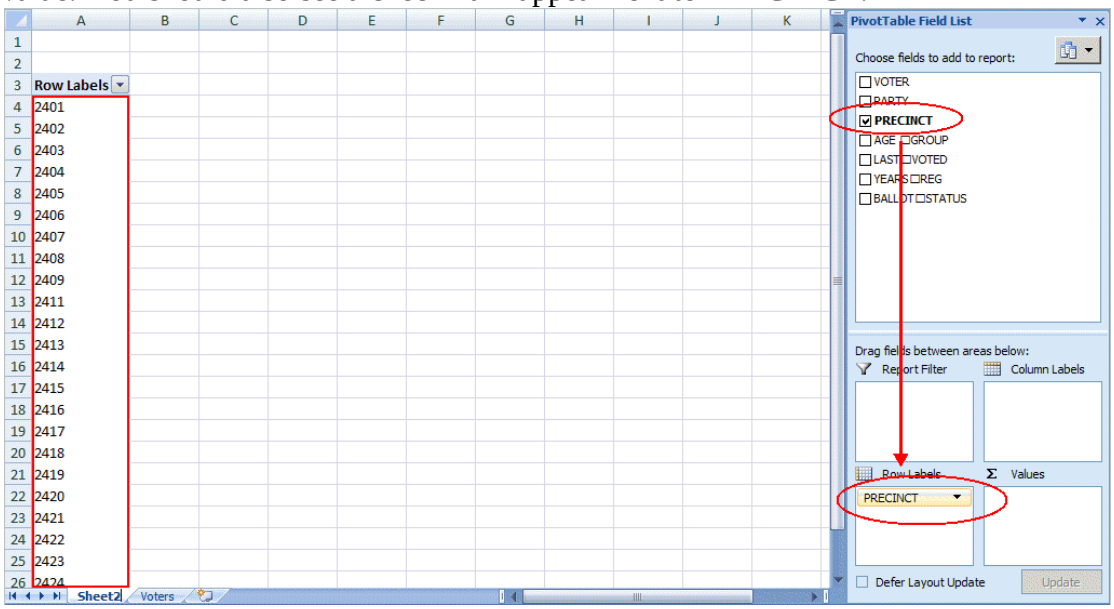
9. Select the radio button for **New Worksheet**.

10. Click **OK**.

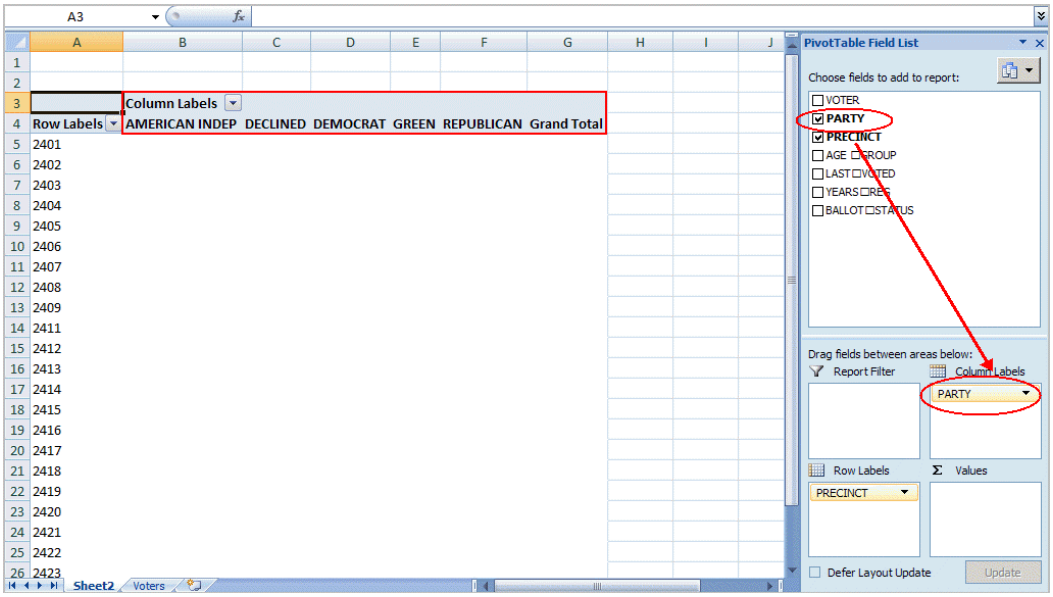
    A new worksheet opens with a blank pivot table. You'll see that the fields from our source spreadsheet were carried over to the **PivotTable Field List**.



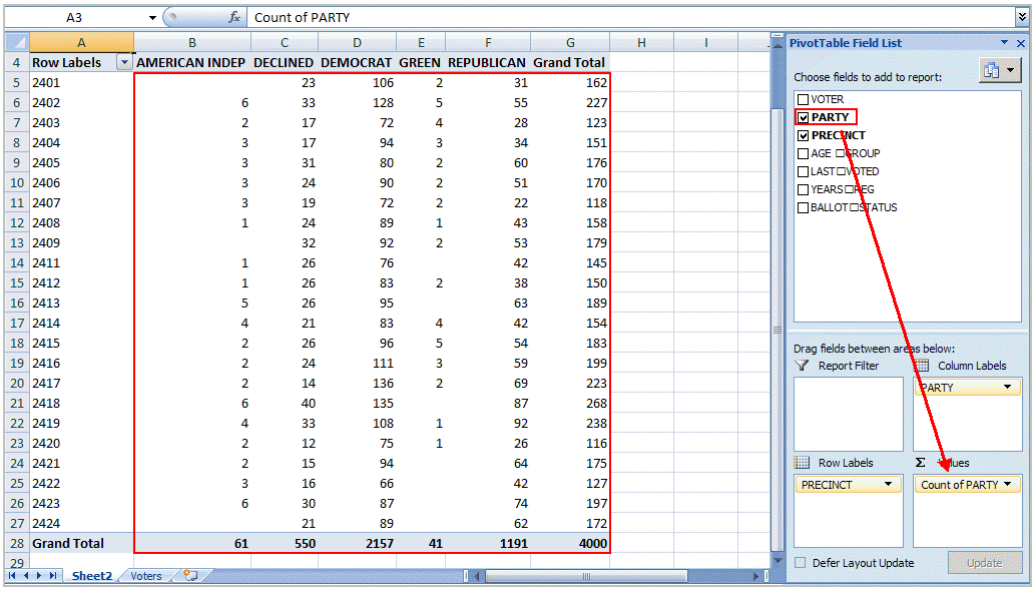
11. Drag an item such as PRECINCT from the PivotTable Field List down to the Row Labels quadrant. The left side of your Excel spreadsheet should show a row for each precinct value. You should also see a checkmark appear next to PRECINCT.



12. The next step is to ask what you would like to know about each precinct. I'll drag the PARTY field from the PivotTable Field List to the Column Labels quadrant. This will provide an additional column for each party. Note that you won't see any numerical data.

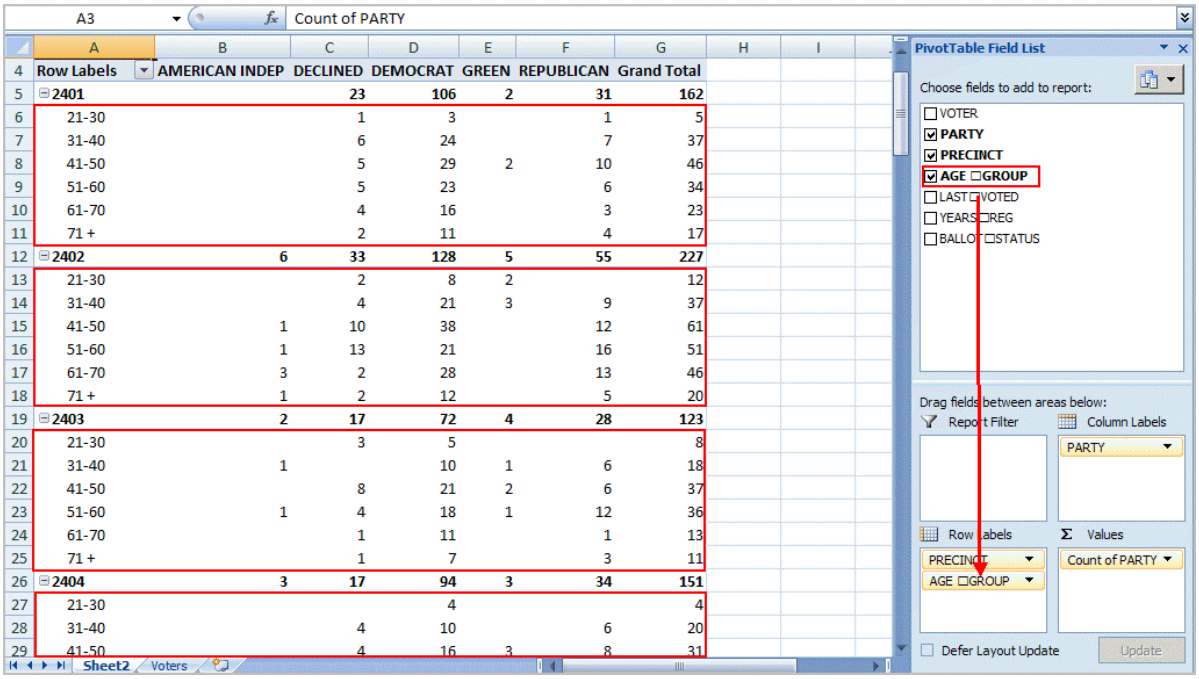


13. To see the count for each party, I need to drag the same field to the Values quadrant. In this case, Excel determines I want a Count of PARTY. I could double-click the entry and choose another Field Setting. Excel has also added Grand Totals.



Additional Groupings and Options

As you build your Excel pivot table, you'll probably think of additional ways to group the information. For example, you might want to know the Age Range of voters by Precinct by Party. In this case, I would drag the AGE GROUP column from the PivotTable Field List down below the PRECINCT value in Row Labels.



Row Labels	AMERICAN INDEP	DECLINED	DEMOCRAT	GREEN	REPUBLICAN	Grand Total
2401		23	106	2	31	162
21-30		1	3		1	5
31-40		6	24		7	37
41-50		5	29	2	10	46
51-60		5	23		6	34
61-70		4	16		3	23
71 +		2	11		4	17
2402	6	33	128	5	55	227
21-30		2	8	2		12
31-40		4	21	3	9	37
41-50	1	10	38		12	61
51-60	1	13	21		16	51
61-70	3	2	28		13	46
71 +	1	2	12		5	20
2403	2	17	72	4	28	123
21-30		3	5			8
31-40	1		10	1	6	18
41-50		8	21	2	6	37
51-60	1	4	18	1	12	36
61-70		1	11		1	13
71 +		1	7		3	11
2404	3	17	94	3	34	151
21-30			4			4
31-40		4	10		6	20
41-50		4	16	3	8	31

Each age group is broken out and indented by precinct. At this stage, you might also be thinking of usability. As with a regular spreadsheet, you may manipulate the fields. For example, you might want to rename ―Grand Total‖ to ―Total‖ or even collapse the age values for one or more precincts. You can also hide or show rows and columns. These features work the same way as a regular spreadsheet.

One area that is different is the pivot table has its own options. You can access these options by right-clicking a cell within and selecting PivotTable Options… For example, you might only want Grand Totals for columns and not rows.

There are also ways to filter the data using the controls next to Row Labels or Column labels on the pivot table. You may also drag fields to the Report Filter quadrant.

EXCERSICE:

Visit UCI data repository and download king-rook-vs-king chess data (http://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King) to answer the following questions using Microsoft Excel Pivot Table and Pivot Chart tool (Give relevant print-outs):

1) How many times the result was draw when the White king File position was _c‘ and the Black King File position was also _c‘?
2) What was the White King Rank position the highest number of times when the result is a win with moves between zero and five (inclusive)? How many times did this occur?
3) Identify a Black King Rank position that required every time more than 5 moves for a win.
4) Draw a pie chart showing the distribution of different values of result.
5) Can you identify some positions that lead to win with few moves?

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 6

TITLE: Study of the Apriori Algorithm (Finding Frequent Item sets Using Candidate Generation).

OBJECTIVE: On completion of this exercise student will able to know about…

1. What is frequent pattern?
2. What is Association rule mining?
3. Study about Association rule mining algorithm (Apriori).
4. Implementation of Apriori algorithm in Weka tools and interprets the output.

THEORY:

What is Apriori algorithm?

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.

The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent item set properties, as we shall see following.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are usedtoexplore (k+1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L1.Next, L1 is used to find L2, the set of frequent 2-item sets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database.

Property of Apriori

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

TheApriori property is based on the following observation. By definition, if an itemset*I* does not satisfy the minimum support threshold, *min sup*, then *I* is not frequent; thatis, *P*(*I*) <*min sup*. If an item *A* is added to the itemset*I*, then the resulting itemset (i.e.,*I* [*A*) cannot occur more frequently than *I*. Therefore, *I* U*A* is not frequent either; that is, *P*(*I* U*A*) <*min sup*.[1]

This property belongs to a special category of properties called antimonotone in the sense that

if a set cannot pass a test, all of its supersets will fail the same test as well.

It is called *antimonotone* because the property is monotonic in the context of failing a test.

AprioriAlgorithm [1]:

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

Input:***D***, a database of transactions;
Min_sup, the minimum support count threshold.

Output: ***L***, frequent itemsets in *D*.

Method:

1. $L1$ = find frequent 1-itemsets(D);
2. for ($k = 2; L_{k-1} \neq \Phi; k$++) {
3. C_k= apriori gen(L_{k-1});
4. for each transaction $t \in D$ f // scan *D* for counts
5. C_t= subset(C_k, t); // get the subsets of t that are candidates
6. for each candidate $c \in Ct$
7. c.count++;
8. }
9. $L_k = \{c \in C \mid c.count \geq min\_sup\}$
10. }
11. return $L = \cup_k L_k$;

procedureapriori gen(L_{k-1}:frequent (k-1)-itemsets)

1. for each itemset$l_1 \in L_{k-1}$
2. for each itemset$l_2 \in L_{k-1}$
3. if ($l_1[1] = l_2[1]$)^($l_1[2] = l_2[2]$)^:::^($l_1[k\text{-}2] = l_2[k\text{-}2]$)^($l_1[k\text{-}1] < l_2[k\text{-}1]$) then {
4. $c = l_1$join l_2; // join step: generate candidates
5. if has infrequent subset(c, L_{k-1}) then
6. delete c; // prune step: remove unfruitful candidate
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 *not belongs to* L_{k-1}then
3. return TRUE;
4. return FALSE;

Generating Association Rules from Frequent Itemsets

Once the frequent itemsets from transactions in a database *D* have been found, it is Straight forward to generate strong association rules from them (where *strong* association rules

satisfy both minimum support and minimum confidence). This can be done using following Equation. for confidence, which we show again here for completeness:

$$confidence(A => B) = \frac{support\_count(A \cup B)}{support(A)}$$

The conditional probability is expressed in terms of itemset support count, where *support_count*(*A* *B*) is the number of transactions containing the itemsets*A* ∪*B*, and*support count*(*A*) is the number of transactions containing the itemset*A*. Based on thisequation, association rules can be generated as follows:

- For each frequent itemset*l*, generate all nonempty subsets of *l*.
- For every nonempty subset *s* of *l*, output the rule "*s* =>*l* -*s*)‖ if (*support count*(*l*) / *support count*(*s*)) ≥ *min_conf*, where *min_conf*is the minimum confidence threshold.

Example:

Apriori. Let's look at a concrete example, based on the AllElectronics transaction database, D, of Table 1. There are nine transactions in this database, that is, |D| = 9. We use Figure 1 to illustrate the Apriori algorithm for finding frequent itemsets in D.

Table 1Transactional data for an AllElectronics branch.

TID	List of item_IDs.
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

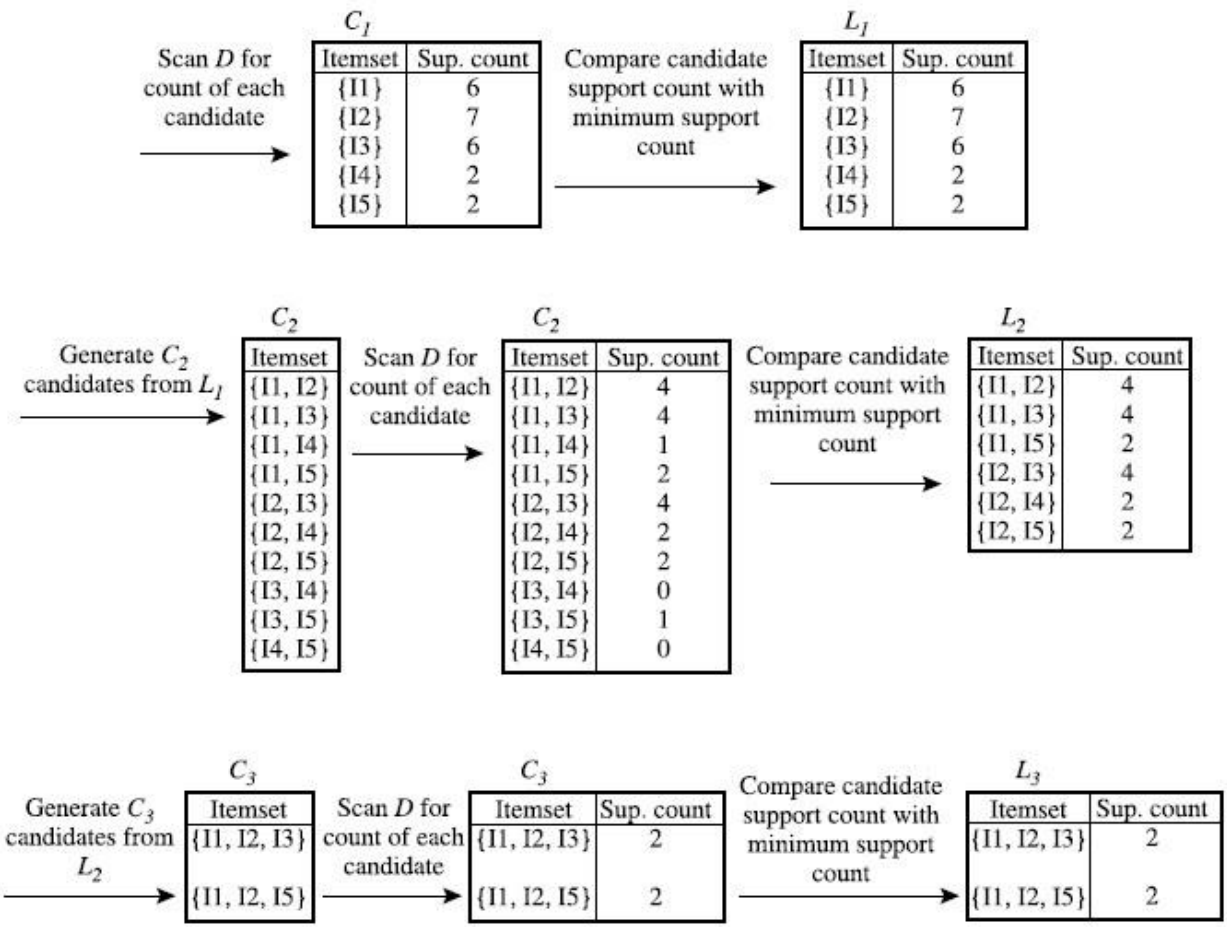


Figure 1 Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

Let‘s try an example based on the transactional data for *All Electronics*shown in Table 1. Suppose the data contain the frequent itemset*l* = fI1, I2, I5g. What are the association rules that can be generated from *l*? Then on empty subsets of *l* are fI1, I2g, fI1, I5g, fI2, I5g, fI1g, fI2g, and fI5g. The resulting association rules are as shown below, each listed with its confidence:

*I*1^*I*2 =>*I*5, *confidence* = 2=4 = 50%
*I*1^*I*5=>*I*2, *confidence* = 2=2 = 100%
*I*2^*I*5=>*I*1, *confidence* = 2=2 = 100%
*I*1=>*I*2^*I*5, *confidence* = 2=6 = 33%
*I*2=>*I*1^*I*5, *confidence* = 2=7 = 29%
*I*5=>*I*1^*I*2, *confidence* = 2=2 = 100%

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong. Note that, unlike conventional classification rules, association rules can contain more than one conjunct in the right-hand side of the rule.

References:

[1] JiaweiHan,MichelineKamber,"Data Mining:Concepts and Techniques",Second Edition, University of Illinois at Urbana-Champaign

EXCERSICE:

1) Take one dataset from the http://archive.ics.uci.edu/ml/or any. And perform Apriori algorithm on that data in Weka tool. Take screen shot.
2) How do we interpret the clustering output?
3) Write down the disadvantage of Apriori algorithm.

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 7

TITLE: Survey of Different Data Mining Tools.

OBJECTIVE: On completion of this exercise student will able to know about…

This practical attempt to support the decision-making process by discussing the historical development and presenting a range of existing state-of-the-art data mining and related tools. Furthermore, the tool categorization based on different user groups, data structures, data mining tasks and methods, visualization and interaction styles, import and export options for data and models, platforms, and license policies.

THEORY:

There are three stages for introduction to data mining tools.

1. The first section Historical Development and State-of-the-Art highlights the historical development of data mining software until present;
2. The criteria to compare data mining software are explained in the second section Criteria for Comparing Data Mining Software.
3. The last section Categorization of Data Mining Software into Different Types proposes a categorization of data mining software and introduces typical software tools for the different types.

Historical Development and State-Of-The-Art

Following the original definition given in Ref [1]

"Data mining is a step in the knowledge discovery from databases (KDD) process that consists of applying data analysis and discovery algorithms to produce a particular enumeration of patterns (or models) across the data. In that same article, KDD is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data."

Today, a large number of standard data mining methods are available, from a historical perspective, these methods have different roots. One early group of methods was adopted from **classical statistics**: the focus was changed from the proof of known hypotheses to the generation of new hypotheses. Examples include methods from Bayesian decision theory, regression theory, and principal component analysis. Another group of methods stemmed from **artificial intelligence**- like decision trees, rule-based systems, and others. The term _machine learning' includes methods such as support vector machines and artificial neural networks. There are several different and sometimes overlapping categorizations; for example, fuzzy logic, artificial neural networks, and evolutionary algorithms, which are summarized as computational intelligence.[2]

The typical life cycle of new data mining methods begins with theoretical papers based on in houses oft ware prototypes, followed by public or on-demand software distribution of successful

algorithms as research prototypes. Then, either special commercial or open source packages containing a family of similar algorithms are developed or the algorithms are integrated into existing open source or commercial packages. Many companies have tried to promote their own stand alone packages, but only few have reached notable market shares. The life cycle of some data mining tools is remarkably short. Typical reasons include internal marketing decisions and acquisitions of specialized companies by larger ones, leading to a renaming and integration of product lines.

The largest commercial success stories resulted from the step-wise integration of data mining methods into established commercial statistical tools. Companies such as SPSS, founded in 1975 with precursors from 1968, or SAS, founded in 1976, have been offering statistical tools for mainframe computers since the 1970s.

Many companies offering business intelligence products have integrated data mining solutions into their database products; one example is Oracle Data Mining (established in 2002). Many of these products are also a product of the acquisition and integration of specialized datamining companies.

In 2008, the worldwide market for business intelligence (i.e., software and maintenance fees) was 7.8 billion USD, including 1.5 billion USD in socalled_advanced analytics', containing data mining and statistics.7 This sector has grown 12.1% between 2007 and 2008, with large players including companies such as SAS (33.2%, tool: SAS Enterprise Miner), SPSS (14.3%, since 2009, an IBM company; tool: IBMSPSS Modeler), Microsoft (1.7%, tool: SQL Server Analysis Services), Teradata (1.5%, tool: Teradata Database, former name TeraMiner), and TIBCO (1.4%, tool: TIBCO Spotfire). [3]

Open-source libraries have also become very popular since the 1990s. The most prominent example is Waikato Environment for Knowledge Analysis (WEKA), see Ref 8. WEKA started in 1994 as a C++ library, with its first public release in 1996. In 1999, it was completely rebuilt as a JAVA package; since that time, it has been regularly updated.

Criteria for Comparing Data Mining Software

In the following, different criteria for comparison of data mining software are introduced. These criteria are based on user groups, data structures, data mining tasks and methods, import and export options, and license models.

A. User Groups

There are many different data mining tools available, which fit the needs of quite different user groups:

1. **Business applications:** This group uses data mining as a tool for solving Commercially relevant business applications such as customer relationship management, fraud detection, and so on.
2. **Applied research:** A user group that applies data mining to research problems, for example, technology and life sciences. Here, users are mainly interested in tools with

Well proven methods, a graphical user interface(GUI), and interfaces to domain-related data formats or databases.

3. **Algorithm development:** Develops new data mining algorithms, and requires tools to both integrate its own methods and compare these with existing methods. The necessary tools should contain many concurrent algorithms.
4. **Education:** For education at universities, data mining tools should be very intuitive, with a comfortable interactive user interface, and inexpensive. In addition, they should
   Allow the integration of in-house methods during programming seminars.

B. Data Structures

An important criterion is the dimensionality of the underlying raw data in the processed dataset. The first data mining applications were focused on handling datasets represented as two-dimensional feature tables. In this classical format, a dataset consists of a set of N examples (e.g., clients of an insurance company) with s features containing real values or usually integer-coded classes or symbols (e.g., income, age, number of contracts, and alike). This format is supported by nearly all existing tools. In some cases, the dataset can be sparse, with only a few nonzero features such as a list of s shopping items for N different customers. The computational and memory effort can be reduced if a tool exploits this sparse structure.

There are different types of data format like feature data (e.g. age and income), texts, time-series data, sequences, images, graphs, 3D images, Videos,3D videos etc.

C. Task and Methods

The most important tasks in data mining are

1. *supervised learning*, with a known output variable in the dataset, including
   - classification: class prediction, with the variable typically coded as an integer output;
   - fuzzy classification: with gradual memberships with values in-between0 and 1 applied to the different classes;
   - regression: prediction of a real-valued output variable, including special cases of predicting future values in a time series out of recent or past values;
2. *unsupervised learning*, without a known output variable in the dataset, including
   - clustering: finds and describes groups of similar examples in the data using crisp of fuzzy clustering algorithms;
   - association learning: finds typical groups of items that occur frequently together in examples;
3. *Semi supervised learning*, whereby the output variable is known only for some examples.

D. Platforms

Data mining tools can be subdivided into standalone and client/server solutions. Client/server solutions dominate, especially in products designed for business users.

They are available for different platforms, including Windows, MAC OS, Linux, or special mainframe supercomputers. There is a growing number of JAVA-based systems that are platform independent for users in research and applied research.

E. Licenses

There exists a wide variety of data mining tools with commercial and open-source licenses. This is particularly true in the business application user group, where commercial software is very attractive due to high software stability, good coupling with other commercial tools for data warehouses, included software maintenance, and the possibility of user training for sophisticated topics. For all other user groups, there is a strong trend toward open-source software, but different types of licenses exist for this.

Categorization of Data Mining Software into Different Types

Following the criteria from the previous section, different types of similar data mining tools can be found.

In addition, for commercial data mining tools, related tools and their group membership are summarized in different tables for commercial (Tables 1 and 2), free, and open-source data mining tools (Table 3). In these tables, very popular tools are marked in bold.

The following types are proposed:[3]

1. Data mining suites (DMS) focus largely on data mining and include numerous methods. They support feature tables and time series, while additional tools for text mining are sometime available. Typical examples include IBM SPSS Modeler, SAS Enterprise Miner, Alice d'Isoft, DataEngine, DataDetective, GhostMiner, Knowledge Studio, KXEN, NAG Data Mining Components, Partek Discovery Suite, STATISTICA, and TIBCO Spotfire.

2. Business intelligence packages (BIs) have no special focus to data mining, but include basic data mining functionality, especially for statistical methods in business applications. Most BI softwares are commercial (IBM Cognos 8 BI,OracleDataMining, SAPNetweaver Business Warehouse, Teradata Database, DB2 Data Warehouse from IBM, and PolyVista), but a few open-source solutions exist (Pentaho).

3. Mathematical packages (MATs) have no special focus on data mining, but provide a large and extendable set of algorithms and visualization routines MATs are attractive to users in algorithm development and applied research because data mining algorithms can be rapidly implemented, mostly in the form of extensions (EXT) and research prototypes (RES). MAT packages exist as commercial (MATLAB and R-PLUS) or open-source tools (R, Kepler).

4. Integration packages (INTs) are extendable bundles of many different open-source algorithms, either as stand-alone software (mostly based on Java; as KNIME, the GUI-version of WEKA, KEEL, and TANAGRA) or as a kind of larger extension package for

tools from the MAT type (such as Gait-CAD, PRTools for MATLAB, and RWEKA for R).

5. EXT are smaller add-ons for other tools such as Excel,Matlab, R, and so forth, with limited but quite useful functionality Data mining libraries (LIBs) implement data mining methods as a bundle of functions. These functions can be embedded in other software tools using an Application Programming Interface (API) for the interaction between the software tool and the data mining functions. A graphical user interface is missing, but some functions can support the integration of specific visualization tools. They are often written in JAVA or C++ and the solutions are platform independent. Open source examples are WEKA (Java-based), MLC++ (C++ based), JAVA Data Mining Package, and LibSVM (C++ and JAVAbased) for support vector machines

6. Specialties (SPECs) are similar to DMS tools, but implement only one special family of methods such as artificial neural networks. Examples are CART for decision trees, Bayesia Lab for Bayesian networks, C5.0, WizRule, Rule Discovery System for rule-based systems, MagnumOpus for association analysis, and JavaNNS, Neuroshell.

7. RES are usually the first—and not always stable—implementations of new and innovative algorithms. They contain only one or a few algorithms with restricted graphical support and without automation support. RES tools are mostly opensource. Examples are GIFT for content-based image retrieval, Himalaya for mining maximal frequent item sets, sequential pattern mining and scalable linear regression trees, Rseslibs for rough sets, and Pegasus for graph mining.

8. Solutions (SOLs) describe a group of tools that are customized to narrow application fields such as text mining, image processing etc.

Table 1 List of Commercial Tools (Part 1) [3]

TOOL	TYPE	LINK
ADAPA (Zementis)	DMS	www.zementis.com
Alice (d'Isoft)	DMS	www.alice-soft.com
Bayesia Lab	SPEC	www.bayesia.com
C5.0	SPEC	www.rulequest.com
CART	SPEC	www.salford-systems.com
Data Applied	DMS	data-applied.com
DataDetective	DMS	www.sentient.nl/?dden
DataEngine	DMS	www.dataengine.de
Datascope	DMS	www.cygron.hu
DB2 Data Warehouse	BI	www.ibm.com/software/data/infosphere/warehouse
DeltaMaster	BI	www.bissantz.com/deltamaster
Forecaster XL	EXT	www.alyuda.com
GhostMiner	DMS	www.fqs.pl/businessintelligence/products/ghostminer
IBM Cognos 8 BI	BI	www.ibm.com/software/data/cognos/data-mining-tools.html
IBM SPSS Modeler	DMS	www.spss.com/software/modeling/modeler
IBM SPSS Statistics	MAT	www.spss.com/software/statistics

iModel	DMS	www.biocompsystems.com/products/imodel
InfoSphere Warehouse	BI	www.ibm.com/software/data/infosphere/warehouse
JMP	DMS	www.jmpdiscovery.com
KnowledgeMiner	SPEC	www.knowledgeminer.net
KnowledgeStudio	DMS	www.angoss.com
KXEN	DMS	www.kxen.com
Magnum Opus	SPEC	www.giwebb.com
MATLAB	MAT	www.mathworks.com
MATLAB Neural Network Toolbox	EXT	www.mathworks.com
Model Builder	DMS	www.fico.com
ModelMAX	SOL	www.asacorp.com/products/mmxover.jsp

Table 2 List of Commercial Tools (Part 2) [3]

TOOL	TYPE	LINK
Molegro Data Modeler	SOL	www.molegro.com
NAG Data Mining Components	LIB	www.nag.co.uk/numeric/DR/DRdescription.asp
NeuralWorks Predict	SPEC	www.neuralware.com/products.jsp
Neurofusion	LIB	www.alyuda.com
Neuroshell	SPEC	www.neuroshell.com
Oracle Data Mining (ODM)	DMS	www.oracle.com/technology/products/bi/odm/index.html
Partek Discovery Suite	DMS	www.partek.com/software
Partek Genomics Suite	SOL	www.partek.com/software
PolyAnalyst	DMS	www.megaputer.com/polyanalyst.php
PolyVista	BI	www.polyvista.com
Random Forests	SPEC	www.salford-systems.com
RapAnalyst	SPEC	www.raptorinternational.com/rapanalyst.html
R-PLUS	MAT	www.experience-rplus.com
SAP Netweaver Business Warehouse (BW)	BI	www.sap.com/platform/netweaver/components/businesswarehouse
SAS Enterprise Miner	DMS	www.sas.com/products/miner
See5	SPEC	www.rulequest.com
SPAD Data Mining	DMS	eng.spadsoft.com
SQL Server Analysis Services	DMS	www.microsoft.com/sql
STATISTICA	DMS	www.statsoft.com/products/data-mining-solutions/G259
SuperQuery	DMS	www.azmy.com
Teradata Database	BI	www.teradata.com
Think Enterprise Data Miner (EDM)	DMS	www.thinkanalytics.com
TIBCO Spotfire	DMS	spotfire.tibco.com
UnicaPredictiveInsight	DMS	www.unica.com
WizRule and WizWhy	SPEC	www.wizsoft.com
XAffinity	SPEC	www.exclusiveore.com

Table 3 List of Free and Open-Source Tools [3]

TOOL	TYPE	LINK
ADaM	LIB	datamining.itsc.uah.edu/adam
CellProfilerAnalyst	SOL	www.cellprofiler.org/index.htm
D2K	DMS	alg.ncsa.uiuc.edu
Gait-CAD	INT	sourceforge.net/projects/gait-cad
GATE	SOL	gate.ac.uk/download
GIFT	RES	www.gnu.org/software/gift
Gnome Data Mine Tools	DMS	www.togaware.com/datamining/gdatamine
Himalaya	RES	himalaya-tools.sourceforge.net
ImageJ	SOL	rsbweb.nih.gov/ij
ITK	SOL	www.itk.org
JAVA Data Mining Package	LIB	sourceforge.net/projects/jdmp
JavaNNS	SPEC	www.ra.cs.uni-tuebingen.de/software/JavaNNS/welcome e.html
KEEL	INT	www.keel.es
Kepler	MAT	kepler-project.org
KNIME	INT	www.knime.org
LibSVM	LIB	www.csie.ntu.edu.tw/ cjlin/libsvm
MEGA	SOL	www.megasoftware.net/m distance.html
MLC++	LIB	www.sgi.com/tech/mlc
Orange	LIB	www.ailab.si/orange
Pegasus	RES	www.cs.cmu.edu/ pegasus
Pentaho	BI	sourceforge.net/projects/pentaho
Proximity	SPEC	kdl.cs.umass.edu/proximity/index.html
PRTools	EXT	www.prtools.org
R	MAT	www.r-project.org
RapidMiner	DMS	www.rapidminer.com
Rattle	INT	rattle.togaware.com
ROOT	LIB	root.cern.ch/root
ROSETTA	SPEC	www.lcb.uu.se/tools/rosetta/index.php
Rseslibs	RES	logic.mimuw.edu.pl/ rses
Rule Discovery System*	SPEC	www.compumine.com
RWEKA	INT	cran.r-project.org/web/packages/RWeka/index.html
TANAGRA	INT	eric.univ-lyon2.fr/ ricco/tanagra/en/tanagra.html
Waffles	LIB	waffles.sourceforge.net
WEKA	DMS, LIB	sourceforge.net/projects/weka
XELOPES Library*	LIB	www.prudsys.de/en/technology/xelopes
XLMiner*	EXT	www.resample.com/xlminer

References:

[1] Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery in databases. *AI Mag* 1996; 17:37–54.

[2] Engelbrecht AP. *Computational Intelligence - An Introduction*. Chichester: John Wiley; 2007.

[3] Ralf Mikut∗ and Markus Reischl,*Data mining tools.*

EXERCISE:

1) Write down the functionality and advantage of the top 5 analytical tool.

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 8

TITLE: Decision Tree Learning.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is Decision tree?
- Basic concept about decision tree.
- How to construct Decision Tree?
- Decision tree algorithm (ID3)

THEORY:

What is decision Tree?

Imagine you only ever do four things at the weekend: go shopping, watch a movie, play tennis or just stay in. What you do depends on three things: the weather (windy, rainy or sunny); how much money you have (rich or poor) and whether your parents are visiting. You say to your yourself: if my parents are visiting, we'll go to the cinema. If they're not visiting and it's sunny, then I'll play tennis, but if it's windy, and I'm rich, then I'll go shopping. If they're not visiting, it's windy and I'm poor, then I will go to the cinema. If they're not visiting and it's rainy, then I'll stay in.

To remember all this, you draw a flowchart which will enable you to read off your decision. We call such diagrams **decision trees**. A suitable decision tree for the weekend decision choices would be as follows:

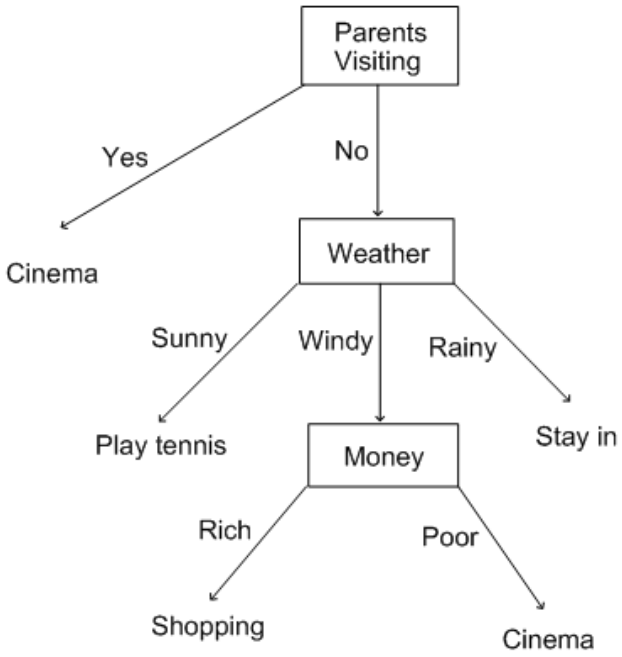


Figure 1

We can see why such diagrams are called trees, because, while they are admittedly upside down, they start from a root and have branches leading to leaves (the tips of the graph at the bottom). Note that the leaves are always decisions, and a particular decision might be at the end of multiple branches (for example, we could choose to go to the cinema for two different reasons). Armed with our decision tree, on Saturday morning, when we wake up, all we need to do is check (a) the weather (b) how much money we have and (c) whether our parent's car is parked in the drive. The decision tree will then enable us to make our decision. Suppose, for example, that the parents haven't turned up and the sun is shining. Then this path through our decision tree will tell us what to do:

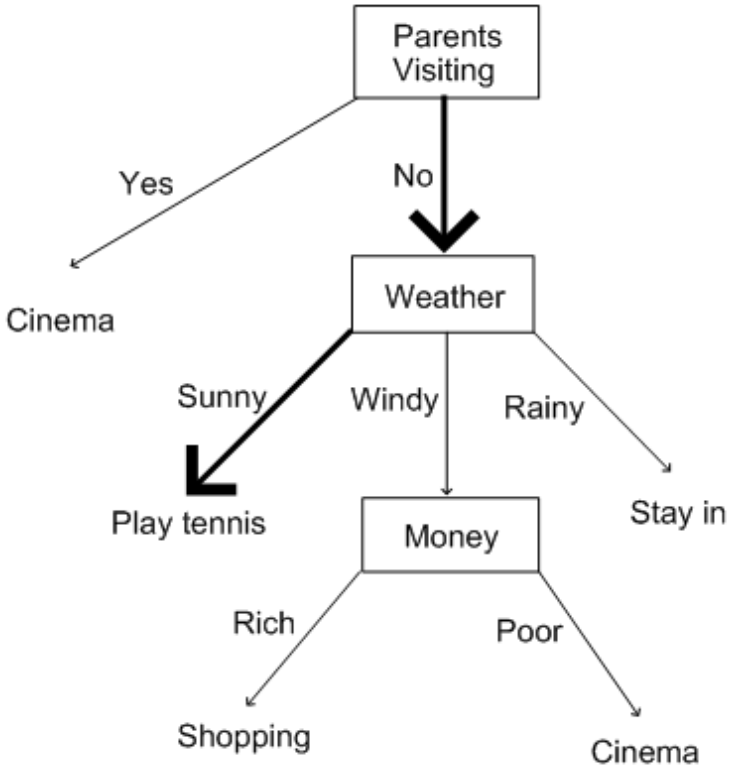


Figure 2

and hence we run off to play tennis because our decision tree told us to. Note that the decision tree covers all eventualities. That is, there are no values that the weather, the parents turning up or the money situation could take which aren't catered for in the decision tree. Note that, in this lecture, we will be looking at how to automatically generate decision trees from examples, not at how to turn thought processes into decision trees.

Reading Decision Trees

There is a link between decision tree representations and logical representations, which can be exploited to make it easier to understand (read) learned decision trees. If we think about it, every decision tree is actually a disjunction of implications (if ... then statements), and the implications are Horn clauses: a conjunction of literals implying a single literal. In the above tree, we can see this by reading from the **root node** to each **leaf node**:

If the parents are visiting, then go to the cinema
Or
if the parents are not visiting *and* it is sunny, then play tennis
Or
If the parents are not visiting *and* it is windy *and* you're rich, then go shopping
Or
If the parents are not visiting *and* it is windy *and* you're poor, then go to cinema
Or
If the parents are not visiting *and* it is rainy, then stay in.

Of course, this is just a re-statement of the original mental decision making process we described. Remember, however, that we will be programming an agent to learn decision trees from example, so this kind of situation will not occur as we will start with only example situations. It will therefore be important for us to be able to read the decision tree the agent suggests.

ID3 Algorithm

Start from root node of decision tree, testing the attribute specified by thisnode, then moving down the tree branch according to the attribute value in thegiven set. This process is the repeated at the sub-tree level.

What is decision tree learning algorithm suited for:

1. Instance is represented as attribute-value pairs. For example, attribute 'Temperature' and its value 'hot', 'mild', 'cool'. We are also concerning to extend attribute -value to continuous-valued data (numeric attribute value) in our project.
2. The target function has discrete output values. It can easily deal with instance which is assigned to a Boolean decision, such as 'true' and 'false', 'p(positive)' and 'n(negative)'. Although it is possible to extend target to real valued outputs, we will cover the issue in the later part of this report.
3. The training data may contain errors. This can be dealt with pruning techniques that we will not cover here.

The 3 widely used decision tree learning algorithms are: ID3, ASSISTANT and C4.5. We will cover ID3 in this experiment.

Decision tree learning is attractive for 3 reasons: (Paul Utgoff& Carla Brodley, 1990)

1. Decision tree is a good generalization for unobserved instance, only if the instances are described in terms of features that are correlated with the target concept.
2. The methods are efficient in computation that is proportional to the number of observed training instances.
3. The resulting decision tree provides a representation of the concept that appeal to human because it renders the classification process self-evident.

ID3 is a no incremental algorithm, meaning it derives its classes from a fixed set of training instances. An incremental algorithm revises the current concept definition, if necessary, with a new sample. The classes created by ID3 are inductive, that is, given a small set of training instances, the specific classes created by ID3 are expected to work for all future instances. The distribution of the unknowns must be the same as the test cases. Induction classes cannot be proven to work in every case since they may classify an infinite number of instances. Note that ID3 (or any inductive algorithm) may misclassify data.

Data Description

The sample data used by ID3 has certain requirements, which are:

- Attribute-value description - the same attributes must describe each example and have a fixed number of values.
- Predefined classes - an example's attributes must already be defined, that is, they are not learned by ID3.
- Discrete classes - classes must be sharply delineated. Continuous classes broken up into vague categories such as a metal being "hard, quite hard, flexible, soft, quite soft" are suspect.
- Sufficient examples - since inductive generalization is used (i.e. not provable) there must be enough test cases to distinguish valid patterns from chance occurrences.

Attribute Selection

1. **How to find entropy?**

   How does ID3 decide which attribute is the best? A statistical property, called information gain, is used. Gain measures how well a given attribute separates training examples into targeted classes. The one with the highest information (information being the most useful for classification) is selected. In order to define gain, we first borrow an idea from information theory called entropy. Entropy measures the amount of information in an attribute.

   Given a collection S of c outcomes

$$\boldsymbol{Entropy(S) = -\sum_{i=1}^{m} p_i log_2(p_i)}$$

   Where
   $\boldsymbol{p_i}$ is the proportion of S belonging to class I.
   S is over c.
   Log_2 is log base 2.
   Note that S is not an attribute but the entire sample set.

   Example 1

   If S is a collection of 14 examples with 9 YES and 5 NO examples then

$$Entropy(S) = -\left(\frac{9}{14}\right) log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) log_2\left(\frac{5}{14}\right) = 0.940$$

Notice entropy is 0 if all members of S belong to the same class (the data is perfectly classified). The range of entropy is 0 ("perfectly classified") to 1 ("totally random").

2. How to find Gain?

Two step process for find gain

a. Find entropy of particular attribute A. i.e. **Entropy(S)**
b. Find entropy of whole dataset. i.e. **Entropy$_A$(S)**

Gain(S, A) is information gain of example set S on attribute A is defined as

$$Gain(S, A) = Entropy(S) - Entropy_A(S)$$

$$Entropy_A(S) = \frac{|S_v|}{|S|} * Entropy(S_v)$$

Where:
S is each value v of all possible values of attribute A
S_v = subset of S for which attribute A has value v
$|S_v|$ = number of elements in S_v
|S| = number of elements in S

Example 2

Suppose S is a set of 14 examples in which one of the attributes is wind speed. The values of Wind can be *Weak* or *Strong*. The classification of these 14 examples are 9 YES and 5 NO. For attribute Wind, suppose there are 8 occurrences of Wind = Weak and 6 occurrences of Wind = Strong. For Wind = Weak, 6 of the examples are YES and 2 are NO. For Wind = Strong, 3 are YES and 3 are NO. Therefore

Entropy(S_{weak}) = - (6/8)*log2(6/8) - (2/8)*log2(2/8) = 0.811

Entropy(S_{strong}) = - (3/6)*log2(3/6) - (3/6)*log2(3/6) = 1.00

$$Gain(S, wind) = Entropy(S) - \frac{|8|}{|14|} * Entropy(S_{weak}) - \frac{|6|}{|14|} * Entropy(S_{strong})$$

= 0.940 - (8/14)*0.811 - (6/14)*1.00
= 0.048

For each attribute, the gain is calculated and the highest gain is used in the decision node.

Example of ID3 algorithm

Suppose we want ID3 to decide whether the weather is amenable to playing baseball. Over the course of 2 weeks, data is collected to help ID3 build a decision tree (see table 1).

The target classification is "should we play baseball?" which can be yes or no.

The weather attributes are outlook, temperature, humidity, and wind speed. They can have the following values:

outlook = { sunny, overcast, rain }
temperature = {hot, mild, cool }
humidity = { high, normal }
wind = {weak, strong }

Examples of set S are:

Table 1

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

We need to find which attribute will be the root node in our decision tree. The gain is calculated for all four attributes:

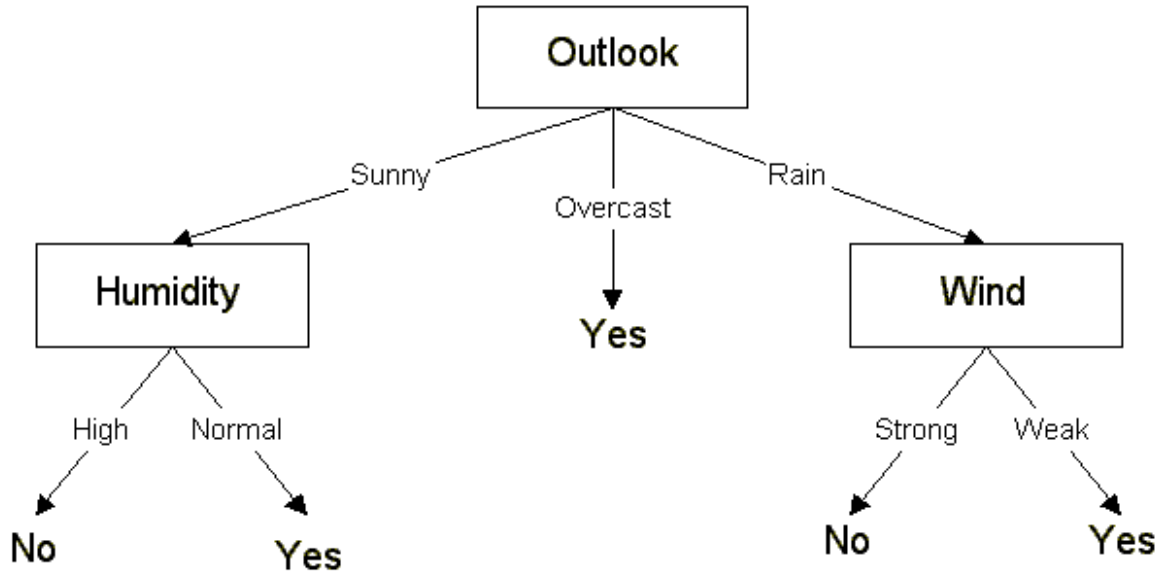
Gain(S, Outlook) = 0.246
Gain(S, Temperature) = 0.029
Gain(S, Humidity) = 0.151
Gain(S, Wind) = 0.048 (calculated in example 2)

Outlook attribute has the highest gain, therefore it is used as the decision attribute in the root node.

Since Outlook has three possible values, the root node has three branches (sunny, overcast, rain). The next question is "what attribute should be tested at the Sunny branch node?" Since we=92ve used Outlook at the root, we only decide on the remaining three attributes: Humidity, Temperature, or Wind.

S_{sunny} = {D1, D2, D8, D9, D11} = 5 examples from table 1 with outlook = sunny
Gain(S_{sunny}, Humidity) = 0.970
Gain(S_{sunny}, Temperature) = 0.570
Gain(S_{sunny}, Wind) = 0.019

Humidity has the highest gain; therefore, it is used as the decision node. This process goes on until all data is classified perfectly or we run out of attributes.



The final decision tree

The decision tree can also be expressed in rule format:

IF outlook = sunny AND humidity = high THEN playball = no
IF outlook = rain AND humidity = high THEN playball = no
IF outlook = rain AND wind = strong THEN playball = yes
IF outlook = overcast THEN playball = yes
IF outlook = rain AND wind = weak THEN playball = yes

ID3 has been incorporated in a number of commercial rule-induction packages. Some specific applications include medical diagnosis, credit risk assessment of loan applications, equipment malfunctions by their cause, classification of soybean diseases, and web search classification.

EXCERSICE:

Consider the customer database described below where an application for a credit card is either approved or rejected. Construct a decision tree (with *Approved* as the decision variable) using the entropy measure.

Case	Income in $K	Own home	Age	Years of employment	Approved
1	>60	Own	35	<5	Yes
2	30-60	Own	35	>5	Yes
3	<30	Rent	35	>5	No
4	<30	Own	35	>5	Yes
5	30-60	Own	35	<5	No
6	>60	Rent	35	>5	Yes
7	<30	Rent	35	<5	No
8	30-60	Rent	35	>5	Yes
9	<30	Own	35	<5	No
10	>60	Own	35	>5	Yes
11	30-60	Rent	35	<5	No
12	>60	Rent	35	<5	Yes

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 9

TITLE: Design & create cube by identifying measures & dimensions for star schema.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is star schema?
- How to create cube by using star schema?

THEORY:

Star schemas

A star schema consists of fact tables and dimension tables.

- Fact tables contain the quantitative or factual data about a business--the information being queried. This information is often numerical, additive measurements and can consist of many columns and millions or billions of rows.
- Dimension tables are usually smaller and hold descriptive data that reflects the dimensions, or attributes, of a business. SQL queries then use joins between fact and dimension tables and constraints on the data to return selected information.

Fact and dimension tables differ from each other only in their use within a schema. Their physical structure and the SQL syntax used to create the tables are the same. In a complex schema, a given table can act as a fact table under some conditions and as a dimension table under others. The way in which a table is referred to in a query determines whether a table behaves as a fact table or a dimension table.

Even though they are physically the same type of table, it is important to understand the difference between fact and dimension tables from a logical point of view. To demonstrate the difference between fact and dimension tables, consider how an analyst looks at business performance:

- A salesperson analyzes revenue by customer, product, market, and time period.
- A financial analyst tracks actuals and budgets by line item, product, and time period.
- A marketing person reviews shipments by product, market, and time period.

The facts--what is being analyzed in each case--are revenue, actuals and budgets, and shipments. These items belong in fact tables. The business dimensions--the by items--are product, market, time period, and line item. These items belong in dimension tables.

For example, a fact table in a sales database, implemented with a star schema, might contain the sales revenue for the products of the company from each customer in each geographic market over a period of time. The dimension tables in this database define the customers, products, markets, and time periods used in the fact table.

A well-designed schema provides dimension tables that allow a user to browse a database to become familiar with the information in it and then to write queries with constraints so that only the information that satisfies those constraints is returned from the database.

Performance of star schemas

Performance is an important consideration of any schema, particularly with a decision-support system in which you routinely query large amounts of data. IBM Red Brick Warehouse supports all schema designs. However, star schemas tend to perform the best in decision-support applications.

Terminology

The terms fact table and dimension table represent the roles these objects play in the logical schema. In terms of the physical database, a fact table is a referencing table. That is, it has foreign key references to other tables. A dimension table is a referenced table. That is, it has a primary key that is a foreign key reference from one or more tables.

Simple star schemas

Any table that references or is referenced by another table must have a primary key, which is a column or group of columns whose contents uniquely identify each row. In a simple star schema, the primary key for the fact table consists of one or more foreign keys. A foreign key is a column or group of columns in one table whose values are defined by the primary key in another table. In IBM Red Brick Warehouse, you can use these foreign keys and the primary keys in the tables that they reference to build STAR indexes, which improve data retrieval performance.

When a database is created, the SQL statements used to create the tables must designate the columns that are to form the primary and foreign keys.

The following figure illustrates the relationship of the fact and dimension tables within a simple star schema with a single fact table and three dimension tables. The fact table has a primary key composed of three foreign keys, Key1, Key2, and Key3, each of which is the primary key in a dimension table. Nonkey columns in a fact table are referred to as data columns. In a dimension table, they are referred to as attributes.

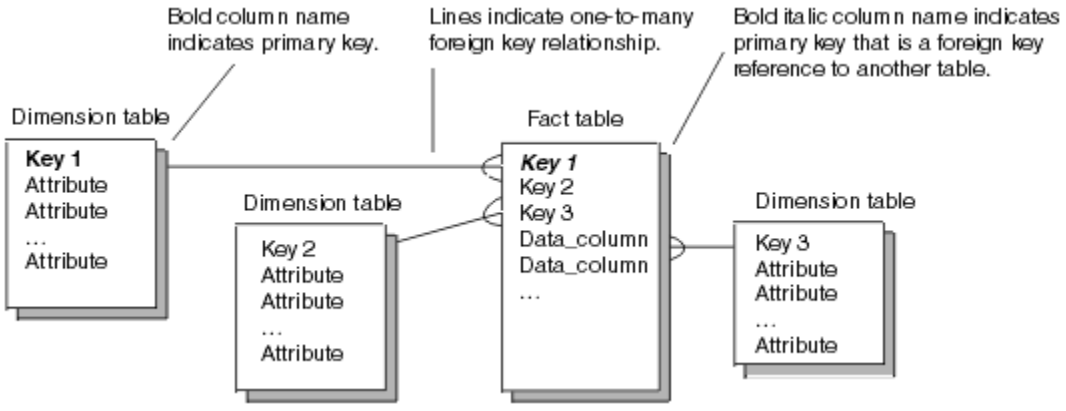


Figure 1 Simple star schema

In the figures used to illustrate schemas:

- The items listed within the box under each table name indicate columns in the table.
- Primary key columns are labeled in bold type.
- Foreign key columns are labeled in italic type.

- Columns that are part of the primary key and are also foreign keys are labeled in bold italic type.
- Foreign key relationships are indicated by lines connecting tables.

Although the primary key value must be unique in each row of a dimension table, that value can occur multiple times in the foreign key in the fact table--a many-to-one relationship.

The following figure 2 illustrates a sales database designed as a simple star schema. In the fact table Sales, the primary key is composed of three foreign keys, Product_id, Period_id, and Market_id, each of which references a primary key in a dimension table.

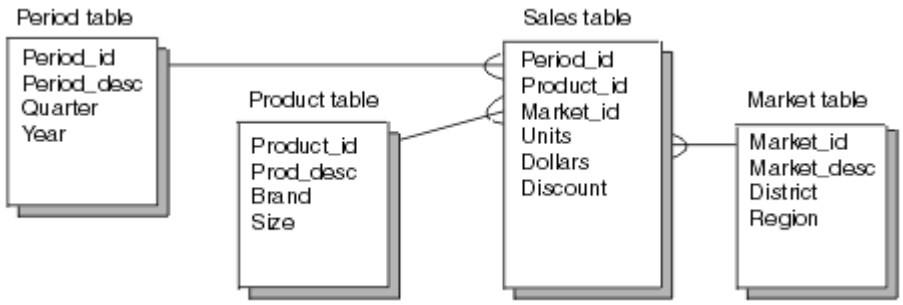


Figure 2

EXCERSICE:

1) Go to following link and perform cube generation in star scheme in SQL server 2000. http://www.databasejournal.com/features/mssql/article.php/1429671/Introduction-to-SQL-Server-2000-Analysis-Services-Creating-Our-First-Cube.htm

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____

EXPERIMENT NO: 10

TITLE: Design & create cube by identifying measures & dimensions for snowflake schema.

OBJECTIVE: On completion of this exercise student will able to know about…

- What is snowflake scheme?
- How to create cube by using snowflake schema?

THEORY:

What is Snowflake schema?

Snowflake schema consists of a fact table surrounded by multiple dimension tables which can be connected to other dimension tables via many-to-one relationship. Snowflake schema is a kind of star schema however it is more complex than a star schema in term of data model. This schema resembles a snowflake therefore it is called snowflake schema.

Snowflake schema is designed from star schema by further normalizing dimension tables to eliminate data redundancy. Therefore in snowflake schema, instead of having big dimension tables connected to a fact table, we have a group of multiple dimension tables. In snowflake schema, dimension tables are normally in the third normal form (3NF). The snowflake schema helps save storage however it increases the number of dimension tables.

Snowflake schema example

The figure below shows an example of snowflake schema that is a snowflaked version of a star schema demonstrated in the star schema article.

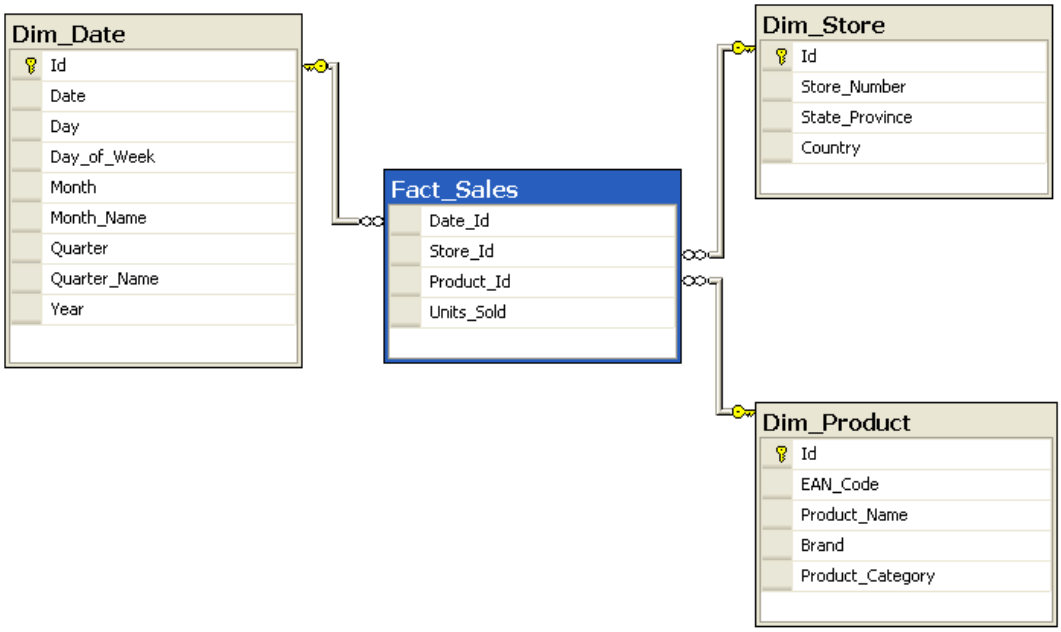


Figure 1 Star Schema Example

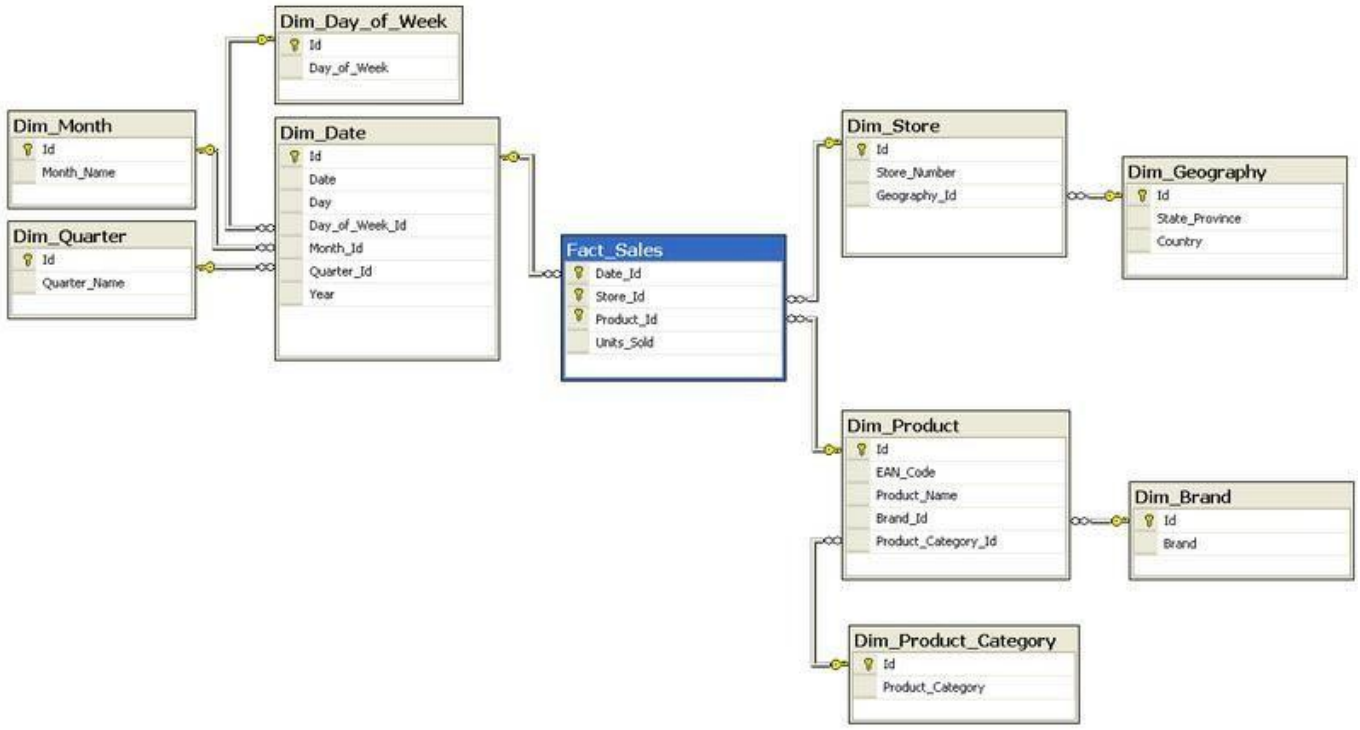


Figure 2 Snowflake Schema Example

Let‘s examine the snowflake schema above in a greater detail:

- DIM_STORE dimension table is normalized to add one more dimension table called DIM_GEOGRAPHY
- DIM_PRODUCT dimension table is normalized to add 2 more dimension tables called DIM_BRAND and DIM_PRODUCT_CATEGORY
- DIM_DATE dimension table is now connecting with three other dimension tables: DIM_DAY_OF_WEEK, DIM_MONTH and DIM_QUARTER.
- Fact table remains the same as star schema.

Useful snowflake schema notes

- The normalization of dimension tables tends to increase number of dimension tables or sub-dimension table that require more foreign key joins when querying the data therefore reduce the query performance.
- The query of snowflake schema is more complex than query of star schema due to multiple joins from dimension table to sub-dimension tables.
- Snowflake schema help to save space by normalizing dimension tables.
- It is more difficult for business users who use data warehouse system using snowflake schema because they have to work with more tables than star schema.

EXCERSICE:

1) Write down the difference between star schema and Snowflake schema.
2) Give the example of star schema and Snowflake schema.
3) Explain the cube generation in snowflake schema using SQL server 2000 take relevant screen shots?

EVALUATION:

Observation & Implementation	Timely completion	Viva	Total
4	**2**	**4**	**10**

Signature: _____

Date: _____