



Saurashtra University

Re – Accredited Grade 'B' by NAAC
(CGPA 2.93)

Atkotiya, Kishorchandra H., 2008, "*Analytical study and computational modeling of statistical methods for data mining*", thesis PhD, Saurashtra University

<http://etheses.saurashtrauniversity.edu/id/eprint/329>

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Saurashtra University Theses Service
<http://etheses.saurashtrauniversity.edu>
repository@sauuni.ernet.in

**ANALYTICAL STUDY AND COMPUTATIONAL
MODELING OF STATISTICAL METHODS
FOR DATA MINING**

**A THESIS IS SUBMITTED TO
SAURASHTRA UNIVERSITY, RAJKOT
FOR THE AWARD OF DOCTOR OF PHILOSOPHY IN
COMPUTER SCIENCE INTERDISCIPLINARY STATISTICS
IN THE FACULTY OF SCIENCE**

**: SUBMITTED BY :
ATKOTIYA KISHORCHANDRA HANSRAJBHAI
J. H. BHALODIA WOMEN'S COLLEGE, RAJKOT**

**UNDER THE GUIDANCE OF
DR. N. N. JANI
PROF. & HEAD,
DEPARTMENT OF COMPTUER SCIENCE,
SAURASHTRA UNIVERSITY, RAJKOT
AND**

**DR. G.C. BHIMANI
ASSOCIATE PROFESSOR,
DEPARTMENT OF BUSINESS MANAGEMENT,
SAURASHTRA UNIVERSITY, RAJKOT**

JUNE – 2008

CERTIFICATE

I hereby certify that **Mr. Atkotiya Kishorchandra Hansrajbhai** has completed his thesis for doctorate degree entitled **“ANALYTICAL STUDY AND COMPUTATIONAL MODELING OF STATISTICAL METHODS FOR DATA MINING”**. I further certify that the research work done by him is of his own and original and is carried out under my guidance and supervision. For the thesis that he is submitting, he has not been conferred any degree, diploma or distinction by either the Saurashtra University or any other University according to best of my knowledge.

Place: Rajkot

Date:

Dr. N. N. Jani
Professor & Head,
Department of Computer Science,
Management,
Saurashtra University,
Rajkot

Dr. G. C. Bhimani
Associate Professor,
Department of Business
Saurashtra University,
Rajkot

CERTIFICATE

I certify that the developed models in this research study are Data Access User interface, Data Extraction, Data Transformation, Data Mining, Data Warehousing and results derived by analysis and described in the thesis has been based on the literature survey, bibliographical references and through study of the web sites in respect of related areas.

Apart from these, all the analysis, hypothesis, inferences and interpretation of data and strategy have been my own and original creation. The model has been prototyped to a domain, which is my own and original creation. Moreover, I declare that the work done in the thesis, either the Saurashtra University or any other university has not conferred any degree, diploma or distinction on me before.

Place: Rajkot

Date:

Atkotiya Kishorchandra Hansrajbhai

To my parents and my wife

ACKNOWLEDGEMENT

I express my profound sense of gratitude to **Dr. N.N. Jani** and **Dr. G.C. Bhimani** my research guides, who provides me undeviating encouragement, indefatigable guidance and valuable suggestions throughout the research study.

I take opportunity to express my deep sense of gratitude to **Dr. K. P. Joshipura**, Vice-Chancellor and **Kalpakhai Trivedi**, Pro-Vice-Chancellor of the Saurashtra University for his consistent encouragement to the research and development.

I express my gratitude to all those officials National Stock Exchange Mumbai, they provided me valuable information and insight into various important issues related to the research study.

I also give my sincere thanks to the Department of Computer Science, Saurashtra University, to provide me platform for the practical study of my research work. I am also thankful to the administrative staff of the department, who has always been a support of inspiration during my entire work.

My deepest thanks to the reviewer of my thesis. I am highly indebted to my parents, my wife and sister, all my relatives and friends who constantly inspired me.

Rajkot

Atkotiya Kishorchandra Hansrajibhai

ABSTRACT

Today, there is tremendous increase of the information available on electronic form. Day by day it is increasing massively. There are enough opportunities for research to retrieve knowledge from the data available in this information. Data mining and applied statistical methods are the appropriate tools to extract knowledge from such data. Although data mining is a very important and growing area, there is insufficient coverage of it in the literature, especially from a statistical viewpoint. Most of the research on data mining are either too technical and computer science oriented or too applied and marketing driven. Aim of this research is to establish a bridge between data mining methods and applications in the fields of business and industry by adopting a coherent and rigorous approach to statistical modeling.

In this research work we suggested various models like, Data Access Interface model, Data Extraction Model, Data Transformation Model and Data Mining model. We also used open source Java based software Weka for the analysis and comparison study of various statistical techniques and algorithms. Combining all these suggested models, computational model is also proposed.

To implement all the models practically, we have use stock market data NSE (National Stock Exchange). Large collection of stock market data being gathered for a implementation of developed applications (tools) based on the suggested models in research study. Data Mining from such a data corpus can lead to interesting results and discoveries of patterns.

LIST OF FIGURES

Sr. No.	Figure No.	Title of Figure	Page No.
1	1.1	Knowledge Discovery in Database (KDD)	32
2	2.1	The data warehouse is subject oriented	41
3	2.2	Data warehouse is integrated	41
4	2.3	OLTP Database and Data Warehouse	44
5	2.4	Three-Tiered Architecture	49
6	2.5	Two-Tiered Architecture	54
7	2.6	MOLAP Architecture	63
8	2.7	ROLAP architecture	64
9	2.8	Extraction and Cleansing of data	68
10	3.1	Phases of data mining process	78
11	3.2	Steps taken in Data Mining	79
12	3.3	Integrated Data Mining Architecture	80
13	3.4	Example of Box Plot Technique	83
14	3.5	Decision Tree Representation	89
15	3.6	Binary Decision Tree with Truth Table	90
16	3.7	Example of Decision Tree	91
17	3.8	Representation of Neural Network	93
18	3.9	Node i	95
19	3.10(a)	Single crossover	98
20	3.10(b)	Multiple crossover	98
21	3.11(a)	Lattice of item sets {A, B, C, D}	101
22	3.11(b)	Subsets of ACD	101
23	3.12	Hierarchy of Association Rule	105
24	3.13(a)	Group of homes (Clustering)	107
25	3.13(b)	Geographic distance based (Clustering)	107
26	3.13(c)	Size – based (Clustering)	108

27	3.14	Classification of Clusters	110
28	3.15(a)	Six clusters	111
29	3.15(b)	Four Clusters	112
30	3.15(c)	Three clusters	112
31	3.15(d)	Two clusters	112
32	3.15(e)	One cluster	113
33	3.16	Dendrogram for Hierarchical Algorithm	113
34	3.17	An Example for hierarchical divisive algorithm	118
35	3.18	An outlier of samples	121
36	4.1	Two-Tiered Application	125
37	4.2	Three-Tiered Application	127
38	4.3	Log in Screen	134
39	4.4	User Screen	134
40	4.5	DBA Screen	134
41	4.6	Table List Screen	135
42	4.7	New Table Screen	135
43	4.8	Fields in New Table	136
44	4.9	Structure of Created Table	136
45	4.10	New Procedure screen	137
46	4.11	Creation of Trigger	137
47	4.12	User Profile Creation	138
48	4.13	Maintaining of Roles	139
49	4.14	Users with Roles and System Privileges	139
50	4.15	Change Password	140
51	5.1	NSE Home Page	160
52	5.2	List of Historical Data	161
53	5.3	Data Extraction Model	162
54	5.4	DateForm Module	163
55	5.5	Range of Date Screen	164
56	5.6	Extracted Data Files	164

57	5.7	Data Transformation Model	169
58	5.8	Data Transformation Tool	170
59	5.9	Transformed Records	171
60	5.10	Data Mining Model	174
61	5.11	Home Page of Data Mining Tool	175
62	5.12	Scripts Bar Chart	178
63	5.13	Bar Chart for Variance	179
64	5.14	Bar Chart of (Open – Close in Plus)	181
65	5.15	Bar Chart of (Open – Close in Minus)	183
66	5.16	Bar Chart for Volume	184
67	5.17	Pie Chart of Variance	187
68	5.18	Consolidated Charts	195
69	5.19	Chart of Scripts	196
70	5.20	Chart of Open rate	196
71	5.21	Chart of High rate	197
72	5.22	Chart of Low rate	197
73	5.23	Chart of Close rate	198
74	5.24	Chart of Last rate	198
75	5.25	Chart of Previous Close rate	199
76	5.26	Chart of Volume	199
77	5.27	Chart of Trading Value	200
78	5.28	Chart of Number of Scripts	200
79	5.29	Chart of Open Rate and High Rate	202
80	5.30	Chart of Low Rate and Close Rate	217
81	5.31	Comparison of Algorithms	245
82	5.32	Comparison of Various Tools	254
83	6.1	Complete Model of Research	261

ANALYTICAL STUDY AND COMPUTATIONAL MODELING OF STATISTICAL METHODS FOR DATA MINING CONTENT

Acknowledgement	
Abstract	
List of Figures	
1. Research Survey and Introduction	17
1.1 Selection of Research Title	
1.2 Survey of the research	
1.3 Research Motivation	
1.4 General objective of the research	
1.5 Data Mining – An Introduction	
1.6 Data Mining – An overview	
1.7 Data Mining and Statistics	
1.8 Organization of the data	
1.9 Data Warehouse	
1.10 Data Webhouse	
1.11 Data Marts	
1.12. Classification of the data	
1.13 Overview of KDD	
1.14 KDD Process Definitions	
1.15 Reason for growth of data mining research	
2. Introduction to Data Warehouse	37
2.1 Types of Systems	

2.2 Difference of Operational and Informational system
2.3 OLTP and DSS Systems
2.4 Introduction to Data Warehouse
2.4.1 Subject-Oriented
2.4.2 Integrated
2.4.3 Time-Variant Data
2.4.4 Nonvolatile Data
2.4.5 Data Granularity
2.5 Data Warehouse S/W and H/W Architecture
2.6 Basic steps to develop DW Architecture
2.7 Architectural Components of D/W (Infrastructure)
2.8 D/W System Architecture
2.8.1 Three-Tiered Architecture
2.8.1.1 Benefits of Three-Tiered Architecture
2.8.1.2 Drawbacks of Three-tiered Architecture
2.8.2 Two-Tiered Architecture
2.9 Data Marts
2.9.1 Data Mart Structure
2.9.2 Usage of Data Mart

2.9.3 Security in a Data Mart
2.10 Data warehouse and Data Mart
2.11 OLAP
2.11.1 OLTP and OLAP Systems
2.11.2 Types of OLAP
2.11.2.1 MOLAP
2.11.2.1.1 Advantages
2.11.2.1.2 Disadvantages
2.11.2.2 ROLAP
2.11.2.2.1 Advantages
2.11.2.2.2 Disadvantages
2.11.2.3 HOLAP
2.12 ETL (Extraction, Transformation, Loading)
2.12.1 Extraction
2.12.2 Transformation
2.12.3 Loading
2.12.4 Data
2.12.5 Pipeline
2.12.6 Component
2.12.7 Popular ETL Tools

2.13 Metadata	
2.13.1 Metadata Architecture	
3. Statistical Techniques Driven Data Mining Process	74
3.1 Foundation of Data Mining	
3.2 Data Mining Process	
3.2.1 Data Understanding	
3.2.2 Data Understanding	
3.2.3 Data Preparation	
3.2.4 Creation of database for data mining	
3.2.5 Exploring the database	
3.2.6 Preparation for creating a data mining model	
3.2.7 Building a data mining model	
3.2.8 Evaluation of data mining model	
3.2.9 Deployment of the data mining model	
3.3 Architecture for Data Mining	
3.4 Data Mining Techniques	
3.4.1 Statistics	

3.4.1.1 Point Estimation
3.4.1.2 Model Based on Summarization
3.4.1.3 Bayes Theorem
3.4.1.4 Hypothesis Testing
3.4.1.5 Regression and Correlation
3.4.2 Machine Learning
3.4.3 Decision Trees
3.4.3.1 Decision Tree Representation
3.4.3.2 Binary Decision Tree
3.4.3.3 Solution of problem using Decision Tree
3.4.4 Neural Networks
3.4.4.1 Single Layer Linear Network
3.4.5 Genetic Algorithm
3.4.5.1 Starting set of individuals, P
3.4.5.2 Crossover technique
3.4.5.3 Mutation algorithm
3.4.5.4 Fitness function

3.4.6. Association Rules
3.4.6.1 Basic Algorithms for Association Rules
3.4.6.1.1 Apriori Algorithm
3.4.6.1.2 Sampling Algorithm
3.4.6.1.3 Partitioning
3.4.6.1.4 Pincer-Search Algorithm
3.4.6.1.5 FP-Tree Growth Algorithm
3.4.6.1.6 Advance Association Rule Techniques
3.4.7 Clustering
3.4.7.1 Hierarchical Algorithms
3.4.7.2 Agglomerative Algorithm
3.4.7.3 Single-Linkage Agglomerative Algorithm:
3.4.7.4 Complete-Linkage Agglomerative Algorithm
3.4.7.5 Average-Linkage Agglomerative Algorithm

3.4.7.6 Divisive Clustering	
3.4.7.7 Partitional Algorithm	
3.4.7.8 K – Means Clustering	
3.4.7.9 Nearest Neighbor Algorithm	
3.4.7.10 Clustering of large database	
3.4.7.11 BIRCH	
3.4.7.12 DBSCAN	
3.4.7.13 CURE	
4. Implementation of Data Access User Interface (Web Based)	125
4.1 One Tier Architecture	
4.2 Two Tier Architecture	
4.3 Three Tier Architecture	
4.3.1. Presentation Tier	
4.3.2 Logic Tier (Business Logic Tier and Data access Tier)	
4.3.3 Data Tier	
4.4 WDAUI Architecture	
4.4.1 Web Browser	

4.4.2 Internet Information Services
4.4.2.1 IIS Installation
4.4.3 HTML (Hypertext Markup Language)
4.4.3.1 CSS (Cascading Style Sheet)
4.4.3.2 JavaScript
4.4.3.3. AJAX
4.4.3.4 ASP (Active Server Pages)
4.4.4 Oracle Database
4.5 RDBMS access using Web Interface
4.5.1 Introduction of Web Interface
4.5.2 Security in Web Interface
4.5.3 How Web Interface works
4.5.4 Coding
4.5.4.1 CSS Code
4.5.4.2 Log In Check Code
4.5.4.3 User Facility Code
4.5.4.4 Creating New Table Code
4.5.4.5 Addition Foreign Key Code
4.6 Limitations of WDAUI

5. Computational Modeling (ETL) and Analytical Study	156
5.1 Overview	
5.2 Extraction of Data (Data Extraction)	
5.2.1 National Stock Exchange (NSE) Organization	
5.2.2 Technology using in NSE	
5.2.3 NSE on Web	
5.2.4 Data Extraction model	
5.2.5 Process of Data Extraction	
5.2.6 Implementation of Data Extraction Tool	
5.2.6 Implementation of Data Extraction Tool	
5.2.6.1 Code of dateform.php file	
5.2.6.2 Code of copyfile.php file	
5.2.7 Advantages of Data Extraction Tool	
5.2.8 Limitations of Data Extraction Tool	
5.3 Data Transformation	
5.3 Data Transformation	
5.3.1 Data Transformation Model	
5.3.2. Process of Data Transformation	

5.3.3 Implementation of Data Transformation Tool	
5.3.3.1 Code of Data Transformation Tool	
5.3.4 Advantages of Data Transformation Tool	
5.4 Data Mining	
5.4.1 Data Mining Model	
5.4.2 Process of Data Mining	
5.4.3 Implementation of Data Mining Tool	
5.5 WEKA Software (Data Mining Software in Java)	
5.5.1 Implementation of Data Set into Weka	
5.5.2 Rules generated using chart	
5.5.3 Analysis of Data using Weka	
5.5.4 Comparison of Various Algorithms	
5.6 Data Mining and Data Warehousing Tools	
5.10.1 SQnL Server 2005 Data Mining Features	
5.6.1 SQL Server 2005 Data Mining Features	
5.6.2 DB2 Intelligent Miner	
5.6.3 Informatica PowerCenter Advanced Edition	
5.6.4 Business Object	
5.6.5 Cognos 8 Business Intelligence	
5.6.6 Comparison of Data Mining Tools	
6. Summary, Conclusion and Future Work	255

6.1 Summary of work
6.2 Conclusion
6.3 Future Work
5.8.1 Selection of different techniques
6.3.1 Selection of different techniques
6.3.2 Managing changing data
6.3.3 Non-standard data types
6.4 Extension work in Research
6.4.1 Web-Based Data Access User Interface (WBDAUI) Model
6.4.2 Data Extraction Model
6.4.3 Data Transformation Model
6.4.4 Data Mining Model
6.4.5 Weka Environment
6.5 Bibliography

1. Research Survey and Introduction

1.1 Selection of Research Title

The increasing availability of data in the current information society has led to the need for valid tools for its modeling and analysis. Data mining and applied statistical methods are the appropriate tools to extract knowledge from such data. Data mining can be defined as the process of selection, exploration and modeling of large databases in order to discover model and pattern that are unknown a priori. It differs from applied statistics mainly in terms of its scope; whereas applied statistics concerns the application of statistical methods to the data at hand, data mining is a whole process of data extraction and analysis aimed at the production of decision rules for specified business goals. In other words, data mining is a business intelligence process.

Although data mining is a very important and growing topic, there is insufficient coverage of it in the literature, especially from a statistical viewpoint. Most of the research on data mining are either too technical and computer science oriented or too applied and marketing driven. Our research aims to establish a bridge between data mining methods and applications in the fields of business and industry by adopting a coherent and rigorous approach to statistical modeling.

Nowadays, each individual and organization – business, family or institution can access a large quantity of data and information about itself and its environment. This data has the potential to predict the evolution of interesting variables or trends in the outside environment, but so far that potential has not been fully exploited. This is particularly true in the business field. There are two main problems, information is scattered within different archive systems that are not connected with one another, producing an inefficient organization of the data. There is a lack of awareness about statistical tools and their potential for information elaboration. This interferes with the production of efficient and relevant data synthesis.

Two developments could help to overcome these problems. First, software and hardware continually, offer more power at lower cost, allowing organization to

collect and organize data in structures that give easier access and transfer. Second, methodologically research, particularly in the field of computing and statistics, has recently led to the development of flexible and scalable procedures that can be used to analyse large data stores. These two developments have meant that data mining is rapidly spreading through many businesses as an important intelligence tool for backing up decisions.

1.2 Survey of the research

As recently as before few years ago, data mining was a new concept for many people and organizations. Data mining products were also new and marred by unpolished interfaces. Only the most innovative or daring early adopters was trying to apply these emerging tools. Now scenario has been changed, today data mining products have matured, and data mining is accessible to a much wider audience. We are even seeing the emergence of specialized vertical market data mining products.

What kinds of organization can use data mining technology to solve their business problem and how users understand to apply these tools effectively to get benefit that is very much important. Data mining extracts hidden information, knowledge and pattern from large volume of data that was previously not available. Data mining tools do more than query and data analysis tools.

Simple query and analysis tools can respond to questions such as, "Do sales of Product A increase in January?" or "Do sales of Product A decrease when there is a promotion on Product B?" In contrast to this, a data mining tool can be asked, "What are the factors that determine sales of Product A?"

Using traditional tools, the analyst starts with a question and assumption, or perhaps just a hunch and explores the data and builds a model, step-by-step, working to prove or disprove a theory. In traditional approach the analyst propose all hypotheses, tests these proposed hypotheses. If requires, then also propose an additional or substitute hypothesis, test it also, and so on, and in this iterative way, a model can be built. Responsibility of an analyst does not disappear entirely with data mining; data mining shifts much of the work of

finding an appropriate model from the analyst to the computer. This has the following potential benefits.

- A model can be generated with requires less manual effort. (It can be more efficient.)
- Larger number of models can be evaluated; it increases the odds of finding a better model.
- The analyst needs less technical expertise because most of the step-by-step procedure is automated.

There is extensive use of statistics in data mining. So nowadays data mining is embarked with statistical techniques is also known as statistical data mining or computational data mining.

1.3 Research Motivation

In modern age all reputed and well-suited organizations are under enormous pressure to respond quickly for continuous changing trends in the competitive market. So it is obvious that in order to achieve this challenge, they need rapid access to different varieties of information before a decision can be framed. For making the right choices for organization, it is very much essential to study and analyze the past data and identify most relevant trends. Before perform any trend analysis it is very much necessary to access to entire relevant information, and naturally this information is mainly stored in very large/huge databases. To build a data warehouse is an easiest approach to gain access large volume of data for extract patterns and effective decision making for business environment. Data warehouse stores historical data and operational data. Operational data is useful to access information and generate some patterns using data mining techniques. These generated patterns can be analyzed to organize and summarize as business intelligence. Today it is business intelligence with quality information at backbone can give the power of with standing and growth.

1.4 General objective of the research

The objective of this research is to do analytical study and computational modeling of statistical methods useful for data mining. How various statistical methods are useful for data mining and how competitive but less expensive computational model can be built? Data mining tools available in market are very expensive and too complex. I want to suggest a model for data mining that built with appropriate statistical methods. This has created a special interested in making comparison of different algorithms of data mining with effort to develop experimental paradigms that allow testing the mining algorithms.

For the different problems and applications, data mining strategies do not follow the same track and gives different pictures in different situations. The variability may cause the implementation complex and success rate not to the anticipated level. This problem is the main reason for failure data mining applications in most of organizations. To smoothen the track, efforts are made to propose the model that keeps the track of data mining and that is expected to cover many different situations.

Today the numbers of organizations are rapidly increasing which are using data mining applications for business intelligence. The model is to be proposed under this research initiative will help the developers for data mining solutions.

1.5 Data Mining – An Introduction

To understand the term ‘Data Mining’ it is useful to look at the literal translation of the word: to mine in English means to extract. The verb usually refers to mining operations that extract from the Earth her hidden, precious resources. The association of this world with data suggests an in-depth search to find additional information, which previously went unnoticed in the mass of data variables. From the viewpoint of scientific research, data mining is relatively new discipline that has developed mainly from studies carried out in other disciplines such as computing, marketing management and statistics. Many of the methodologies used in data mining come from two branches of research, one developed in the

machine learning community and the other developed in the statistical community, particularly in multivariate and computational statistics.

Machine learning is connected to computer science and artificial intelligence and is concerned with finding relations and regularities in data that can be translated into general truths. The aim of machine learning is the reproduction of the data-generating process, allowing analysts to generalize from the observed data to new, unobserved cases. Rosenblatt (1962) introduced the first machine-learning model, called the perception. Following on from this, neural networks developed in the second half of the 1980s. During the same period, some researchers perfected the theory of decision trees used mainly for dealing with problems of classification. Statistics has always been about creating models for analyzing data, and now there is the possibility of using computers to do it. From the second half of the 1980s; given the increasing importance of computational models as the basis for statistical analysis, there was also a parallel development of statistical methods to analyze real multivariate applications. In the 1990s statisticians began showing interest in machine learning methods as well, which led to important developments in methodology.

Towards the end of the 1980s machine learning methods started to be used beyond the fields of computing and artificial intelligence. In particular, they were used in database marketing applications where the available databases were used for elaborate and specific marketing campaigns. The term knowledge discovery in databases (KDD) was coined to describe all those methods that aimed to find relations and regularity among the observed data. Gradually the term KDD was expanded to describe the whole process of extrapolating information from a database, from the identification of the initial business aims to the application of the decision rules. The term 'data mining' was used to describe the component of the KDD process where the learning algorithms were applied to the data.

This terminology was first formally put forward by Usama Fayyad at the First International Conference on Knowledge Discovery and Data Mining held in Montreal in 1995 and still considered one of the main conferences on this topic.

It was used to refer to a set of integrated analytical technique divided into several phases with the aim of extrapolation previously unknown knowledge from massive sets of observed data that do not appear to have any obvious regularity or important relationships. As the term 'data mining' slowly established itself, it became a synonym for the whole process of extrapolating knowledge. The previous definition omits one important aspect – the ultimate aim of data mining. In data mining the aim is to obtain results that can be measured in terms of their relevance for the owner of the database – business advantage. Here is a more complete definition of data mining:

Data Mining is the process of selection, exploration, and modeling of large quantities of data to discover regularities or relations that are at first unknown with the aim of obtaining clear and useful results for the owner of the database.

To apply data mining methodology means following an integrated methodological process that involves translating the business needs into a problem, which has to be analyzed, retrieving the database needed to carry out the analysis, and applying a statistical technique implemented in a computer algorithm with the final aim of supportive important results useful for taking a strategic decision. The business needs, setting off what has been called 'the virtuous circle of knowledge' introduced by data mining (Berry and Linoff, 1997).

Data mining is not just about the use of a computer algorithm or a statistical technique: it is a process of deriving business intelligence that can be used together with what is provided by information technology to support business decision.

1.6 Data Mining – An overview

The emergence of data mining is closely connected to developments in computer technology, particularly the evolution and organization of databases, which have recently made great leaps forward. Some new terms are clarified as below.

Query and reporting tools are simple and very quick to use: they help to explore business data at various levels. Query tools retrieve the information and reporting tools present it meaningfully. They allow the results of analysis to be transmitted across a client-server network, intranet or even on the internet. The networks allow sharing, so that the data can be analyzed by the most suitable platform. This makes it possible to exploit the analytical potential of remote servers and receive an analysis report on local PCs. A client-server network must be flexible enough to satisfy varied types of remote requests, from a simple reordering of data to ad hoc queries using Structured Query Language (SQL) for extracting and summarizing data in the database.

Data retrieval, like data mining, extracts interesting data and information from archives and databases. The difference is that, unlike data mining, the criteria for extracting information are decided beforehand so they are exogenous from the extraction itself. A classic example is a request from the marketing department of a company to retrieve all the personal details of clients who have bought product A and product B at least once in that order. This request may be based on the idea that there is some connection between having bought A and B together at least once but without any empirical evidence. The names obtained from this exploration could then be the targets of the next publicity campaign. In this way the success percentage (i.e. the customers who will actually buy the products advertised compared to the total customers contacted) will definitely be much higher than otherwise. Once again, without a preliminary statistical analysis of the data, it is difficult to predict the success percentage and it is impossible to establish whether having better information about the customers' characteristics would give improved results with a smaller campaign effort.

Data mining is different from data retrieval because it looks for relations and association between phenomena that are known beforehand. It also allows the effectiveness of a decision to be judged on the data, which allows a relational evaluation to be made, and on the objective data available. It is not to be confused with data mining methods used to create multidimensional reporting tools. Online Analytical Processing (OLAP). OLAP is usually a graphical

instrument used to highlight relations between the variables available following the logic of a two dimensional report. OLAP is an important tool for business intelligence. The query and reporting tools describe what a database contains, but OLAP is used to explain why certain relation exists.

OLAP is not a substitute for data mining: the two techniques are complementary and used together they can create useful strategies. OLAP can be used in the processing stages of data mining. This makes understanding the data easier, because it becomes possible to focus on the relevant data, identifying special cases or looking for principal interrelations. The final data mining results, expressed using specific summary variables, can be easily represented in an OLAP hypercube.

Following is the simple sequence that shows the evolution of business intelligence tools used to extrapolate knowledge from a database:

QUERY AND REPORTING DATA RETRIEVEL OLAP DATA MINING

Above sequence indicates that Query and Reporting has the lowest information capacity and Data Mining has highest information capacity. This suggests a trade-off between information capacity and ease of implementation. Lack of information is one of the greatest obstacles to achieving efficient data mining.

The creation of a data warehouse can eliminate many of these problems. Efficient organization of the data in a data warehouse coupled with efficient and scalable data mining allows the data to be used correctly and efficiently to support business decision.

1.7 Data Mining and Statistics

Statistics has always been about creating methods to analyze data. The main difference between statistical methods and machine learning methods is that statistical methods are usually developed in relation to the data being analyzed but also according to a conceptual reference paradigm. Although this has made the statistical methods coherent and rigorous, it has also limited their ability to adapt quickly to the new methodologies arising from new information technology

and new machine learning applications. Statisticians have recently shown an interest in data mining and this could help its development.

For a long time statisticians saw data mining as a synonymous with 'data fishing', 'data dredging', or 'data snooping'. In all these cases data mining had negative connotations. This idea came about because of two main criticisms. First, there is not just one theoretical reference model but also several models in competition with each other; these models are chosen depending on the data being examined. The criticism of this procedure is that it is always possible to find a model, however complex, which will adapt well to the data. Second, the great amount of data available may lead to non-existent relation being found among the data.

Although these criticisms are worth considering, we shall see that the modern methods of data mining pay great attention to the possibility of generalizing results. This means that when choosing a model, the predictive performance is considered and the more complex models are penalized. It is difficult to ignore the fact that many important findings are not known beforehand and cannot be used in developing a research hypothesis. This happens in particular when there are large databases.

This last aspect is one of the characteristics that distinguished data mining from statistical analysis. Whereas statistical analysis traditionally concerns itself with analyzing primary data that has been collected to check specific research hypotheses, data mining can also concern itself with secondary data collected for other reasons. This is the norm, for example, when analyzing company data that comes from a data warehouse. Furthermore, statistical data can be experimental data, but in data mining the data is typically observational data.

Berry and Linoff (1997) distinguish two analytical approaches to data mining. They differentiate top-down analysis (confirmative) and bottom-up analysis (explorative). Top-down analysis aims to confirm or reject hypothesis and tries to widen our knowledge of a partially understood phenomenon; it achieves this principally by using the traditional statistical methods. Bottom-up analysis is where the user looks for useful information previously unnoticed, searching

through the data and looking for ways of connecting it to create hypotheses. The bottom-up approach is typical of data mining. In reality the two approaches are complementary. In fact, the information obtained from a bottom-up analysis, which identifies important relations and tendencies, cannot explain why these discoveries are useful and to what extent they are valid. The confirmative tools of top-down analysis can be used to confirm the discoveries and evaluate the quality of decision based on those discoveries.

There are at least three other aspects that distinguish statistical data analysis from data mining. First, data mining analyses great masses of data. This implies new consideration for statistical analysis. For many applications it is impossible to analyze or even access the whole database for reasons of computer efficiency. Therefore it becomes necessary to have a sample of the data from the database being examined. This sampling must take account of the data mining aims, so it cannot be performed using traditional statistical theory. Second many databases do not lead to the classic forms of statistical organization, for example data that comes from the internet. This creates a need for appropriate analytical methods from outside the field of statistics. Third, data mining results must be of some consequence. This means that constant attention must be given to business results achieved with the data analysis models.

So it can be concluded that there are reasons for believed that the data mining is nothing new from statistical viewpoint. But there are also reasons to support the idea that, because of their nature, statistical methods should be able to study and formalize the methods used in data mining. This means that on one hand we need to look at the problems posed by data mining from a viewpoint of statistics and utility, while on the other hand we need to develop a conceptual paradigm that allows the statisticians to lead the data mining methods back to a scheme of general and coherent analysis.

1.8 Organization of the data

Data analysis requires that the data is organized into an ordered database. The data is analyzed and it depends greatly on how the data is organized within the database. In information society there is an abundance of data and a growing need for an efficient way of analyzing it. However, an efficient analysis presupposes a valid organization of the data.

It has become strategic for all medium and large organizations to have unified information system called a data warehouse; this integrates, for example, the accounting data with data arising from the production process, the contacts with the suppliers(supply chain management), and the sales trends and the contacts with the customers(customer relationship management). Another example is the increasing diffusion of electronic trade and commerce and, consequently, the abundance of data about websites visited along with any payment transactions. In this case it is essential for the services supplier, through the internet, to understand who the customers are in order to plan offers. This can be done if the transactions, which correspond to clicks on the web, are transferred to an ordered database, usually called a webhouse.

1.9 Data Warehouse

The data warehouse can be defined as 'An integrated collection of data about a collection of subjects of subjects (units), which is not volatile in time and can support decision taken by the management'.

From this definition, the first characteristic of a data warehouse is the orientation to the subjects. This means that data in a data warehouse should be divided according to subjects rather than by business. For example, in the case of an insurance company the data put into the data warehouse should probably be divided into Customer, Policy and Insurance Premium rather than into Civil Responsibility, Life and Accident. The second characteristic is data integration, and it is certainly most important. The data warehouse must be able to integrate itself perfectly with the multitude of standard used by the different application from which data is collected. For example, various operational business

applications could codify the sex of the customer in different way and the data warehouse must be able to recognize these standards unequivocally before going on to store the information.

Third, a data warehouse can vary in time since the temporal length of a data warehouse usually oscillates between 5 to 10 years; during this period the data collected is no more than a sophisticated series of instant photos taken at specific moments in time. At the same time, the data warehouse is not volatile because data is added rather than updated. In other words, the set of photos will not change each time the data is updated but it will simply be integrated with a new photo. Finally, a data warehouse must produce information that is relevant for management decision.

This means a data warehouse is like a container of all the data needed to carry out business intelligence operations. It is the main difference between a data warehouse and other business databases. The data contained in the operational databases is used to carry out relevant statistical analysis for the business (related to various management decisions) is almost impossible. On the other hand, a data warehouse is built with this specific aim in mind.

There are two ways to approach the creation of the data warehouse. The first is based on the creation of a single centralized archive that collects all the business information and integrates it with information coming from outside. The second approach brings together different thematic databases, called data marts, that are initially not connected among themselves, but which can evolve to create a perfectly interconnected structure. The first approach allows the system administrators to constantly control the quality of the data introduced. But it requires careful programming to allow for future expansion to receive new data and to connect to other databases. The second approach is initially easier to implement and is therefore the most popular solution at the moment. Problems arise when the various data marts are connected among each other, as it becomes necessary to make a real effort to define, clean and transform the data to obtain a sufficiently uniform level. That is until it becomes a data warehouse in the real sense of the word.

In a system that aims to preserve and distribute data, it is also necessary to include information about the organization of the data itself. This idea is called metadata and it can be used to increase the security level inside the data warehouse. Although it may be desirable to allow vast access to information, some specific data marts and some details might require limited access. Metadata is also essential for management, organization and the exploitation of the various activities. For an analyst it may be very useful to know how the profit variable was calculated, whether the sales areas were divided differently before a certain date, and how a multi-period event was split in time. The metadata therefore helps to increase the value of the information present in the data warehouse because it becomes more reliable.

Another important component of a data warehouse system is a collection of data marts. A data mart is a thematic database, usually represent in a very simple form that is specialized according to specific objectives.

1.10 Data Webhouse

The data webhouse developed rapidly during the 1990s, when it was very successful and accumulated widespread use. The advent of the web with its revolutionary impact has forced the data warehouse to adapt to new requirements. In this new era the data warehouse becomes a web data warehouse or, more simply, data webhouse. The web offers an immense source data about people who use their browser to interact to websites. Despite the fact that most of the data related to the flow of users is very coarse and very simple, it gives detailed information about how internet users surf the net. This huge and undisciplined source can be transferred to the data webhouse, where it can be put together with more conventional sources of data that previously formed the data warehouse.

Another change concerns the way in which the data warehouse can be accessed. It is now possible to exploit all the interfaces of the business data warehouse that already exist through the web just by using the browser. With this it is possible to carry out various operations, from simple data entry to ad hoc queries through the web. In this way the data warehouse becomes completely

distributed. Speed is a fundamental requirement in the design of a webhouse. However, in the data warehouse environment some requests need a long time before they will be satisfied. Slow time processing is intolerable in an environment based on the web. A webhouse must be quickly reachable at any moment and any interruption, however brief, must be avoided.

1.11 Data marts

A data mart is a thematic database that was originally oriented towards the marketing field. Indeed, its name is a contraction of marketing database. In this sense it can be considered a business archive that contains all the information connected to new and/or potential customers. In other words, it refers to a database that is completely oriented to managing customer relations. As we shall see, the analysis of customer relationship management data is probably the main field where data mining can be applied. In general, it is possible to extract from a data warehouse as many data marts as there are aims we want to archive in business intelligence analysis. However, a data mart can be created, although with some difficulty, even when there is no integrated warehouse system. The creation of thematic data structures like data marts represents the first and fundamental move towards an informative environment for the data mining activity

1.12 Classification of the data

A data mart should be organized according to two principles: the statistical units, the elements in the reference population that are considered important for the aims of the analysis (e.g. the supply companies, the customers, the people who visit the site) and the statistical unit (e.g. the amounts customers buy, the payment methods they use, the socio-demographic profile of each customer).

The statistical units can refer to the whole reference population (e.g. All the customers of the company) or they can be a sample selected represent the whole population. There is a large body of work on the statistical theory of sampling and sampling strategies. If we consider an adequately representative sample rather than a whole population, there are several advantages. It might be

expensive to collect complete information about the entire population and the analysis of great masses of data could waste a lot of time in analysing and interpreting the results.

The statistical variables are the main source of information to work on in order to extract conclusions about the observed units and eventually to extend these conclusions to a wider population. It is good to have a large number of variables to achieve these aims, but there are two main limits to having an excessively large number. First of all, for efficient and stable analyses the variables should not duplicate information. For example, the presence of the customers' annual income makes monthly income superfluous. Furthermore, for each statistical unit the data should be correct for all the variables considered. This is difficult when there are many variables, because some data can go missing: missing data causes problems for the analysis.

Once the units and the interest variables in the statistical analysis of the data have been established, each observation is related to a statistical unit, and a distinct value (level) for each variable is assigned. This process is known as classification. In general it leads to two different types of variable: qualitative and quantitative. Qualitative variables are typically expressed as an adjectival phrase, so they are classified into levels, sometimes known as categories. Some examples of qualitative variables are sex, postal code and brand preference. Qualitative data is nominal if it appears in different categories have an order that is either explicit or implicit.

The measurement at a nominal level allows us to establish a relation of equality or inequality between the different levels ($=$, \neq). Examples of nominal measurements are the eye colour of a person and legal status of a company. Ordinal measurements allow us to establish an order relation between the different categories but they do not allow any significant numeric assertion (or metric) on the difference between categories. More precisely, we can affirm which category is bigger or better but we cannot say by how much ($=$, $>$, $<$). Examples of ordinal measurements are computing skills of a person and the credit rate of a company.

Quantitative variables are linked to intrinsically numerical quantities, such as age and income. It is possible to establish connections and numerical relations among their levels. They can be divided into discrete quantitative variables when they have a finite number of levels, and continuous quantitative variables if the levels cannot be counted. A discrete quantitative variable is the annual revenues of a company.

Very often the ordinal level of a qualitative variable is marked with a number. This does not transform the qualitative variable into a quantitative variable, so it is not possible to establish connections and relations between the levels themselves.

1.13 Overview of KDD

The term Knowledge Discovery in Databases abbreviated as KDD refers to the broad process of extracts knowledge from huge data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization. The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

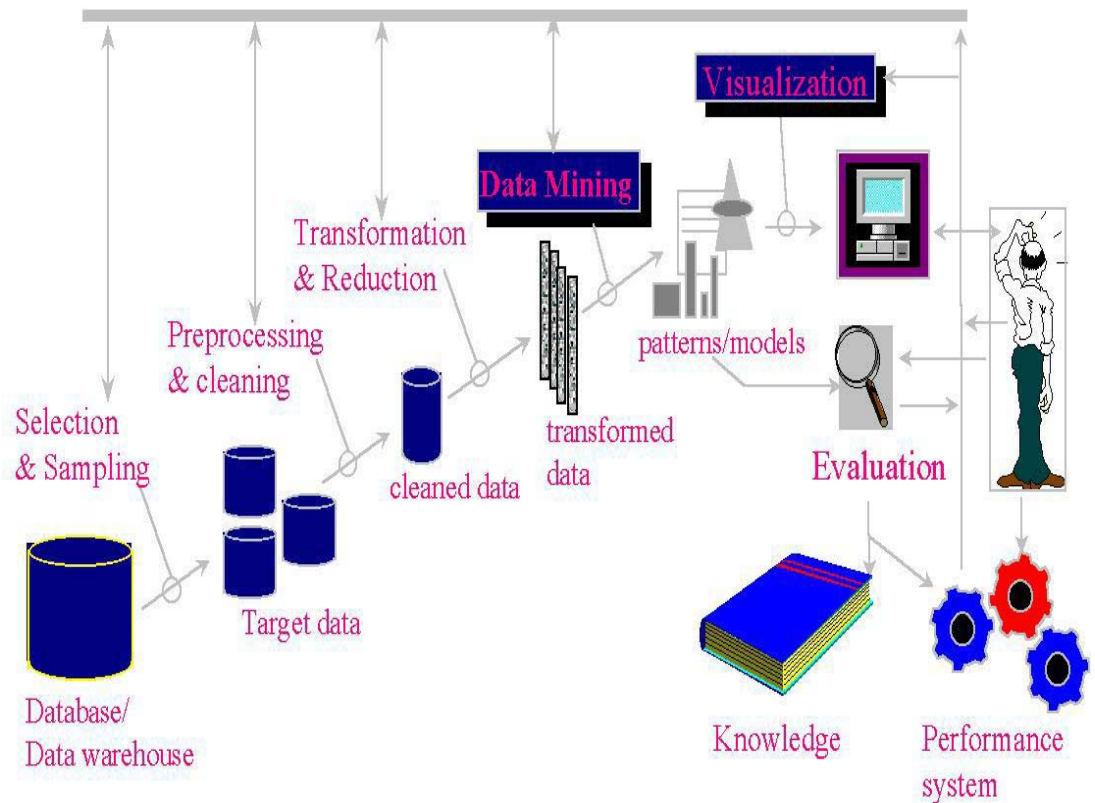


Fig 1.1:KDD

Reference:

Fayyad, Piatetsky-Shapiro, Smyth, "From Data Mining to Knowledge Discovery: An Overview", in Fayyad, Piatetsky-Shapiro, Smyth, Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, Menlo Park, CA, 1996, pp.1-34

The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. Developing an understanding of

- the application domain
- the relevant prior knowledge
- the goals of the end-user

2. Creating a target data set: selecting a data set, or focusing on subset of variables, or data samples, on which discovery is to be performed.

3. Data cleaning and preprocessing.

- Removal of noise or outliers.
- Collecting necessary information to model or account for noise.
- Strategies for handling missing data fields.
- Accounting for time sequence information and known changes.

4. Data reduction and projection.

- Finding useful features to represent the data depending on the goal of the task.
- Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations of the data.

5. Choosing the data mining task.

- Deciding whether the goal of the KDD process is classification, regression, clustering etc.

6. Choosing the data mining algorithm(s).

- Selecting method(s) to be used for searching for patterns in the data.
- Deciding which models and parameters may be appropriate.
- Matching a particular data mining method with the overall criteria of the KDD process.

7. Data mining.

- Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.

8. Interpreting mined patterns.

9. Consolidating discovered knowledge

1.14 KDD Process Definitions

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

Data:

Set of facts, F .

Pattern:

An expression E in a language L describing facts in a subset FE of F .

Process:

KDD is a multi-step process involving data preparation, pattern searching, knowledge evaluation, and refinement with iteration after modification.

Valid:

Discovered patterns should be true on new data with some degree of certainty. Generalize to the future (other data).

Novel:

Patterns must be novel (should not be previously known).

Useful:

Actionable; patterns should potentially lead to some useful actions.

Understandable:

Patterns must be made understandable in order to facilitate a better understanding of the underlying data.

1.15 Reason for growth of data mining research

The amount of digital data has been exploding during the past decade, while the number of scientist, engineers and analysts available to analyze the data has been static. To bridge this gap requires the solution of fundamentally new research problems, which can be grouped into the following broad challenges:

- (a) Developing algorithms and systems to mine large, massive and high
- (b) Dimensional data sets
- (c) Developing algorithms and systems to mine new types of data
- (d) Developing algorithms, protocols and other infrastructure to mine distributed data
- (e) Improving the ease of use of data mining systems
- (f) Developing appropriate privacy and security models for data mining

In order to respond to those challenges, there is requirement of applied, multidisciplinary and interdisciplinary research in data mining and knowledge discovery.

1.16 Contributions

The major contributions of this thesis are summarized as under:

- Proposal of a framework and model, for hierarchical organization and management of Data for resource and implicit knowledge discovery.
- Presentation of strategies for mediating between different data views.
- Proposal of automatic transferring the different OLTP data to Data Warehouse.
- Proposal of architecture for a data mining and OLAP system from Data Warehousing cubes.
- Comparisons of different statistical techniques used for Data Mining.
- Comparisons of different Data Mining Algorithms and their uses.

2. Introduction to Data Warehouse

2.1 Types of Systems

Perhaps the most important concept that has come out of the Data Warehouse movement is the recognition that there are two fundamentally different types of information systems in all organizations: operational systems and informational systems.

"Operational systems" are just what their name implies; they are the systems that help us run the enterprise operation day-to-day. These are the backbone systems of any enterprise, our "order entry", "inventory", "manufacturing", "payroll" and "accounting" systems. Because of their importance to the organization, operational systems were almost always the first parts of the enterprise to be computerized. Over the years, these operational systems have been extended and rewritten, enhanced and maintained to the point that they are completely integrated into the organization. Indeed, most large organizations around the world today couldn't operate without their operational systems and the data that these systems maintain.

On the other hand, there are other functions that go on within the enterprise that have to do with planning, forecasting and managing the organization. These functions are also critical to the survival of the organization, especially in our current fast-paced world. Functions like "marketing, planning", "engineering planning" and "financial analysis" also require information systems to support them. But these functions are different from operational ones, and the types of systems and information required are also different. The knowledge-based functions are informational systems.

"Informational systems" have to do with analyzing data and making decisions, often major decisions, about how the enterprise will operate, now and in the future. And not only do informational systems have a different focus from operational ones, they often have a different scope. Where operational data needs are normally focused upon a single area, informational data needs often

span a number of different areas and need large amounts of related operational data.

In the last few years, Data Warehousing has grown rapidly from a set of related ideas into architecture for data delivery for enterprise end- user computing.

2.2 Difference of Operational and Informational system

	OPERATIONAL	INFORMATIONAL
Data Content	Current Value	Archived, derived, Summarized
Data Structure	Optimized for transactions	Optimized for complex queries
A c c e s s Frequency	High	Medium to low
Access Type	Read, update, delete	Read
Usage	Predictable, repetitive	Ad hoc, random, heuristic
Response Time	Sub-seconds	Several seconds to minutes
Users	Large number	Relatively small number

2.3 OLTP and DSS Systems

One of the interesting differences between the operational environment and the data warehouse environment is that of the transaction that is executed in each environment. In the operational environment when a transaction executes, the execution entails very little data. As few as two or three rows of data may be required for the execution of an operational transaction. A really large operational transaction may access up to twenty-five rows of data. But the number of rows that is accessed is modest. It is necessary to keep the row size small in the operational environment if consistent, good online response time is to be maintained.

The transaction profile in the DSS data warehouse environment is very different. The transactions run in the DSS environment may access thousands and even

hundreds of thousands of rows of data. Depending on what the DSS analyst is after, the data warehouse transaction may access huge amounts of data.

The response time in the DSS environment is very different from the response time found in the OLTP environment. Depending on what is being done in the DSS data warehouse environment, response time may vary from a few seconds all the way up to several hours.

There is then a marked difference in the transaction profile found in the DSS data warehouse environment and the operational transaction processing environment.

One by-product of this extreme difference in transaction profiles is that the definition of response time differs from one environment to another. In the operational environment, transaction response time is the length of time from the initiation of the transaction until the moment in time when results are FIRST returned to the end user.

In the DSS data warehouse environment, there are two response times. One response time is the length of time from the moment when the transaction is initiated until the first of the results are returned. And the second measurable response time is the length of time from the moment of the initiation of the transaction until the moment when the LAST of the results are returned. The difference between these two variables can be considerable.

Both sets of response time are needed in order to effectively measure system performance in the DSS data warehouse environment.

2.4 Introduction to Data Warehouse

The Data Warehousing is the only viable solution for providing strategic information. The information delivery for the new system environment called the data warehouse. So, followings are the functional definition of the data warehouse.

The data warehouse is an informational environment that

- Provides an integrated and total view of the enterprise.
- Makes the enterprise's current and historical information easily available for decision-making.
- Makes decision-support transaction possible without hindering operational systems
- Renders the organization's information consistent.
- Presents a flexible and interactive source of strategic information.

The data warehouse is : Subject-Oriented, Integrated, Non-Volatile, and Time variant collection of data.

2.4.1 Subject-Oriented

In operational systems, data is stored by individual applications. In the data sets for an order processing applications, data is kept for that particular application. These data sets provides the data for all the functions for entering orders, checking stock, verifying customer's credit, and assigning the order for shipment. But these data sets contain only the data needed for those functions relating to this particular application. Some data sets containing data about individual orders, customers, stock status and detailed transactions, but these are structured around the processing of orders.

In enterprise, data sets are organized around individual applications to support those particular operational systems. These individual data sets a have to provide data for the specific application to perform the specific function efficiently. Therefore, the data sets for each application need to be organized around that specific application.

In the data warehouse, subjects store data, not by applications. If business subject stores the data, what are business subjects? Business subjects differ from enterprise to enterprise. These are the subjects critical for the enterprise. For a manufacturing company, sales, shipments, and inventory are critical business subjects. For a retail store, sale at the check-out counter is a critical subject.

Following figure distinguish between how data is stored in operational system and in the data warehouse. In the operational system shown, data for each application is organized separately by application: order processing, consumer loans, customer billing, accounts receivable, claims processing and saving accounts. For example *claim* is a critical business subject for an insurance company, claims under automobile insurance policies organized in that application. Similarly, claims data for workers compensation insurance is organized in the Workers Comp Insurance application. But in the data warehouse for an insurance company, claims data are organized around the subject of claims and not by individual application of Auto Insurance and Workers Comp.

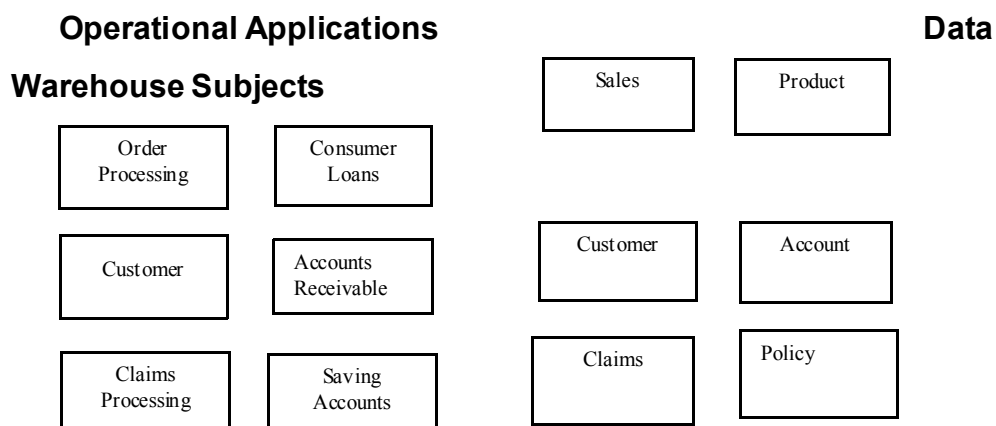


Fig 2.1: The data warehouse is subject oriented

In a data warehouse, there is no application flavor. The data in a data warehouse cut across applications.

2.4.2 Integrated

All the relevant data should pull together from various applications for proper decision making. The data in data warehouse comes from several operational systems. Source data are in different database files, and data segments. These are disparate applications, so the operational platforms and operating systems could be different. The file layouts, character code representation, and field naming conventions all could be different. This means that there is a single key structure and a single structure of data to be found in the warehouse where there might have been many forms of the same data in the applications. In the data warehouse there is a single structure for customer. There is a single structure for product. There is a single structure for transaction, and so on.

Before the data from various disparate sources can be usefully stored in a data warehouse, the inconsistencies should be removed, various data elements should be standardized and meaning data names in each source application should clear. Before moving the data into the data warehouse, one should go through the process of transformation, consolidation, and integration of the source data.

Following figure illustrates a simple process of data integration for a banking institution. In this example the data fed into the subject area of *account* in the data warehouse comes from three different operational applications. Naming conventions could be different; attributes for data items could be different. The account number in Saving Account Application could be eight bytes long, but only six bytes in the Checking Account application.

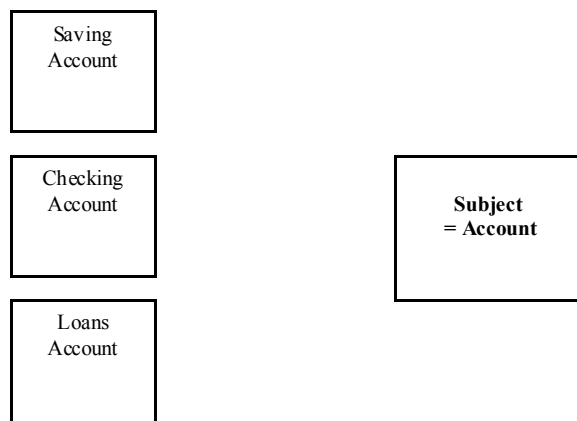


Fig 2.2: Data warehouse is integrated

2.4.3 Time-Variant Data

A data warehouse, because of the very nature of its purpose, has to contain historical data, not just current value. Data is stored as snapshots over past and current periods. Every data structure in the data warehouse contains the time element. Data warehouse contains historical snapshots of the operational data. This aspect of the data warehouse is quite significant for both the design and the implementation phase.

Time variant records are records that are created as of some moment in time. Every record in the data warehouse has some form of time valiancy attached to it. The easiest way to understand time variant records is to contrast time variant records against standard data base records. Consider a standard data base record. With the world changes, so change the values inside the database record. Data is updated, deleted, and inserted inside the standard database record. Now contrast the data warehouse record with the standard database record. Data is loaded into the data warehouse record. The moment when the data is loaded into the warehouse is usually a part of the warehouse record. And data is accessed inside the data warehouse record. But once placed inside the data warehouse, data is not changed there. Data inside the warehouse becomes an environment where the environment can be typified as load and access.

For example, in a data warehouse containing units of sale, the quantity stored in each file record for table row relates to a specific time element. Depending on the level of the details in the data warehouse, the sales quantity in a record may relate to a specific date, week, month, or quarter.

The time-variant nature of the data in a data warehouse, allows for analysis of the past, relates information to the present, enables forecasts for the future.

2.4.4 Nonvolatile Data:

Data extracted from the various operational systems and pertinent data obtained from outside sources are transformed, integrated, and stored in the data warehouse. The data in the data warehouse is not intended to run the day-to-day

business. When one want to process the next order received from a customer, it is not necessary to look into the data warehouse to find the current stock status. The operational order entry application is meant for that purpose. In the data warehouse, it is kept the extracted stock status data as snapshots over time. It is not necessary to update the data warehouse every time a single record is processed.

Data from the operational systems are moved into the data warehouse at specific intervals. Depending on the requirements of the business, these data movements take place twice a day, once a day, once a week, or once in two weeks. In fact, in a typical data warehouse, data movements to different data sets may take place at different frequencies. The changes to the attributes of the products may be moved once a week. Any revision to geographical setup may be moved once a month. The units of sales may be moved once a day. There should be planning and scheduling of data movements or data loads based on the requirements of users.

As illustrated in following figure, every business transaction does not update the data in the data warehouse. The business transactions update the operational system databases in real time. It is added, changed, or deleted data from an operational system as each transaction happens but do not usually update the data in the data warehouse. The data cannot be deleted in the data warehouse in real time. Once the data is captured in the data warehouse, an individual transaction cannot be run to change the data there. Data updates are commonplace in an operational database; not so in a data warehouse. The data in a data warehouse is not as volatile as the data in an operational database is. The data in a data warehouse is primarily for query and analysis.

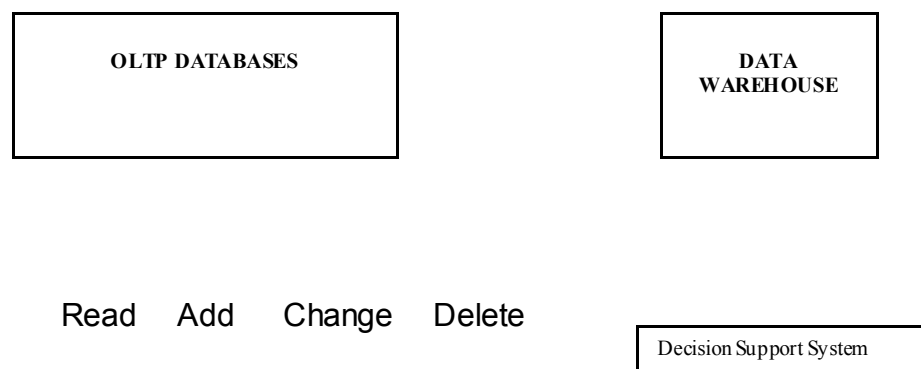


Fig 2.3: OLTP Database to Data Warehouse

2.4.5 Data Granularity

In an operational system, data is usually kept at the lowest level of detail. In a point-of-sale system for a grocery store, the units of sale are captured and stored at the level of units of a product per transaction at the check-out counter. In an order entry system, the quantity ordered is captured and stored at the level of units of a product per order received from the customer. Whenever it is needed summary data, it can be added up the individual transactions. If it is required that how many units of a product is ordered in a month, all the orders entered for the entire month for that product must be read and then add up. Operational system keep summary of data.

Data in the warehouse is granular. This means that data is carried in the data warehouse at the lowest level of granularity. So it can be found summarized data at different levels. Data granularity in a data warehouse refers to the level of detail. The lower level of detail provides the finer the data granularity. Granularity levels can be decided based on the data types and the expected system performance for queries.

2.5 Data Warehouse S/W and H/W Architecture

The architecture of a data warehouse by necessity is complex, and includes many elements. The reason for this is that a data warehouse is an amalgamation of many different systems. Integration of diverse elements is its primary concern, and to accomplish this integration, many different systems and processes are necessary.

Most software development projects require selection of the technical infrastructure, and this is true for the warehouse as well. Basic technical infrastructure includes operating system, hardware platform, database management system, and network. The DBMS selection becomes a little more complicated than a straightforward operational system because of the unusual challenges of the data warehouse, especially in its capability to support very complex queries that cannot be predicted in advance.

How does the data get into the data warehouse? The warehouse requires ongoing processes to feed it; these processes require their own infrastructure. Many time, IS shops overlook this aspect when they plan for the data warehouse. Data layers need to be understood and planned for. Data cleansing usually involves several steps; where will the "staging area" be stored? How will ongoing data loads, cleansing, and summarizing be accomplished?

Backup and recovery are interesting challenges in the data warehouse, mainly because data warehouses are usually so large.

How will users get information out of the warehouse? The choice of query tool becomes very important, and depends upon a multiplicity of factors.

2.6 Basic steps to develop DW Architecture:

It is very much important to understand and discuss the basic steps to develop data warehouse architecture. Each and every one of these steps needs to be performed in order to have the best opportunity of succeeding. There are six important steps to develop effective data warehouse architecture developments are as follows:

1. The first and most important step of developing effective data warehouse architecture is to enlist the full support and commitment of project sponsor/executive of the company.
2. Appointed staff in architecture team must be strongly skilled Personnel. It is not necessarily the technology you choose for your architecture, it is the personnel you have designing and developing the architecture that makes the project successful.
3. Prototype/benchmark all the technologies you are interested in using. Design and develop a prototype that can be used to test all of the different technologies that are being considered.
4. The architecture team should be given enough time to build the architecture infrastructure before development begins. For a large

organization, this can be anywhere from six months to a year or more.

5. The development staff must be trained on the use of the architecture before development begins. Spend time letting the development team get full exposure to the capabilities and components of the architecture.
6. Provide freedom to the architecture team to enhance and improve the architecture as the project moves forward. No matter how much time is spent up for developing architecture, it will not be perfect the first time around.

2.7 Architectural Components of D/W (Infrastructure):

In data warehouse architecture includes a number of factors. Primarily it includes the integrated data that is the centerpiece. The architecture includes everything that is needed to prepare the data and store it. On the other hand, it also includes All Students the means for delivering information from your data warehouse. The architecture is further composed of the rules, procedures, and functions that enable your data warehouse to work and fulfill the business requirements. Finally, the architecture is made up of the technology that empowers the data warehouse.

The data warehouse consists of the following architectural components, which compose the data warehouse infrastructure:

- System infrastructure: Hardware, software, network, database management system, and personnel components of the infrastructure.
- Metadata layer: Data about data. This includes, but is not limited to, definitions and descriptions of data items and business rules.
- Data discovery: The process of understanding the current environment so it can be integrated into the warehouse.
- Data acquisition: The process of loading data from the various Sources.

This is described in more detail in the ongoing maintenance section later in this chapter.

- Data distribution: The dissemination/replication of data to distributed data marts for specific segmented groups.
- User analysis: Includes the infrastructure required to support user queries and analysis. This is described in more detail in the "User Access" section, later in this chapter.

2.8 D/W System Architecture

The system architecture is the overall blueprint that is to be followed when building data warehouse platform. It is the underlying foundation that governs many of the decisions will be needed to make when building and managing data warehouse platform. Given this, it is no surprise that there are probably as many different data warehouse architectures as there are data warehouses. However, they can usually be grouped into one of two main categories: a three- tiered architecture or a two-tiered architecture.

2.8.1 Three-Tiered Architecture

In a three-tiered architecture, the first tier is comprised of operational systems that are already in place. These are the transaction processing systems that collect the data of all the events that occur within enterprise. The data collected by these systems is fed into your data warehouse. The second and third tiers of this architecture are the data warehouse and the data marts, respectively.

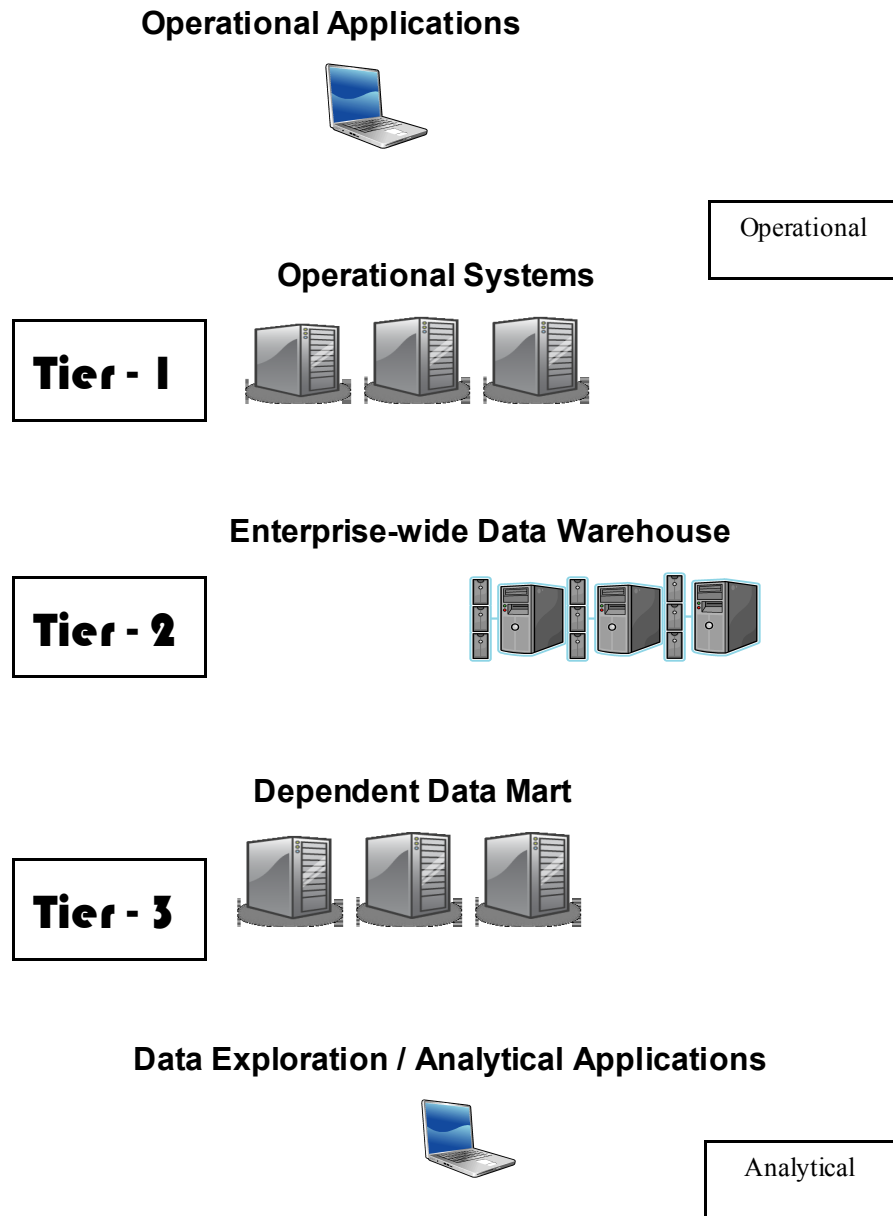


Fig 2.4: Three-Tiered Architecture

There two different types of functions that must be addressed when building a large-scale data warehouse environment that is data consolidation and data analysis.

Data consolidation refers to the process of transforming, extracting, and cleaning the data from disparate and unconsolidated operational systems into one consolidated repository. Data analysis refers the process of end-users access, manipulate, and generally analyze the data looking for useful insights. For these two different needs, it is much more scalable to split these functions

into two different tiers. In addition, form of functional parallelism is used here to improve scalability, and different tasks are assigned to different computers. It is very much simple and easier to address different needs if have one tier focused on optimally solving data consolidation issues and another tier focused on optimally solving data exploration issues.

In three-tier architecture, a tier for the data warehouse is responsible for the consolidation activities and its function is to take data from the various operational systems, consolidate the data, and then feed portions of the consolidated data into the various data marts, and the data marts are responsible for the exploration activities. Since these data marts get their data from the data warehouse, and it is referred as “dependent data marts”, they are dependent on the existence of an enterprise data warehouse.

Data warehouses and data marts can be distinguished as: the data warehouse is fed by multiple operational systems, and it performs the required extractions and transformations. On the other hand, data marts only need to extract data from a single source that is already consolidated in the data warehouse. The data marts also occasionally include additional external data, but the amount of consolidation of data performed by data mart is very much less than in comparison with the data warehouse.

The data warehouse stores its information in a form that is called **application generic**, and it is used to feed multiple data marts, each of which is focused on a different set of business problems. Data warehouse designer, always wants to keep the data stored in the data warehouse tier in its most flexible form, which is the not summarized, means in detail level form. So the design a database schema for the second tier that has much of the flavor of a traditional third-normal form schema. On other hand, data marts need to store their information in a form that is **application specific** and tailored to meet the needs of the explorer or farmer. It means, there is requirement of summarizations, subsets, and/or samples in data mart that are specific to the particular business unit that is using the data mart.

It is also necessary that the data in data marts must be easily accessible by the end-user community. Traditional third-normal form schemas are excel in minimizing data redundancy, and fairly poor models for end users to try to understand and analyze, and is, therefore, usually a poor choice for data marts. Instead of third-normal form, data marts should use dimensional models and star schema designs, which make heavy use of redundancy to make it far easier for end-users to navigate their way through large volumes of data without getting lost.

The choice of optimal hardware and DBMS may different for the data warehouse and the data marts. The data warehouse tier has to act as a repository or enormous amounts of data that span many different organizations and subject areas. In addition, more data from both existing and new subject areas is constantly being added to the data warehouse. So the data warehouse tier must be able to feed an ever-growing number of application-specific data marts. This means that the data warehouse tier is desired that it must be an industrial-strength, highly scalable, enterprise-class hardware and DBMS.

However, the data marts need only focus on a single business problem. Data mart will see growth in their particular subject area, because all the new transactions related to that subject area are continuously being collected from the operational systems, but the magnitude of this growth is far less. Therefore the data mart tier is usually a smaller but scalable, department level hardware and DBMS solution.

2.8.1.1 Benefits of Three-Tiered Architecture

The major benefits of three-tiered data warehouse architecture are high performance and scalability. The high performance comes from the fact that the inclusion of data marts allows to partition different query workloads across different data marts. It means that the workload levels of users of other data marts will not affect users of one data mart. For instance, if users of the sales data mart are executing very complex and long-running queries that are highly resource intensive, it does not effect on the performance seen by users of the completely separate finance data mart. It results the increase in end-user

satisfaction levels is enormous. General experience is that, end users get frustrated by their own queries slowing down their machine, but nothing infuriates end users more than having someone else in some other department bring a machine that everyone must share to its knees. When the workload is separated into physically separate marts, different sets of users can be prevented from adversely affecting each other.

Three-Tier architecture is also very much scalable. As explained above, extraction and consolidation functions are assigned to one-tier and end-user query and data analysis functions to another tier. This is a straightforward use of the concept of functional parallelism that is described previously as one effective method for improving scalability. Second and third tiers can be individually scaled up as well. The data warehouse tier is a large and highly scalable, and it can be scaled up by adding more resources like processors, disks and I/O controllers to it. Scaling up of the data mart tier can be done by simply adding more data marts to service new user populations, address new subject areas, or focus on providing a new type of functionality such as data mining. But data marts are usually very cheaper to build compare to data warehouses. It is also very easy to add another data mart when it is needed to explore a new business area.

2.8.1.2 Drawbacks of Three-Tiered Architecture

The main drawback is the multi-subject; enterprise-wide data warehouse is always at the center of this architecture. Designing of this data warehouse is a complex process. It is very much time consuming process to consolidate various subject areas, this can involve many long meetings and debates with the representatives from various organizations. So, there is quite a large time investment involved with building an enterprise-wide data warehouse. Defining which business problems you want to solve, finding where all the data required to solve those problems is located, writing all the required extraction, cleansing, and transformation routines, loading all the data into the database, and then tuning the resulting system is no trivial task. In fact, to build average enterprise-wide data warehouse takes about 18-24 months to build. Finally, the

cost of such a system is not trivial, often reaching into the many millions of rupees.

The complexity of the project and the required time and cost investments are prohibitive for many organizations. The dynamic, organic nature of a data warehouse environment is that, it doesn't really make sense to build something that large as at very first step. Need of organization may change by the time when data warehouse is to be delivered finally. It is required to spent a large amount of effort building a wonderful system that helps to give answers of the questions, and there are no longer the most important questions that need answering.

2.8.2 Two-Tiered Architecture

Generally, to build a two-tiered architecture, two way are used. The first involves just building the enterprise-wide data warehouse without the data marts, and all the end-users can have direct access of data warehouse. In this architecture, there is no need of separate data mart hardware to store copies of the data because that already exists in the central data warehouse, main benefit is that cost and some amount of time may reduce to build such architecture compare to Three-Tier architecture. Data marts are generally not as complicated to build, so the timesavings would not be dramatic role some time. But there are some limitations with this approach. First, when building of the central data warehouse is started, at that time majority of the complexity, time, cost, and risk are main factors. Second is that, when all departments and all users will be sharing a single database, to separate workloads among different user groups is very much crucial task. With these limitations, it can be concluded that there are not advantages to build such architecture.

Another most common approach is to build Two-Tiered architecture is to build the data marts without building the centralized data warehouse. But these data marts do not depend on the existence of a consolidated data warehouse, so it can be referred as **independent data mart**.

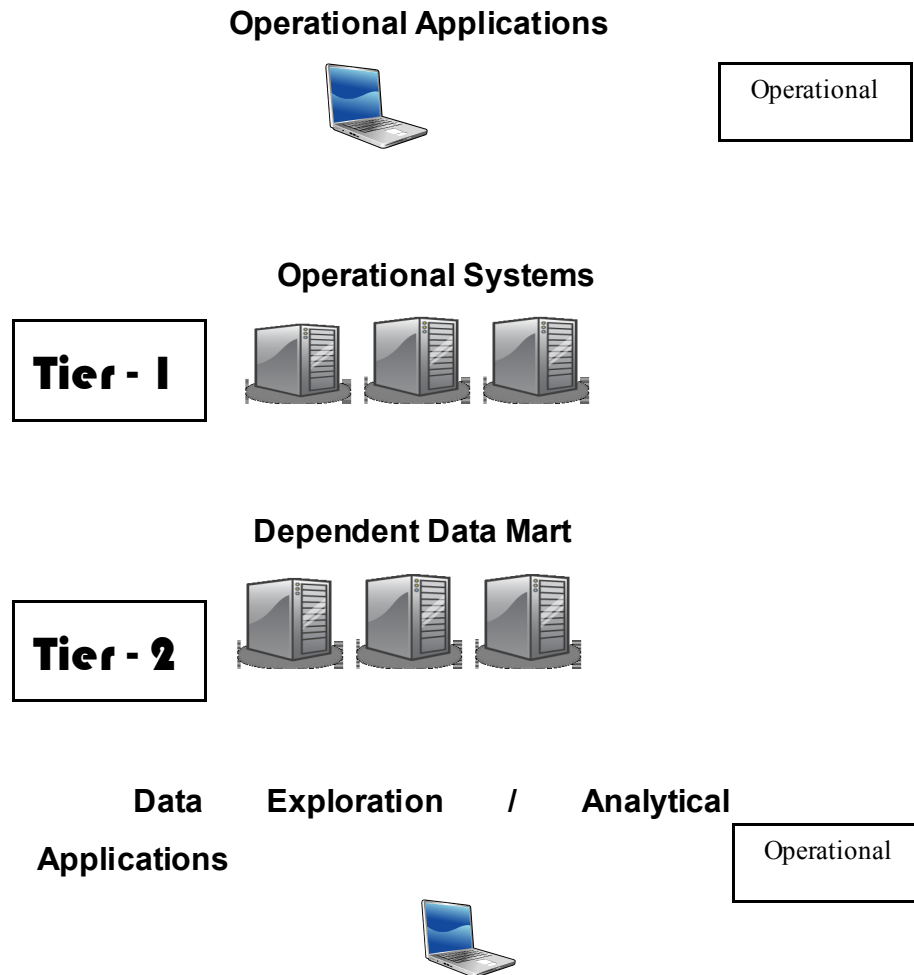


Fig 2.5: Two-Tiered Architecture

But there are some advantages of building a two-tiered environment using the independent data mart approach.

- The data mart traditionally will only have data pertaining to one or two subject areas, so there is much less complexity involved in the design and implementation of this architecture.
- This architecture is dealing with fewer data sources and less data, so amount of time is reduced to build such architecture in comparison of three-tiered architecture.
- As discussed earlier, the hardware required for the data mart, are generally much smaller departmental machines, not enterprise-class machines like in three-tiered architecture.

2.9 Data Marts

A data mart is a powerful and natural extension of a data warehouse to a specific functional or departmental usage. The data warehouse provides granular data and various data marts interpret and structure the granular data to suit their needs. Data mart is a container or structure in which gets the data from enterprise data warehouse. The data mart is the place, where the end user has the most interaction with the enterprise data warehouse environment. Various data marts are designed after knowing the requirements of the different departments that own the data mart. All the data mart looks different from each other, Because a different department owns each data mart.

The detailed data is found in the enterprise data warehouse, while very little detailed data is found in the data mart, because enterprise data warehouse is the source of data inside the data mart. The data stored in enterprise data warehouse is reshaped according to departmental requirement and then data is sent to the data mart. Data is often summarized and/or otherwise aggregated when it moves into the data mart.

The data stored in the data mart can be described as residing in star joins (star schemas) or snowflake structures. These star joins reflect the different ways that the departments look at their data.

The world of data mart revolves around technology and structures that are designed for end user access and analysis. The leading of these structures is the cube or the multi dimensional structure. Most of the queries are well structured before the query is submitted to the data mart. Many reports emanate from the data mart. The data mart can also spawn smaller desktop versions of the data mart that can be fed and maintained at an individual's workstation. The data mart is reconcilable back to the enterprise data warehouse. The only legitimate source of data for the data mart other than the enterprise data warehouse is external data.

2.9.1 Data Mart Structure

A star join or a snowflake structure is most suitable for the data structure of the data mart. There are two basic components of a star join structure. One is a fact table and another is supporting dimension tables. A fact table represents the data that is the most populous in the data mart. Stocking data are typically most populous data. In bank, transactions done by ATM are also populous data.

The fact table is a composition of many types of data that have been pre-joined together. The fact table contains:

- A primary key reflecting the entity for which the table has been built, such as an order, a transaction of stock, a transaction through ATM, etc.
- Information about primary key.
- Foreign keys relating the fact table to the dimensions.
- Non-key foreign data that is carried with the foreign key. This non-key foreign data is included if it is regularly used in the analysis of data found in the fact table.

The fact table is highly indexed. In some cases every column in the fact table is indexed. There may be 30 to 40 indexes on the fact table. The highly indexing results that data in a fact table is highly accessible. However, the amount of resources required for the loading of the indexes must be factored into the equation. By rule, fact tables are not updated any way. They have data loaded into them, but once a record is loaded properly, there is no need to go into the record and alter any of its contents.

The dimension tables surround the fact tables. The dimension tables contain the data that are non-populace. The dimension tables are related to the fact tables through a foreign key relationship. Typical dimension tables might be product list, customer list, vendor lists, etc., depending of course on the data mart being represented. Dimension data are the dimensions along which data can be analyzed. In the pet store, sales data can be analyzed by salesperson or by type of pet. Therefore, “salesperson” and “type of pet” are dimension data. Generally, time is a dimension element in most data warehouses.

Followings are the differences between fact data and dimension data.

Fact Data	Dimension Data
<ul style="list-style-type: none"> • Millions or billions of rows • Multiple foreign keys • Numeric • Does not change 	<ul style="list-style-type: none"> • Tens to a few million rows • One primary key • Textual descriptions • Frequently modified

The source of the data found in the data mart is the enterprise data warehouse. All data should pass through the enterprise data warehouse before finding its way into the data mart, but there is one exception. The exception is data that is specific only to the data mart that is used nowhere else in the environment. External data often fits this category. If however, the data is used anywhere else in the DSS environment, then it must pass through the enterprise data warehouse.

Generally the data mart contains two kinds of data, which is detailed data and summary data. The detailed data in the data mart is contained in the star join, as previously described. It is noteworthy that the star join may well represent a summarization as it passes out of the enterprise data warehouse. In that sense, the enterprise data warehouse contains the most elemental data while the data mart contains a higher level of granularity. However, from the point of view of the data mart user, the star join data is as detailed as data gets.

The second kind of data that the data mart contains is summary data. It is very common for the users to create summaries from the data found in the star join (detailed data). A typical summary might be monthly sales totals of sales territory. Because summaries are kept on an ongoing basis, history of the data is stored in the data mart. But the preponderance of history that is kept in the data mart is stored at the summary level. Very little history is kept at the star join level.

Whenever it is required, the data mart can be refreshed from the enterprise data warehouse. However, refreshment can be done either much more or much less

frequently, depending on the needs of the department that owns the data mart.

2.9.2 Usage of Data Mart

The data mart is the most versatile of the data structures. The data mart allows the data to be examined from the standpoint of many perspectives - from a detailed perspective, from a summarized perspective, across much data, across few occurrences of data.

The primary user of the data called a farmer. The farmer knows what will happen, when the query is submitted. The farmer does the same activity repeatedly against different occurrences of data. The data is structured inside the data mart so that it is optimal for the access of the farmer. The farmer spends a fair amount of time with the DWA for the gathering and synthesizing requirements before the data mart is built.

The farmer looks at summarized data, at exception based data, at data created on a periodic basis, and other types of data. The farmer also looks upon the data mart as a mission critical component of the environment.

Another use of the data mart is as a spawning ground for insightful analysis by the explorer community. While the data mart is not designed to support exploration, many of the insights, which merit deeper exploration, are initiated in the data mart. The explorer has the inspiration at the data mart, does some cursory exploration there, and then moves down to the explore warehouse for the detailed analysis that is required for exploration. The data mart lacks the foundation for exploration because data is structured along the lines of a department, because data is usually summarized and the explorer needs detail, and because the data mart has a limited amount of historical data. Never the less, the data mart is a fertile breeding ground for insight.

2.9.3 Security in a Data Mart

When there is secretive information in the data mart it needs to be secured well. Typically, secretive information includes financial information, medical records and human resource information etc. the data mart administrator should make

necessary security arrangements such as: firewalls, log on/off security, application based security, DBMS security, encryption and decryption. The information cost of security depends upon its exclusiveness.

2.10 Data warehouse and Data Mart

Finally Data warehouse and Data Mart can be distinguished considering following factors:

Data Warehouse	Data Mart
<ul style="list-style-type: none"> • Corporate/Enterprise-wide • Union of all data marts • Data received from staging area • Queries on presentation resource 	
<ul style="list-style-type: none"> • Structure for corporate view of data 	
<ul style="list-style-type: none"> • Departmental • A single business process • Star-join (facts & dimensions) • Technology optimal for data access and analysis • Structure to suit the departmental view of data 	

2.11 OLAP (Online Analytical Processing)

Online Analytical Processing (OLAP) systems, contrary to the regular, conventional online transaction processing (OLTP) systems, are capable of analyzing online a large number of past transactions or large number of data records (ranging from mega bytes to giga bytes and tera bytes) and summarize them on the fly. This type of data is usually multidimensional in nature. This multi-dimensionality of the key driver for OLAP technology, which happens to be central to data warehousing.

Multidimensional data may be stored in spreadsheet, cannot be processed by conventional SQL type DBMS. For a complex real-world problem, the data is

usually multidimensional in nature. Even though one can manage to put such data in a conventional relational database in normalized tables, the semantics of multidimensionality will be lost and any processing of such data in the conventional SQL will not be capable of handling it effectively. As such, multidimensional query on such a database will explode into a large number of complex SQL statements each of which may involve full table scan, multiple joins, aggregation, sorting and also large temporary table space for storing temporary results.

Finally, the end-result may consume large computing resources in terms of disk space, memory, CPU time, which may not be available and even if they are available the query may take very long time. For example, conventional DBMS may not be able to handle three months' moving average or net present value calculations. These situations call for extensions to ANSI SQL, a near non-feasible requirement.

In addition to response time and other resources, OLAP is a continuously iterative process and preferably interactive one. Drilling down from summary aggregative levels to lower level details may be required to be done on an ad hoc basis by user. Such drilling down may lead the user to detect certain patterns in the data. The user may put forward yet another OLAP query based on these patterns.

This process makes it impossible to handle or tackle for a conventional database.

2.11.1 OLTP and OLAP Systems

Conventional OLTP database applications are developed to meet the day-to-day database transactional requirements and operational data retrieval needs of the entire user community. On the other hand, the data warehousing based OLAP tools are developed to meet the information exploration and historical trend analysis requirements of the management or executive user communities. The conventional regular database transactions or OLTP transactions are short, high volume, provide concurrent and online update,

insert, delete in addition to retrieval queries and other procedures, processing or reporting. These transactions in batch mode may be ad hoc, online or pre-planned. On the other hand, OLAP transactions are long (infrequent or occasional updates or refreshing the data warehouse), but the more efficient in processing a number of ad hoc queries. Information in a data warehouse frequently comes from different operational source systems (which are usually conventional OLTP database systems) and is interpreted, filtered, wrapped, summarized and organized in an integrated manner, making it more suitable for trend analysis and decision support data retrieval.

Following is the comparison between OLTP and OLAP systems

- | OLTP | OLAP |
|---|--|
| <ul style="list-style-type: none"> • Only current data available (old data) is replaced by current data by updating) • Short transactions (single granularity or more) • Online update/insert/delete transactions • High volume of transactions in a given period | <ul style="list-style-type: none"> • Concurrency control and transactions recovery |
| <ul style="list-style-type: none"> • Largely online ad hoc queries, requiring low level of indexing • Both current and historic data available (current is appended to historic data) • Long database transactions | <ul style="list-style-type: none"> • Batch update/insert/delete transactions. • Low volume transactions, periodic refreshing |
| <ul style="list-style-type: none"> • No concurrent transactions and therefore no recovery upon failure required • Largely pre-determined queries requiring high level of indexing | |

A very popular and early approach for achieving analytical processing is 'star schema' or 'collection model'. This approach is based on the common denomination of the user requirements. In this model, a small number of user-referenced or joint data sets are generated from the detailed data sets. This involves de-normalization of the data followed by integration it is shown in figure of three-tiered architecture of data warehouse.

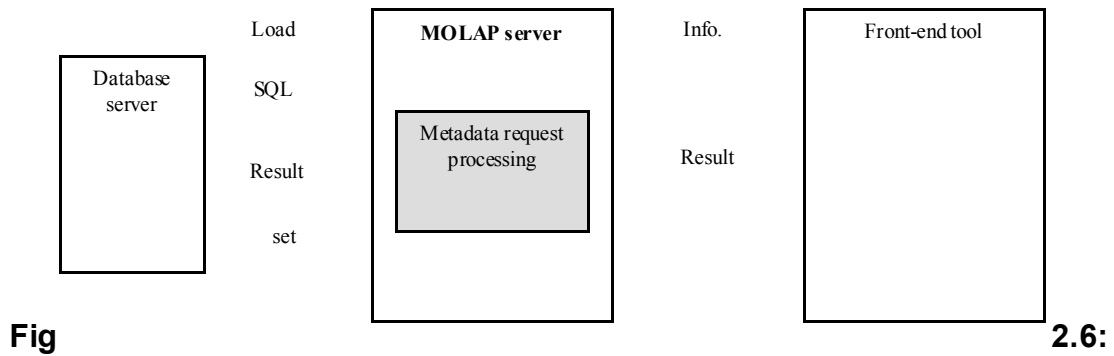
2.11.2 Types of OLAP

In the OLAP world, there are mainly two different types: Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). Hybrid OLAP (HOLAP) refers to technologies that combine MOLAP and ROLAP.

2.11.2.1 MOLAP

MOLAP is the more traditional way of OLAP analysis. MOLAP-based products organize, navigate and analyze data typically in an aggregated form. They require tight coupling with the applications and type depend upon a multidimensional database (MDDDB) system. Efficient implementations store the data in a way similar to the form in which it is utilized by using improved store techniques so as to minimize storage. Many efficient techniques are used as sparse data storage management on disk so as to improve the response time. Applications requiring iterative and comprehensive time series analysis of trends are well suited for MOLAP technology.

Some of the problems faced by users are related to maintaining support to multiple subject areas in an RDBMS. As shown in following figure, some vendors can solve these problems by maintaining access from MOLAP tools to detailed data in RDBMS.



MOLAP architecture

This can be very useful for organizations with performance-sensitive multidimensional analysis requirements and that has built or is in the process of building a data warehouse architecture that contains multiple subject areas. An example would be creation of sales data measured by several dimensions (product, sales region) to be stored and maintained in a persistent structure. This structure would be provided to reduce the application overhead performing calculations and building aggregations during application initialization. These structures can be automatically refreshed at predetermined intervals established by an administrator.

2.11.2.1.1 Advantages

- Excellent performance: MOLAP cubes are built for fast data retrieval, and are optimal for slicing and dicing operations.
- Ability to perform complex calculations: All calculations have been pre-generated when the cube is created. Hence, complex calculations are not only doable, but they return quickly.

2.11.2.1.2 Disadvantages

- Limitation of handling large amount of data: Because all calculations are performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is

possible. But in this case, only summary-level information will be included in the cube itself.

- Requirement of additional investment: Cube technologies are often proprietary and do not already exist in the organization. Therefore, to adopt MOLAP technology, chances are additional investments in human and capital resources are needed.

2.11.2.2 ROLAP

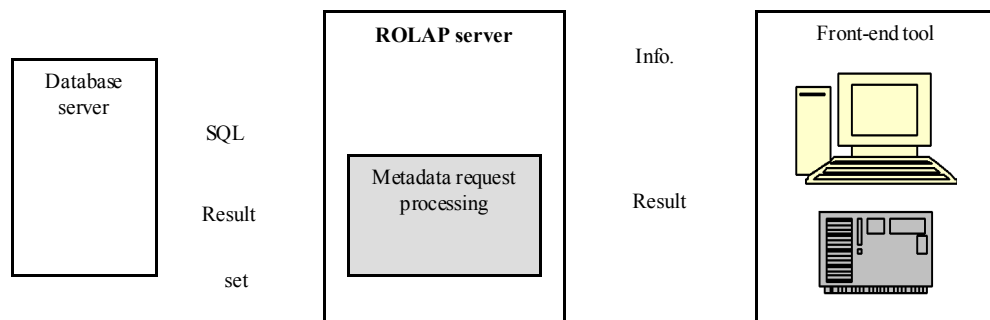


Fig 2.7: ROLAP architecture

This approach enables multiple multidimensional views of two-dimensional relational tables to be created, avoiding structuring data around the desired view. Some products in this segment have supported strong SQL engines to support the complexity of multidimensional analyst. This includes creating multiple SQL statements to handle user requests, being 'RDBMS aware' and also being capable of generating SQL statement based on the optimizer of the DBMS engine. While flexibility is the attractive feature of ROLAP, there exist products, which require the use of de-normalized database designs (as star schema). However, of late there is a noticeable change of realignment in ROLAP technology. Firstly, there is a shift towards pure middle-ware technology so as to simplify the development of multidimensional applications. Secondly, the sharp delineation between ROLAP and other approaches as hybrid-OLAP is fast disappearing. Thus vendors of ROLAP tools and RDBMS products are now eager to provide multidimensional persistent structure with facilities to assist in the administrations of these structures.

2.11.2.2.1 Advantages

- Handling large amounts of data: The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on amount of data.
- Leverage of functionalities inherent in the relational database: Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities.

2.11.2.2.2 Disadvantages

- Slow Performance: Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, so it is obvious that the query time can be long if the underlying data size is large.
- Limited by SQL functionalities: Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

2.11.2.3 HOLAP

HOLAP technologies attempt to combine the advantages of MOLAP and ROLAP, which are other possible implementation of OLAP. HOLAP allows storing part of the data in the MOLAP store and another part of the data in ROLAP store. The degree of control that cube designer has over this partitioning varies from product to product. For summary-type information, HOLAP leverages cube technology for faster performance. When detail information is needed, HOLAP can "drill through" from the cube into the

underlying relational data.

Vertical Partitioning: In this mode HOLAP stores aggregations in MOLAP for fast query performance, and detailed data in ROLAP to optimize time of cube processing.

Horizontal Partitioning: In this mode HOLAP store comes slice of data, usually the more recent one (eg. sliced by Time Dimension) in MOLAP for fast query performance, and older data in ROLAP. Moreover, some dices in MOLAP and others in ROLAP can be stored, that leverages the fact that in large cuboids, there will be dense and sparse sub regions.

Popular Tools:

- Micro Analysis Services
- Micro Strategy DSS Web
- Cognos PowerPlay
- BI Accelerator
- SAP AG
- IBI Focus Fusion
- Pilot Software
- Arbor Essbase Web
- Information Advantage Web

2.12 ETL (Extraction, Transformation, Loading)

ETL is process that involves extracting data from outside sources, transforming it to fit business needs, and loading into the data warehouse.

ETL is important, as it is the way data actually gets loaded into the warehouse. ETL can also be used for the integration with legacy systems.

2.12.1 Extraction

The first part of an ETL process is to extract the data from the source systems. Most data warehousing projects consolidate data from different source systems. Each separate system may also use a different data organization /

format. Common data source formats are relational databases and flat files, but may include non-relational database structures such as IMS or other data structures such as VSAM or ISAM. Extraction converts the data into a format for transformation processing.

2.12.2 Transformation

The transformation stage applies a series of rules or functions to the extracted data to derive the data to be loaded. Some data sources will require very little manipulation of data. In other cases, one or more of the following transformations types may be required.

- Selecting only certain columns to load or selecting that null columns not to be loaded.
- Translating coded values, if the source system stores 1 for male and 2 for female, but the warehouse stores M for male and F for female, this is called data cleansing.
- Encoding free-form values, it is mapping of “Male” ,“1” and “Mr” into M.
- Deriving new calculated values, like $\text{sale_amt} = \text{quantity} * \text{rate}$.
- Joining together data from multiple sources, like lookup data and merging of data.
- Summarizing multiple rows of data, means to summarize total sale for each store for each region.
- Generating surrogate key values.
- Transporting or pivoting, it turns multiple columns into multiple rows or vice versa.
- Splitting a column into multiple columns.



Fig 2.8: Extraction and Cleansing of data

2.12.3 Loading

The load phase loads the data into the data warehouse. Depending on the requirements of the organization, this process ranges widely. Some data warehouses might weekly overwrite existing information with cumulative, updated data, while other data warehouse or part of them might add new data hourly. The timing and scope to replace or append are strategic design choices dependent on the time available and business needs. More complex systems can maintain a history and audit trail of all changes to the data.

ETL processes can be quite complex, and significant operational problems can occur with improperly designed ETL systems.

The range of data values or data quality in an operational system may be outside the expectations of designers at the time validation and transformation rules are specified. Data profiling of a source during data analysis is recommended to identify the data conditions that will need to be managed by transform rules specifications.

The scalability of an ETL system across the lifetime of its usage needs to be established during analysis. This includes understanding the volumes of data that will have to be processed within Service Level Agreements (SLAs). The

time available to extract from source systems may be change, which may mean the same amount of data may have to be processed in less time. Some ETL systems have to scale to process terabytes of data to update data warehouse with tens of terabytes of data. Increasing volumes of data may require designs that can scale from daily batch to intra-day micro-batch to integration with message queues for continuous transformation and update.

A recent development in ETL software is the implementation of parallel processing. This has enabled a number of methods to improve overall performance of ETL processes when dealing with large volumes of data. There are three main types of parallelisms as implemented in ETL applications

2.12.4 Data

By splitting a single sequential file into smaller data files to provide parallel access.

2.12.5 Pipeline

Allowing the simultaneous running of several components on the same data stream. An example would be looking up a value on record 1 at the same time as adding together two fields on record 2.

2.12.6 Component

The simultaneous running of multiple processes on different data streams in the same job. Sorting one input file while performing a duplication on another file would be an example of component parallelism.

All the three types of parallelism are usually combined in a single job. An additional difficulty is making sure the data being uploaded is relatively consistent. Since all have different update cycles, an ETL system may be required to hold back certain data until all sources are synchronized.

Likewise, where a warehouse may have to be reconciled to the contents in a source system or with the general ledger, establishing synchronization and

reconciliation point is necessary.

2.12.7 Popular ETL Tools

- Data Junction
- Essential Data Stage
- Ab Initio
- Informatica

2.13 Meta Data

Metadata (data about data) describes the details about the data in a data warehouse or in a data mart. When structured into a hierarchical arrangement, metadata is more properly called an ontology or schema. Both terms describe, “what exists” for some purpose or to enable some action. For instance, the arrangement of subject heading in a library catalog serves not only as a guide to finding books on a particular subject in the stacks, but also as a guide to what subjects “exist” in the library’s own ontology and how more specialized topics are related to or derived from the more general subject headings.

Metadata is frequently stored in a central location and used to help organizations standardize their data. This information is typically stored in a metadata registry.

Following are the components of metadata for a data warehouse or data mart.

- Description of sources of the data.
- Description of customization that may have taken as the data passes from data warehouse into data mart.
- Descriptive information about data mart, its tables, attributes and relationships, etc.
- Definitions of all types.

The metadata of a data mart is created and updated from the load programs that move data from data warehouse to data mart. The linkages and

relationships between metadata of data warehouse and metadata of data mart have to be well established or well understood by the analyst using the metadata.

This description is essential to establish drill down capability between the two environments. With this linkage, the analyst using data mart metadata can easily find the heritage of the data in data warehouse. Further, the DSS analyst also requires understanding how calculations are made and how data is selected for the data mart environment. The metadata pertaining to the individual data mart should also be available to the end-user for effective usage.

Distributed metadata resides at the data mart also. Using distributed metadata in the data mart, the end user can see:

- The metadata that applies to the local data mart
- The metadata that may resides elsewhere but which is relevant to the local data mart
- Locally protected metadata that is not open and available for any other departments
- Metadata residing at architectural entities other than data marts, such as ODS, enterprise data warehouse, etc.

The distributed metadata is very much useful for the data mart user, using this data mart user can look at both technical and business metadata.

2.13.1 Metadata Architecture

There are two basic architectures for metadata in the DSS data warehouse environment. Those architectures are a centralized architecture and a distributed architecture.

The classical metadata architecture is a centralized architecture. In centralized architecture metadata is stored and managed centrally. The rights of creation and update are invested in a central administrator. The great appeal of the centralized approach is that data can be uniformly defined and used over the

enterprise. Once defined centrally, the metadata will have no conflict in its definition.

The centralized approach to metadata architecture has the problem of not accommodating the need for local autonomy of metadata. Local autonomy of metadata refers to the need to create and manage metadata entirely within the confines of a department. There are other difficulties with the centralized approach to metadata as well. Some of these difficulties are:

- Most or all of the metadata in the enterprise must be accommodated and captured before any of the metadata is useful.
- The administrator of the centralized metadata must be taught the business of the department before the metadata becomes useful.
- The centralized metadata infrastructure cannot be built incrementally.

It is very difficult for the centralized metadata to be kept up to date once captured, and so forth.

A variant form of the centralized metadata architecture is the centralized replicated form of architecture. In this replicated form of metadata architecture, the metadata is gathered centrally, as in the case of the classical centralized metadata architecture. But once gathered there, the metadata can be copied out to any other person or environment that requests. Once copied, the person or organization that has requested the metadata can alter or otherwise manipulate the metadata. There are no controls or conditions on the metadata once copied.

Yet another alternative is that of the distributed metadata architecture. In the distributed metadata mode, metadata resides independently at the many different locales in the DSS environment, such as at the data mart environment, the ODS environment, the enterprise data warehouse environment, etc. Metadata resides and is managed locally. This means that a data mart, for example, creates, updates, and deletes its own metadata. There is local control and ownership of metadata. However, the metadata that is owned and managed locally can be shared. Other corporate entities - other data marts,

other ODS, other enterprise data warehouse, and so forth - can access the metadata as it is stored locally. In doing so, careful record is made of who the owner of the metadata is. If an organization is sharing metadata, it cannot alter the metadata that does not belong to it. The preservation of the rights of ownership of metadata is called the system of record across all participating corporate entities. The system of record is the backbone of the distributed metadata environment.

3. Statistical Techniques Driven Data Mining Process

3.1 Foundation of Data Mining

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time. Data mining takes this evolutionary process beyond retrospective data access and navigation to prospective and proactive information delivery. Data mining is ready for application in the business community because it is supported by three technologies that are now sufficiently mature:

- Massive data collection
- Powerful multiprocessor computers
- Data mining algorithms

Data mining derives its name from the similarities between searching for valuable business information in a large database — for example, finding linked products in gigabytes of store scanner data — and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities.

Large amount of data generated by organizations worldwide is mostly unorganized. If data is organized one can generate/extract meaningful and useful information to convert unorganized data into organized data. Normally the concept of DBMS is implemented though a database in management systems is embedded with a query language popularly known as SQL server. The use of SQL particularly in unorganized large databank is not always adequate to meet the end user requirements. Data mining is the technique of abstracting meaningful information from large and unorganized databanks. It involves the process of performing automated abstraction and generating predictive information from large databanks. The abstraction of meaningful

large databanks can also be known as knowledge discovery. The data mining process uses a variety of analysis tools to determine the relationship between data and the databank and to use the same to make valid prediction. Data mining techniques are a result of integration of various techniques from multiple disciplines such as statistics, machine learning, pattern recognition, neural networks, image processing, etc.

In other words it can be described as Competitive advantage requires abilities. Abilities are built through knowledge. Knowledge comes from data. The process of extracting knowledge from data is called Data Mining.

3.2 Data Mining Process

Data mining is an iterative process that typically involves the following phases. The general phases in the data mining process to abstract knowledge are outlined as under:

1 Problem Definition

This initial phase is for understanding the problem and domain environment in which the problem occurs. At this stage data mining experts, business experts, and domain experts work closely together to define the project objectives and the requirements. Problem definition specifies the limits within which problem need to be solved. The object is clear then it is translated into a data mining problem definition. In the problem definition phase, data mining tools are not yet required.

2 Data Understanding

The data-understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data, or to detect interesting subsets to form hypotheses for hidden information.

3

Data Preparation

The data preparation phase covers all activities to construct the final dataset (data that will be fed into the modeling tool(s)) from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection as well as transformation and cleaning of data for modeling tools.

4 Creation of database for data mining

This phase is to create a database where the data to be mined are stored for knowledge acquisition. Creating a database does not require to create a specialized database management system. Even a flat file or a spreadsheet to store data. Data warehouse is also a kind of data storage where large amount of data is stored for data mining. The creation of data mining database consumes about 50% to 90% of the overall data mining process.

5 Exploring the database

This phase is to select and examine Important data sets of a data mining database in order to determine their feasibility to solve the problem. Exploring the database is a time-consuming process and requires a good user interface and computer system with good processing speed.

6 Preparation for creating a data mining model

This phase is to select variables to act as predictors. New variables are also built depending upon the existing variables along with defining the range of variables in order to support imprecise information.

7 Building a data mining model

This phase is to create multiple data mining models and to select the best of these models. Building a data mining model is an interactive process, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. The selected data mining model can be a decision tree, an artificial neural network, or an association rule model. Typically, there

are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often selected.

8 Evaluation of data mining model

At this stage the built model or models that appear to have high quality, from a data analysis perspective. It is important to evaluate the accuracy of the selected data mining model. In data mining, the evaluating parameter is data accuracy in order to test working model. This is because the information generated in the simulated environment varies from the external environment. The errors that occur during the evaluation phase need to be recorded and the cost and time involved in rectifying the error need to be estimated. External validation is also needed to be performed in order to check whether the selected model performs correctly when provided real world values. A key objective is to determine if there is some important issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

9 Deployment of the data mining model

This phase is to deploy the built data mining model and evaluate with external working environment. A monitoring system should monitor the working of the model and generate reports about its performance. The information in the report helps to enhance the performance of selected data mining model. The purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the end-user can use it. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process.

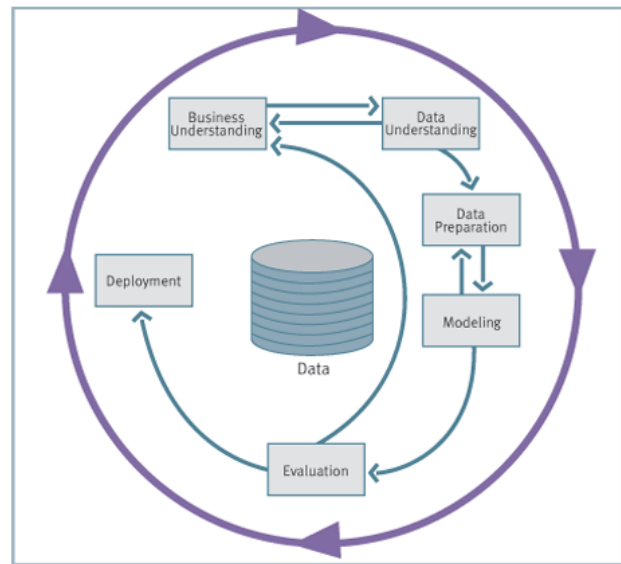


Fig 3.1: Phases of data mining process

3.3 Architecture for Data Mining

Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry

keying errors.

Data mining techniques can yield the benefits of automation on existing software and hardware platforms, and can be implemented on new systems as existing platforms are upgraded and new products developed. When data mining tools are implemented on high performance parallel processing systems, they can analyze massive databases in minutes. Faster processing means that users can automatically experiment with more models to understand complex data. High speed makes it practical for users to analyze huge quantities of data. Larger databases, in turn, yield improved predictions.

To best apply these advanced techniques, they must be fully integrated with a data warehouse as well as flexible interactive business analysis tools. Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Furthermore, when new insights require operational implementation, integration with the warehouse simplifies the application of results from data mining. The resulting analytic data warehouse can be applied to improve business processes throughout the organization, in areas such as promotional campaign management, fraud detection, new product rollout, and so on. Following figure illustrates architecture for advanced analysis in a large data warehouse.

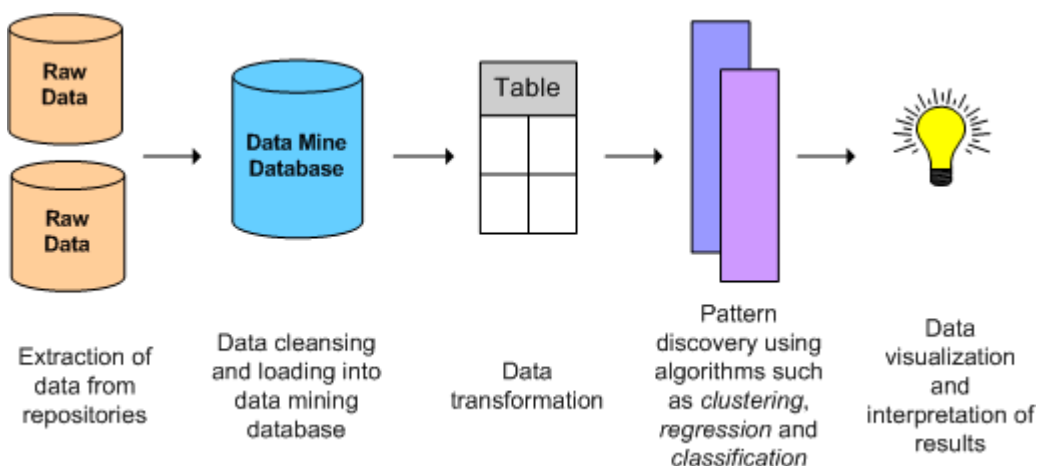


Fig 3.2: Steps taken in Data Mining

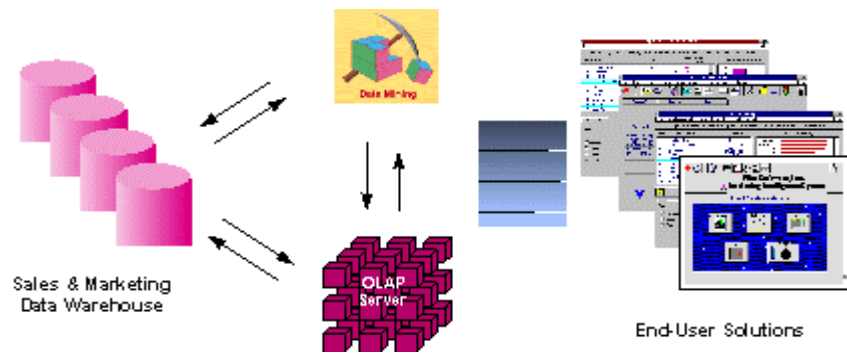


Fig 3.3: Integrated Data Mining Architecture

The ideal starting point is a data warehouse containing a combination of internal data tracking all customer contact coupled with external market data about competitor activity. Background information on potential customers also provides an excellent basis for prospecting. This warehouse can be implemented in a variety of relational database systems: Sybase, Oracle, Redbrick, and so on, and should be optimized for flexible and fast data access.

An OLAP (On-Line Analytical Processing) server enables a more sophisticated end-user business model to be applied when navigating the data warehouse. The multidimensional structures allow the user to analyze the data as they want to view their business summarizing by product line, region, and other key perspectives of their business. The Data Mining Server must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure. An advanced, process-centric metadata template defines the data mining objectives for specific business issues like campaign management, prospecting, and promotion optimization. Integration with the data warehouse enables operational decisions to be directly implemented and tracked. As the warehouse grows with new decisions and results, the organization can continually mine the best practices and apply them to future decisions.

This design represents a fundamental shift from conventional decision support systems. Rather than simply delivering data to the end user through query and reporting software, the Advanced Analysis Server applies users' business models directly to the warehouse and returns a proactive analysis of the most relevant information. These results enhance the metadata in the OLAP Server by providing a dynamic metadata layer that represents a distilled view of the data. Reporting, visualization, and other analysis tools can then be applied to plan future actions and confirm the impact of those plans.

3.4 Data Mining Techniques

There are many different techniques used to perform data mining tasks. These techniques not only require specific types of data structure, but also imply certain types of algorithmic approaches. Data mining techniques provide a way to use data mining tasks in order to predict solution sets for a problem and a level of confidence about the predicted solution in terms of consistency of prediction and in terms of frequency of correct predictions. In this chapter, we briefly introduced some of the common data mining techniques. Data mining techniques include:

1. Statistics
2. Machine learning
3. Decision Trees
4. Neural Networks
5. Genetic Algorithms
6. Association Rules
7. Clustering

1 Statistics

There have been many statistical concepts that are the basis for data mining techniques.

1 Point Estimation

Point estimation refers to the process of estimating a population parameter, θ ,

by an estimate of the parameter, θ . This can be done to estimate mean, variance, standard deviation, or any other statistical parameter. Often the estimate of the parameter for a general population may be made by actually calculating the parameter value for a population sample. An estimator technique may also be used to estimate or predict the value of missing data. The *bias* of an estimator is the difference between the expected value of the estimator and the actual value:

$$\text{Bias} = E(\theta) - \theta$$

An *unbiased* estimator is one whose bias is 0. While point estimator for small data sets may actually be unbiased, for larger database applications we could expect that most estimators are biased.

One measure of the effectiveness of an estimate is the *mean square error* (MSE), which is defined as the expected value of the squared difference between the estimate and the actual value:

$$\text{MSE}(\theta) = E(\theta - \theta)^2$$

The *square error* is often examined for a specific prediction to measure accuracy rather than to look at the average difference.

2 Model Based on Summarization

There are many basic concepts that provide an abstraction and summarization of the data as a whole. The basic well-known statistical concepts such as *mean*, *standard deviation*, *median* and *mode* are simple models of the underlying population. Fitting a population to a specific *frequency distribution* provides an even better model of the data. Of course, doing this with large databases that have multiple attributes, have complex and/or multimedia attributes, and are constantly changing is not practical.

There are also many well-known techniques to display the structure of the data graphically: For example, *histogram* shows the distribution of the data. A *Box*

plot is more sophisticated technique that illustrates several different features of the population at once.

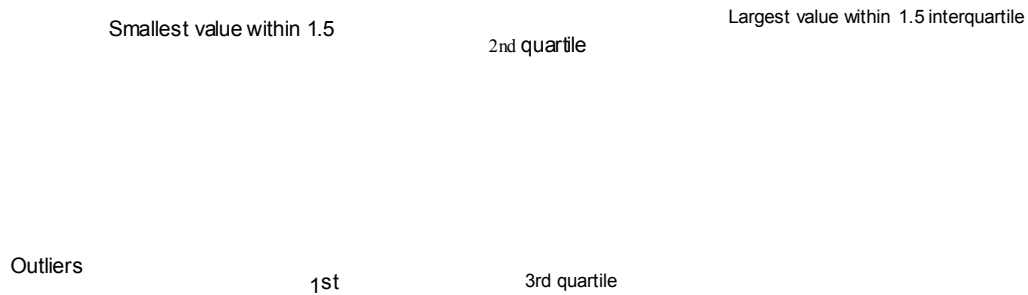


Fig 3.4: Example of Box Plot Technique

Above figure shows a sample box plot. The *total range* of the data values is divided into four equal parts called *quartiles*. The box in the center of the figure shows the range between the first, second, and third quartiles. The line in the box shows the median. The lines extending from either end of the box are the values that are a distance of 1.5 of the interquartile range from the first and third quartiles, respectively. Outliers are shown as points beyond these values.

Another visual technique to display data is called a *scatter diagram*. This is a graph on a two-dimensional axis of points representing the relationships between x and y values. By plotting the actually observable (x, y) points as seen in a sample, a visual image of some derivable may be seen.

3 Bayes Theorem

With statistical inference, information about a data distribution are inferred by examining data that follow that distribution. Given set of data $X = \{x_1, x_2, x_3, \dots, x_n\}$, a data mining problem is to uncover properties of the distribution from which the set comes. Bayes rule is a technique to estimate the likelihood of a property given the set of data as evidence or input. Suppose that either hypothesis h_1 or hypothesis h_2 must occur, but not both. Also suppose that x_i is an observable event.

In Bayes Rule or Bayes Theorem $P(h_1 / x_i)$ is called the posterior probability,

while $P(h_1)$ is the prior probability associated with hypothesis h_1 . $P(x_i)$ is the probability of the occurrence of data value x_i and $P(x_1 / h_1)$ is the conditional probability that, given a hypothesis, the tuple satisfies it.

$$P(h_1 / x_i) = \frac{P(x_i / h_1) P(h_1)}{P(x_i)}$$

Bayes rule allows us to assign probabilities of hypothesis of given a data value, $P(h_1 / x_i)$.

3.4.1.4 Hypothesis Testing

Hypothesis testing attempts to find a model that explains the observed data by first creating a hypothesis and then testing the hypothesis against the data. This is in contrast to most data mining approaches, which create the model from the actual data without guessing what it is first. The actual data itself drive the model creation. The hypothesis usually is verified by examining a data sample. If the hypothesis holds for the sample, it is assumed to hold for the population in general. Given a population, the initial or assumed hypothesis to be tested, **H₀** is called the *null hypothesis*. Rejection of the null hypothesis causes another hypothesis, **H₁** called the *alternative hypothesis*, to be made.

One technique to perform hypothesis testing is based on the use of the chi-squared statistic. Actually, there is a set of procedures referred to as chi-squared. These procedures can be used to test the association between two observed variable values and to determine if a set of observed variable values is statistically significant. A hypothesis is first made, and then the observed values are compared based on this hypothesis. Assuming that **O** represents the observed data and **E** is the expected values based on the hypothesis, the *chi-squared statistic* is defined as:

$$\text{Chi-squared statistic} = \sum (O - E)^2$$

E

When comparing a set of observed variable values to determine statistical significance, the values are compared to those of the expected case. This may be the uniform distribution. We could look at the ratio of the difference of each observed score from the expected value over the expected value. However, since the sum of these scores will always be 0, this approach cannot be used to compare different samples to determine how they differ from the expected values. The solution to this is the same as we saw with mean square error – 0. Statistical tables allow the actual value to be evaluated to determine its significance.

5 Regression and Correlation

Both *bivariate regression and correlation* can be used to evaluate the strength of a relationship between two variables. Regression is generally used to predict future values based on past values by fitting a set of points to a curve. Correlation, however, is used to examine the degree to which the values for two variables behave similarly.

Linear regression assumes that a linear relationship exists between the input data and the output data. The common formula for a linear relationship is used in the model:

$$y = c_0 + c_1x_1 + c_2x_2 + c_3x_3 + \dots + c_nx_n$$

Here there are n input variables, which are called *predictor or regressions*; one output variable (the variable being predicted), which is called the *response*; and $n+1$ constants, which are chosen during the modeling process to match the input samples. This is sometimes called *multiple linear regression* because there is more than one predictor.

Two different data variables X and Y , may behave very similarly, *Correlation* is the problem of determining how much alike the two variables actually are. One standard formula to measure linear correlation is the *correlation coefficient* 'r'.

Given two variables X and Y , the correlation coefficient is a real value $r \in [-1, 1]$. A positive number indicates a positive correlation, whereas a negative number indicates a negative correlation. Here negative correlation indicates that one variable increases while the other decreases in value. The closer the value of r to 0, the smaller the correlation. A perfect relationship exists with a value 1 or -1, whereas no correlation exists with a value of 0. When looking at a scatter plot of the two variables, the closer the values are to a straight line, the closer the r -value is to 1 or -1. The value for r is defined as

$$r = \frac{\sum (x_i - X) (y_i - Y)}{\sqrt{\sum (x_i - X)^2 \sum (y_i - Y)^2}}$$

Where X and Y are the means of X and Y , respectively. When two data variables have a strong correlation, they are similar. Thus, the correlation can be used to define similarity for clustering or classification.

1 Machine Learning

Machine Learning is a process capable of independently acquiring data and integrating that data to generate useful knowledge. The concept of machine learning is implemented by way of computing software system that act as human being who learns from experience, analyses the observations made and self-improves providing increased efficiency and effectiveness.

The processes by which most modern predictive models are developed are *adaptive*, using numerical methods that adjust model parameters based upon analysis of data and model performance. The machine learning process generates a predictive model through mechanical analysis of data and introspection; model parameters are determined without human involvement.

Machine learning is the area of Artificial Intelligence (AI) that examines how to write programs that can learn. In data mining, machine learning is often used for

prediction or classification. With machine learning, the computer makes a prediction and then, based on feedback as to whether it is correct, “learns” from this feedback. It learns through examples, domain knowledge, and feedback. When a similar situation arises in the future, this feedback is used to make the same prediction or to make a completely different prediction. Statistics are very important in machine learning programs because the results of the predictions must be statistically significant and must perform better than a naive prediction. Applications that typically use machine learning techniques include speech recognition, training moving robots, classification of astronomical structures, and game playing.

When machine learning is applied to data mining tasks, a model is used to represent the data (such as a graphical structure like a neural network or a decision tree). During the learning process, a sample of the database is used to train the system to properly perform the desired task. Then the system is applied to the general database to actually perform the task. This predictive modeling approach is divided into two phases. During the training phase, historical or sampled data are used to create a model that represents those data. It is assumed that this model is representative not only for this sample data, but also for the database as a whole and for future data as well. The testing phase then applies this model to the remaining and future data.

2 Decision Trees

Predicting future outcomes and identifying factors that can produce a desired effect are often the main goals of data analysis and data mining. Decision trees are one of the most popular methods of predictive modeling for data mining purposes because they provide interpretable rules and logic statements that enable more intelligent decision-making. Decision tree is a tree-shaped structure, which represents a predictive model used in classification, clustering, and prediction tasks. Decision trees use a “divide and conquer” technique to split the problem search space into subsets. In data mining and machine learning, a decision tree is a predictive model that is a mapping from observations about an item to conclusions about its target value. More

descriptive names for such tree models are classification tree. In a decision tree, each branch of the tree represents a classification question while the leaves of the tree represents the partition of the classified information. Decision trees are generally suitable for tasks related to clustering and classification. It helps generate rules that can be used to explain the decision being taken. Decision Trees are excellent tools for helping to choose between several courses of action. They provide a highly effective structure within which it can be laid out options and investigate the possible outcomes of choosing those options. They also help you to form a balanced picture of the risks and rewards associated with each possible course of action.

3.4.3.1 Decision Tree Representation

A decision tree is an arrangement of tests that prescribes an appropriate test at every step in an analysis. In general, decision trees represent a disjunction of conjunctions of constraints on the attribute-values of instances. Each path from the tree root to a leaf corresponds to a conjunction of attribute tests, and the tree itself to a disjunction of these conjunctions.

More specifically, decision trees classify instances by sorting them down the tree from the root node to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute.

An instance is classified by starting at the root node of the decision tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated at the node on this branch and so on until a leaf node is reached.

Diagram

- Each non-leaf node is connected to a test that splits its set of possible answers into subsets corresponding to different test results.
- Each branch carries a particular test result's subset to another node.
- Each node is connected to a set of possible answers.

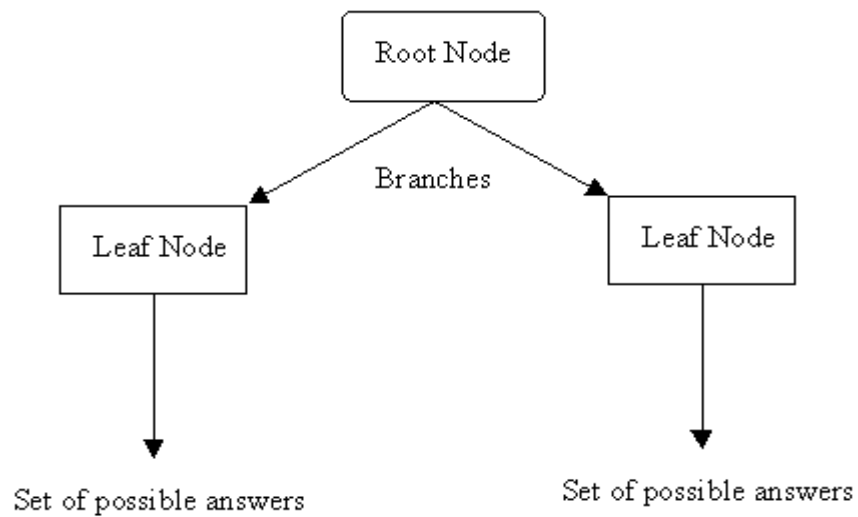
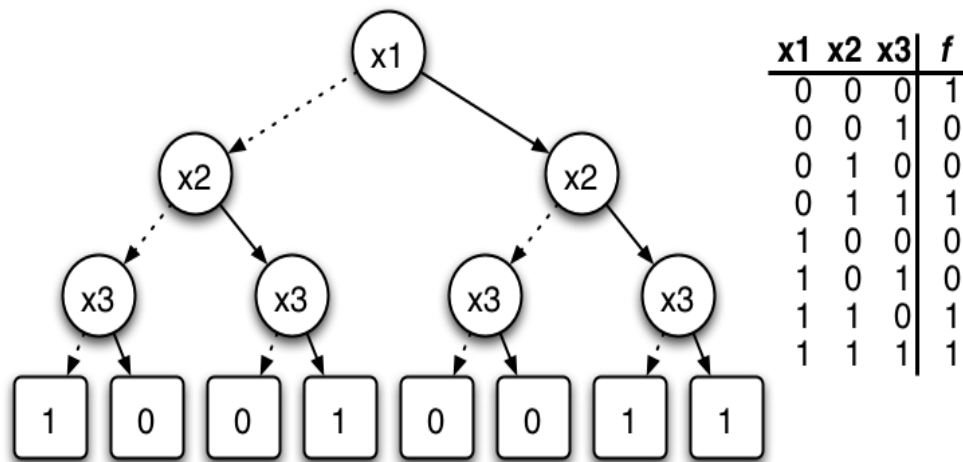


Fig 3.5: Decision Tree Representation

3.4.3.2 Binary Decision Tree

The following figure shows a binary decision tree and a truth table, each representing the function $(x_1, x_2, x_3, x_3, x_4)$. In the tree on the left, the value of the function can be determined for a given variable assignment by following a path down the graph to a terminal. In the figure below, a dotted (solid) line represents an edge to a low (high) child. Therefore, to find $(x_1=0, x_2=1, x_3=1)$, begin at x_1 , traverse down the dotted line to x_2 (since x_1 has an assignment to 0), then down two solid lines (since x_2 and x_3 each have an assignment to one). This leads to the terminal, 1 which is the value of $f(x_1=0, x_2=1, x_3=1)$.

The binary decision tree of the left figure can be transformed into a binary decision diagram by maximally reducing it according to the two reduction rules. The resulting binary decision diagram is shown in the right figure.



Binary decision tree and truth table for the function $f(x_1, x_2, x_3) = -x_1 * -x_2 * -x_3 + x_1 * x_2 + x_2 * x_3$

Fig 3.6: Binary Decision Tree with Truth Table

3.4.3.3 Solution of problem using Decision Tree

Following example of golf club is illustrating the decision tree. The manager of golf club having some trouble with his customer attendance. There are days when everyone wants to play golf and the staff is overworked. On other days, for no apparent reason, no one plays golf and staff has too much slack time. Manager's objective is to optimize staff availability by trying to predict when people will play golf. To accomplish that he needs to understand the reason people decide to play and if there is any explanation for that. He assumes that weather must be an important underlying factor, so he decides to use the weather forecast for the upcoming week. So during two weeks he has been recording following factors.

- The outlook, whether it was sunny, overcast or raining.
- The temperature (in Fahrenheit) .
- The relative humidity in percent.

- Whether it was windy or not
- Whether people attended the golf club on that day.

The dataset is to be compiled into a table containing 14 rows and 5 columns as shown below.

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

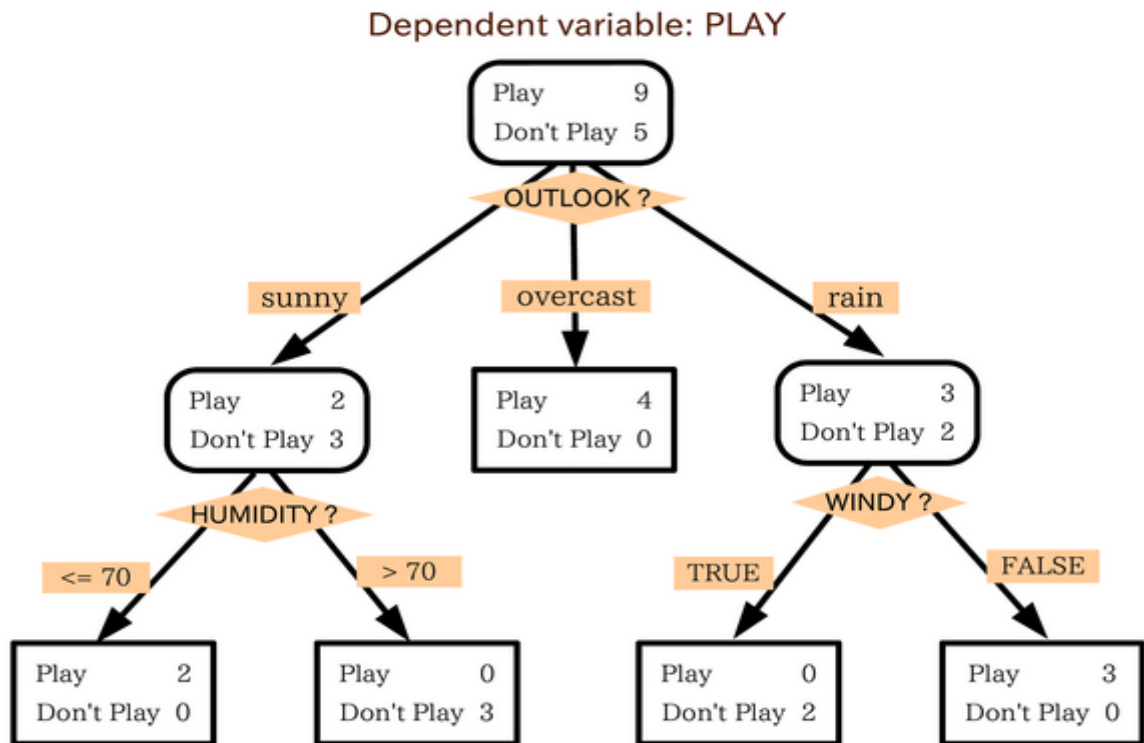


Fig 3.7: Example of Decision Tree

A decision tree is a model of the data that encode the distribution of the class label in terms of the predictor attributes. It is a directed acyclic graph in form of a tree. The top node represents all the data. the classification tree algorithm concludes that the best way to explain the dependent variable, play, is by using the variable “outlook”. Using the categories of the variable outlook, three different groups were found.

- One that plays golf when the whether is sunny.
- One that plays when the weather is clouded.
- One that plays when it's raining.

So it can be concluded that, when the outlook is overcast, people always play golf, and there are some fanatical who plays golf even in the rain. Then sunny group should be divided into two groups. It is also seen from the decision tree that the people do not like to play golf when humidity is higher than seventy percent.

Rain category can also be divided into two and it is founded that people will also not play golf, if it is windy.

Lastly, short solution of this problem given by the classification tree is that, most of staff should be dismissed on days that are sunny and humid or on rainy days that were windy, because almost no one is going to play golf on those days. On days when lot of people will play golf, more staff can be hired.

Following are the decision tree algorithms.

- CART
- ID3
- C4.5
- CHAID
- Rainforest
- Approximate Methods
- CLOUDS

3 Neural Networks

Neural networks is non-predictive mode, often referred to as *artificial neural networks* to distinguish them from biological neural networks, are modeled after the working of the human brain. The *neural networks* are actually an information processing system that consists of a graph representing the processing system as well as various algorithms that access that graph. As with the human brain, the *neural networks* consists of many connected processing elements. The *neural networks*, then, is structured as a directed graph with many nodes (processing elements) and arcs (interconnections) between them. The nodes in the graph are like individual neurons, while the arcs are their interconnections. Each of these processing elements functions independently from the other and

uses only local data (input and output to the node) to direct its processing. This feature facilitates the use of *neural networks* in a distributed and/or parallel environment.

The *neural networks* approach, like decision trees, requires that a graphical structure be built to represent the model and then that the structure be applied to the data. The *neural networks* can be viewed as a directed graph with source (*input*), sink (*output*), and internal (*hidden*) nodes. The input nodes exist in an *input layer*, while the output nodes exist in an *output layer*. The hidden nodes exist over one or more *hidden layer*.

A typical feedforward network has neurons arranged in a distinct layered topology. The input layer is not really neural at all: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks that are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better.

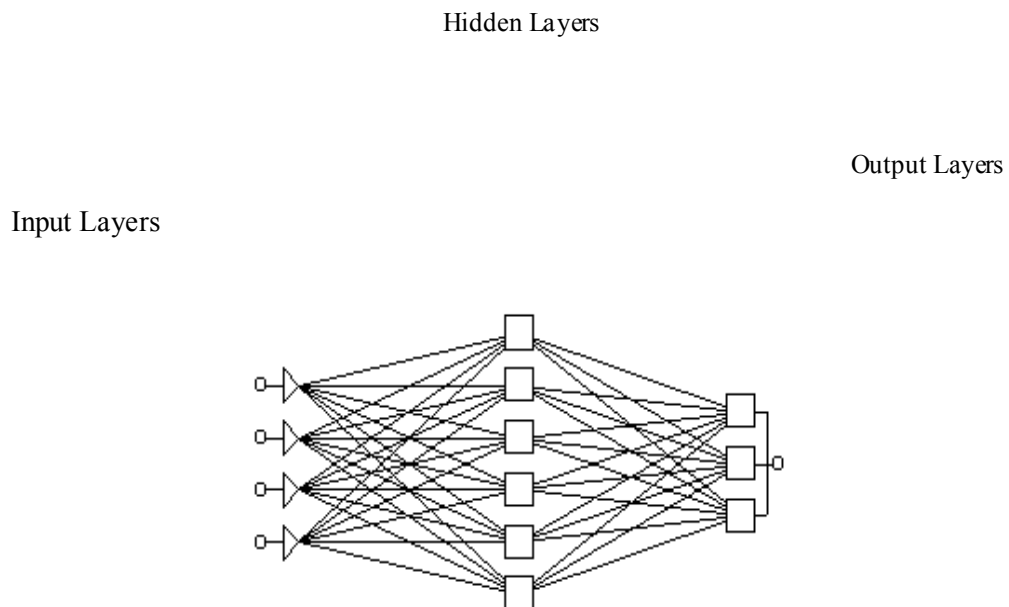


Fig 3.8: Representation of Neural Network

When the network is executed (used), the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer, and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.

To perform the mining task, a tuple is input through the input nodes and the output node determines what the prediction is. Unlike decision trees, which have only input node (the root of the tree), the *neural networks* has one input node for each attribute value to be examined to solve the data mining function. Unlike decision trees, after a tuple is processed, the *neural networks* may be changed to improve future performance. Although the structure of the graph does not change, the labeling of the edges may change.

In addition to solving complex problems, *neural networks* can “learn” from prior applications. That is, if a poor solution to the problem is made, the network is modified to produce a better solution to this problem the next time. The major drawback to the use of *neural networks* is the fact that they are difficult to explain to the end users (unlike decision trees, which are easy to understand). Also, unlike decision trees, *neural networks* usually work only with numeric data.

A Neural network (NN) model is a computational model consisting of three parts:

1. Neural network graph that defines the data structure of the neural network.
2. Learning algorithm that indicates how learning takes place.
3. Recall techniques that determine how information is obtained from the network.

Neural network have been used in pattern recognition, speech recognition and synthesis, medical applications (diagnosis, drug design), fault detection, problem diagnosis, robot control and computer vision. Although *Neural*

networks can solve problems that seem more elusive to other AI techniques, they have a long training time (time during which the learning takes place) and thus are not appropriate for real-time applications. *Neural networks* may contain many processing elements and thus can be used in massively parallel systems.

Artificial neural networks can be classified based on the type of connectivity and learning. The basic type of connectivity is *feedforward*, where connections are only to layers later in the structure. Alternatively, a *neural network* may be *feedback* where some links are back to earlier layers. Learning can be either supervised or unsupervised.

3.4.4.1 Single Layer Linear Network

Following figure shows a sample node, i , in a neural network. Here there are k input arcs coming from nodes 1,2,3, ..., k . with weights of $w_1, w_2, w_3, \dots, w_{ki}$ and input values of $x_{1i}, x_{2i}, x_{ki}, \dots, x_{ki}$. The values that flow on these arcs are shown on dashed arcs because they do not really exist as part of the graph itself. There is one output value y_i produced. During propagation this value is output on all output arcs of the node. The activation function, f_i , is applied to the inputs, which are scaled by applying the corresponding weights. The weights in the *Neural network* may be determined in two ways. In simple cases where much is known about the problem, the weights may be predetermined by a domain expert. The more common approach is to have them determined via a learning process.

The structure of the *Neural network* may also be viewed from the perspective of matrices. Input and weight on the arcs into node i are

$$[x_{1i}, x_{2i}, x_{ki}, \dots, x_{ki}]^T, [w_{1i}, w_{2i}, w_{3i}, \dots, w_{ki}]$$

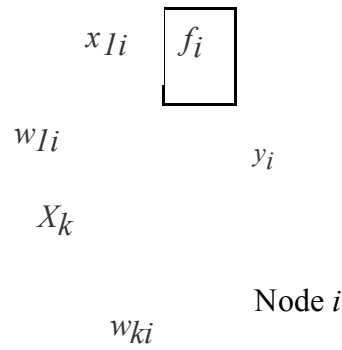


Fig 3.9: Node i

There is one output value from node i , y_i , which is propagated to all output arcs during the propagation process. Using summation to combine the inputs, then, the output of a node is

$$y_i = f_i \left[\sum_{j=1}^k w_{ji} x_{ji} \right] = f_i [w_{1i}, w_{2i}, w_{3i}, \dots, w_{ki}]$$

Here f_i is the activation function. Because *neural networks* are complicated, domain experts and data mining experts are often advised to assist in their use. This in turn complicates the process.

Overfitting occurs when the *neural network* is trained to fit one set of data almost exactly. The error that occurs with the given training data is quite small. However, when new data are examined, the error is very large. In effect, the *neural network* has “memorized” the training set, and cannot generalize to more data. Larger and more complicated *neural network* can be trained to represent more complex functions. To avoid overfitting, smaller *neural networks* are advisable. However, this is difficult to determine beforehand. Another approach that can be used to avoid overfitting is to stop the learning process early.

3.4.5 Genetic Algorithm

Genetic algorithms are example of *evolutionary computing* methods and are optimization-type algorithms. Given a population of potential problem solution,

evolutionary computing expands this population with new and potentially better solutions. The basis for evolutionary computing algorithms is biological evolutions, where over time evolution produces the best or “fittest” individuals. Chromosomes, which are DNA strings, provide the abstract model for a living organism. Subsections of the chromosomes, which are called *genes*, are used to define different traits of the individual. During reproduction, genes from the parents are combined to produce the genes for the child.

When using genetic algorithms to solve a problem, the first thing, and perhaps the most difficult task, that must be determined is how to model the problem as a set of individuals. In real world, individuals may be identified by a complete encoding of the DNA structure. An individual typically is viewed as an array or tuple of values. Based on the recombination (crossover) algorithms, the values are usually numeric and may be binary strings. These individuals are like a DNA encoding structure for each individual represents an encoding of the major feature needed to model the problem. Each individual in the population is represented as a string of characters from the given alphabet.

A genetic algorithm (GA) is a computational model consisting of following parts:

1. Starting set of individuals, P .
2. Crossover technique
3. Mutation algorithm
4. Fitness function

3.4.5.1 Starting set of individuals, P

Given an alphabet A , an **individual** or **chromosome** is a string $I = I_1, I_2, I_3, \dots, I_n$ where $I_j \in A$. Each character in the string, I_j , is called a **gene**. The values that each character can have are called **alleles**. A **population**, is a set of individuals.

Although individuals are often represented as bit strings, any encoding is possible. An array with non-binary characters could be used, as could more

complicated data structures including trees and arrays. The only real restriction is that the genetic operators (mutation, crossover) must be defined.

3.4.5.2 Crossover technique

In genetic algorithms, reproduction is defined by precise algorithms that indicate how to combine the given set of individuals to produce new ones. These are called *crossover* algorithms. Given two individuals (*parents*) from the population, the crossover technique generates new individuals (*offspring* or *children*) by switching subsequences of the strings. Following figure illustrates the process of crossover. The locations indicating the crossover points are shown in the figure with the vertical lines.

000 000	000 111	000 000 00	000 111 00
111 111	111 000	111 111 11	111 000 11
Parents	Childrens	Parents	Childrens

Fig 3.10: (a) Single crossover

(b) Multiple crossover

In figure 3.10(a) crossover is achieved by interchanging the last three bits of the two strings. In figure 3.10(b) the center three bits are interchanged. The figure shows single and multiple crossover points. There are many variations of the crossover approach, including determining crossover points randomly. A crossover probability is used to determine how many new offspring are created via crossover. In addition, the actual crossover point may vary within one algorithm.

3.4.5.3 Mutation algorithm

As in nature, however, mutations sometimes appear, and these also may be present in genetic algorithms. The mutation operation randomly changes characters in the offspring. A very small probability of mutation is set to determine whether a character should change.

Since genetic algorithms attempt to model nature, only the strong survive. When new individuals are created, a choice must be made about which individuals will

survive. This may be the new individuals, the old ones, or more likely a combination of the two. The third major component of genetic algorithms, then, is the part that determines the best (or fittest) individuals to survive.

3.4.5.4 Fitness function

One of the most important components of a genetic algorithm is determining how to select individuals. A fitness function, f , is used to determine the best individuals in a population. This is then used in the selection process to choose parents. Given an objective by which the population can be measured, the fitness function indicates how well the goodness objective is being met by an individual.

Given a population, P , a **fitness function**, f is a mapping $f:P \rightarrow R$. The simplest selection process is to select individuals based on their fitness.

$$p_{l_j} = \frac{f(l_j)}{\sum f(l_j)}$$

Here p_{l_j} is the probability of selecting individual l_j . This type of selection is called *roulette wheel selection*. One problem with this approach is that it is still possible to select individuals with a very low fitness value. In addition, when the distribution is quite skewed with a small number of extremely fit individuals, these individuals may be chosen repeatedly. In addition, as the search continues, the population becomes less diverse so that the selection process has little effect.

3.4.6. Association Rules

Association rule discovery techniques are generally applied to databases of transactions where each transaction consists of a set of items. In such a framework the problem is to discover all associations and correlations among data items where the presence of one set of items in a transaction implies (with a certain degree of confidence) the presence of other items. Association rules provide information of this type in the form of "if-then" statements. These rules

are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature.

In addition to the antecedent (the "if" part) and the consequent (the "then" part), an association rule has two numbers that express the degree of uncertainty about the rule. In association analysis the antecedent and consequent are sets of items (called itemsets) that are disjoint (do not have any items in common).

The first number is called the **support** for the rule. The support is simply the number of transactions that include all items in the antecedent and consequent parts of the rule. (The support is sometimes expressed as a percentage of the total number of records in the database.)

The other number is known as the confidence of the rule. **Confidence** is the ratio of the number of transactions that include all items in the consequent as well as the antecedent (namely, the support) to the number of transactions that include all items in the antecedent.

For example, if a supermarket database has 100,000 point-of-sale transactions, out of which 2,000 include both items A and B and 800 of these include item C, the association rule "If A and B are purchased then C is purchased on the same trip" has a support of 800 transactions (alternatively $0.8\% = 800/100,000$) and a confidence of $40\% (=800/2,000)$. One way to think of support is that it is the probability that a randomly selected transaction from the database will contain all items in the antecedent and the consequent, whereas the confidence is the conditional probability that a randomly selected transaction will include all the items in the consequent given that the transaction includes all the items in the antecedent.

Lift is one more parameter of interest in the association analysis. Lift is nothing but the ratio of Confidence to Expected Confidence. Expected Confidence in this case means, using the above example, "confidence, if buying A and B does not enhance the probability of buying C." It is the number of transactions that include the consequent divided by the total number of transactions. Suppose the number of total number of transactions for C is 5,000. Thus Expected

Confidence is $5,000 / 1,00,000 = 5\%$. For our supermarket example the Lift = Confidence/Expected Confidence = $40\% / 5\% = 8$. Hence Lift is a value that gives us information about the increase in probability of the "then" (consequent) given the "if" (antecedent) part.

3.4.6.1 Basic Algorithms for Association Rules

3.4.6.1.1 Apriori Algorithm:

The Apriori algorithm is the most well known association rule algorithm and is used in most commercial products. It uses the following property, which we call the *large itemset property*: Any subset of a large itemset must be large.

The large itemsets are also said to be *downward closed* because if an itemset satisfies the minimum support requirements, so do all subsets of its subsets. Looking at the contrapositive of this, if we know that an itemset is small, we need not generate any superset of it as candidates because they also must be small. We use the lattice shown in figure (a) to illustrate the concept of this property. In this case there are four items {A, B, C, D}. The lines in the lattice represent the subset relationship, so the large itemset property says that any set in a path above an itemset must be large if the original itemset is large. In figure (b) the nonempty subsets of ACD are seen as {AC, AD, CD, A, C, D}.

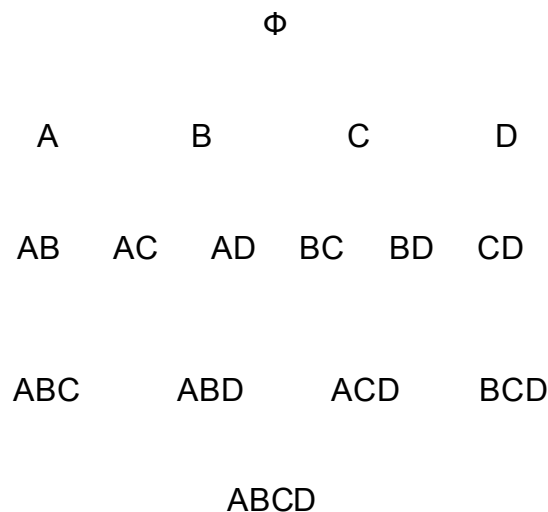


Fig 3.11(a): Lattice of itemsets {A, B, C, D}

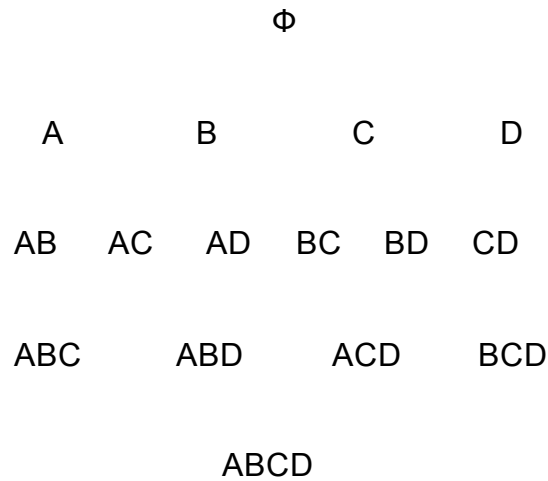


Fig 3.11(b): Subsets of ACD

The basic idea of the Apriori algorithm is to generate candidate itemsets of a particular size and then scan the database to count these to see if they are large. During scan i , candidates of size i , C_i are counted. Only those candidates that are large are used to generate candidates for the next pass. That is L_i are used to generate C_{i+1} . An itemset is considered as a candidate only if All Students its subsets also are large. To generate candidates of size $i+1$, joins are made of large itemsets found in the previous pass.

An algorithm called Apriori-Gen is used to generate the candidate itemsets for each pass after the first. All Students singleton itemsets are used as candidates in the first pass. Here the set of large itemsets of the previous pass, L_{i-1} , is joined with itself to determine the candidates. Individual itemsets must have All Students but one item in common in order to be combined.

3.4.6.1.2 Sampling Algorithm

To facilitate efficient counting of itemsets with large databases, sampling of the database may be used. The original sampling algorithm reduces the number of database scans to one in the best case and two in the worst case. The database sample is drawn such that it can be memory-resident. Then any algorithm, such as Apriori, is used to find the large itemsets for the sample. These are viewed as *potentially large* (PL) itemsets and used as candidates to be counted using the

entire database. Additional candidates are determined by applying the *negative border* function, BD^- , against the large itemsets from the sample. The entire set of candidates is then $C = BD^-(PL) \cup PL$. the negative border function is a generation of the Apriori-Gen algorithm. It is defined as the minimal set of itemsets that are not in PL, but whose subsets are All Students in PL.

3.4.6.1.3 Partitioning

Various approaches to generating large itemsets have been proposed based on a partitioning of the set of transactions. In this case, D is divided into p partitions $D_1, D_2, D_3, \dots, D_p$. Partitioning may improve the performance of finding large itemsets in several ways:

- By taking advantage of the large itemset property, we know that a large itemset must be larger in at least one of the partitions. This idea can help to design algorithms more efficiently than those based on looking at the entire database.
- Partitioning algorithms may be able to adapt better to limited main memory. Each partition can be created such that it fits into main memory. In addition, it would be expected that the number of itemsets to be counted per partition would be smaller than those needed for the entire database.
- By using partitioning, parallel and/or distributed algorithms can be easily created, where a separate machine could handle each partition.
- Incremental generation of association rules may be easier to perform by treating the current state of the database as one partition and treating the new entries as a second partition

The basic partition algorithm reduces the number of database scans to two and divides the database into partitions such that each can be placed into main memory. When it scans the database, it brings that partition of the database into main memory and counts the items in that partition alone. During the first database scan, the algorithm finds all large itemsets in each partition. Although

any algorithm could be used for this purpose, the original proposal assumes that some level-wise approach, such as Apriori.

3.4.6.1.4 Pincer-Search Algorithm

Discovering frequent itemsets is a key problem in important data mining applications, such as the discovery of association rules, strong rules, episodes, and minimal keys. Typical algorithms for solving this problem operate in a bottom-up, breadth-first search direction. The computation starts from frequent 1-itemsets (the minimum length frequent itemsets) and continues until all maximal (length) frequent itemsets are found. During the execution, every frequent itemset is explicitly considered. Such algorithms perform well when all maximal frequent itemsets are short. However, performance drastically deteriorates when some of the maximal frequent itemsets are long. We present a new algorithm, which combines both the bottom-up and the top-down searches. The primary search direction is still bottom-up, but a restricted search is also conducted in the top-down direction. This search is used only for maintaining and updating a new data structure, the maximum frequent candidate set. It is used to prune early candidates that would be normally encountered in the bottom-up search. A very important characteristic of the algorithm is that it does not require explicit examination of every frequent itemset. Therefore, the algorithm performs well even when some maximal frequent itemsets are long. As its output, the algorithm produces the maximum frequent set, i.e., the set containing all maximal frequent itemsets, thus specifying immediately all frequent itemsets. We evaluate the performance of the algorithm using well-known synthetic benchmark databases, real-life census, and stock market databases. The improvement in performance can be up to several orders of magnitude, compared to the best previous algorithms.

3.4.6.1.5 FP-Tree Growth Algorithm

FP-growth algorithm is an efficient algorithm for mining frequent patterns. It scans database only twice and does not need to generate and test the candidate sets that is quite time consuming. The efficiency of the FP-growth algorithm outperforms previously developed algorithms. But, it must recursively

generate huge number of conditional FP-trees that requires much more memory and costs more time. In this paper, we present an algorithm, CFPmine, that is inspired by several previous works. CFPmine algorithm combines several advantages of existing techniques. One is using constrained sub-trees of a compact FP-tree to mine frequent pattern, so that it is doesn't need to construct conditional FP-trees in the mining process. Second is using an array-based technique to reduce the traverse time to the CFP-tree. And an unified memory management is also implemented in the algorithm. The experimental evaluation shows that CFPmine algorithm is a high performance algorithm. It outperforms Apriori, Eclat and FP-growth and requires less memory than FP-growth.

3.4.6.1.6 Advance Association Rule Techniques

1. Generalized Association Rules

Using a concept of hierarchy that shows the set relationship between different items, generalized association rules allow rules at different levels. Association rules can be generalized for any and all levels in the hierarchy. A *generalized association rule*, $X \rightarrow Y$, is defined like a regular association rule with the restriction that no item in Y may be above any item X . When generating generalized association rules, all possible rules are generated using one or more given hierarchies. Several algorithms have been proposed to generate generalized rules. The simplest would be to expand each transaction by adding all items above it in any hierarchy.



Fig 3.12: Hierarchy of Association Rule

Above figure shows a partial concept hierarchy for food. This hierarchy shows that Wheat Bread is a type of Bread, which is a type of grain. An association rule of the form Bread \rightarrow Peanut Butter has a lower support and threshold than one of the form Grain \rightarrow Peanut Butter. There obviously are more transactions containing any type of grain than transactions containing Bread. Likewise, Wheat Bread \rightarrow Peanut butter has a lower threshold and support than Bread \rightarrow Peanut Butter.

2. Multiple-Level Association Rules

A variation of generalized rules is *multiple-level association rules*. With multiple-level rules, itemsets may occur from any level in the hierarchy. Using a

variation of the Apriori algorithm, the concept hierarchy is traversed in a top-down manner and large itemsets are generated. When large itemsets are found at level l , large itemsets are generated for level $l + 1$. large k -itemsets at one level in the concept hierarchy are used as candidates to generate large k -itemsets for children at the next level.

Modification to the basic association rule ideas may be changed. We expect that there is more support for itemsets occurring at higher levels in the concept hierarchy. Thus, the minimum support required for association frequency of itemsets at higher levels is much greater than the frequency of itemsets at lower level. Thus, for the reduced minimum support concept, the following rules apply:

- The minimum support for all nodes in the hierarchy at the same level is identical.
- If α_i is the minimum support for level i in the hierarchy and α_{i-1} is the minimum support for level $i-1$, then $\alpha_{i-1} > \alpha_i$.

3.4.7 Clustering

Clustering is an example of data mining task that fits in the descriptive model of data mining. The use of clustering enables you to create new groups and classes based on the study of patterns and relationship between values of data in a data bank. It is similar to classification but does not require you to predefine the groups or classes. Clustering technique is otherwise known as unsupervised learning or segmentation. All those data items that resembles more closely with each other are clubbed together in a single group, also known as clusters.

Clustering is similar to classification in that data are grouped. However, unlike classification, the groups are not predefined. Instead, the grouping is accomplished by finding similarities between data according to characteristics found in the actual data. the groups are called *clusters*, some researchers view clustering as a special type of classification. Here we follow a more conventional

view in that the two are different. Many definitions for cluster have been proposed:

- Set of like elements, elements from different clusters are not alike.
- The distance between points in a cluster is less than the distance between a point in the cluster and any point outside it.

The term similar to clustering is *database segmentation*, where like tuples in a database are grouped together. This is done to partition or segment the database into components that then give the user a more general view of the data.

As illustrated in the following figure, a given set of data may be clustered on different attributes. Here a group of homes in a geographic area is shown. The first type of clustering is based on the location of the home. Homes that are geographically close to each other are clustered together. In the second clustering, homes are grouped based on the size of the house.

Fig 3.13(a): Group of homes

Fig 3.13(b): Geographic distance based

Fig 13.3(c): Size – based

Clustering has been used in many domains, including biology, medicine, anthropology, marketing, and economics. Clustering applications include plant and animal classification, diseases classification, image processing, pattern recognition, and document retrieval. One of the first domains in which clustering was used was biological taxonomy. Recent uses include examining Web log data to detect usage patterns.

When clustering is applied to a real-world database, many interesting problems occur:

- Outlier handling is difficult. Here the elements do not naturally fall into any cluster. They can be viewed as solitary clusters. However, if a clustering algorithm attempts to find larger clusters, these outliers will be forced to be placed in some cluster. This process may result in the creation of poor clusters by combining two existing clusters and leaving the outlier in its own cluster.
- Dynamic data in the database implies that cluster membership may change over time.
- Interpreting the semantic meaning of each cluster may be difficult. However, with clustering, this may not be the case. Thus, when the clustering process finishes creating a set of clusters, the exact meaning of each cluster may not be obvious. Here is where a domain expert is needed to assign a label or interpretation for each cluster.

- There is no one correct answer to a clustering problem. In fact, many answers may be found. The exact number of clusters required is not easy to determine. Again, a domain expert may be required. For example, suppose we have a set of data about plants that have been collected during a field trip, without any prior knowledge of plant classification, if we attempt to divide this set of data into similar groupings, it would not be clear how many groups should be created.
- Another related issue is what data should be used for clustering. Unlike learning during a classification process, where there is some a priori knowledge concerning what the attributes of each classification should be, in clustering we have no supervised learning to aid the process. Indeed, clustering can be viewed as similar to unsupervised learning.

It can be summarized some basic features of clustering against the classification:

- The number of clusters is not known,
- There may not be any a priori knowledge concerning the clusters.
- Cluster results are dynamic.

The clustering problem is stated as shown in following definition. Here we assume that the number of clusters to be created is an input value, k . the actual content (and interpretation) of each cluster, K_j , $1 \leq j \leq k$, is determined as a result of the function definition. Without loss of generality, we will view that the result of solving a cluster problem is that a set of clusters is created: $K = \{K_1, K_2, K_3, \dots, K_k\}$.

Given a database $D = \{t_1, t_2, t_3, \dots, t_n\}$ of tuples and an integer value k , the **clustering problem** is to define a mapping $f : D \rightarrow \{1, 2, 3, \dots, k\}$ where each

t_i is assigned to one cluster K_j , $1 \leq j \leq k$. A **cluster**, K_j , contains precisely those tuples mapped to it; that is, $K_j = \{t_j \mid f(t_j) = K_j, 1 \leq j \leq k, \text{ and } t_j \in D\}$.

A classification of the different types of clustering algorithms is shown in following figure. Clustering algorithms themselves may be viewed as hierarchical or partitional. With *hierarchical* clustering, a nested set of clusters is created.

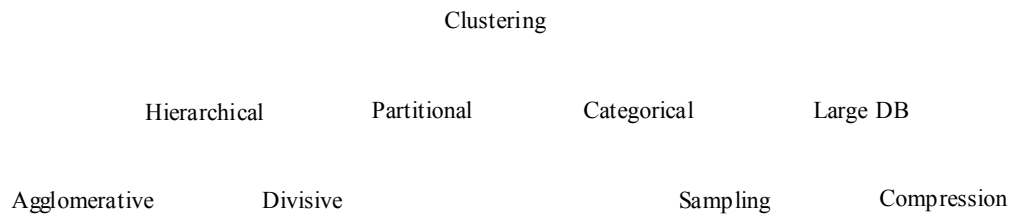


Fig 3.14: Classification of Clusters

Each level in the hierarchy has separate set of clusters. At the lowest level, each item is in its own unique cluster. At the highest level, All Students items belong to the same cluster. With hierarchical clustering, the desired number of clusters is not input. With *partitional* clustering, the algorithm creates only one set of clusters. These approaches use the desired number of clusters to drive how the final set is created. Traditional clustering algorithms tend to be targeted to small numeric databases that fit into memory. These are, however, more recent clustering algorithms that look at categorical data and are targeted to larger, perhaps dynamic, databases. Algorithms targeted to larger databases may adapt to memory constraints by either sampling the database or using data structures, which can be compressed or pruned to fit into memory regardless of the size of the database. Clustering algorithms may also differ based on whether they produce overlapping or non-overlapping clusters. Even though we consider only non-overlapping clusters, it is possible to place an item in multiple clusters. In turn, non-overlapping clusters can be viewed as extrinsic or intrinsic. *Extrinsic* techniques use labeling of the items to assist in the classification process. These algorithms are the traditional classification supervised learning algorithms in which a special input training set is used. *Intrinsic* algorithms do not use any a

priori category labels, but depend only on the adjacency matrix containing the distance between objects.

3.4.7.1 Hierarchical Algorithms

Hierarchical clustering algorithms actually create sets of clusters. Hierarchical algorithms differ in how the sets are created. A tree data structure, called a *dendrogram*, can be used to illustrate the hierarchical clustering technique and the sets of different clusters. The root in a dendrogram tree contains one cluster where All Students elements are together. The leaves in the dendrogram each consist of a single element cluster. Internal nodes in the dendrogram represent new clusters formed by merging the clusters that appear as its children in the tree each level in the tree is associated with the distance measure that was used to merge the clusters. All clusters created at a particular level were combined because the children clusters had a distance between them less than the distance value associated with this level in the tree.

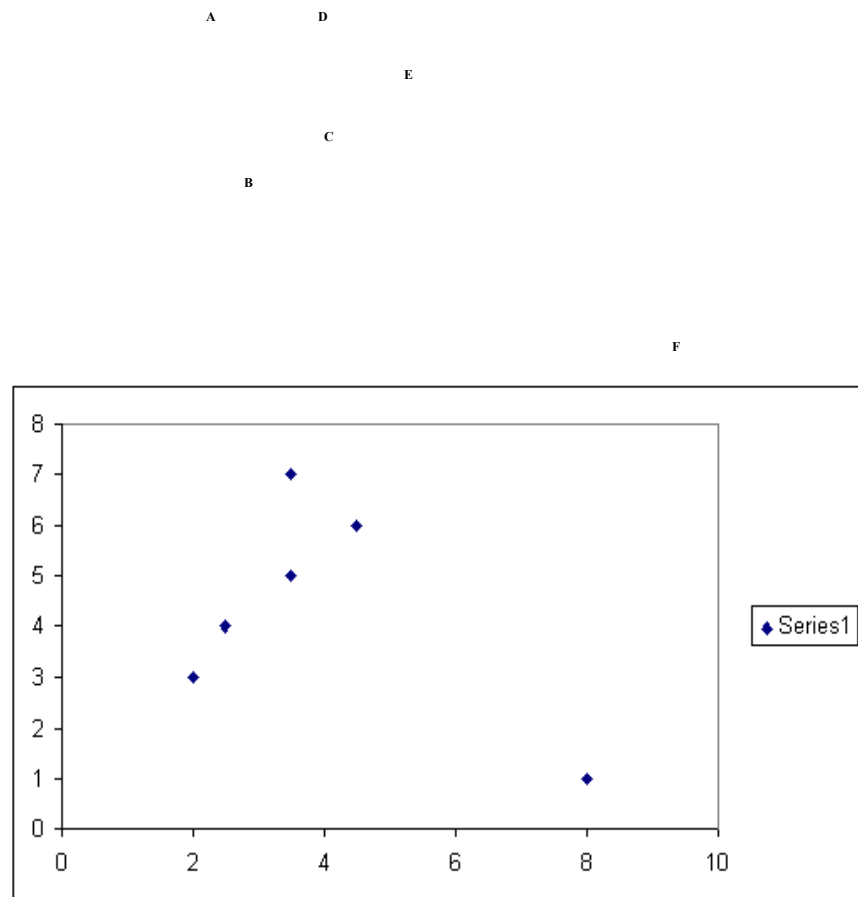


Fig 3.15(a): Six clusters

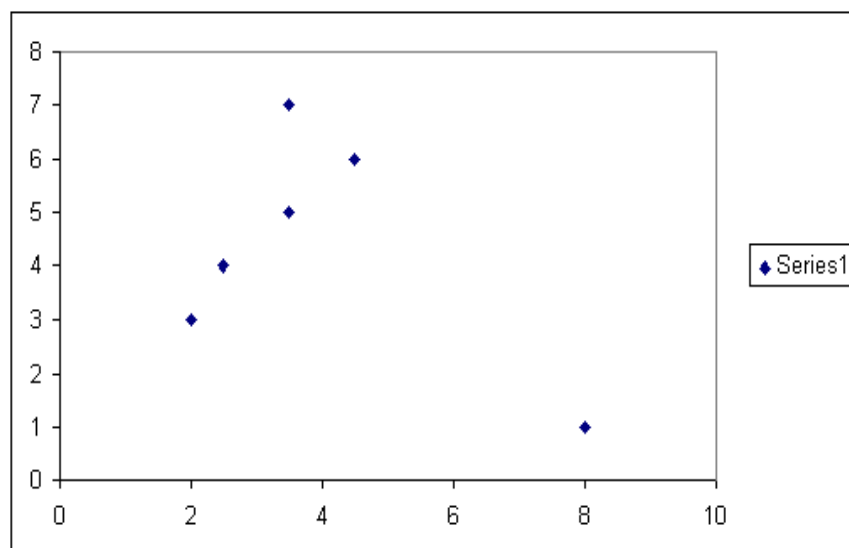


Fig 3.15(b): Four Clusters

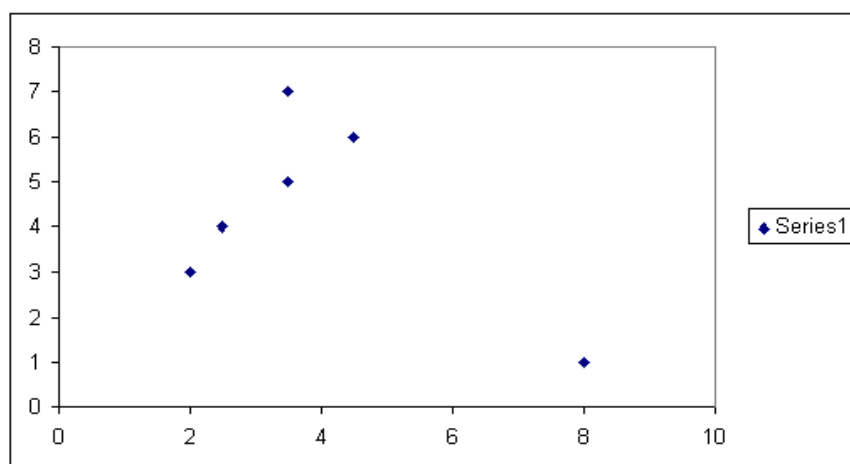


Fig 3.15(c): Three clusters

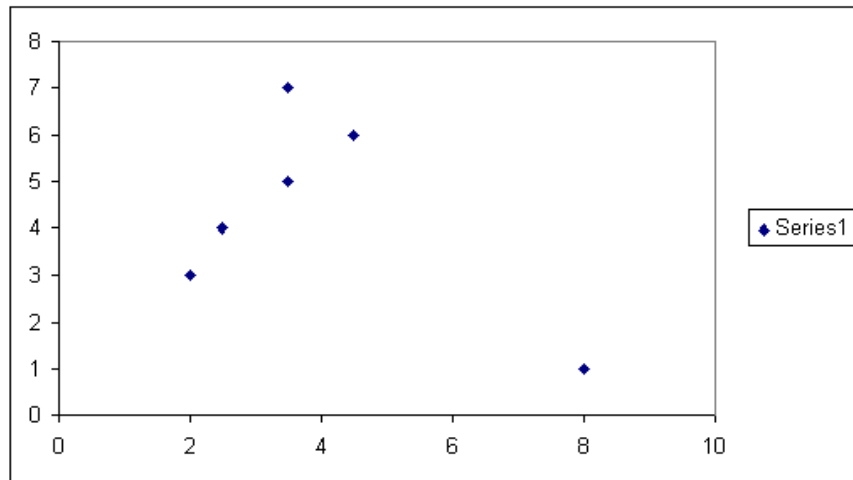


Fig 3.15(d): Two clusters

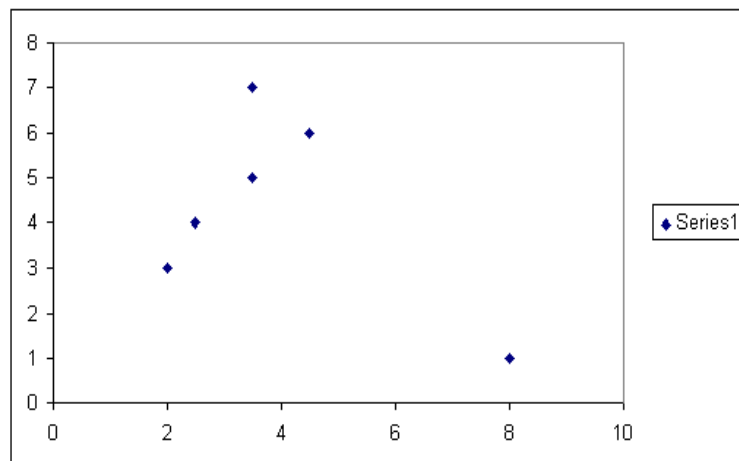


Fig 3.15(e): One cluster

Above figure shows six elements { A, B, C, D, E, F}, to be clustered. Part (a) to (e) of the figure shows five different sets of clusters. In part (a) each cluster is viewed to consist of a single element. Part (b) illustrates four clusters. Here there are two sets of two-element clusters. These clusters are formed at this level

because these two elements are closer to each other than any of the other elements. Part (c) shows a new cluster formed by adding a close element to one of the two-element clusters. In part (d) the two-element and three-element clusters are merged to give a five-element cluster. This is done because these two clusters are closer to each other than to the remote element cluster, {F}. At the last stage, part (e), all six elements are merged.

A B C D E F

Fig 3.16: Dendrogram for above Example

3.4.7.2 Agglomerative Algorithm

Agglomerative algorithms start with each individual item in its own cluster and iteratively merge clusters until all item belongs in one cluster. Different agglomerative algorithms differ in how the clusters are merged at each level.

All agglomerative approaches experience excessive time and space constraints.

The space required for the adjacency matrix is $O(n^2)$ where there are n items to cluster. Because of the iterative nature of the algorithm, the matrix (or subset of it) must be accessed multiple times. Let's consider the following algorithm.

- (1) Start with clusters, and a single sample indicates one cluster.
- (2) Repeat step 3 until cluster number is what we want or 1.
- (3) Find the most similar clusters and , then merge them into one cluster.

What is the most similar cluster pair? Compute distances between each pair of clusters to judge which two clusters have prior opportunity to merge. There are several ways to calculate the distances between cluster and cluster .

3.4.7.3 Single-Linkage Agglomerative Algorithm:

The single link technique is based on the idea of finding maximal connected components in a graph. A *connected component* is a graph in which there exists a path between any two vertices. With the single link approach, two clusters are merged if there is at least one edge that connects the two clusters; that is, if the minimum distance between any two points is less than or equal to the threshold distance being considered. For this reason it is often called the *nearest neighbor* cluster technique.

In single-linkage agglomerative algorithm, defining the distance between two clusters is the shortest distance between a sample in one cluster and a sample in the other cluster.

3.4.7.4 Complete-Linkage Agglomerative Algorithm:

Although the complete link algorithm is similar to the single link algorithm, it looks for cliques rather than connected components. A *clique* is a maximal graph in which there is an edge distance between any clusters so that two clusters are merged if the maximum distance is less than or equal to the distance threshold. In this algorithm, we assume the existence of a procedure, *clique*, which finds all cliques in a graph. As with the single link algorithm, this is expensive because it is an $O(n^2)$ algorithm.

In the complete-linkage agglomerative algorithm, defining the distance between two clusters is the longest distance between a sample in one cluster and a sample in the other cluster.

3.4.7.5 Average-Linkage Agglomerative Algorithm:

In the average-linkage agglomerative algorithm, defining the distance between two clusters is the average distance between a sample in one cluster and a sample in the other cluster ($\frac{1}{n_1 n_2}$ is the number of the cluster).

3.4.7.6 Divisive Clustering

With divisive clustering, all items are initially placed in one cluster and clusters are repeatedly split in two until all items are in their own cluster. The idea is to split up clusters where some elements are not sufficiently close to other elements.

One simple example of a divisive algorithm is based on the Missing Spanning Tree (MST) version of the single link algorithm. Here, however, we cut out edges from MST from the largest to the smallest. Let's see following algorithm.

- (1) Start with just only one cluster. That is, all samples in this one cluster.
- (2) Repeat step 3, 4, 5, 6 until cluster number is the number of samples or what we want.
- (3) Calculate diameter of each cluster. Diameter is the maximal distance between samples in the cluster. Choose one cluster having maximal diameter of all clusters to split.
- (4) Find the most dissimilar sample from cluster . Let depart from the original cluster to form a new independent cluster (now cluster doesn't include sample). Assign all members of cluster to .
- (5) Repeat 6 until members of cluster and don't change.
- (6) Calculate similarities from each member of to cluster and , and let the member owning the highest similarities in move to its similar cluster or . Update members of and .

Here we take a simple example to describe the method above. First, the distance matrix of 5 samples is

.

Our processing steps are as follows:

1. Because there is only one cluster, this cluster has maximal diameter. For a start, we split this cluster.

2. Calculate average distances from one sample to the others. For example, the average distance from x_1 to x_2 , x_3 and x_4 is $\frac{d(x_1, x_2) + d(x_1, x_3) + d(x_1, x_4)}{3}$, and the others:

Sample x_4 has maximal average distance, so extract x_4 from the cluster.

Now we have 2 clusters: $\{x_1, x_2, x_3\}$ and $\{x_4\}$.

3. Find average distances from x_1 , x_2 and x_3 to clusters $\{x_1, x_2, x_3\}$ and $\{x_4\}$.

The distance from x_1 to cluster $\{x_1, x_2, x_3\}$ is minimum, so put x_1 into cluster $\{x_1, x_2, x_3\}$.

Now clusters are updated to $\{x_1, x_2, x_3, x_4\}$ and $\{x_4\}$. Repeat step 6 of the algorithm to check if members of each cluster are updated.

The distance from x_1 to cluster C_1 is also minimum and cluster members don't change again. Go to step 3 of the algorithm. Now there are 2 clusters C_1 and C_2 .

4. The diameter of the cluster C_1 is:

$$=2.$$

The diameter of cluster C_2 is

.

We choose the cluster C_1 to split (has maximal diameter of all clusters).

5. Calculate average distances from one sample to the others in cluster C_1 .

.

So split C_1 into C_{11} and C_{12} . The average distances from x_1 and x_2 to clusters C_{11} and C_{12} are:

Because minimum distance is 3, cluster members of each cluster don't update. Go to step 3 of the algorithm.

6. Now we have 3 clusters C_1 , C_2 , and C_3 . Their diameters are 2, 0, and 3. Because there is only one sample in cluster C_2 , don't think about this cluster. We decide split the cluster C_3 .

7. Split C_3 into C_4 and C_5 . Because cluster members of each cluster don't update, go to step 3.

8. Now we have 4 clusters C_1 , C_2 , C_4 , and C_5 . Only the cluster C_1 has more than one sample and have maximal diameter, so split

9. Split C_1 into C_6 and C_7 . Each sample represents one cluster; so stop (see following figure).

Fig 3.17: An example for hierarchical divisive algorithm

3.4.7.7 Partitional Algorithm

Nonhierarchical or partitional clustering creates the clusters in one step as opposed to several steps. Only one set of clusters is created, although several different sets of clusters may be created internally within the various algorithms. Since only one set of clusters is output, the user must input the desired number, k , of clusters. In addition, some metric or criterion function is used to determine the goodness of any proposed solution. This measure of quality could be average distance between clusters or some other metric. The solution with the best value for the criterion functions is the clustering solution used. One common measure is a squared error metric, which measures the squared distance from each point to the centroid to the associated cluster:

$$\sum_{m=1}^k \sum_{t_{mi}} \text{dis}(C_m, t_{mi})$$

A problem with partitional algorithms is that they suffer from a combinatorial explosion due to the number of possible solutions. Clearly, searching all possible clustering alternatives usually would not be feasible. For example, given a measurement criteria, a naïve approach could look at all possible sets of k clusters. There are $S(n, k)$ possible combinations to examine.

$$S(n, k) = \sum_{m=1}^k (-1)^{k-1} \binom{k}{m} m^n$$

$$K!$$

3.4.7.8 K – Means Clustering

K – means is an iterative clustering algorithm in which items are moved among sets of clusters until the desired set is reached. As such, it may be viewed as a type of squared error algorithm, although the convergence criteria need not be defined based on the square error. A high degree of similarity among elements in clusters is obtained, while a high degree of dissimilarity among elements in

different clusters is achieved simultaneously. The *cluster mean* of $K_j = \{t_{j1}, t_{j2}, t_{j3}, \dots, t_{jm}\}$ is defined as

$$m_j = \frac{\sum_{i=1}^m t_{ji}}{m}$$

This definition assumes that each tuple has only one numeric value as opposed to a tuple with many attribute values. The K-means algorithm requires that some definition of cluster mean exists, but it does not have to be this particular one. Here the means is defined identically to our earlier definition of centroid. This algorithm assumes that the desired number of clusters, k , is an input parameter.

- (1) Initialize k cluster centers to be seed points (These centers can be randomly produced or use other ways to generate).
- (2) For each sample, find the nearest cluster center, put the sample in this cluster and re-compute centers of the altered cluster (Repeat n times).
- (3) Exam all samples again and put each one in the cluster identified with the nearest center (don't re-compute any cluster centers). If members of each cluster haven't been changed, stop. If changed, go to step 2.

Let's consider following example.

Samples in 2-D

We want to have 2 clusters of the data. Our steps are:

1. Set initial points. Because , we select 2 points, and , as center points.
2. is near , so put into cluster 1. Now 2 clusters are and . Others are as follows:

	and	N e a r which center	New 2 Cluster members

3. Now new 2 centers are and . For each sample, find its nearest center (don't re-compute the centers). Sample and are near . Sample , and are near . Members of each cluster don't change. So stop.

K-means algorithm has advantages such as easy to implement and converge in finite iterations. But when samples have outliers, which are sufficiently far removed from the rest of the data (see following figure), they will have influences on the results.

Fig 3.18: An outlier of samples

Another important problem of K-means algorithm is selecting initial seed points because clustering results always depend on initial seed points and partitions. To prevent this problem, one way we can run many conditions of different initial points to decide which condition is the best. For a fixed number of clusters, the goal of K-means algorithm is to minimize the square error E , so that stop condition can switch to finding minimum of square error instead of observing the change of members.

3.4.7.9 Nearest Neighbor Algorithm

An algorithm similar to the single link technique is called the *nearest neighbor algorithm*. With this serial algorithm, items are iteratively merged into the existing clusters that are closest. In this algorithm a threshold, t , is used to determine if items will be added to existing clusters or if a new cluster is created. The nearest neighbor prediction algorithm simply stated is: Objects that are “near” to each other will have similar prediction values as well. Thus if you know the prediction value of one of the objects you can predict it for its nearest neighbors.

3.4.7.10 Clustering of large database

The clustering algorithms presented in the preceding sections are some of the classic clustering techniques. When clustering is used with dynamic databases, these algorithms may not be appropriate. First, they all assume that sufficient main memory exists to hold the data to be clustered and the data structures needed to support them. With large databases containing thousands of items (or more), these assumptions are not realistic. In addition, performing I/Os continuously through the multiple iterations of an algorithm is too expensive. Because of these main memory restrictions, the algorithms do not scale up to large databases. Another issue is that some assume that the data are present all at once. These techniques are not appropriate for dynamic databases. Clustering techniques should be able to adapt as the database changes.

Recent research at Microsoft has examined how to efficiently perform the clustering algorithms with large databases. The basic idea of this scaling approach is as follows:

1. Read a subset of the database into main memory.
2. Apply clustering technique to data in memory.
3. Combine results with those from prior samples.
4. The in-memory data are then divided into three different types” those items that will always be needed even when the next sample is brought in, those that can be discarded with appropriate updates to data being kept in order to answer the problem, and those that will be saved in a compressed format. Based on the type, each data item is then kept, deleted, or compressed in memory
5. If termination criteria are not met, then repeat from step 1.

This approach has been applied to the K-means algorithm and has been shown to be effective.

3.4.7.11 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH is designed for clustering a large amount of metric data. It assumes that there may be a limited amount of main memory and achieves a linear I/O time requiring only one database scan. It is incremental and hierarchical, and it uses an outlier handling technique. Here points that are found in sparsely populated areas are removed. The basic idea of the algorithm is that a tree is built that captures needed information to perform clustering. The clustering is then performed in the tree itself, where labeling of nodes in the tree contain the needed information to calculate distance values. A major characteristics of the BIRCH algorithm is the use of the *clustering feature*, which is a triple that contains information about a cluster. The clustering feature provides a summary of the information about one cluster. By this definition it is clear that BIRCH applies only to numeric data.

3.4.7.12 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

The approach used by DBSCAN is to create clusters with a minimum size and density. Density is defined as a minimum number of points within a certain distance of each other. This handles the outlier problem by ensuring that an outlier will not create a cluster. One input parameter, *Min Pts*, indicates the minimum number of points in any cluster. In addition, for each point in a cluster there must be another point in the cluster whose distance from it is less than a threshold input value, *Eps*. The *Eps* – *neighborhood* or *neighborhood* of a point is the set of points within a distance of *Eps*. The desired number of clusters, *k*, is not input but rather is determined by the algorithm itself.

3.4.7.13 CURE (Clustering Using Representatives) Algorithm

One objective for the CURE clustering algorithm is to handle outlier well. It has both hierarchical component and a partitioning component. First, a constant number of points, *c*, are chosen from each cluster. These well-scattered points are the shrunk toward the cluster's centroid by applying a shrinkage factor, α . When α is 1, all points are shrunk to just one – the centroid. These points represent the cluster better than a single point could. With multiple representative points, clusters of unusual shapes (not just a sphere) can be better represented. CURE then uses a hierarchical clustering algorithm. At each step in the agglomerative algorithm, clusters with the closest pair of representative points are chosen to be merged. The distance between them is defined as the minimum distance between any pair of points in the representative sets from the two clusters.

CURE handles limited main memory by obtaining a random sample to find initial clusters. The random sample is partitioned, and each partition is then partially clustered 1. These resulting clusters are then completely clustered to second pass. The sampling and partitioning are done solely to ensure that the data (regardless of database size) can fit into available main memory. When clustering of the sample is complete, the labeling of data on disk is performed. A data item is assigned to the cluster with the closest representative points.

4. Implementation of Data Access User Interface (Web Based)

4.1 One Tiered Architecture

In one tier all the Processing is done in one node that means the User Interface, Server (business logic) and Database resides on one node.

4.2 Two Tiered Architecture

Only two tiers one is presentation tier and data tier. Here the presentation tier directly interacts with the data base server. This presentation tier takes the responsibility of the logical tier. It is typically used in small environments.

The most basic type of client-server architecture sometimes referred to as two-tier.

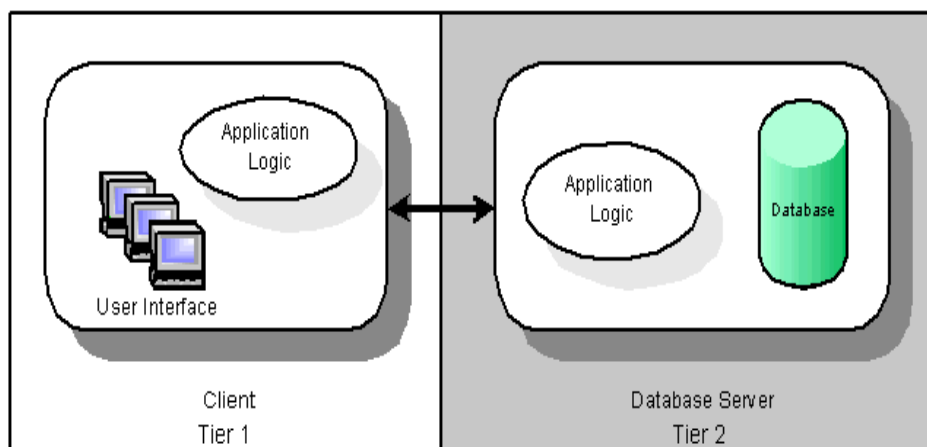


Fig 4.1 Two-Tiered Application

4.3 Three Tiered Architecture

The 3-Tier architecture has the following three tiers.

- Presentation Tier
- Logic Tier / Business Logic Tier / Data access Tier
- Data Tier

4.3.1. Presentation Tier

Presentation Tier is responsible for communication with the users and it will interact with the business layer.

4.3.2 Logic Tier (Business Logic Tier and Data access Tier)

The second tier is the Logical tier. This is again classified into two parts. One is business layer and second is data access layer.

Some architecture consider this logical tier as business tier and they are not yet divided into two parts.

This business layer contains code for some general calculations. And will interact with data access layer and it returns the data to presentation tier, which are received from data access layer.

Data access layer interacts with the data tier. This layer returns data to business layer. This layer contains the code to interact with the database.

4.3.3 Data Tier

The third tier is data tier. It is responsible for store, retrieve and update the data. Stored procedures come under this Tier.

4.4 WDAUI Architecture

A WDAUI tool uses the three-tier architecture. 3-tier architecture is used for larger, more interactive web tool. In web 3-tier architecture include web browser, web server and database.

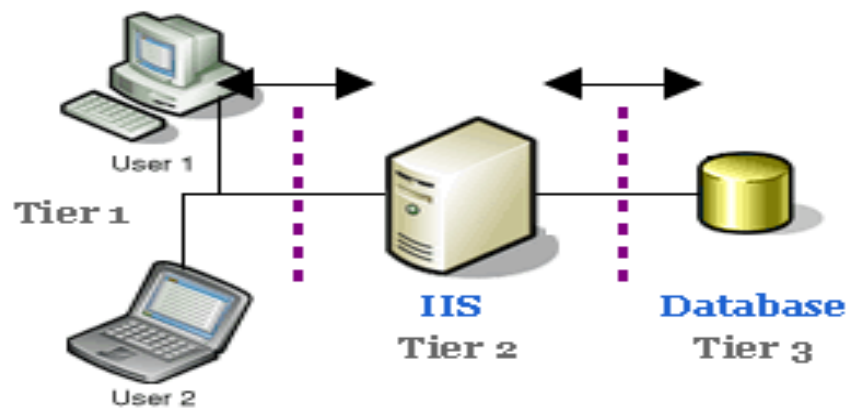


Fig 4.2: Three-Tiered Application

The WDAUI architecture consists of the following components:

- A web browser
- The IIS web server, powered by Microsoft
- HTML documents, CSS, JavaScript, AJAX, ASP
- Oracle Database

4.4.1 Web Browser

This is client user interface component. Any browser that must support tables frames and should be JavaScript enabled.

4.4.2 Internet Information Services

Internet Information Services (IIS) makes it easy to publish information on the Internet or on intranet. IIS includes a broad range of administrative features for managing Web sites and Web server. With programmatic features like Active

Server Pages (ASP), it can be created and deployed scalable and flexible Web applications.

4.4.2.1 IIS Installation

During installation, IIS installs optional components like Common Files, Documentation, and the Internet Information Services snap-in. It is not necessary to install all the optional components; however, deselecting specific components can decrease IIS functionality or disable IIS services. If we are unfamiliar with the optional components and how they affect IIS, install IIS with the default settings. After installation of IIS, we can view Installing IIS Optional Components in the IIS online documentation for more information.

1. Click Start, click Control Panel, and click Add or Remove Programs.
2. Click Add/Remove Windows Components. The Windows Components Wizard appears.
3. Follow the on-screen instructions to install, remove, or add components to IIS.

4.4.3 HTML (Hypertext Markup Language)

HTML is a Hyper Text Markup Language that is used to develop web pages. HTML developed a few years ago as a subset of SGML (Standard Generalized Mark-up Language). Any HTML document is also valid for SGML.

HTML is not a programming language like C, C++ and Java etc. It is a cross platform markup language that is design to be flexible enough to display text and other elements like graphical on a variety of views. Web browser interprets HTML file.

4.4.3.1 CSS (Cascading Style Sheet)

Cascading Style Sheet use for the design attractive web pages. That solve the limitation of the HTML tags and add more attributes.

Styles sheets define HOW HTML elements are to be displayed, just like the font tag and the color attribute in HTML 3.2. Styles are normally saved in external .css

files. External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document!

CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer you can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are updated automatically.

4.4.3.2 JavaScript

JavaScript is a client side scripting language that adds significant power to HTML files without the need for server-based CGI programs. All popular web clients interpret JavaScript code.

The embedded JavaScript coding provides a mechanism for client side caching of user-entered data during a transaction, and simple client side validation of user-entered data. Execution of simple JavaScript code logic at the client side results in reduced network traffic between the web browser client and the web server.

Using Java script following tasks can be performed.

- JavaScript gives HTML designers a programming tool - HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
- JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element

- JavaScript can be used to validate data - A JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing
- JavaScript can be used to detect the visitor's browser - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- JavaScript can be used to create cookies - A JavaScript can be used to store and retrieve information on the visitor's computer

3. AJAX

AJAX stands for Asynchronous Java Script and XML. AJAX is a type of programming made popular in 2005 by Google. AJAX is not a new programming language, but a new way to use existing standards.

AJAX is based on JavaScript and HTTP Request. JavaScript can communicate directly with the server, using the JavaScript XMLHttpRequest object. Using this object JavaScript can trade data with a web server without reloading the web page.

AJAX use asynchronous data transfer between the web browser and web server, allowing web pages to request small bits of information from the server instead of whole pages.

AJAX technique makes Internet applications smaller, faster and more users friendly.

4.4.3.4 ASP (Active Server Pages)

Active Server Pages (ASP) is a tool for creating dynamic and interactive web pages.

ASP is a Microsoft technology, which works by allowing us to use the functionality of a programming language that will generate the HTML for the web page dynamically.

Advantages of ASP:

- Dynamically edit, change or add any content of a Web page.
- Response to user queries or data submitted from HTML forms.
- Access any data or databases and return the results to a browser.
- The advantages of using ASP instead of CGI and Perl are those of simplicity and speed.
- Provides security since your ASP code can not be viewed from the browser.
- Since ASP files are returned as plain HTML, they can be viewed in any browser.
- ASP programming can minimize the network traffic.

4.4.4 Oracle Database

The following example illustrates an Oracle configuration where the user and associated server process are on separate machines (connected via a network).

1. An instance is currently running on the computer that is executing Oracle (often called the *host* or *database server*).
2. A computer running an application (a *local machine* or *client workstation*) runs the application in a user process. The client application attempts to establish a connection to the server using the proper Net8 driver.
3. The server is running the proper Net8 driver. The server detects the connection request from the application and creates a (dedicated) server process on behalf of the user process.
4. The user executes a SQL statement and commits the transaction. For example, the user changes a name in a row of a table.

5. The server process receives the statement and checks the shared pool for any shared SQL area that contains an identical SQL statement. If a shared SQL area is found, the server process checks the user's access privileges to the requested data and the previously existing shared SQL area is used to process the statement; if not, a new shared SQL area is allocated for the statement so that it can be parsed and processed.
6. The server process retrieves any necessary data values from the actual datafile (table) or those stored in the system global area.
7. The server process modifies data in the system global area. The DBWn process writes modified blocks permanently to disk when doing so is efficient. Because the transaction committed, the LGWR process immediately records the transaction in the online redo log file.
8. If the transaction is successful, the server process sends a message across the network to the application. If it is not successful, an appropriate error message is transmitted.
9. Throughout this entire procedure, the other background processes run, watching for conditions that require intervention. In addition, the database server manages other users' transactions and prevents contention between transactions that request the same data.

4.5 RDBMS access using Web Interface

4.5.1 Introduction of Web Interface

Any client machine want to access oracle database, then it is necessary to install Oracle client at workstation. Further more Oracle client software is not user friendly, users have to remember and follow the syntax of statements, commands and functions to create tables, users, profiles, triggers, etc. So person at the client machine must be aware of Oracle.

This web interface constructs the bridges between Oracle server and workstation on which one wants to use oracle database. It makes easy to access Oracle database without much prior knowledge of Oracle. This web interface is user friendly so user does not have to remember syntax of statements, commands and functions to create tables, users, profiles, triggers, etc. On any client machine, if a web browser is installed, using this web interface oracle database can be accessed from server. To access Oracle database, there is requirement of higher hardware configuration at client machine. But advantage of this tool is that client machine does not require higher hardware configuration to access Oracle database. This way cost of client machine can be decreased. So it is very much beneficial for medium-scale organizations.

4.5.2 Security in Web Interface

In an organization, database security is a big and crucial issue. All the RDBMS software provides its own security but some time there is requirement of more security. Oracle has its own securities, like user authentication and etc. This software also provides web session security. In which if any user is idle for 30 seconds, then automatically session is destroyed, and then user have to log in again. This way this tool provides database security and web security.

This tool provides also DBA level functionality like create profile, role, new users, etc at any client machine. For this it is not compulsory for DBA to log in at server machine.

4.5.3 How Web Interface Works

This is log in screen, here user enters user name, password and HOST or Server name of Oracle database. This is same as log in screen of oracle database.

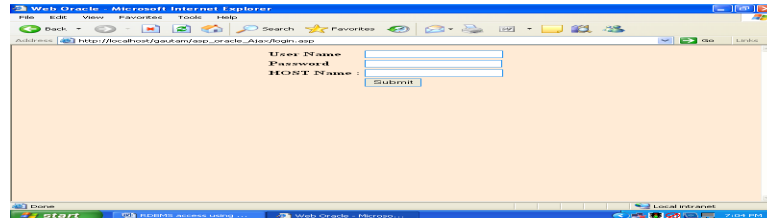


Fig 4.3: Log in Screen

After log in into Oracle database, following screen is displayed. In this screen user can create Table, Procedure, Trigger. Password can also be changed from this screen.

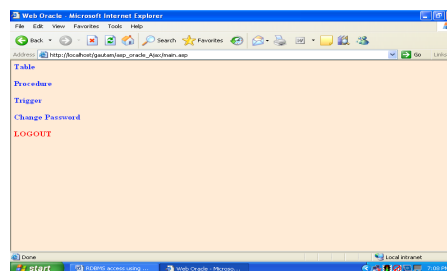


Fig 4.4: User Screen

If user log in as DBA, following screen will be displayed. From here DBA can create new user, edit user, create role, edit role, create profile, edit profile etc.

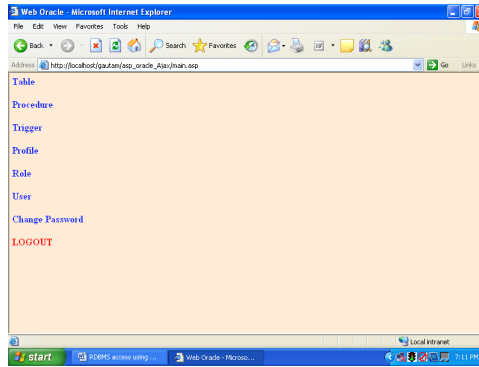


Fig 4.5 DBA Screen

When user clicks on Table, following screen is displayed, in which all the existing tables are displayed. User can create a new table by clicking on Create New Table.

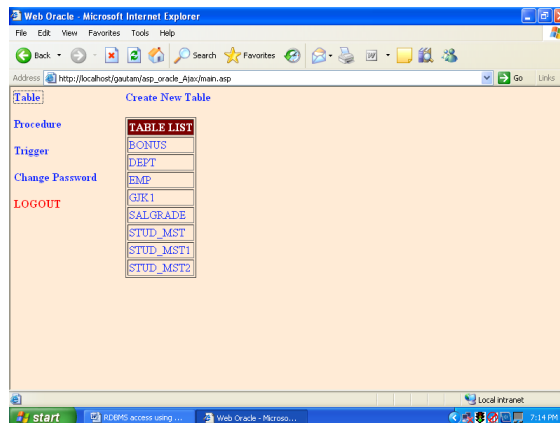


Fig 4.6: Table List Screen

When a new table is created, user has to provide table name and total number of fields. It is displayed in following screen.

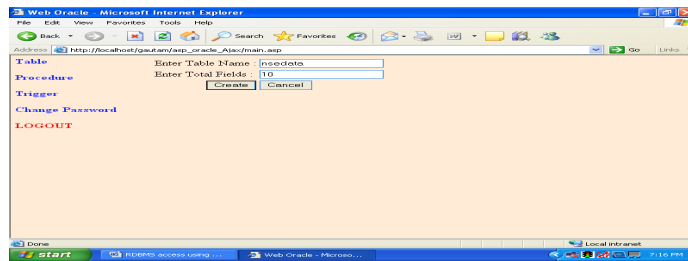


Fig 4.7 New Table Screen

After specifying table name and number of fields, when Create button is clicked, it will display following screen in which field name, data type, constraints like Primary Key, Unique Key, Check can be specified.

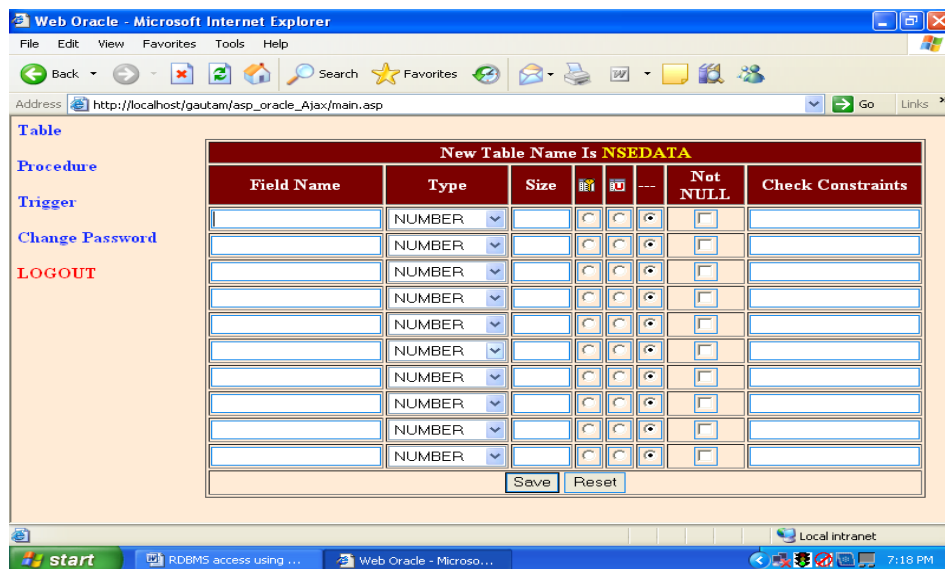


Fig 4.8 Fields in New Table

When Save button is clicked, then following screen is displayed. From here, use can display table structure, Browse stored data, and Insert new rows and table can be dropped. From here table structure can be modified also.

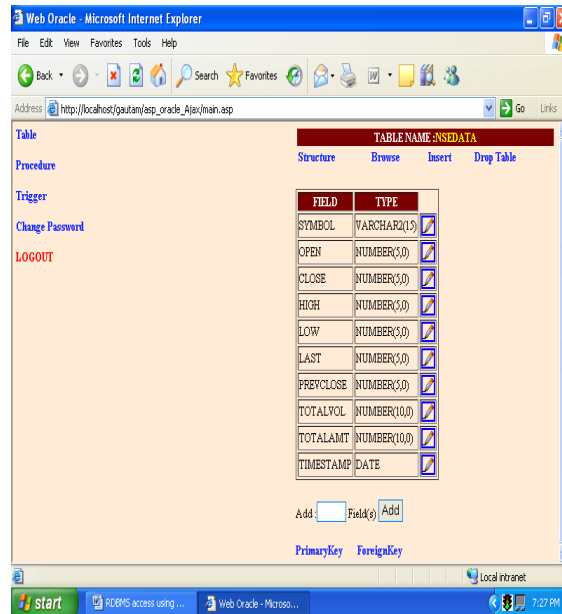


Fig 4.9: Structure of Created Table

When user click on Procedure, existing procedure is display, it can be modified or deleted also. A new procedure can be created.

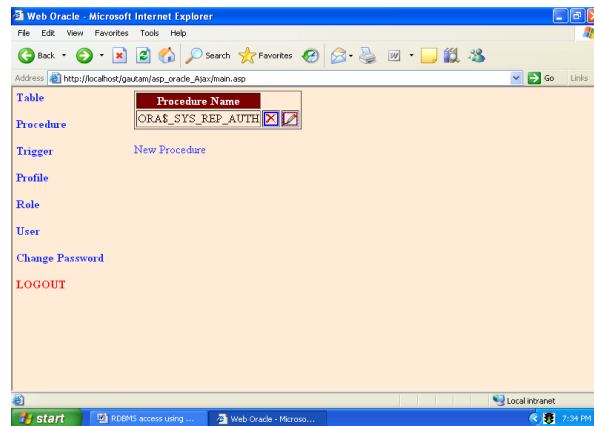


Fig 4.10: New Procedure screen

When user clicks on Trigger, existing triggers are displayed, it can be modified or deleted also. A new trigger can be created from here.

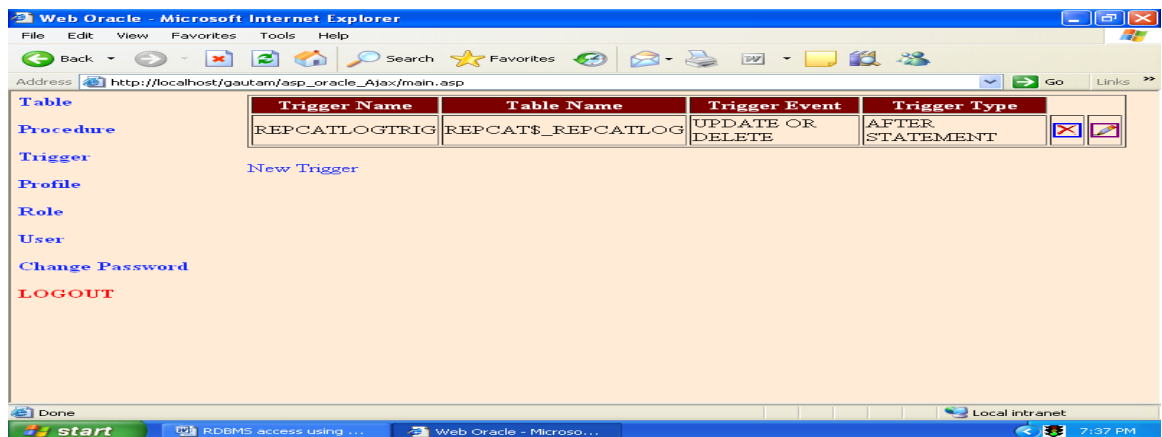


Fig 4.11: Creation of Trigger

DBA wants to create, edit existing profile and delete profile by clicking on Profile.

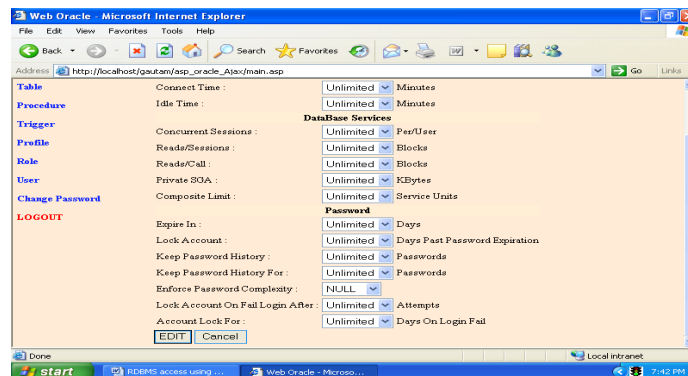
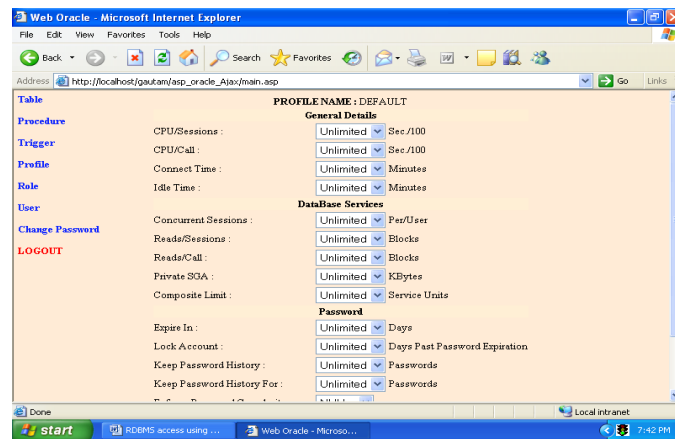


Fig 4.12: User Profile Creation

DBA can create new roles, edit roles and delete roles by clicking on Role. Following screen displays various roles.

	ROLE	Other Grant Role	System Privileges	
Table	AQ_ADMINISTRATOR_ROLE	Roles	System Privileges	
Procedure	AQ_USER_ROLE	Roles	System Privileges	
Trigger	CONNECT	Roles	System Privileges	
Profile	DBA	Roles	System Privileges	
Role	DELETE_CATALOG_ROLE	Roles	System Privileges	
User	EXECUTE_CATALOG_ROLE	Roles	System Privileges	
Change Password	EXP_FULL_DATABASE	Roles	System Privileges	
LOGOUT	HS_ADMIN_ROLE	Roles	System Privileges	
	IMP_FULL_DATABASE	Roles	System Privileges	
	RECOVERY_CATALOG_OWNER	Roles	System Privileges	
	RESOURCE	Roles	System Privileges	
	SELECT_CATALOG_ROLE	Roles	System Privileges	
	SNMPAGENT	Roles	System Privileges	

[New Role](#)

Fig 4.13: Maintaining of Roles

DBA can create new user, modify and delete existing user by clicking on User. Following screen displays the list of existing users with roles and system privileges. From here by clicking on Role or System Privileges of any user, the role and privileges assigned to that user is displayed. It can also be modified.

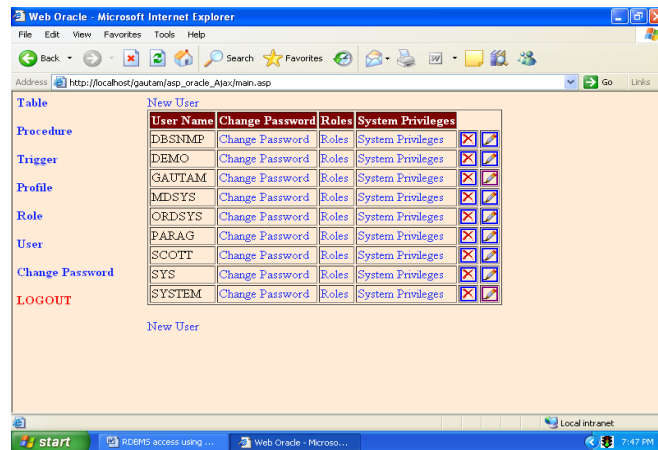


Fig 4.14: Users with Roles and System Privileges

By clicking on Change Password, user or DBA can change its own password.

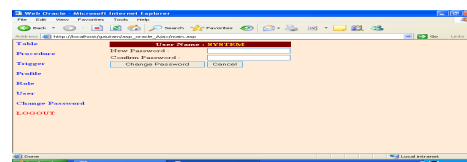


Fig 4.15: Change Password

4.5.4 Coding

4.5.4.1 CSS Code

Following CSS code is used to make web page attractive and easy to modify.

```
.txtborder
{
    BORDER-BOTTOM: #1188ee 1px solid;
    BORDER-LEFT: #1188ee 1px solid;
```

```

    BORDER-RIGHT: #1188ee 1px solid;
    BORDER-TOP: #1188ee 1px solid
}
INPUT
{
    BORDER-BOTTOM: #1188ee 1px solid;
    BORDER-LEFT: #1188ee 1px solid;
    BORDER-RIGHT: #1188ee 1px solid;
    BORDER-TOP: #1188ee 1px solid
}

BODY
{
    BACKGROUND-COLOR: #ffe6d6;
    MARGIN: 5px
}
A:hover
{
    COLOR: #004522;
    TEXT-DECORATION: underline
}
A
{
    COLOR: #1127ff;
    TEXT-DECORATION: none
}
table
{
    border-color: gray
}
td
{
    border-color: gray;

```

```
}  
th  
{  
  border-color: gray;  
  background-color: #7C0000;  
  color: white;  
}
```

4.5.4.2 Log In Check Code

Following code is written for log in check. This code checks that user is authenticated or not. If it is authenticated, then session variables are created to store user name, password and connection object, that is useful for security purpose. If user is not authenticated, then again log in page appears with message "Invalid Username or Password".

```
<%
    On Error Resume Next

    set session("con")= server.CreateObject("ADODB.Connection")
    'Response.Write      "Provider=MSDAORA.1;User      ID="      &
Request.Form("uname")&" ;password="      &      Request.Form("upass")&" ;Data
Source="& Request.Form("sename")
    session("con").Open      "Provider=MSDAORA.1;User      ID="      &
Request.Form("uname")&" ;password="      &      Request.Form("upass")&" ;Data
Source="& Request.Form("sename")
    if Err.number<>0 then
        Response.Redirect ("login.asp?msg=1")
    end if
    SESSION("uname")=Request.Form("uname")
    SESSION("upass")=Request.Form("upass")
    session("tablename")=""
    Response.Redirect ("main.asp")
%>
```

4.5.4.3 User Facility Code

This code displays the links for user facilities depending on user type. If user is DBA, then user profile, user, role, etc. is displayed.

```
<%
    Response.Write "<b><a href=gettable.asp target=f2>Table </a></b>"
        Response.Write          "<br><br><b><a          href=procedure.asp
target=f2>Procedure </a></b>"
        Response.Write  "<br><br><b><a href=trigger.asp target=f2>Trigger
</a></b>"

        set rsrole= session("con").execute("SELECT GRANTED_ROLE FROM
USER_ROLE_PRIVS WHERE GRANTED_ROLE='DBA' ")

        if not rsrole.eof then
            session("auth")="DBA"
            Response.Write          "<br><br><b><a          href=profile.asp
target=f2>Profile </a></b>"
            Response.Write  "<br><br><b><a href=role.asp target=f2>Role
</a></b>"
            Response.Write "<br><br><b><a href=userlist.asp target=f2>User
</a></b>"
        end if
        Response.Write          "<br><br><b><a
href=user_password.asp?auth=local&usr=" & Session("uname") & "
target=f2>Change Password </a></b>"

        Response.Write          "<br><br><b><a          href=logout.asp?id=1><font
color=RED>LOGOUT</font> </a></b>"

%>
```

4.5.4.4. Creating New Table Code

Following is code for creating new table. First we tried to store a new created table structure in XML file.

This XML file occupies the web space. Similarly it occupies space for all the created and existing tables. This way very much web space is required for all the tables. To over take this problem, idea to store table structure as XML file is dropped. It is indicated in comment lines in coding.

```
<%
    on error resume next

    if Request.Form("s1")<>"Save" then
'   Response.Write "Table Name:=" & Request.Form("tabname")
'   Response.Write "<br>Table Field:=" & Request.Form("tabfld")

%>
<form name=ftabdesign method=post>
<br>
<table border=1>
    <tr>
        <th colspan=8> New Table Name Is <font
color=yellow><%=ucase(Request.Form("tabname")) %></font>
    </tr>
    <tr>
        <th>Field Name</th>
        <th>Type</th>
        <th>Size</th>
        <th WIDTH=10px></th>
        <th WIDTH=10px></th>
        <th WIDTH=15px>---</th>
        <th>Not NULL</th>
```



```

        <th>Check Constraints</th>
    </tr>
<%
    for i=1 to Request.Form("tabfld")

        Response.Write "<tr>"
        Response.Write "<td><input type=text name=fname"&i &"> </td>"
        Response.Write "        <td><select            name=ftype"&i            &">
<option>NUMBER</option>                <option>VARCHAR2</option>
<option>DATE</option> </TD>"
        Response.Write "<td><input type=text SIZE=4 name=fsize"&i &"></td>"
        Response.Write "        <td><input    type=radio    value='Primary    Key'
name=fprimary" &i &"></td>"
        Response.Write "<td><input type=radio value='Unique' name=fprimary" &i
&"></td>"
        Response.Write "<td><input type=radio value='N' checked name=fprimary"
&i &"></td>"
        Response.Write "<td align=center><input type=checkbox name=fnotnull" &i
&"></td>"
        Response.Write "<td><input type=text name=fconstrain" &i &"></td>"
        Response.Write "</tr>"

    next

    Response.Write "        <input    type=hidden    name=tabname    value=" &
Request.Form("tabname")&">"
    Response.Write "        <input    type=hidden    name=tabfld    value=" &
Request.Form("tabfld")&">"
%>

    <tr>

        <td colspan=8 align=center>

<input type=submit value="Save" name=s1>

```

```

        <input type=Reset value="Reset" name=r1>
    </td>
</tr>
</table>

</form>
<%

```

else 'create table query in same form when submit the strucutre of table

```

q="create table "& Request.Form("tabname") & " ("
for i=1 to Request.Form("tabfld")

        fnm="fname"& i
        ft="ftype" & i
        fs="fsize" & i
        fp="fprimary" & i
        fn="fnotnull" & i
        fc="fconstrain" & i

        q=q & Request.Form(fnm) & " " & Request.Form(ft)
'        xmlarr(0)=0 'array for xmlfile datatype only

        if Request.Form(ft)="VARCHAR2" or Request.Form(ft)="NUMBER" and
Trim(Request.Form(fs))<>"" then

                q=q & " (" & Request.Form(fs)& ")"
'                $xmlarr[$i]="(" & Request.Form("fsize").$i & ")"
                else
'                $xmlarr[$i]="("

        end if

" FOR PRIMARY KEY

```

```

if Request.Form(fp)<>"N" then

    if Request.Form(fp)="Unique" then

        '                                     q=q."  CONSTRAINT
UK_".Request.Form['tablename']. "_".Request.Form["fname".$i]. "
".Request.Form["fprimary".$i];
        q=q & " " & Request.Form(fp)

    else

        '                                     $q=$q."  CONSTRAINT PK_".Request.Form['tablename']. "
".Request.Form["fprimary".$i];
        q=q & " " & Request.Form(fp)
    end if
end if

" FOR NOT NULL
    if Request.Form(fn)<>"" and UCase(Request.Form(fn))="ON" then

        '//                                     $q=$q."  CONSTRAINT
NN_".Request.Form['tablename']. "_".Request.Form["fname".$i]. " NOT NULL";
        q=q & " NOT NULL"
    end if

" For CHECK Constraint

    if Trim(Request.Form(fc))<>"" then

```

```

//                                $q=$q." CONSTRAINT
CH_".Request.Form["tablename"]."_".Request.Form["fname"].$i." CHECK
(".Request.Form["fconstrain"].$i.");
    q=q & " CHECK (" & Request.Form(fc) & ")"
end if

q=q & ", "

next

q = Mid (q,1,len(q)-2)& ")"
'Response.Write q

session("con").Execute q

if Err.number<>0 then

%>
<script language="javascript">

alert("<%=mid(Err.description,1,Len(Err.description)-1) %>");
    history.back();

</script>
<%
else
    Response.Redirect "tablestructure.asp?table=" &
UCASE(Request.Form("tablename"))
end if
'& "="& Err.description & "="& Err.source & "="& Err.helpcontext
& "="& Err.helpfile

```

```

' // ADD TABLE IN XML FILE
'
'      set xml = server.CreateObject("Microsoft.XMLDOM")
'      xml.Load server.MapPath(session("uname")&".xml")
'
'      set n= xml.getElementsByTagName("TABLES").item(0)
'      nodenm="T1Y2G3P_"& (n.childNodes.length + 2)
'
'      set newnode=xml.CreateElement(nodenm)
'      set atr=xml.CreateAttribute("id")
'      atr.Text=Request.Form("tabname")
'
'      newnode.setAttributeNode atr
'
'      set tabnode= n.AppendChild (newnode)
'
'      cnt=1
'      for i=1 to Request.Form("tabfld")
'          set fldnode=xml.CreateElement("field"&cnt)
'          fldnode.Text=Request.Form("ftype"&i)
'
'          set fldatr=xml.CreateAttribute("fldid")
'          fldatr.Text=Request.Form("fname"&i)
'
'          fldnode.setAttributeNode fldatr
'          tabnode.AppendChild fldnode
'
'          cnt=cnt+1
'      next
'      xml.Save server.MapPath(session("uname")&".xml")
'
' end if
'
' %>

```


4.5.4.5. Addition Foreign Key Code

This code adds foreign key into the table. AJAX technology is used to create this facility. Main advantage of AJAX is that, entire page is not refreshed, but only selected portion of field can be refreshed. Without AJAX, to create this facility is not possible. It saves time to refresh the data on web page.

```
<link rel="stylesheet" type="text/css" href="style.css">
<script type="text/javascript">
    var objxml
        function getXmlObject()
        {
            var xmlHttp=null
            try
            {
                xmlHttp=new XMLHttpRequest()
            }
            catch(e)
            {
                try
                {
                    xmlHttp=new
ActiveXObject("Msxml2.XMLHTTP")
                }
                catch(e)
                {
                    try
                    {
                        xmlHttp=new
ActiveXObject("Microsoft.XMLHTTP")
                    }
                    catch(e)
                    {

```

```

                                alert("Your Web Browser Does Not
Support AJAX")
                                }
                                }
                                }

                                return xmlhttp
                                }

function fillcmb()
{
    var cmb= document.getElementById("fld")

    for(i=cmb.length-1;i>=0;i--)
    {
        cmb.remove(i)
    }

    objxml=getXmlObject();
    if (objxml==null)
    {
        alert("Object Does Not Created")
        return
    }
    else
    {
        url="getcolumn.asp?table="+
document.getElementById("tablename").value
        objxml.onreadystatechange=dispinfo
        objxml.open ("GET",url ,true)
        objxml.send(null)
    }
}

```



```

    }

    function dispinfo()
    {
        var obj= document.getElementById("fld")
        if (objxml.readyState==4)
        {
            eval(objxml.responseText)
            obj.value=objxml.responseText
        }
    }
</script>

</HEAD>
<BODY bgcolor="#f0f8ff">
<%
        server.Execute "commanlink.asp"
    if Request.Form("s1")<>"" then
        qa="ALTER TABLE "& SESSION("tablename") &" ADD
FOREIGN KEY ("& Request.Form("fldname") &") REFERENCES "&
Request.Form("tabname") &" ("& Request.Form("fld") &")"
        session("con").execute qa
        Response.Redirect "ADDFOREIGNKEY.asp"
    else
        Response.Write "<table border=1>"
        Response.Write "<tr><th>Foreign Key Field</th><th>Reference
Table</th><th>Reference Field</th></tr>"

        qf="select column_name,a.constraint_name from
user_cons_columns a,user_constraints b where a.constraint_name =
b.constraint_name and constraint_type = 'R' AND A.TABLE_NAME = "&
SESSION("tablename")&"

```

```

        set rsqf= session("con").execute (qf)
    while not rsqf.EOF

        qm="SELECT      TABLE_NAME,      COLUMN_NAME      FROM
USER_CONS_COLUMNS  WHERE  CONSTRAINT_NAME  IN  (SELECT
R_CONSTRAINT_NAME   FROM      USER_CONSTRAINTS   WHERE
CONSTRAINT_TYPE='R' AND TABLE_NAME='"& SESSION("tablename") &"
AND COLUMN_NAME='"& rsqf(0)&"')"
```

set rsqm=session("con").execute (qm)

```

    Response.Write "<tr>"
    Response.Write "<td>"& rsqf(0)&"</td>"
    Response.Write "<td>"& rsqm(0)&"</td>"
    Response.Write "<td>"& rsqm(1)&"</td>"
    Response.Write      "<td><a      href=deleteforeignkey.asp?keyid="&
rsqf(1)&"><img src='./icon/b_drop.png' alt='Delete Foreign Key'></a></td>"

    Response.Write "</tr>"
    rsqf.MoveNext
wend
Response.Write "</table>"

%>
<form method=post>
<table border=1>
<tr>
    <th>Field Name</th>
    <th>Master Table Name</th>
    <th>Primary Key</th>

```

```

</tr>
<tr>
<%

```

```

q="select column_name from user_tab_columns where table_name='"&
SESSION("tablename") &"'"

```

```

set rsfld= session("con").execute (q)

```

```

Response.Write "<td><select name=fldname>"
while not rsfld.eof

```

```

Response.Write "<option>"& rsfld(0)&"</option>"
rsfld.MoveNext
wend
Response.Write "</select></td>"

```

```

q="select tname from tab where TABTYPE='TABLE' and TNAME!='" &
SESSION("tablename")&"'"

```

```

set rstab= session("con").execute (q)

```

```

Response.Write "<td><select name=tabname id=tablename
onchange='fillcmb()'">"

```

```

' set xml = server.CreateObject("Microsoft.XMLDOM")
' xml.Load server.MapPath(session("uname")&".xml")
' set n= xml.getElementsByTagName("TABLES").item(0)
' for i=0 to n.childNodes.length - 1
' tabnm=n.childNodes(i).getAttribute("id")
' if ucase(tabnm)<>ucase(session("tablename")) then
' Response.Write "<option value='"&
n.childNodes(i).NodeName &"'">" & tabnm & "</option>"

```

```

'          end if
'      next
while not rstab.eof

    Response.Write "<option value="& rstab(0) &">"& rstab(0)&"</option>"
        rstab.MoveNext
wend

Response.Write "</select></td>"
Response.Write "<input type=hidden name=ftabname id=ftabname1>"
'For Retrive Correct Table Name set By JavaScript
Response.Write "<td><select name=fld></select>"

%>
<!-- <td><input type=text name=mfldname></td> -->

</table>
<br>
<td><input    type=hidden    name=xmlname    id=xmlfile    value=<%=
SESSION("uname")& ".xml" %>></td>
    <input type=submit value="Submit" name=s1>
</form>
<%
    end if
%>

```

4.6 Limitations of WDAUI

Right now WDAUI is working with Oracle database only. This is only the limitation of this tool. But this is not permanent limitation. Now we are in position to access Oracle database from any client machine (without installing client software) using this web data access tool (WDAUI). Our future work is to extend features of this tool to access data from more than one database. In which user can select any one database software by clicking on option button. Our data access tool will work with the selected database. When this feature will be

established, it will very much useful for users of any database software using this three-tiered application.

5. Computational Modeling (ETL) and Analytical Study

5.1 Overview

In this chapter computational modeling includes the ETL (Extraction, Transformation and Loading). First part in this chapter is data extraction. Operational data is often scattered in different environment, different forms and stored at different location. So it is necessary to collect and pull the data at our location that is useful to us. We have been developed a web-based tool to collect and pull the operational data from various environments.

After collection of data it is very much necessary to transform the data in our database that we are using. The collected and pulled data from different environment may be stored in different type of databases and different formats. It is very much difficult and sometime not possible to handle various types of databases and different format of storage in single system. So here data is transformed from different databases into our database. We have developed a tool to transform and gather these operational data into our database. This way the data warehouse is created and data is stored.

Next phase is the charting of data. We have used web technology software to plot our data in chart that is stored in our database. From chart some pattern extraction is possible. Charts can be compared with various statistical results that is also derived from our data. Then after also another patterns can be generated and some conclusion can be extracted.

We have used two softwares to analyze various data mining algorithms. One is SQL Server analysis service that provides very much powerful and fast service to generate the result. We also have used Weka software that is Java based software used for data mining. In both SQL Server analysis service and Weka there are many algorithms are ready to use for different techniques of data mining. We have used these algorithms in which our database is source and generated result is analyzed.

5.2 Extraction of Data (Data Extraction)

Systems in the data warehouse environment are not built under the system development life cycle (SDLC). There are two major components to building a data warehouse.

1. The design of the interface from operational systems.
2. Design of the data warehouse.

The requirements for the data warehouse cannot be known until it is partially populated and in use and design approaches that have worked in the past will not necessarily suffice in subsequent data warehouse. Data warehouse are constructed in a heuristics manner, where one phase of development depends entirely on the results attained in the previous phase.

We have decided to use stock exchange data of equities for our research practical work. Because the data NSE (nseindia.com) provides historical data.

5.2.1 National Stock Exchange (NSE) Organization

The National Stock Exchange of India Limited has genesis in the report of the High Powered Study Group on Establishment of New Stock Exchanges, which recommended promotion of a National Stock Exchange by financial institutions (FIs) to provide access to investors from all across the country on an equal footing. Based on the recommendations, NSE was promoted by leading Financial Institutions at the behest of the Government of India and was incorporated in November 1992 as a tax-paying company unlike other stock exchanges in the country.

On its recognition as a stock exchange under the Securities Contracts (Regulation) Act, 1956 in April 1993, NSE commenced operations in the Wholesale Debt Market (WDM) segment in June 1994. The Capital Market (Equities) segment commenced operations in November 1994 and operations in Derivatives segment commenced in June 2000.

5.2.2 Technology using in NSE

Across the globe, developments in information, communication and network technologies have created paradigm shifts in the securities market operations. Technology has enabled organizations to build new sources of competitive advantage, bring about innovations in products and services, and to provide for new business opportunities. Stock exchanges all over the world have realized the potential of IT and have moved over to electronic trading systems, which are cheaper, have wider reach and provide a better mechanism for trade and post trade execution.

NSE believes that technology will continue to provide the necessary impetus for the organization to retain its competitive edge and ensure timeliness and satisfaction in customer service. In recognition of the fact that technology will continue to redefine the shape of the securities industry, NSE stresses on innovation and sustained investment in technology to remain ahead of competition. NSE's IT set-up is the largest by any company in India. It uses satellite communication technology to energize participation from around 320 cities spread all over the country. In the recent past, capacity enhancement measures were taken up in regard to the trading systems so as to effectively meet the requirements of increased users and associated trading loads. With upgradation of trading hardware, NSE can handle up to 6 million trades per day in Capital Market segment. In order to capitalize on in-house expertise in technology, NSE set up a separate company, NSE.IT, in October 1999. This is expected to provide a platform for taking up new IT assignments both within and outside India and attaining global exposure.

NEAT is a state-of-the-art client server based application. At the server end, all trading information is stored in an in-memory database to achieve minimum response time and maximum system availability for users. The trading server software runs on a fault tolerant STRATUS mainframe computer while the client software runs under Windows on PCs.

The telecommunications network uses X.25 protocol and is the backbone of the automated trading system. Each trading member trades on the NSE with other

members through a PC located in the trading member's office, anywhere in India. The trading members on the various market segments such as CM / F&O, WDM are linked to the central computer at the NSE through dedicated 64Kbps leased lines and VSAT terminals. The Exchange uses powerful RISC -based UNIX servers, procured from Digital and HP for the back office processing. The latest software platforms like ORACLE 7 RDBMS, GUPTA - SQL/ORACLE FORMS 4.5 Front - Ends, etc. have been used for the Exchange applications. The Exchange currently manages its data center operations, system and database administration, design and development of in-house systems and design and implementation of telecommunication solutions.

NSE is one of the largest interactive VSAT based stock exchanges in the world. Today it supports more than 3000 VSATs. The NSE- network is the largest private wide area network in the country and the first extended C- Band VSAT network in the world. Currently more than 9000 users are trading on the real time-online NSE application. There are over 15 large computer systems which include non-stop fault-tolerant computers and high end UNIX servers, operational under one roof to support the NSE applications. This coupled with the nation wide VSAT network makes NSE the country's largest Information Technology user.

In an ongoing effort to improve NSE's infrastructure, a corporate network has been implemented, connecting all the offices at Mumbai, Delhi, Calcutta and Chennai. This corporate network enables speedy inter-office communications and data and voice connectivity between offices.

In keeping with the current trend, NSE has gone online on the Internet. Apart from having a 2mbps link to VSNL and our own domain for internal browsing and e-mail purposes, we have also set up our own Web site. Currently, NSE is displaying its live stock quotes on the web site (www.nseindia.com), which are updated online.

5.2.3 NSE on Web

National Stock Exchange provides historical and live data on web in figures as well as chart. Following is the home page of nseindia.org.



Fig 5.1: NSE Home Page

We have used National Stock Exchange of India (NSE) data for our practical work implementation. NSE provides historical data for Equities, Derivatives, Members, WDM, Corporate Bonds, Press, Publications, RDM, Indices, NSCCL and SLBS. The following web page displays the various categories of historical data provided by NSE.

The official website of NSE provides various historical data. To pull these data we have to select the information for which we require historical data and enter the date in dd-mm-yyyy format in the input box given below in following web page.

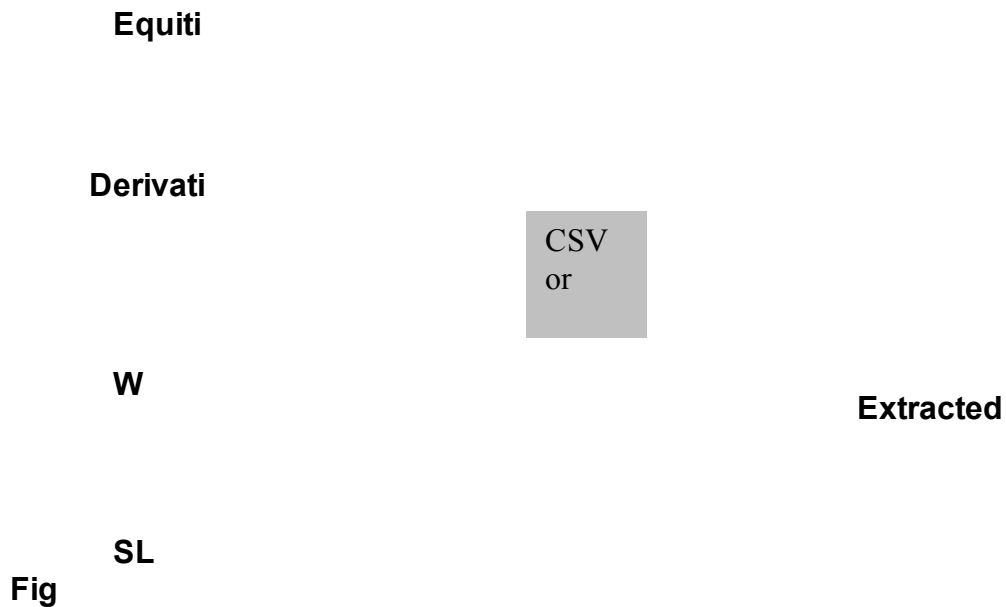
EQUITIES	WDM	NSCCL
<input checked="" type="radio"/> Bhavcopy	<input type="radio"/> Bhavcopy	<input type="radio"/> SPAN Risk Parameter files
<input type="radio"/> Market Activity Report	<input type="radio"/> Daily Reports	<input type="radio"/> VaR Margin Rate files
<input type="radio"/> Security-wise Delivery Position	<input type="radio"/> Weekly Reports	<input type="radio"/> Security Category files
<input type="radio"/> Client Funding	<input type="radio"/> Monthly Reports	<input type="radio"/> VaR Multiplier files
<input type="radio"/> Margin Trading	<input type="radio"/> Zero Coupon Yield Curve	<input type="radio"/> Daily Volatility files
<input type="radio"/> Category-wise Turnover	<input type="radio"/> NSE VaR for Govt. sec.	<input type="radio"/> Daily Settlement Price files
<input type="radio"/> ALBM Yield Statistics	<input type="radio"/> Monthly Debt Update	<input type="radio"/> Daily Client-wise Position Limit files
DERIVATIVES	CORPORATE BONDS	SLBS
<input type="radio"/> Bhavcopy	<input type="radio"/> Bhavcopy	<input type="radio"/> Var Margin Rate File
<input type="radio"/> Market Activity Report	PRESS	<input type="radio"/> List of Eligible securities
<input type="radio"/> Monthly Derivative Update	<input type="radio"/> NSE Newsletters	<input type="radio"/> Bhavcopy
<input type="radio"/> Exercise	PUBLICATIONS	<input type="radio"/> Positions data
<input type="radio"/> Interest Rate Derivatives	<input type="radio"/> NSE Factbook	<input type="radio"/> Transaction details
<input type="radio"/> Security in ban period	<input type="radio"/> ISMR	<input type="radio"/> Client level Position Limits
MEMBERS	RDM	<input type="radio"/> Market wide Position Limits
<input type="radio"/> Member Disablement Database - CM	<input type="radio"/> Bhavcopy	<input type="radio"/> FII / MF position limits
<input type="radio"/> Member Disablement Database - F&O	INDICES	<input type="radio"/> Participantwise position limits
<input type="radio"/> Member proprietary positions & margins - CM	<input type="radio"/> IISL Monthly Update	
<input type="radio"/> Member proprietary positions & margins - F&O	<input type="radio"/> Impact Cost	
Date (dd-mm-yyyy) : <input type="text"/> <input type="button" value="Get"/>		

Fig 5.2: List of Historical Data

NSE provides the data in two formats that is CSV and DBF. The data in CSV format can be viewed in Excel and the data in DBF format can be opened in FoxPro. The main draw back of this system is that at a time we can pull the data of only one day. For the data of another day we have to input another date and we can retrieve the data of that particular date. In our application we have get the rid on this draw back.

5.2.4 Data Extraction model

NSE provides various historical data as shown in Fig 5.2. From there we can collect the data from nseindia.org into our computer system.



Fig

5.3: Data Extraction Model

As we discussed earlier, the main drawback of nseindia.org is that, it provides only single day historical data. If we want to extract the historical data for more than one day, then it is not possible from nseindia.org, there is no facility to extract the historical data for more than one day.

In our Data Extraction model we get rid from this main drawback of nseindia.org.

5.2.5 Process of Data Extraction

This Data Extraction Model is developed in PHP. To execute this Data Extraction Model, first we have to start Internet Information Services (IIS). When IIS is started following screen will be displayed.

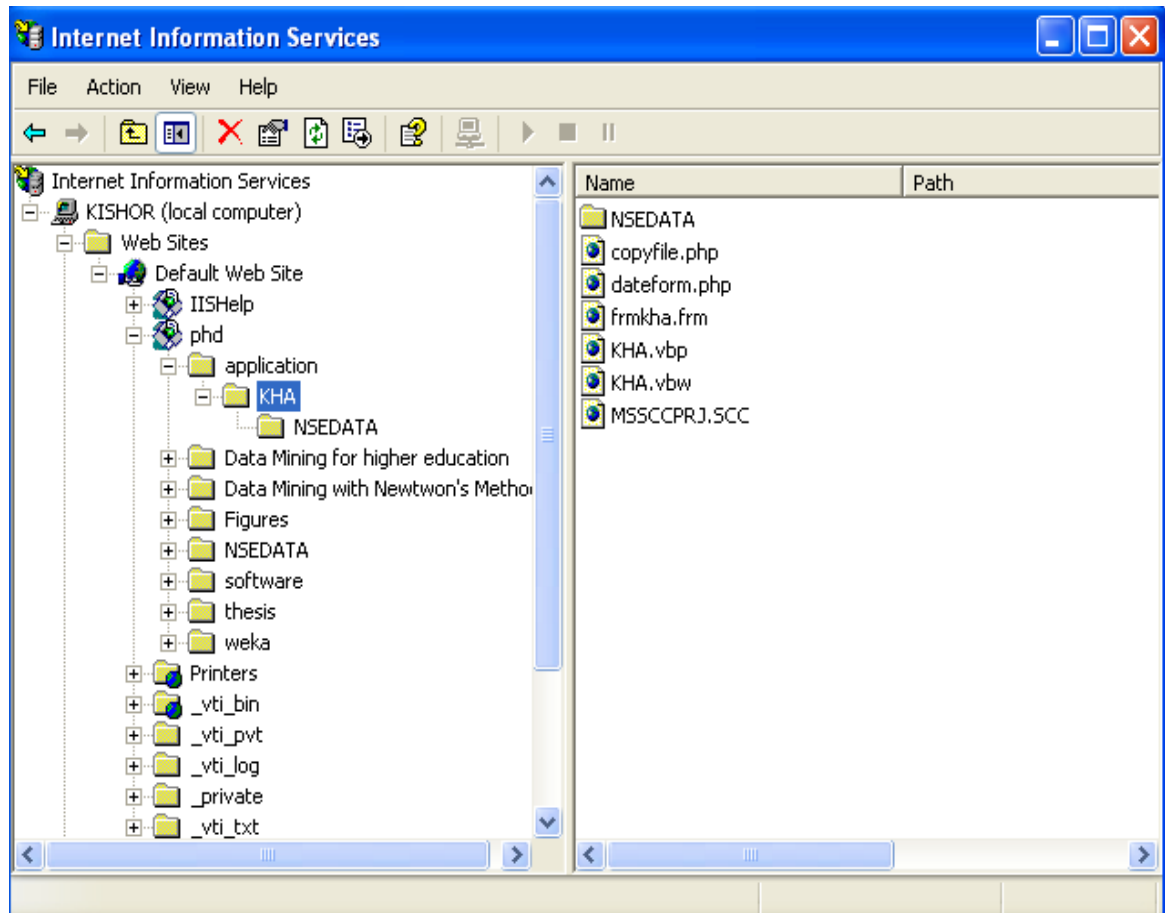
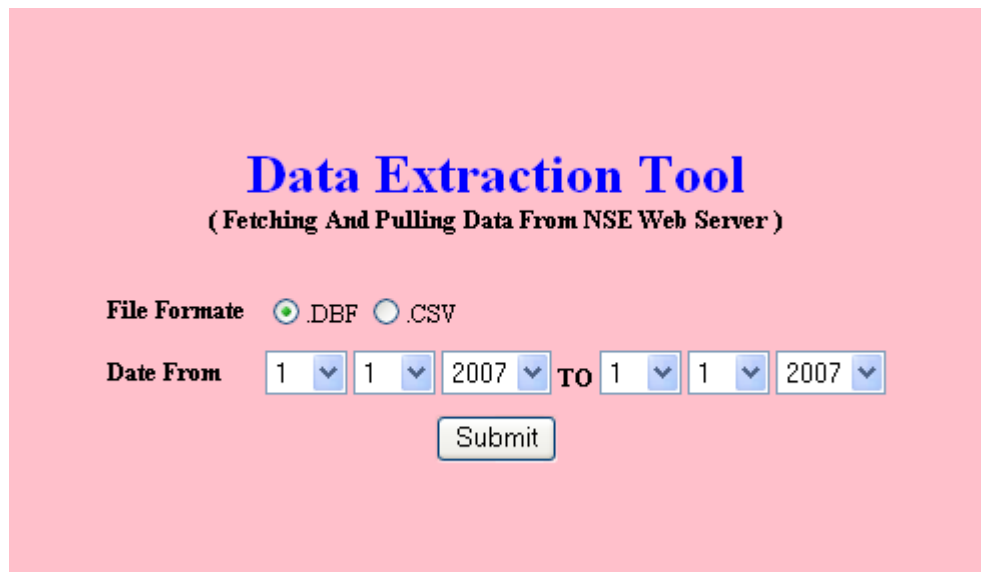


Fig 5.4 DateForm Module

In right window pan, right click on dateform.php and select “Browse” option. It will display following web page.



The screenshot shows a web form titled "Data Extraction Tool" in blue text, with a subtitle "(Fetching And Pulling Data From NSE Web Server)" in black. Below the title, there are two radio buttons for "File Formate": ".DBF" (selected) and ".CSV". Underneath, the "Date From" section contains three dropdown menus for day, month, and year, followed by the word "TO" and another three dropdown menus for day, month, and year. The date range is set to "1 1 2007" to "1 1 2007". A "Submit" button is located below the date fields.

Fig 5.5: Range of Date Screen

In this web page we can specify the range of date for the Bhavcopy of equities. It provides two options either CSV or DBF. We can select any one format out of them. Whatever format is selected, it will extract the data for each day and save as separate file at specified location in our computer. For example 2007_04_02bhav.dbf is the file of 2nd April 2007. Similarly it will save all the separate files for the specified date of range.

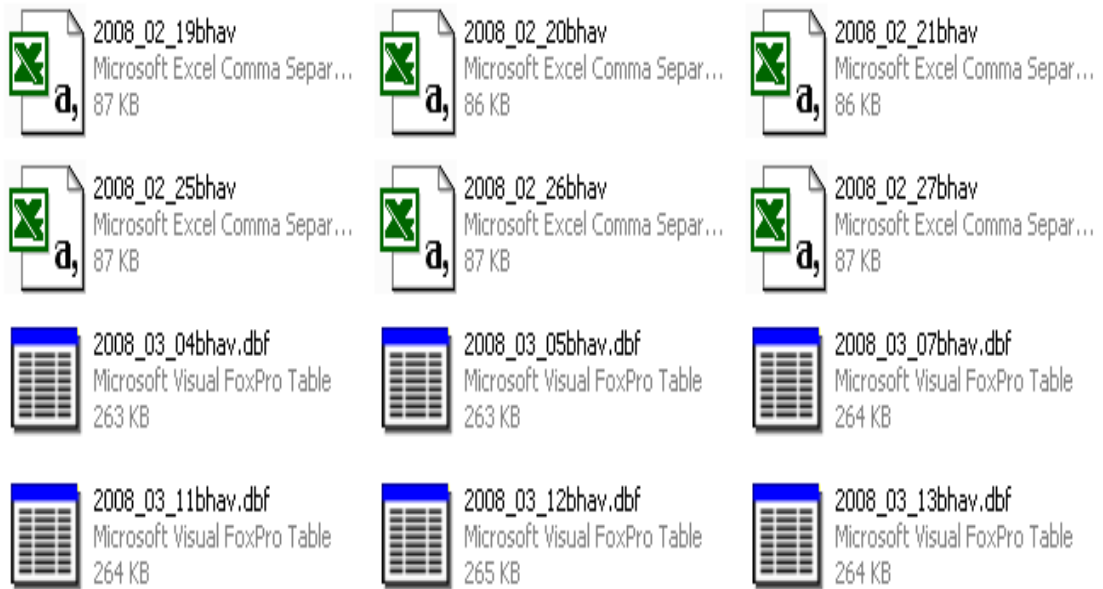


Fig 5.6: Extracted Data Files

All the extracted data files from nseindia.com are displayed in figure 5.6. These files either in CSV format or DBF format whatever format we have been selected. If any file exists previously and we extract that file again then this module will refresh that file, it also gives warning message for that. This way past and historical data can be refreshed also if necessary.

5.2.6 Implementation of Data Extraction Tool

There are two files in PHP developed for data extraction from nseindia.com to host machine. One file is dateform.php another is copyfile.php.

The dateform.php file displays the range between two dates and allows selecting the file format either CSV or DBF. After entering range of date and selecting file format, when "Submit" button is clicked, automatically copyfile.php is executed in background and it extracts the data from web server of NSE and saves into our local machine.

5.2.6.1 Code of dateform.php file

```

<html>
<head>
    <title>Date Range</title>
</head>
<body bgcolor="#FFFFFF">
<form action=copyfile.php>
<table border=0>
    <tr>
        <td><b>Date From </b></td>
        <td>
            <select id=cmbdayfrom name=cmbdayfrom>
                <?
                    for ($i=1;$i<=31;$i++)
                    {
                        echo "<option value=\".$i.\">".$i."</option>";
                    }
                ?>
            </select>
            <select id=cmbmonthfrom name=cmbmonthfrom>
                <?
                    for ($i=1;$i<=12;$i++)
                    {
                        echo "<option value=\".$i.\">".$i."</option>";
                    }
                ?>
            </select>
            <select id=cmbyearfrom name=cmbyearfrom>
                <?
                    for ($i=2007;$i<=2010;$i++)
                    {
                        echo "<option value=\".$i.\">".$i."</option>";

```

```

    }
?>
</select>
<b>TO</b>
<select id=cmbday name=cmbdayto>
<?
    for ($i=1;$i<=31;$i++)
    {
        echo "<option value=\".$i.\">\".$i.\"</option>\";
    }
?>
</select>
<select id=cmbmonth name=cmbmonthto>
<?
    for ($i=1;$i<=12;$i++)
    {
        echo "<option value=\".$i.\">\".$i.\"</option>\";
    }
?>
</select>
<select id=cmbyear name=cmbyearto>
<?
    for ($i=2007;$i<=2010;$i++)
    {
        echo "<option value=\".$i.\">\".$i.\"</option>\";
    }
?>
</select>
</td>
</tr>
<tr>
    <td colspan=2 align=center><input type=submit value="Submit"></td>
</tr>

```

```

</table>
</form>
</body>
</html>

```

5.2.6.2 Code of copyfile.php file

```

<html>
<head>
    <title>File Copy</title>
</head>
<body bgcolor="#FFFFFF">
<?
/
/
copy("http://www.nseindia.com/content/historical/EQUITIES/2007/AUG/cm28A
UG2007bhav.dbf","E:\gautam\NSEDATA\Database\check.dbf");
    set_time_limit(0); //set script execution time
    $dt1=mktime(0,0,0,$_GET["cmbmonthfrom"],$_GET["cmbdayfrom"],$_GET["c
mbyearfrom"]);
    $dt2=mktime(0,0,0,$_GET["cmbmonthto"],$_GET["cmbdayto"],$_GET["cmbye
arto"]);
    echo "Date From :".date("d-M-Y",$dt1)." To ".date("d-M-Y",$dt2);
    for($i=$dt1;$i<=$dt2;$i=$i+24*60*60)
    {
        $d=date("w",$i);
        if ($d!=0 && $d!=6)
        {
            $urldata="http://www.nseindia.com/content/historical/EQUITIES/" .date("Y",$i)."/"
            .strtoupper(date("M",$i))."/cm".date("d",$i).strtoupper(date("M",$i)).date("Y",$i).
            "bhav.dbf";
            $dest="D:\\PHD\\NSEDATA\\Database\\".date("Y",$i)."_".date("m",$i)."_".date(
            "d",$i)."bhav.dbf";
            echo "<br>".$urldata."==" . $dest;
            copy ($urldata,$dest);

```

```

    }
}
echo "File Copies Completed ";
?>
</body>
</html>

```

5.2.7 Advantages of Data Extraction Tool

Advantage of this Data Extraction tool is that it works very fast and it occupies very less bytes of memory. This tool is also machine independent and for further development we want to put it open source software category.

There is no limitation on the range of date to be specified for extraction of data. This tool also works very fast, it takes about 10 seconds to extract, pull and fetch the data of one day, almost 265 Kb of data. These are main advantages of this tool. This facility is not available in nseindia.org. But for our practical implementation it is very much necessary to extract the data this way.

5.2.8 Limitations of Data Extraction Tool

Limitation of this tool is that extracted data is stored as separated file for each date. This way the data is stored as separate file is meaningless. To make it meaningful, Data Transformation is very much necessary. Another disadvantage is that, during the extraction and fetching of data from web server of NSE, if our Internet connection is lost accidentally or anyway, in such case manually we have to see that how many files are pulled. Then after again we have to specify the range of date for remaining days. When Internet speed is slow, we experienced that the fetched file is corrupted, that is also main disadvantage of this tool. We could not find solution for these advantages.

5.3 Data Transformation

After extracting the data, next phase is to transform the data. It is not necessary that extracted data available in our desired format. The extracted data may be available in various formats. This Data Transformation model is developed to

transform the data from CSV or DBF format to Microsoft Access (MS-Access), SQL Server or Oracle database. This way this model is very much important to transform the data from various formats to our desired format. The Data Transformation tool is developed in Microsoft Visual Basic.

5.3.1 Data Transformation Model

Format - 1

Format - 2

DATA

Format - 3

TRANSFORMED DATA

Format - 4

Fig 5.7: Data Transformation Model

As shown in figure 5.7, the pulled, fetched or extracted data may available different formats, like CSV, DBF, MDB and others. Operational data pulled and fetched from various places are always in various format. For our study we have to transform all these data into our own format in data warehouse.

As we discussed earlier it is necessary to transform these operational data scattered in various files into our format. The developed Data Transformation tool works very efficiently.

5.3.2. Process of Data Transformation

This tool is developed in Microsoft Visual Basic for practical work implementation of Data Transformation module. This can be implemented in

many programming languages also. After extraction of data, when we execute this Data Transformation tool, following screen is displayed.

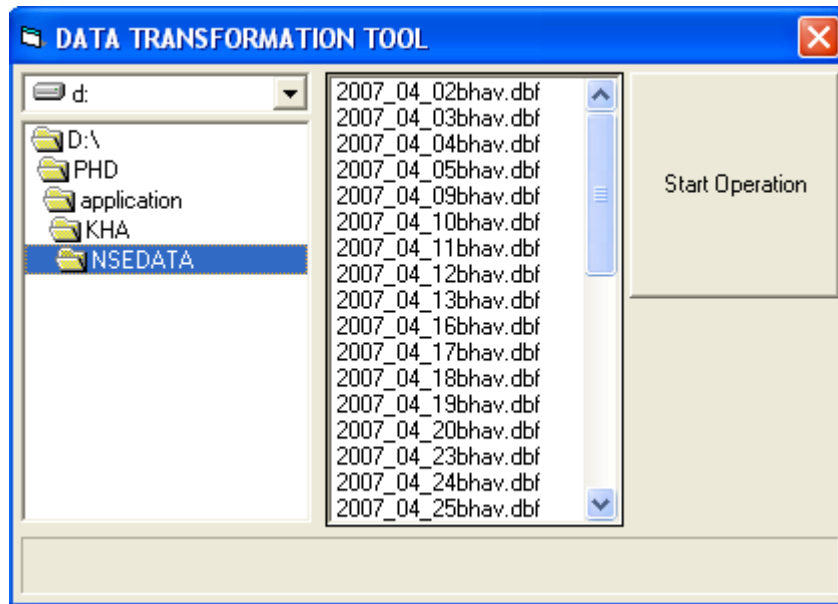


Fig 5.8: Data Transformation Tool

In this form when we select drive and folder, the content of the selected folder is displayed in right side window. In Figure 5.8, drive D is selected and NSEDATA folder is selected in D drive. The DBF files stored in the selected folder are displayed in right side window. These files are operational data that is extracted, pulled and fetched. Our tool will transform these operational data stored into different DBF files into MDB (Microsoft Database File) format. One MDB file is created there when we click on “Start” button, it will start Transformation of data from DBF to MDB.

Microsoft Access database file KHA.MDB is already created. This tool will transform all the operational data scattered into various DBF file will be gathered, converted and transformed into table NSEDATA in KHA.DBF file. After successfully transformation of all the data it will display following message,

it shows that how much total number of records is transformed into access database.



Fig 5.9: Transformed Records

5.3.3 Implementation of Data Transformation Tool

Here legacy system is in FoxPro database files or Excel CSV files. Data warehouse is in Microsoft SQL Server or Microsoft Access. To execute this tool Microsoft Visual Basic, Microsoft Access and Microsoft SQL server 2005 must be installed. This tool is developed in Visual Basic, It converts the records stored into various DBF files into single Access or SQL table. For this Microsoft Visual FoxPro driver is used. There are about 10 fields in DBF file, when it is transformed into Access or SQL database, one more field is added that is time data.

This tool transforms data very fast; in our study we have extracted the DBF or CSV files for one year, one file for each working day. Each file is almost 265 Kb of memory and almost 235 working days. This tool will transform the data 62275 Kb data that is of one year for our practical work.

5.3.3.1 Code of Data Transformation Tool

```

Option Explicit
Dim cn As New rdoConnection
Dim rs As rdoResultset
Dim SQL As String
Dim cnAccess As New ADODB.Connection
Dim rsAccess As New ADODB.Recordset
Dim SQLAccess As String
Dim FileName As String
Dim d As String, m As String, y As String
Dim i As Integer
Dim incr As Integer
Private Sub cmdStart_Click()
cn.Connect = "SourceType=DBF;SourceDB=" & Dir1.Path &
";Driver={Microsoft Visual FoxPro Driver}"
cn.CursorDriver = rdUseOdbc
cn.EstablishConnection "rdDriverNoPrompt"
cnAccess.Open "provider=microsoft.jet.oledb.4.0;data source=" & Dir1.Path &
"kha.mdb"
rsAccess.Open "select * from nsetable", cnAccess, adOpenStatic,
adLockOptimistic
If Not (rsAccess.EOF And rsAccess.BOF) Then
cnAccess.Execute "delete from nsetable"
End If
For i = 0 To File1.ListCount - 1
FileName = File1.List(i)
If Mid(FileName, Len(FileName) - 3) = ".dbf" Then
SQL = "select * from " & Mid(FileName, 1, Len(FileName) - 4)
Set rs = cn.OpenResultset(SQL, rdOpenKeyset, rdConcurRowVer)
rs.MoveFirst
Do While Not rs.EOF
rsAccess.AddNew

```



```

rsAccess.Fields(0) = rs(0)
rsAccess.Fields(1) = rs(1)
rsAccess.Fields(2) = rs(2)
rsAccess.Fields(3) = rs(3)
rsAccess.Fields(4) = rs(4)
rsAccess.Fields(5) = rs(5)
rsAccess.Fields(6) = rs(6)
rsAccess.Fields(7) = rs(7)
rsAccess.Fields(8) = rs(8)
rsAccess.Fields(9) = rs(9)
rsAccess.Fields(10) = rs(10)
d = Mid(fileName, 9, 2)
m = Mid(fileName, 6, 2)
y = Mid(fileName, 1, 4)
rsAccess.Fields(11) = d & m & y
rsAccess.Update
rs.MoveNext
Loop
End If
incr = incr + CInt(100 / File1.ListCount)
If incr > 100 Then
    incr = 100
End If
ProgressBar1.Value = incr
Next
MsgBox rsAccess.RecordCount & " Records Transferred In Access Database"
Set rs = Nothing
cn.Close
rsAccess.Close
cnAccess.Close
End Sub
Private Sub Dir1_Change()
File1.Path = Dir1.Path

```

```

End Sub
Private Sub Drive1_Change()
Dir1.Path = Drive1.Drive
End Sub

```

5.3.4 Advantages of Data Transformation Tool

Advantage of this Data Transformation tool is that it works very fast and it occupies very less bytes of memory. This tool is also machine independent and for further development we want to put it open source software category.

There is no limitation on the number of files to be transformed from legacy system into data warehouse. This tool also works very fast, it takes about 2 minutes to fetch and transform the data of one year. We have been studied many ready made software for this facility, but we could not found this type of data transformation. For our practical implementation it is very much necessary to transform the data this way.

5.4 Data Mining

After extraction and transformation of the data from legacy system, the data is being useful for the mining purpose. The mining of the data is only possible, if the data is available in desired format and stored into data warehouse.

5.4.1 Data Mining Model

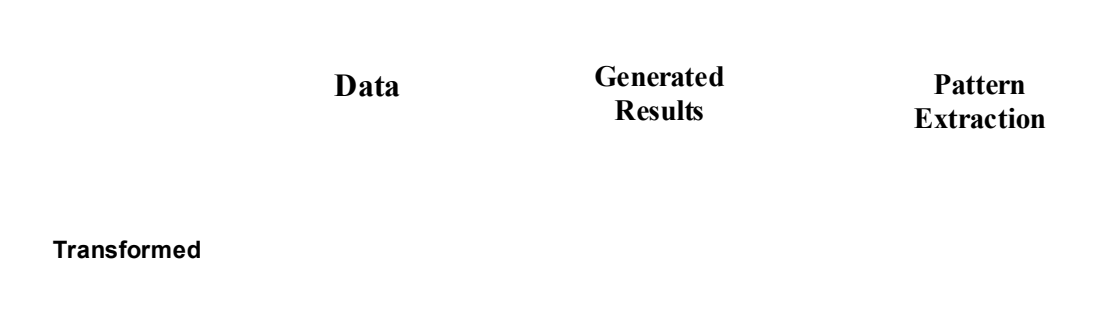


Fig 5.10: Data Mining Model

As shown in Fig 5.10 the mining of transformed data is performed and results are generated. These results are studied and the experts can extract patterns.

5.4.2 Process of Data Mining

To implement the data mining process, a web-based tool has been developed using PHP, HTML, and DHTML technology for the mining of NSE historical data. Transformed and stored operational data in Access database is connected with this tool using ODBC connection. This tool performs statistical methods like Euclidean Distance, Variance, and Correlation-coefficient etc. This is visualize tool, generates various charts like bar chart, pie chart, etc. This tool also generates numerical results. These charting result and numerical results are useful to extract patterns.

When this data-mining tool is executed, following screen will be displayed.

Data Mining of NSE Historical Data	
Data Mining Graphical View	1 Bar Chart (Open, Close, High, Low)
	2 Bar Chart (Variance)
	3 1. Bar Chart For Positive Difference (Open - Close) 2. Bar Chart For Negative Difference (Open - Close)
	4 Bar Chart (Volume)
	5 Pie Chart (Variance)
Data Mining Numeric View	1 Display Open, Close, Euclidean Distance, Variance (Open) and Average (Open)

Fig 5.11: Home Page of Data Mining Tool

This data-mining tool facilitates the graphical view and numerical view of mining data. Graphical view provides data in charting,

1. First Bar chart shows open, close, high and low rate of the selected script for the selected range of date.
2. Second Bar chart shows variance of the various selected script for the selected range of date.

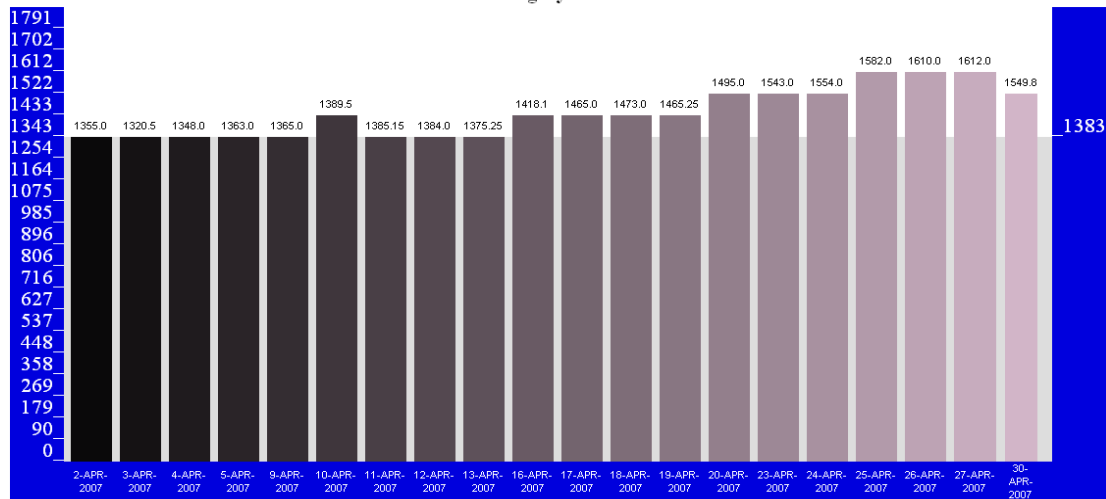
3. Third Bar chart shows positive difference of open and close rate of the selected script for the selected range of date.
4. Fourth Bar chart shows negative difference of open and close rate of the selected script for the selected range of date.
5. Fifth chart is Pie chart shows the variance of selected one or more scripts for the given range of date.

Numerical view results the open, close, Euclidean distance of open, close and Average trading quantity of the selected script for the selected range of date. It also calculates the average of open rate, close rate, high rate, low rate and variance of open, close, high and low rate.

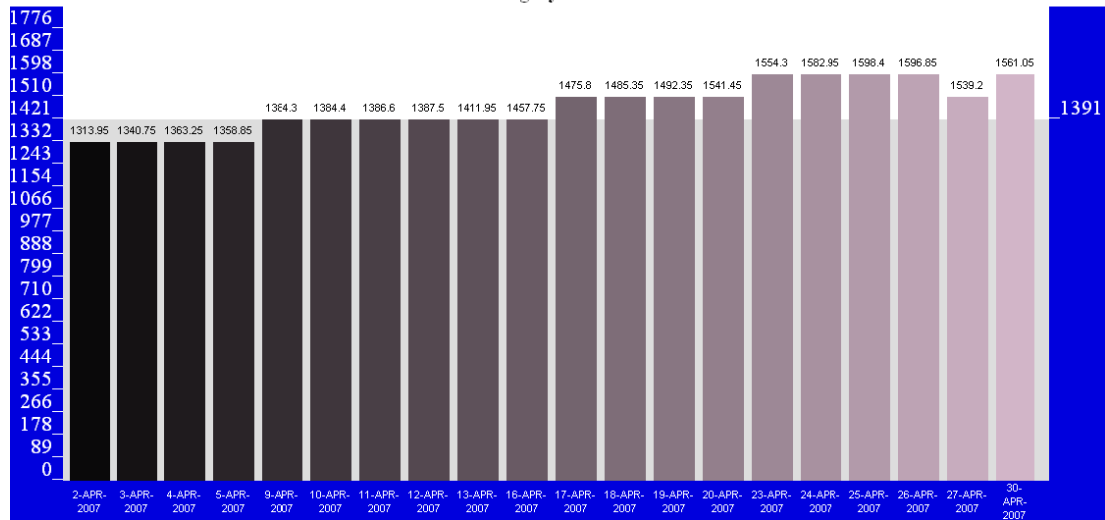
5.4.3 Implementation of Data Mining Tool

<h1>Script's Bar-Chart</h1>	
Select a Company : RELIANCE ▼	
Start Date 2007-04-01	to End Date 2007-04-30
Catagory: OPEN ▼	
search	

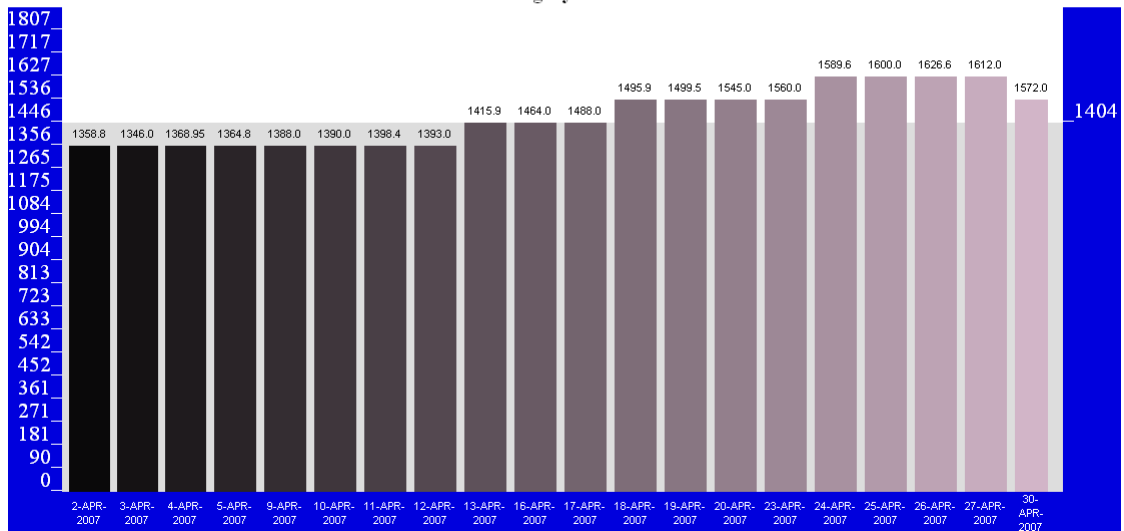
Script :RELIANCE
2007-04-01 to 2007-04-30
Catagory :OPEN



Script :RELIANCE
2007-04-01 to 2007-04-30
Category :CLOSE



Script :RELIANCE
2007-04-01 to 2007-04-30
Category :HIGH



Script :RELIANCE
2007-04-01 to 2007-04-30
Category :LOw

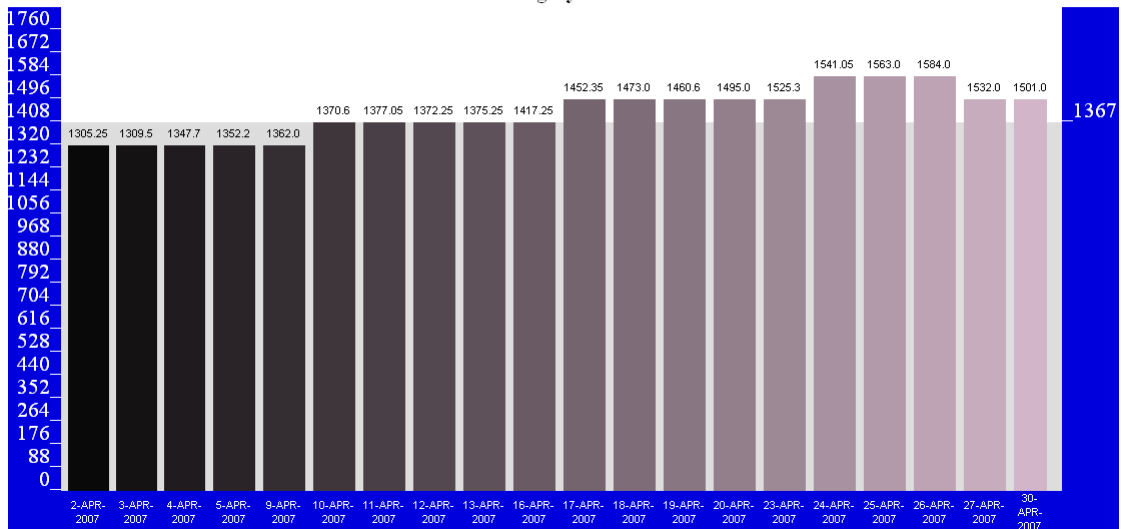


Fig 5.12: Scripts Bar Chart


The above script's bar charts are generated for the script Reliance for the range of date 1st April 2007 to 30th April 2007. It also gives the average of open, close, high and low rate.

Category	Average Rate
Open	1383
Close	1391
High	1404
Low	1367

From above mined data it can be analyzed that there is not remarkable movement in the script Reliance for the selected range of dates.

Variance Bar-Chart

Select Scripts :

<input type="checkbox"/> 3IINFOTECH	
<input type="checkbox"/> 3MINDIA	
<input type="checkbox"/> AARTIDRUGS	
<input type="checkbox"/> AARTIIND	
<input type="checkbox"/> AARVEEDEN	
<input type="checkbox"/> ABAN	
<input type="checkbox"/> ABB	
<input type="checkbox"/> ABGSHIP	
<input type="checkbox"/> ABIRLANUVO	
<input type="checkbox"/> ABSHEKINDS	

Start Date: **to End Date:**

Variance Bar-Chart

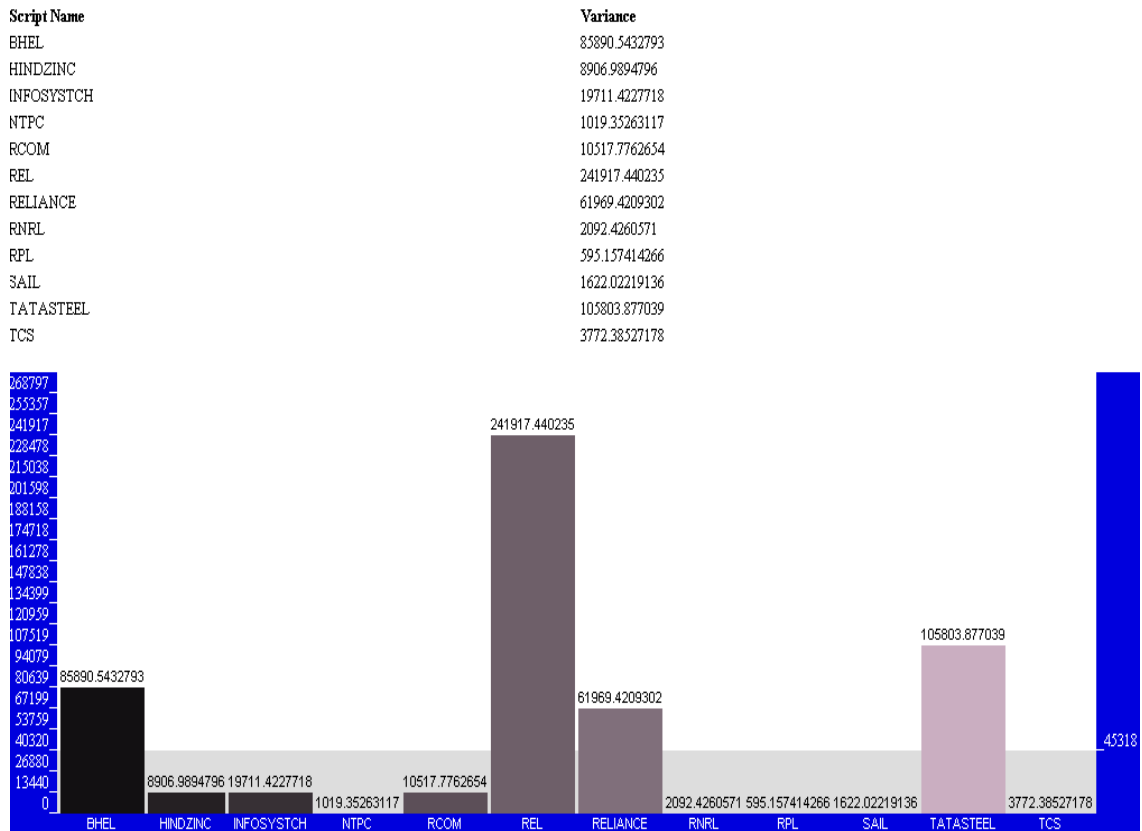


Fig 5.13: Bar Chart for Variance

Above script's Bar chart is generated for the variance of the selected scripts for the selected range of dates, that is 1st January 2008 to 16th May 2008. Here variance of open rate is calculated; if variance is more then fluctuation in the open rate is higher.

Script Name	Variance
-------------	----------

BHEL	85890.5433
HINDZINC	8906.9895
INFOSYSTCH	19711.4228
NTPC	1019.3526
RCOM	10517.7763
REL	241917.4402
RELIANCE	61969.4209
RNRL	2092.4261
RPL	595.1574
SAIL	1622.0222
TATASTEEL	105803.8770
TCS	3772.3853

Above mined data results the variance of the various selected script of the open rate for the range of date 1st January 2008 to 16th May 2008. Variance of the script REL is highest, it can be said that this script has very much fluctuation in open rate for the given range of date. The script RPL is lowest variance, it means this script has not much fluctuation in open rate compare to the other selected scripts.

Script(Open - Close in Plus) Bar-Chart

Script :RELIANCE

2008-01-01 to 2008-01-31

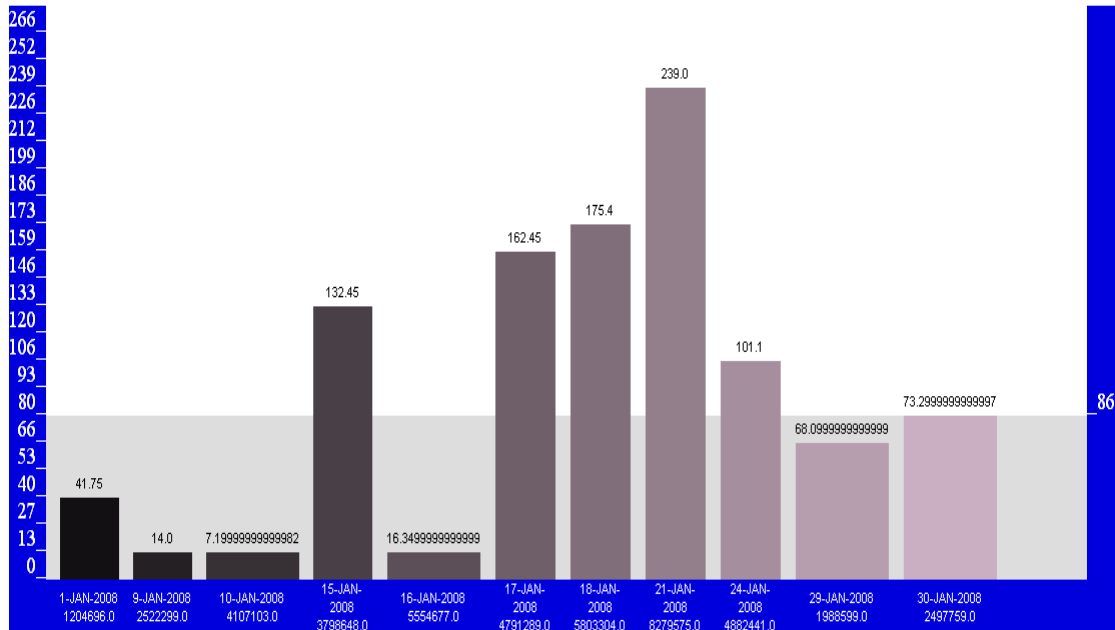


Fig 5.14: Bar Chart of (Open – Close in Plus)

OPEN	CLOSE	Euclidean Distance	Trading Quantity	TIMESTAMP
2890	2848.25	41.75	1204696	1-JAN-2008
3047.65	3033.65	14	2522299	9-JAN-2008
3035	3027.8	7.2	4107103	10-JAN-2008
3298	3165.55	132.45	3798648	15-JAN-2008

3110	3093.65	16.35	5554677	16-JAN-2008
3159	2996.55	162.45	4791289	17-JAN-2008
2975	2799.6	175.4	5803304	18-JAN-2008
2779.8	2540.8	239	8279575	21-JAN-2008
2590	2488.9	101.1	4882441	24-JAN-2008
2645	2576.9	68.1	1988599	29-JAN-2008
2545.2	2471.9	73.3	2497759	30-JAN-2008

This Bar chart displays the positive Euclidean Distance of open and close rate for the month January 2008 for the script Reliance. It also displays the positive difference of open and close rate with trading quantity in figure for the script Reliance. If Euclidean Distance is positive, it means rate of the script is fall down. The above table shows the dates when price of the script was fall.

We are interesting to compare the Euclidean Distance with Trading Quantity. The correlation coefficient of Euclidean distance and Trading Quantity is 0.67. From this result it can be concluded that there is positive correlation coefficient between Euclidean Distance and trading quantity. When rate is fall down during the day, trading quantity is also decrease 67% and rate is rise, trading quantity also increase 67%.

Script(Open - Close in Minus) Bar-Chart

Script :RELIANCE
2008-01-01 to 2008-01-31

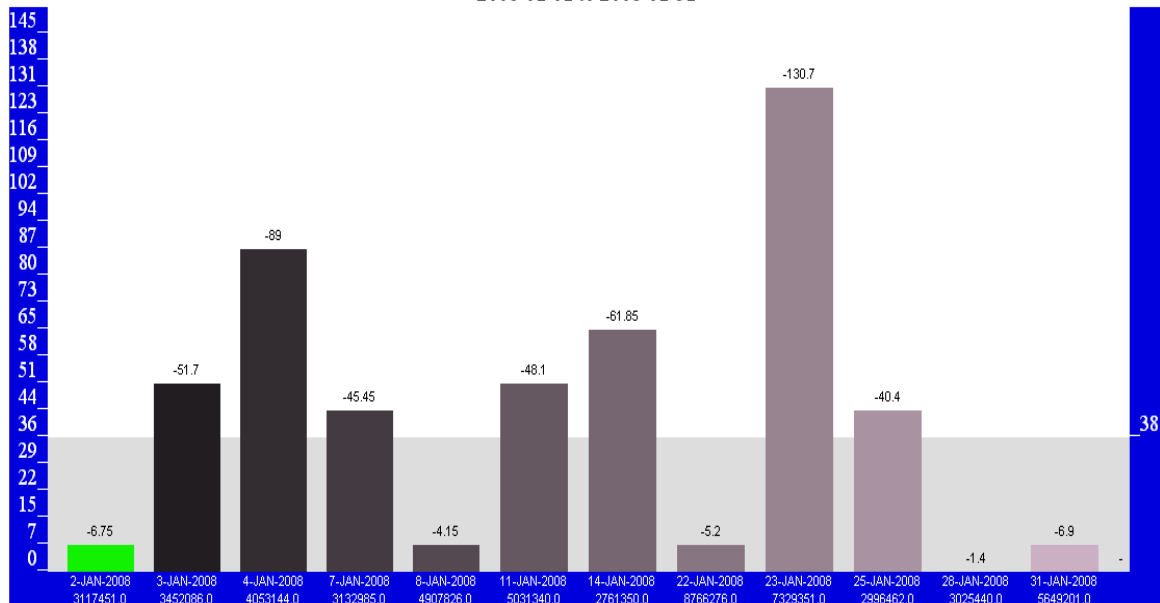


Fig 5.15: Bar Chart of (Open – Close in Minus)

OPEN	HIGH	Euclidean Distance	Trading Quantity	TIMESTAMP
2855	2884	6.75	3117451	2-JAN-2008
2852	2930	51.7	3452086	3-JAN-2008
2904	3019.8	89	4053144	4-JAN-2008
2974.8	3030	45.45	3132985	7-JAN-2008

3050.5	3084	4.15	4907826	8-JAN-2008
3079.5	3140	48.1	5031340	11-JAN-2008
3159	3240	61.85	2761350	14-JAN-2008
2352.25	2488	5.2	8766276	22-JAN-2008
2420	2825	130.7	7329351	23-JAN-2008
2575	2625	40.4	2996462	25-JAN-2008
2565	2594.7	1.4	3025440	28-JAN-2008
2472	2569.7	6.9	5649201	31-JAN-2008

This Bar chart displays the negative Euclidean Distance of open and close rate for the month January 2008 for the script Reliance. It also displays the negative difference of open and close rate with trading quantity in figure for the script Reliance. If Euclidean Distance is negative, it means rate of the script is increased. The above table shows the dates when price of the script was increased.

We are interesting to compare the Euclidean Distance with Trading Quantity. The correlation coefficient of Euclidean distance and Trading Quantity is 0.098. From this result it can be concluded that there is positive correlation coefficient between Euclidean Distance and trading quantity. When rate is increased during the day, trading quantity is also increase 9.8% and rate is fall down during the day, trading quantity also decrease 9.8%.

From above results finally it can be concluded that for the script Reliance in the month of January 2008, if rate of the share is fall down during the day then it

effects on the trading volume. If the rate of the share is increasing during the day then it does not affect much on trading volume.

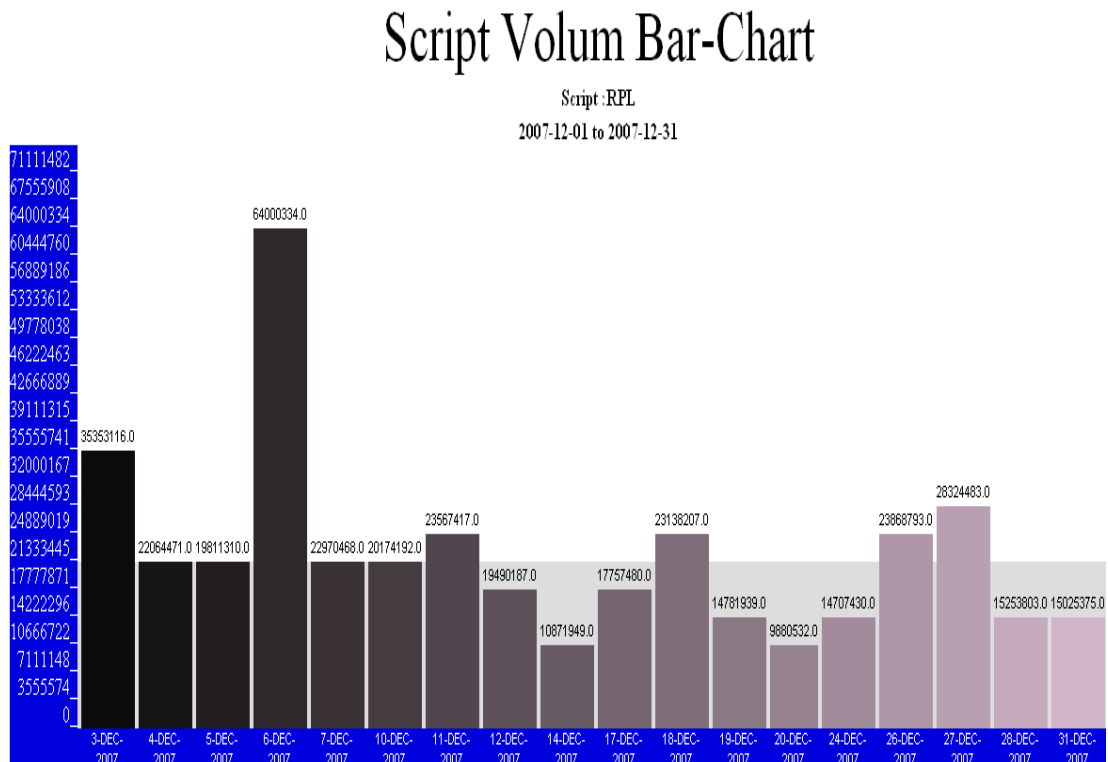


Fig 5.16: Bar Chart for Volume

This Bar chart displays the volume (Trading Quantity) for the script RPL for the month December 2007. Average volume of the script for this month is 21107447. Highest volume is 64000334 and lowest volume is 9880532. It can be said from the graph and volume data that this script has much fluctuation in volume in this month.

OPEN	HIGH	LOW	CLOSE	TOTTRDQTY	TIMESTAMP
215	226.4	215	223.25	35353116	3-DEC-2007
225	225	217.15	218.8	22064471	4-DEC-2007
219	224.8	218.15	223.35	19811310	5-DEC-2007
225.4	236.7	224.35	227	64000334	6-DEC-2007
231.65	232.4	221.2	225.4	22970468	7-DEC-2007
225	228.65	221.7	227.25	20174192	10-DEC-2007
227	232.3	226.5	227.85	23567417	11-DEC-2007
223.9	230.65	222	229.45	19490187	12-DEC-2007
225	225	220.75	221.65	10871949	14-DEC-2007
223	223	205.25	208.25	17757480	17-DEC-2007
209	218	203.55	210.6	23138207	18-DEC-2007
213.8	215.4	207.35	210.55	14781939	19-DEC-2007
211	212.8	207	208.1	9880532	20-DEC-2007
211.3	217.85	210	216.55	14707430	24-DEC-2007
219.8	225.45	217.35	223.9	23868793	26-DEC-2007

226	227.45	218.15	219.45	28324483	27-DEC-2007
217.45	224.4	216.2	222.8	15253803	28-DEC-2007
224.8	226.6	223	223.4	15025375	31-DEC-2007

Following table results the correlation coefficient between open, close, high and low rate with the total trading volume. Variance is also calculated and compared with the correlation coefficient.

	OPEN	HIGH	LOW	CLOSE
Correlation Coefficient	0.240314	0.619976	0.340396	0.372023
Variance	41.83948	37.99893	48.26033	47.56095

From above table, it can be analyzed that, there is positive correlation between open, high, low and close rate with the volume. There is remarkable positive correlation between high rate and volume. For this script for this range of the dates, when high rate is increasing, volume also increasing and decreasing in high rate will decrease volume 60%.

Another interesting thing is that when correlation coefficient is remarkable positive for high rate compare to open, low and close rate; variance for the high rate is remarkable low compare to open, low and close rate.

Variance Pie-Chart(1-April-2007 to 16-May-2008)

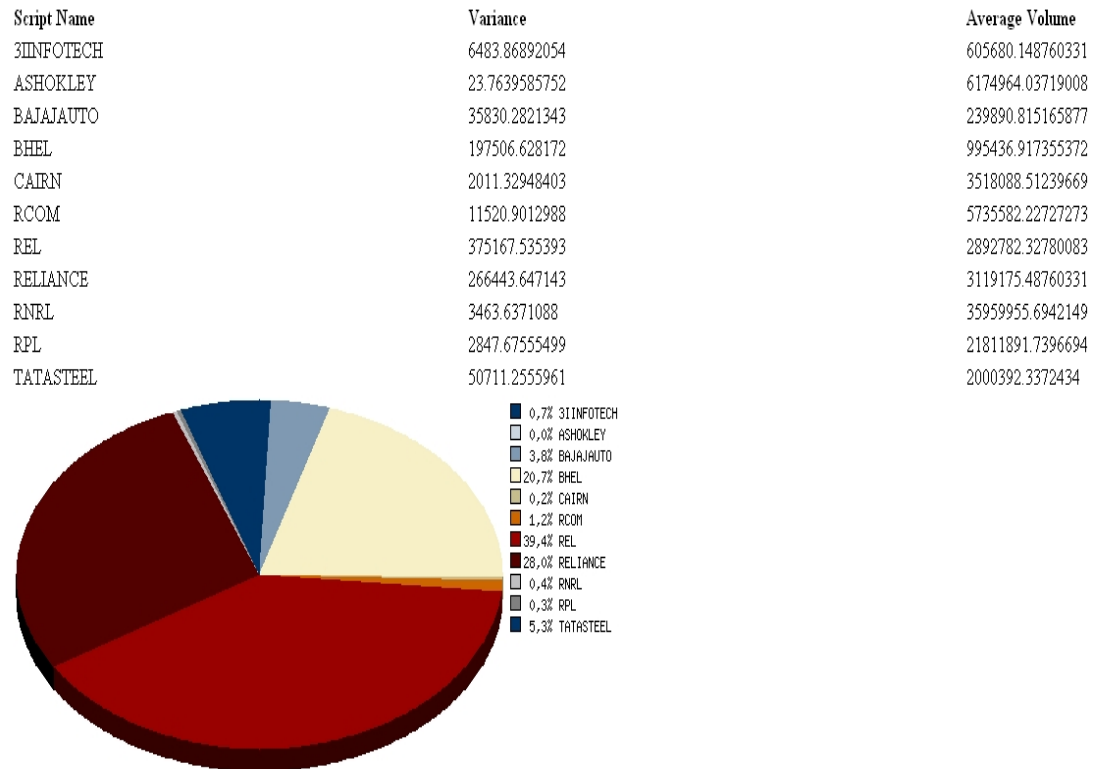


Fig 5.17: Pie Chart of Variance

Script Name	Variance	Percentage	Average Volume
3IINFOTECH	6483.87	0.70	605680.15
ASHOKLEY	23.76	0.00	6174964.04
BAJAJAUTO	35830.28	3.80	239890.82
BHEL	197506.63	20.70	995436.92
CAIRN	2011.33	0.20	3518088.51
RCOM	11520.90	1.20	5735582.23
REL	375167.54	39.40	2892782.33
RELIANCE	266443.65	28.00	3119175.49
RNRL	3463.64	0.40	35959955.69
RPL	2847.68	0.30	21811891.74
TATASTEEL	50711.26	5.30	2000392.34

This pie chart displays the variance of open rate of the selected scripts for the selected time period, which is 1st April 2007 to 16th May 2008. We can see that there is maximum variance is of the script REL (39.40%) and minimum variance is of the script ASHOKLEY (0.0%). Maximum average volume is of the script RNRL (35959955.69) and minimum variance is of the script BAJAJAUTO (239890.82).

It can be concluded that the script RNRL having variance is 3463.64 but volume of this script is highest. Similarly the variance of REL is 375167.54 which is maximum but volume is only 2892782.33.

Percentage of the variance of RNRL in REL is 0.92 and percentage of average volume of REL in RNRL is 8.04.

This analysis says that the script having minimum variance having high volume and maximum variance having low volume for these selected scripts for the selected time period.

Numerical View

Company Name	Date	Open	Close	Euclidean Distance	Trading Volume
RPL	1-NOV-2007	250.0	261.2	11.2	144509671.0
RPL	2-NOV-2007	246.65	269.6	22.95	70988169.0
RPL	5-NOV-2007	271.0	267.6	-3.4	75749282.0
RPL	6-NOV-2007	271.7	220.15	-51.55	150327863.0
RPL	7-NOV-2007	215.0	218.8	3.8	85111710.0
RPL	8-NOV-2007	215.0	221.05	6.05	46105589.0
RPL	9-NOV-2007	226.9	223.2	-3.7	9470210.0
RPL	12-NOV-2007	214.7	218.25	3.55	27332368.0
RPL	13-NOV-2007	220.0	217.05	-2.95	34261680.0
RPL	14-NOV-2007	225.0	215.2	-9.8	65742523.0
RPL	15-NOV-2007	224.7	212.7	-12	40662603.0
RPL	16-NOV-2007	208.7	214.45	5.75	33600352.0
RPL	19-NOV-2007	215.95	208.8	-7.15	28154379.0
RPL	20-NOV-2007	205.65	207.85	2.2	28848961.0
RPL	21-NOV-2007	209.0	203.45	-5.55	38220473.0
RPL	22-NOV-2007	208.0	208.8	0.8	53963652.0
RPL	23-NOV-2007	215.0	209.6	-5.4	35372893.0
RPL	26-NOV-2007	213.0	204.05	-8.95	60244235.0
RPL	27-NOV-2007	201.0	198.15	-2.85	26772972.0
RPL	28-NOV-2007	199.9	192.1	-7.8	30827741.0
RPL	29-NOV-2007	193.8	215.6	21.8	176590962.0
RPL	30-NOV-2007	214.7	217.9	3.2	66118769.0
Category:Open		Avg:221.152272727273	Variance:418.011244835		
Category:Close		Avg:219.343181818182	Variance:421.734214641		
Category:High		Avg:232.109090909091	Variance:680.062381657		
Category:Low		Avg:211.702272727273	Variance:372.65203491		

Numerical View

Company Name	Date	Open	Close	Euclidean Distance	Trading Volume
RPL	3-DEC-2007	215.0	223.25	8.25	35353116.0
RPL	4-DEC-2007	225.0	218.8	-6.2	22064471.0
RPL	5-DEC-2007	219.0	223.35	4.35	19811310.0
RPL	6-DEC-2007	225.4	227.0	1.6	64000334.0
RPL	7-DEC-2007	231.65	225.4	-6.25	22970468.0
RPL	10-DEC-2007	225.0	227.25	2.25	20174192.0
RPL	11-DEC-2007	227.0	227.85	0.85	23567417.0
RPL	12-DEC-2007	223.9	229.45	5.55	19490187.0
RPL	14-DEC-2007	225.0	221.65	-3.35	10871949.0
RPL	17-DEC-2007	223.0	208.25	-14.75	17757480.0
RPL	18-DEC-2007	209.0	210.6	1.6	23138207.0
RPL	19-DEC-2007	213.8	210.55	-3.25	14781939.0
RPL	20-DEC-2007	211.0	208.1	-2.9	9880532.0
RPL	24-DEC-2007	211.3	216.55	5.25	14707430.0
RPL	26-DEC-2007	219.8	223.9	4.1	23868793.0
RPL	27-DEC-2007	226.0	219.45	-6.55	28324483.0
RPL	28-DEC-2007	217.45	222.8	5.35	15253803.0
RPL	31-DEC-2007	224.8	223.4	-1.4	15025375.0
Category:Open		Avg:220.727777777778	Variance:39.5150617284		
Category:Close		Avg:220.422222222222	Variance:47.1139540466		
Category:High		Avg:225.158333333333	Variance:38.5052891137		
Category:Low		Avg:216.369444444444	Variance:47.7183879755		

Numerical View

Company Name	Date	Open	Close	Euclidean Distance	Trading Volume
RPL	1-JAN-2008	223.8	226.3	2.5	18703429.0
RPL	2-JAN-2008	227.7	228.1	0.4	17069475.0
RPL	3-JAN-2008	227.0	232.65	5.65	32976607.0
RPL	4-JAN-2008	233.4	244.75	11.35	41386541.0
RPL	7-JAN-2008	244.0	250.75	6.75	36890718.0
RPL	8-JAN-2008	255.0	246.8	-8.2	51083028.0
RPL	9-JAN-2008	246.4	232.1	-14.3	27562683.0
RPL	10-JAN-2008	234.0	217.95	-16.05	28546978.0
RPL	11-JAN-2008	222.0	220.3	-1.7	34347022.0
RPL	14-JAN-2008	215.2	225.45	10.25	19739668.0
RPL	15-JAN-2008	226.8	219.85	-6.95	15355074.0
RPL	16-JAN-2008	215.0	221.15	6.15	23692899.0
RPL	17-JAN-2008	224.0	219.2	-4.8	17709867.0
RPL	18-JAN-2008	215.0	208.6	-6.4	18409433.0
RPL	21-JAN-2008	204.0	171.95	-32.05	46871071.0
RPL	22-JAN-2008	165.0	147.0	-18	44940240.0
RPL	23-JAN-2008	155.0	168.95	13.95	35414678.0
RPL	24-JAN-2008	175.0	160.85	-14.15	32459074.0
RPL	25-JAN-2008	171.0	173.05	2.05	16980174.0
RPL	28-JAN-2008	170.0	168.25	-1.75	17746186.0
RPL	29-JAN-2008	172.0	171.45	-0.55	18034658.0
RPL	30-JAN-2008	173.0	161.05	-11.95	19050710.0
RPL	31-JAN-2008	163.0	158.8	-4.2	32151513.0
Category:Open		Avg:206.839130434783	Variance:928.075425331		
Category:Close		Avg:203.273913043478	Variance:1110.87933796		
Category:High		Avg:212.560869565217	Variance:973.840179155		
Category:Low		Avg:194.789130434783	Variance:1365.81825921		

Numerical View

Company Name	Date	Open	Close	Euclidean Distance	Trading Volume
RPL	1-APR-2008	157.25	167.05	9.8	19717128.0
RPL	2-APR-2008	169.5	165.4	-4.1	25137021.0
RPL	3-APR-2008	165.6	169.7	4.1	20331087.0
RPL	4-APR-2008	170.5	167.15	-3.35	27476191.0
RPL	7-APR-2008	168.7	171.45	2.75	19452114.0
RPL	8-APR-2008	172.7	168.9	-3.8	13824881.0
RPL	9-APR-2008	168.0	170.3	2.3	11825194.0
RPL	10-APR-2008	171.0	175.75	4.75	39022873.0
RPL	11-APR-2008	181.0	182.05	1.05	34569359.0
RPL	15-APR-2008	184.9	189.8	4.9	31677402.0
RPL	16-APR-2008	194.3	184.85	-9.45	34561389.0
RPL	17-APR-2008	189.25	187.15	-2.1	14141058.0
RPL	21-APR-2008	188.0	191.95	3.95	12762927.0
RPL	22-APR-2008	190.05	191.8	1.75	14332379.0
RPL	23-APR-2008	192.0	195.35	3.35	27615333.0
RPL	24-APR-2008	195.5	193.25	-2.25	20489976.0
RPL	25-APR-2008	193.85	194.8	0.95	18330120.0
RPL	28-APR-2008	196.0	193.65	-2.35	11575067.0
RPL	29-APR-2008	194.0	202.4	8.4	27025484.0
RPL	30-APR-2008	204.0	201.3	-2.7	22057269.0
Category:Open		Avg:182.305	Variance:165.463975		
Category:Close		Avg:183.2025	Variance:157.4905675		
Category:High		Avg:186.6275	Variance:152.555147125		
Category:Low		Avg:178.5625	Variance:166.163976106		

Numerical View

Company Name	Date	Open	Close	Euclidean Distance	Trading Volume
RPL	2-MAY-2008	203.6	203.0	-0.6	11532763.0
RPL	5-MAY-2008	202.0	200.9	-1.1	11361036.0
RPL	6-MAY-2008	202.0	200.1	-1.9	18425469.0
RPL	7-MAY-2008	198.7	199.85	1.15	13507243.0
RPL	8-MAY-2008	198.0	196.85	-1.15	9941846.0
RPL	9-MAY-2008	195.2	181.0	-14.2	32457605.0
RPL	12-MAY-2008	181.7	181.85	0.15	38836663.0
RPL	13-MAY-2008	183.2	178.35	-4.85	23069287.0
RPL	14-MAY-2008	175.0	184.0	9	25971300.0
RPL	15-MAY-2008	185.7	187.05	1.35	16505675.0
RPL	16-MAY-2008	188.5	187.2	-1.3	16624023.0
Category:Open		Avg:192.145454545455		Variance:86.3570247934	
Category:Close		Avg:190.922727272727		Variance:86.0403493614	
Category:High		Avg:194.963636363636		Variance:73.6655276279	
Category:Low		Avg:187.677272727273		Variance:125.510667801	

Month	Open		Close		High		Low	
	Average	Variance	Average	Variance	Average	Variance	Average	Variance
November	221	418	219	421	232	680	211	376
December	220	39	220	47	225	38	216	47
January	206	928	203	1110	212	973	194	1365
April	182	165	183	157	186	152	178	166
May	192	86	190	86	194	73	187	125

These are the numerical results of open, close, high and low rates of the selected script for the selected range of dates.

Average and variance are calculated month wise for the various categories. These numerical results are very interesting for analysis, we can observe that in the month of December variance of open, close high and low rate is very less, compared to other months. For the month of December it is ranging from 38 to 47. In which close and low rate has the same variance that is 47. For this month it can be said that this script is very steady, and there is not remarkable fluctuation in script RPL. The open, close, high and low rates also indicate about this. Maximum and minimum Euclidean Distance for the open and close rate are 8.25 and -14.75 respectively.

If we observe the data of the month of January, variance is very high for all the categories, compared to other months. It is ranging from 928 to 1365. Variance is higher and fluctuating much for the open, close, high and low categories. It can be said that the script is not steady and there is remarkable fluctuation in this month for this script. Maximum and minimum Euclidean Distance for the open and close rate are 13.95 and -32.05 respectively.

One another point is also highlighted here, total trading volume does not affect to the fluctuation on rate of the any category.

5.5 WEKA Software (Data Mining Software in Java)



An exciting and potentially far-reaching development in computer science is the invention and application of methods of machine learning. These enable a computer program to automatically analyze a large body of data and decide what information is most relevant. This crystallized information can then be used to automatically make predictions or to help people make decisions faster and more accurately.

The overall goal of our project is to build a state-of-the-art facility for developing machine learning (ML) techniques and to apply them to real-world data mining problems. The team of Weka project incorporated several standard ML techniques into a software "workbench" called WEKA, for Waikato Environment for Knowledge Analysis. With it, a specialist in a particular field is able to use ML to derive useful knowledge from databases that are far too large to be analyzed by hand. WEKA's users are ML researchers and industrial scientists, but it is also widely used for teaching.

The objectives of Weka project are to

- Make ML techniques generally available;
- Apply them to practical problems that matter to New Zealand industry;
- Develop new machine learning algorithms and give them to the world;
- Contribute to a theoretical framework for the field.

This machine-learning package is publically available and presents a collection of algorithms for solving real-world data mining problems. The software is written entirely in Java and includes a uniform interface to a number of standard ML techniques. So we can feel free to browse around.

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well suited

for developing new machine learning schemes. Weka is open source software issued under the GNU General Public License.

5.5.1 Implementation of Data Set into Weka

We have applied our data set into Weka for analysis and found some interesting results and patterns.

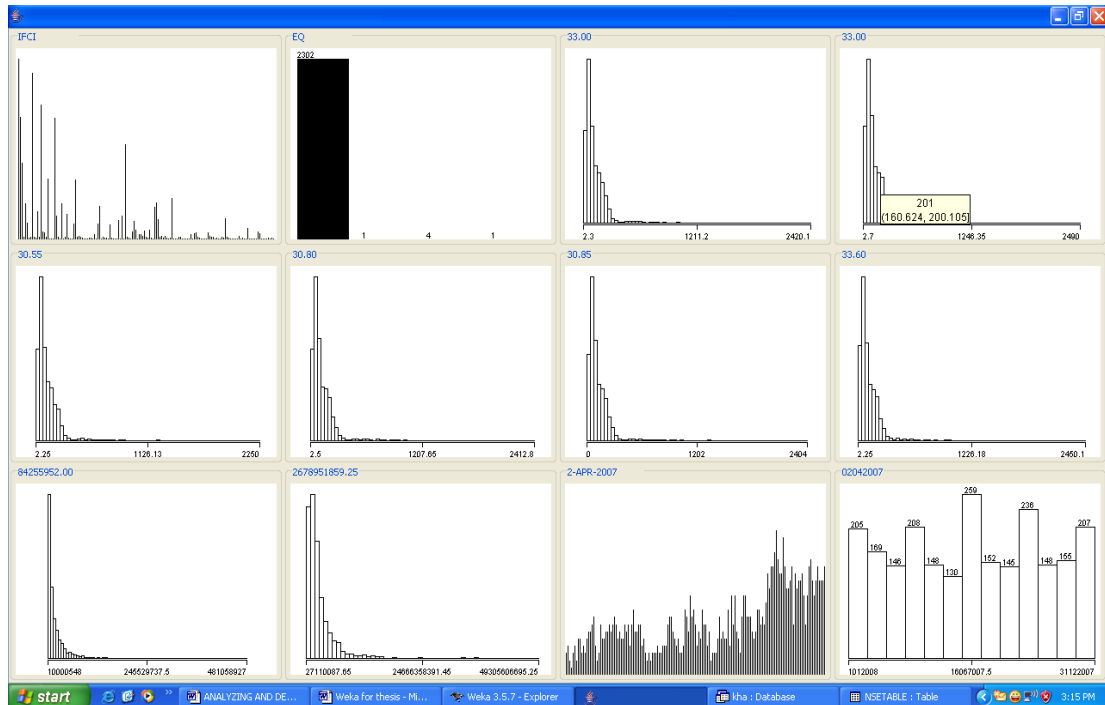


Fig 5.18: Consolidated Charts

Above figure shows the result of Data Visualization using the WEKA (an open source data mining tool developed using JAVA). For this example the data is taken from our data warehouse. We have not used all the data that we have extracted and transformed because we have experienced that Weka software does not work with very large data set. In our data warehouse we have total

279912 rows of stock market data that is data from 1st April 2007 to 16th May 2008. But we have used only those rows whose total trading volume is more than 100000000 shares (100 millions). Total number of rows meeting this criterion is 2308. It means to make 2308 records useful to Weka, it is converted into the ARFF format and then applied to Weka.

By visualizing the result one can easily see that there are 201 scripts whose high rate is ranging from Rs.160.624 to Rs.200.105. Taking individual look of every chart the data can be analyzed as follows.

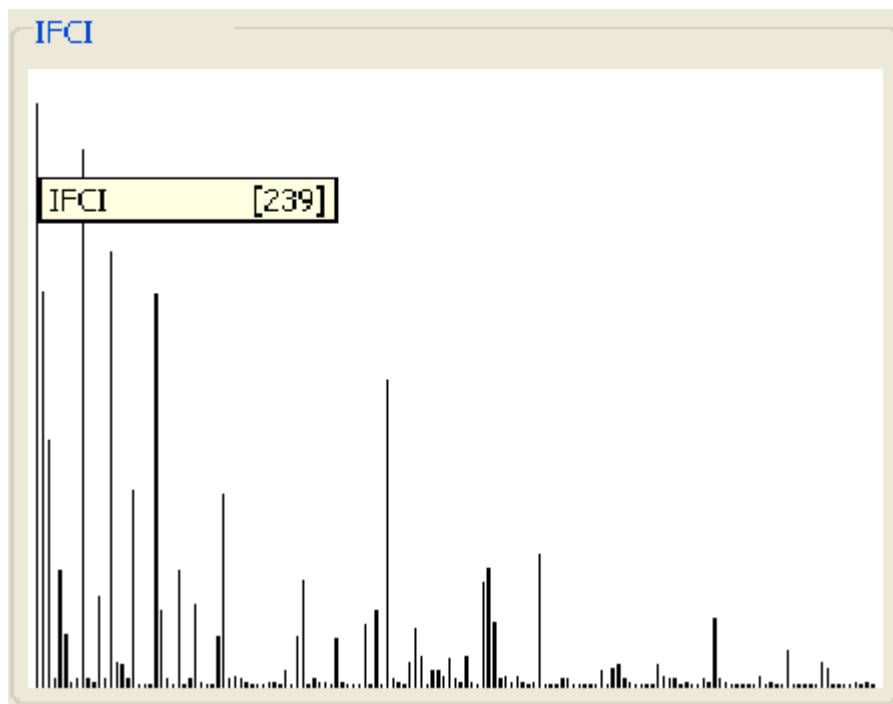


Fig 5.19: Chart of Scripts

Above figure visualizes that out of total 2308 rows, the script IFCI appears 239 times, so it can be said that the script IFCI has 239 times trading volume more than 10 million of shares during the period 1st April 2007 to 16th May 2008.

Above chart displays the number of occurrences for all the scripts whose trading volume is more than 10 million.

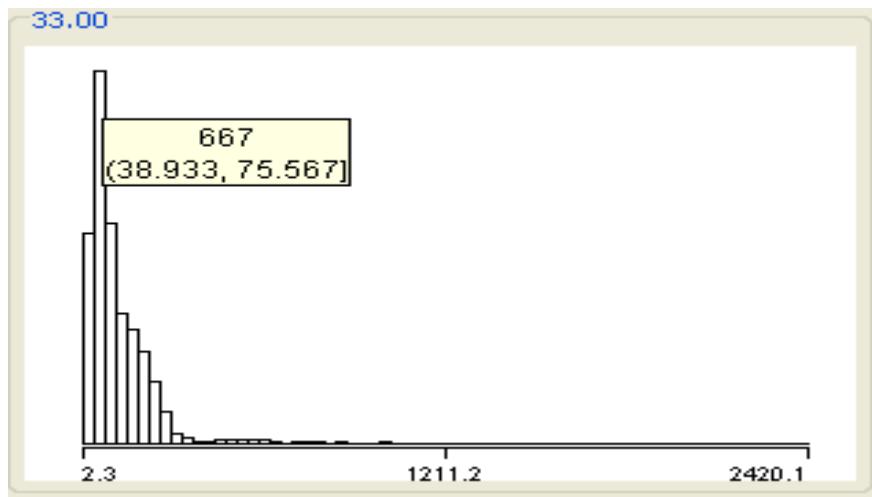


Fig 5.20: Chart of Open rate

Above figure visualize that there are 667 scripts whose trading volume more than 10 million of shares having open rate between 38.933 to 75.567 when share market opened during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 667.

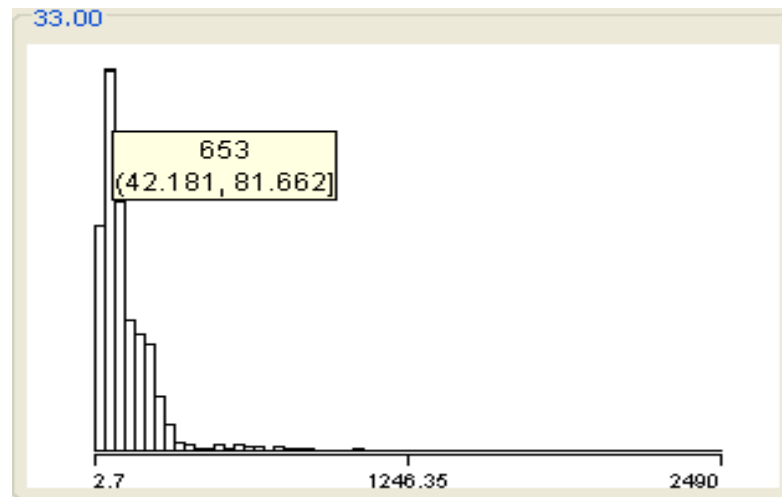


Fig 5.21: Chart of High rate

Above figure visualize that there are 653 scripts whose trading volume more than 10 million of shares having high rate between 42.181 to 81.662 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 653.

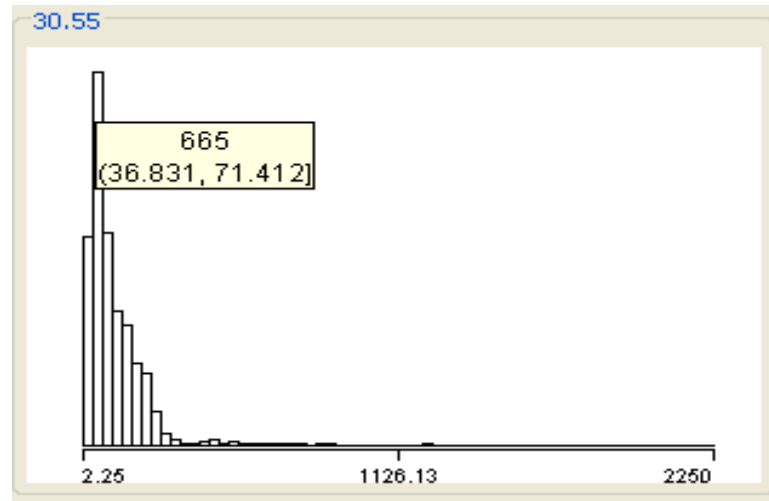


Fig 5.22: Chart of Low rate

Above figure visualize that there are 665 scripts whose trading volume more than 10 million of shares having low rate between 36.831 to 71.412 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 665.

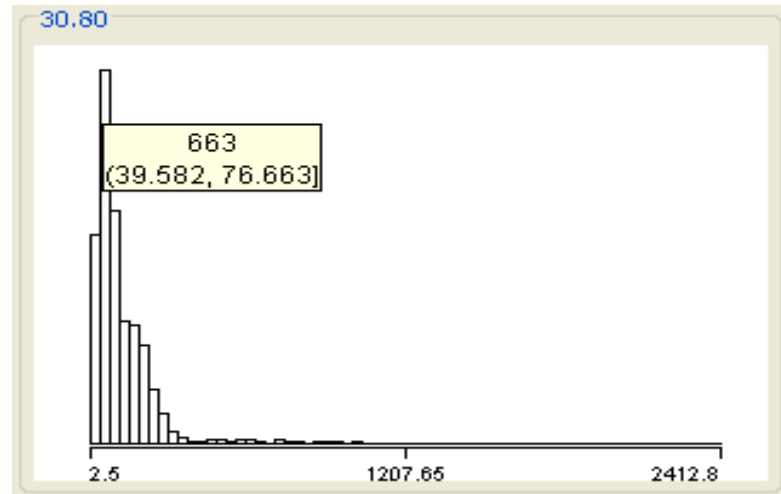


Fig 5.23: Chart of Close rate

Above figure visualize that there are 663 scripts whose trading volume more than 10 million of shares having closing rate between 39.582 to 76.663 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 663.

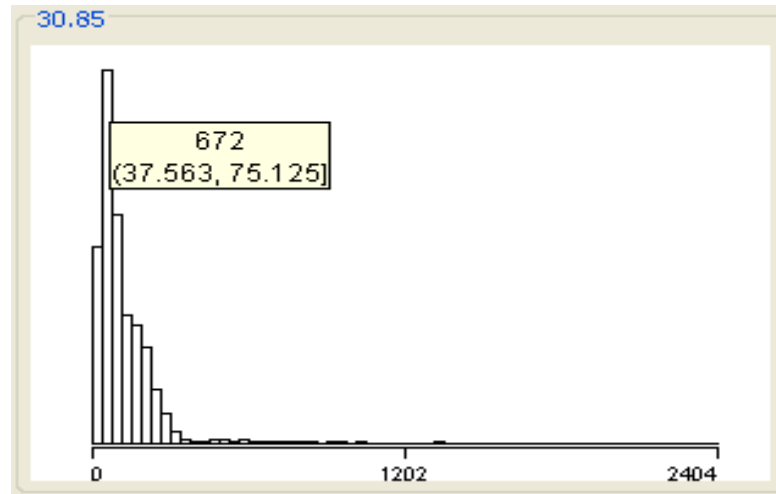


Fig 5.24: Chart of Last rate

Above figure visualize that there are 672 scripts whose trading volume more than 10 million of shares having last rate between 37.563 to 75.125 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 672.

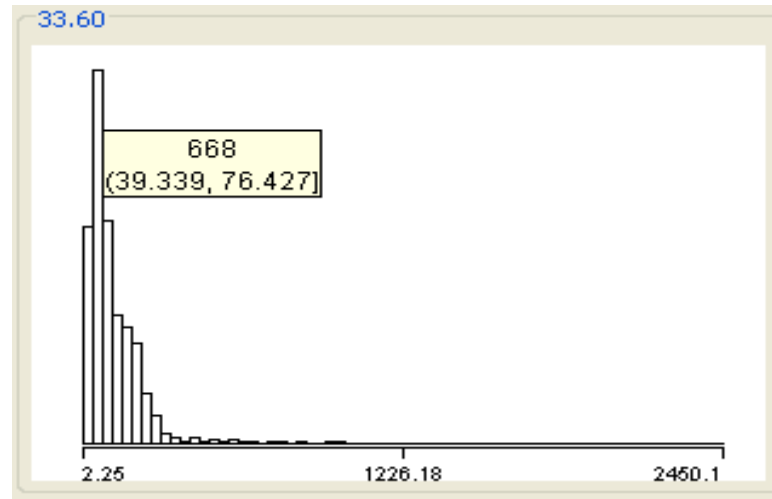


Fig 5.25: Chart of Previous Close rate

Above figure visualize that there are 668 scripts whose trading volume more than 10 million of shares having previous closing rate between 39.339 to 76.427 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 668.

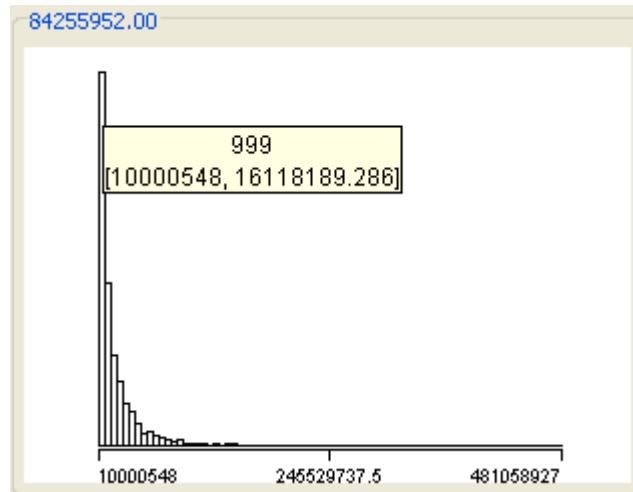


Fig 5.26: Chart of Volume

Above figure visualize that there are 999 scripts whose trading volume more than 10 million of shares having trading quantity between 10000548 to 16118189.286 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 999.

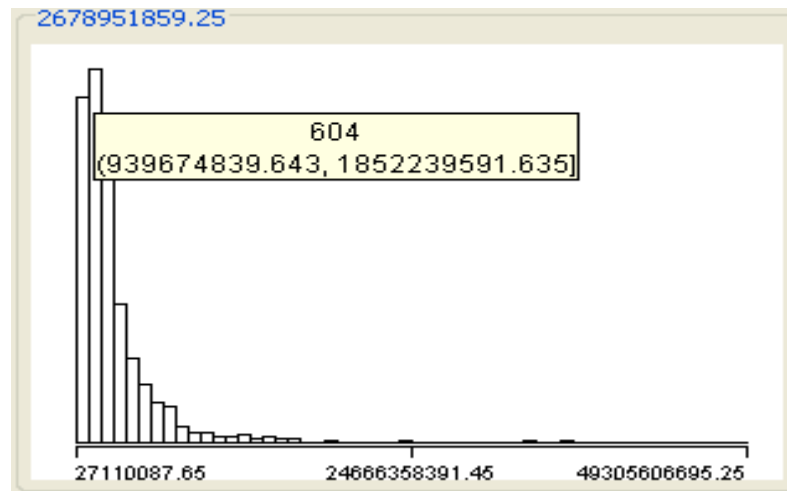


Fig 5.27: Chart of Trading Value

Above figure visualize that there are 604 scripts whose trading volume more than 10 million of shares having trading value between 939674839.643 to 1852239591.635 during the day, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 604.

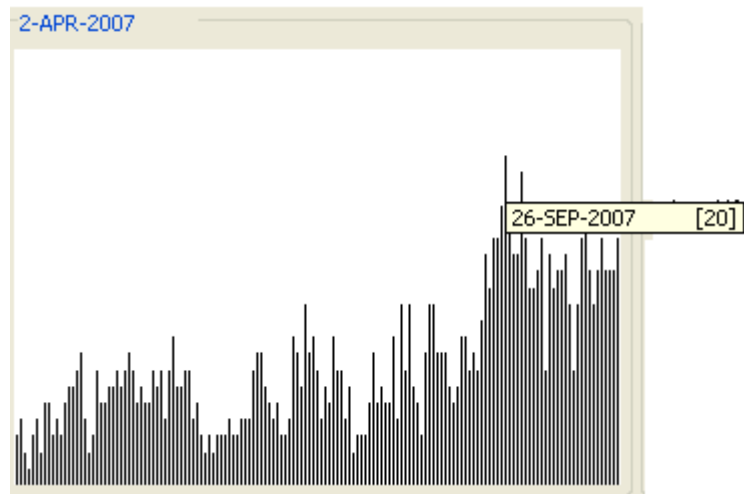


Fig 5.28: Chart of Number of Scripts

Above figure visualize that there are 20 scripts whose trading volume more than 10 million of shares on 26th September 2007, during the period 1st April 2007 to 16th May 2008. Here it is visualized that there are maximum number of scripts are 20.

From Fig 5.28 following data are found, from these data we found some interesting pattern.

Action	Qty	L o w e r Range	H i g h e r Range	Euclidean Distance
Open	667	38.933	75.567	36.634
High	653	42.181	81.662	39.481
Low	665	36.831	71.412	34.581
Close	663	39.582	76.663	37.081
Last	672	37.563	75.125	37.562
Previous Close	668	39.339	76.427	37.088

From above table following consolidated chart can be generated, it visualized the data graphically.

From above data and chart it can be concluded that for the period 1st April 2007 to 16th May 2008, quantity of scripts for the open rate, high rate, low rate, close rate, last rate and previous close rate does not fluctuated very much. The

variation between them is not remarkable; it is proved from Euclidean distance also.

5.5.2 Rules generated using chart

We can say for total trading quantity of any script is more than 10 million shares per day, maximum number of scripts for open rate, high rate, low rate, close rate, last rate and previous close rate is ranging from 653 to 672, lower range is ranging from 36.831 to 42.181, higher range is ranging from 71.412 to 81.662 and Euclidian distance is ranging from 34.581 to 39.481 also.

It can be concluded that for maximum number of scripts does not have major variations in their rate.

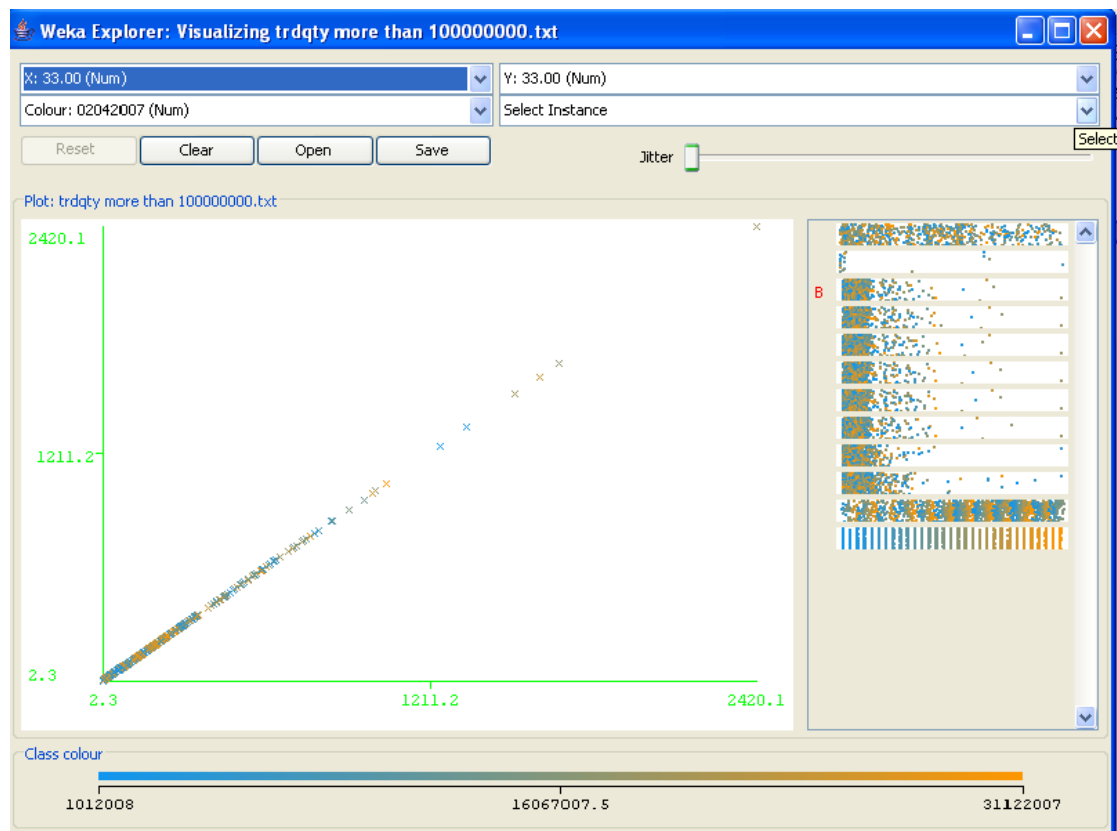


Fig 5.29: Chart of Open Rate and High Rate

Above figure shows the chart for Open rate and High Rate of scripts. On X- axis the data for Open rate is plotted and on Y-axis the data of High Rate is plotted. Below is the data, which is indicated one of the point of the chart. Using this instance information one can generate the rule too.

Plot: Master Plot

Instance: 192

SCRIPT: MIC

EQ: EQ

OPEN: 262.5

HIGH: 368.0

LOW: 253.0

CLOSE: 338.15

LAST: 337.3

PREVCLOSE: 150.0

TOTALTRQTY: 2.0186521E7

TOTALTRVALUE: 6.8326596157E9

TIMESTAMP: 30-MAY-2007

Plot: Master Plot

Instance: 1040

SCRIPT: SAIL

EQ: EQ

OPEN: 266.0

HIGH: 279.6

LOW: 2262.0

CLOSE: 3276.4

LAST: 275.35

PREVCLOSE: 1262.9

TOTALTRQTY: 1.7033345E7

TOTALTRVALUE: 4.6128225416E9

TIMESTAMP: 29-OCT-2007

Plot: Master Plot

Instance: 1067

SCRIPT: SAIL

EQ: EQ

OPEN: 261.0

HIGH: 267.4

LOW: 2256.6

CLOSE: 3261.0

LAST: 259.8

PREVCLOSE: 1261.0

TOTALTRQTY: 1.128556E7

TOTALTRVALUE: 2.95249406265E9

TIMESTAMP: 31-OCT-2007

Plot: Master Plot

Instance: 1081

SCRIPT: SAIL

EQ: EQ

OPEN: 264.5

HIGH: 272.0

LOW: 2 252.6

CLOSE: 3 258.3

LAST: 257.3

PREVCLOSE: 261.0

TOTALTRQTY: 1.5676043E7

TOTALTRVALUE: 4.1185041526E9

TIMESTAMP: 1-NOV-2007

Plot: Master Plot

Instance: 1107

SCRIPT: RPL

EQ: EQ

OPEN: 271.0

HIGH: 281.8

LOW: 264.05

CLOSE: 267.6

LAST: 267.65

PREVCLOSE: 269.6

TOTALTRQTY: 7.5749282E7

TOTALTRVALUE: 2.0718391704E10

TIMESTAMP: 5-NOV-2007

Plot: Master Plot

Instance: 1285

SCRIPT: SAIL

EQ: EQ

OPEN: 260.9

HIGH: 273.0

LOW: 257.0

CLOSE: 263.0

LAST: 263.0

PREVCLOSE: 262.3

TOTALTRQTY: 1.0622007E7

TOTALTRVALUE: 2.8327480084E9

TIMESTAMP: 20-NOV-2007

Plot: Master Plot

Instance: 1360

SCRIPT: SAIL

EQ: EQ

OPEN: 266.0

HIGH: 272.45

LOW: 261.1

CLOSE: 264.0

LAST: 264.25

PREVCLOSE: 265.25

TOTALTRQTY: 1.0220733E7

TOTALTRVALUE: 2.7319939647E9

TIMESTAMP: 27-NOV-2007

Plot: Master Plot

Instance: 1434

SCRIPT: SAIL

EQ: EQ

OPEN: 264.95

HIGH: 287.85

LOW: 262.5

CLOSE: 285.0

LAST: 284.15

PREVCLOSE: 262.7

TOTALTRQTY: 2.0417605E7

TOTALTRVALUE: 5.65558182045E9

TIMESTAMP: 4-DEC-2007

Plot: Master Plot

Instance: 1440

SCRIPT: GMRINFRA

EQ: EQ

OPEN: 261.25

HIGH: 266.3

LOW: 257.0

CLOSE: 263.1

LAST: 262.4

PREVCLOSE: 258.95

TOTALTRQTY: 1.0357085E7

TOTALTRVALUE: 2.71740211745E9

TIMESTAMP: 5-DEC-2007

Plot: Master Plot

Instance: 1463

SCRIPT: GMRINFRA

EQ: EQ

OPEN: 265.0

HIGH: 269.8

LOW: 254.65

CLOSE: 257.45

LAST: 257.0

PREVCLOSE: 263.1

TOTALTRQTY: 1.6716605E7

TOTALTRVALUE: 4.3268903113E9

TIMESTAMP: 6-DEC-2007

Plot: Master Plot

Instance: 1484

SCRIPT: GMRINFRA

EQ: EQ

OPEN: 259.75

HIGH: 260.0

LOW: 240.1

CLOSE: 243.8

LAST: 244.05

PREVCLOSE: 257.45

TOTALTRQTY: 1.7857966E7

TOTALTRVALUE: 4.3922063069E9

TIMESTAMP: 7-DEC-2007

Plot: Master Plot

Instance: 1526

SCRIPT: SAIL

EQ: EQ

OPEN: 271.0

HIGH: 281.5

LOW: 271.0

CLOSE: 277.2

LAST: 276.1

PREVCLOSE: 270.8

TOTALTRQTY: 1.0231161E7

TOTALTRVALUE: 2.8356424296E9

TIMESTAMP: 11-DEC-2007

Plot: Master Plot

Instance: 1575

SCRIPT: ESSAROIL

EQ: EQ

OPEN: 261.0

HIGH: 298.9

LOW: 255.25

CLOSE: 281.3

LAST: 277.0

PREVCLOSE: 255.1

TOTALTRQTY: 2.5977786E7

TOTALTRVALUE: 7.32023110305E9

TIMESTAMP: 18-DEC-2007

Plot: Master Plot

Instance: 1613

SCRIPT: SAIL

EQ: EQ

OPEN: 263.0

HIGH: 271.35

LOW: 255.75

CLOSE: 264.5

LAST: 263.5

PREVCLOSE: 259.05

TOTALTRQTY: 1.0123188E7

TOTALTRVALUE: 2.6609455879E9

TIMESTAMP: 20-DEC-2007

Plot: Master Plot

Instance: 1622

SCRIPT: SAIL

EQ: EQ

OPEN: 269.7

HIGH: 271.5

LOW: 265.0

CLOSE: 268.75

LAST: 267.85

PREVCLOSE: 264.5

TOTALTRQTY: 1.0885466E7

TOTALTRVALUE: 2.9257632661E9

TIMESTAMP: 24-DEC-2007

Plot: Master Plot

Instance: 1782

SCRIPT: NTPC

EQ: EQ

OPEN: 270.0

HIGH: 272.85

LOW: 259.35

CLOSE: 264.9

LAST: 267.0

PREVCLOSE: 268.8

TOTALTRQTY: 1.2215701E7

TOTALTRVALUE: 3.2303941998E9

TIMESTAMP: 8-JAN-2008

Plot: Master Plot

Instance: 1795

SCRIPT: NTPC

EQ: EQ

OPEN: 266.25

HIGH: 279.4

LOW: 262.1

CLOSE: 277.15

LAST: 276.0

PREVCLOSE: 264.9

TOTALTRQTY: 1.6240067E7

TOTALTRVALUE: 4.45392137295E9

TIMESTAMP: 9-JAN-2008

Plot: Master Plot

Instance: 1819

SCRIPT: NTPC

EQ: EQ

OPEN: 267.1

HIGH: 274.45

LOW: 262.1

CLOSE: 272.25

LAST: 274.45

PREVCLOSE: 265.85

TOTALTRQTY: 1.0923531E7

TOTALTRVALUE: 2.93383916765E9

TIMESTAMP: 11-JAN-2008

Plot: Master Plot

Instance: 1856

SCRIPT: NTPC

EQ: EQ

OPEN: 270.15

HIGH: 270.5

LOW: 252.45

CLOSE: 258.45

LAST: 259.35

PREVCLOSE: 274.0

TOTALTRQTY: 2.9114178E7

TOTALTRVALUE: 7.60495780775E9

TIMESTAMP: 16-JAN-2008

Plot: Master Plot

Instance: 1866

SCRIPT: NTPC

EQ: EQ

OPEN: 259.95

HIGH: 263.25

LOW: 253.25

CLOSE: 255.5

LAST: 255.0

PREVCLOSE: 258.45

TOTALTRQTY: 1.8489209E7

TOTALTRVALUE: 4.79902367045E9

TIMESTAMP: 17-JAN-2008

Plot: Master Plot

Instance: 1883

SCRIPT: ESSAROIL

EQ: EQ

OPEN: 271.0

HIGH: 271.0

LOW: 176.0

CLOSE: 184.7

LAST: 181.0

PREVCLOSE: 271.0

TOTALTRQTY: 1.4912146E7

TOTALTRVALUE: 3.1567433135E9

TIMESTAMP: 21-JAN-2008

Plot: Master Plot

Instance: 2076

SCRIPT: ESSAROIL

EQ: EQ

OPEN: 264.0

HIGH: 274.45

LOW: 261.05

CLOSE: 263.45

LAST: 263.25

PREVCLOSE: 261.9

TOTALTRQTY: 1.1843308E7

TOTALTRVALUE: 3.17086769455E9

TIMESTAMP: 16-APR-2008

Plot: Master Plot

Instance: 2189

SCRIPT: JPASSOCIAT

EQ: EQ

OPEN: 267.0

HIGH: 274.9

LOW: 267.0

CLOSE: 271.7

LAST: 270.15

PREVCLOSE: 266.0

TOTALTRQTY: 1.4977885E7

TOTALTRVALUE: 4.07125799615E9

TIMESTAMP: 30-APR-2008

Instance	Script	Date	Open	High	Euclidean Distance
192	MIC	30-May-07	362.50	368.00	5.50
1040	SAIL	29-Oct-07	266.00	279.60	13.60
1067	SAIL	31-Oct-07	261.00	267.40	6.40
1081	SAIL	1-Nov-07	264.50	272.00	7.50
1107	RPL	5-Nov-07	271.00	281.80	10.80
1285	SAIL	20-Nov-07	260.00	273.00	13.00
1360	SAIL	27-Nov-07	266.00	272.45	6.45
1434	SAIL	4-Dec-07	264.95	286.85	21.90
1440	GMRINFRA	5-Dec-07	261.25	266.30	5.05
1463	GMRINFRA	6-Dec-07	265.00	269.80	4.80
1484	GMRINFRA	7-Dec-07	259.75	260.00	0.25
1526	SAIL	11-Dec-07	271.00	281.50	10.50
1575	ESSAROIL	18-Dec-07	261.00	298.90	37.90
1613	SAIL	20-Dec-07	263.00	271.35	8.35
1622	SAIL	24-Dec-07	269.70	271.50	1.80
1782	NTPC	8-Jan-08	270.00	272.85	2.85
1795	NTPC	9-Jan-08	266.25	279.40	13.15
1819	NTPC	11-Jan-08	267.10	274.45	7.35
1856	NTPC	16-Jan-08	270.15	270.50	0.35
1866	NTPC	17-Jan-08	259.95	263.25	3.30
1883	ESSAROIL	21-Jan-08	271.00	271.00	0.00
2076	ESSAROIL	16-Apr-08	264.00	274.45	10.45
2189	JPASSOCIAT	30-Apr-08	267.00	274.90	7.90

From above data table it is shown that high rate of the script is always greater than open rate. We can also observe that the script appearing less number of times having higher Euclidean distance in most of cases, these scripts are highlighted.

Script	Number of Occurrence	Variance of Euclidean Distance	Variance of Open rate	Variance of High rate	Euclidean Distance of variance of Open rate and High rate
ESSAROIL	3	383.1858	26.33333	231.3525	205.0192
GMRINFRA	3	7.300833	7.3125	24.66333	17.35083
NTPC	5	25.07	17.17925	35.06425	17.885
SAIL	9	33.1959	13.14944	38.32757	25.17812

Here we can see that Euclidean Distance of variance of open rate and variance of high rate is increasing with the number of occurrence of script, except in the case of ESSAROIL. This is because of zero Euclidean distance of ESSAROIL on 21st Jan 2008. On this date this script may be suspended or there may be some other reason of this. So this script should be excluded from our data set.

Correlation coefficient between variance of open rate and variance of high rate is 0.796727; it means when variance of open rate is increasing; with it variance of high rate is also increasing and number of occurrences of scripts are also increasing.

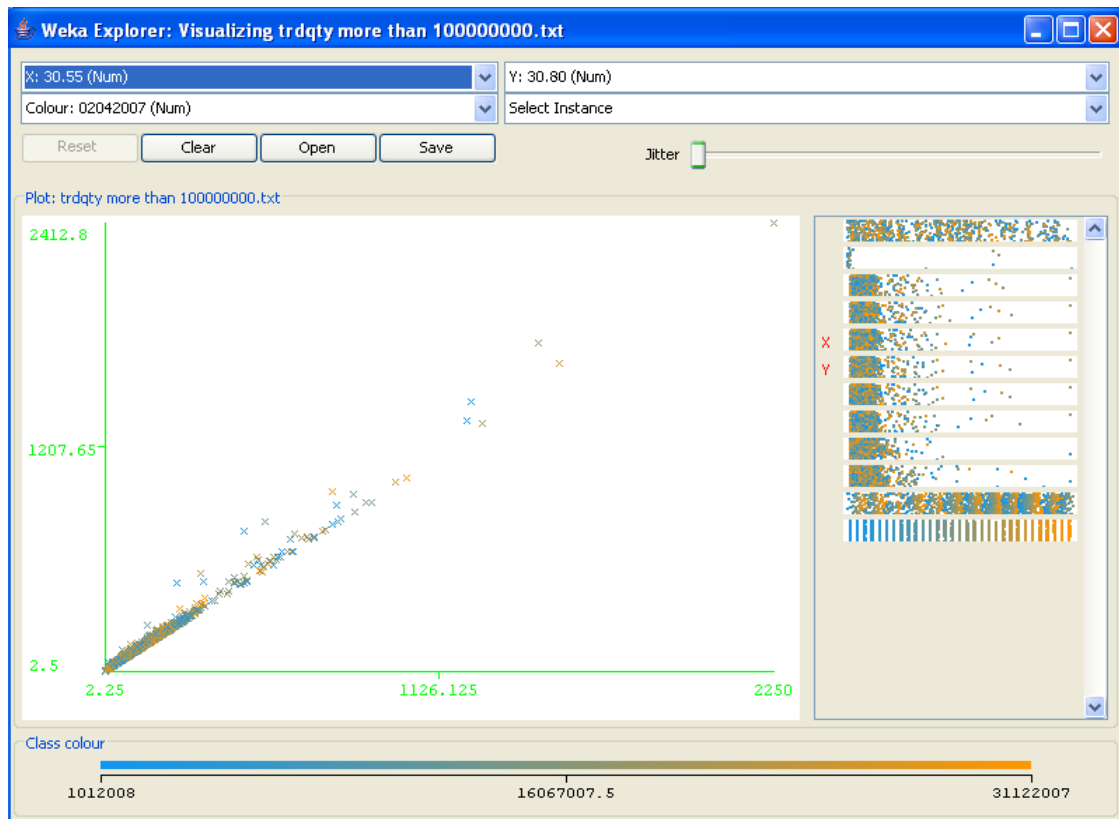


Fig 5.30: Chart of Low Rate and Close Rate

Above figure shows the chart for Low rate and Close Rate of scripts. On X- axis the data for Low rate is plotted and on Y-axis the data of Close Rate is plotted. Below is the data, which is indicated one of the point of the chart. Using this instance information one can generate the rule too.

Plot : Master Plot

Instance: 18

SCRIPT: ORBITCORP

EQ: EQ

OPEN: 113.0

HIGH: 137.6

LOW: 110.15

CLOSE: 128.2

LAST: 129.7

PREVCLOSE: 110.0

TOTALTRQTY: 1.4577737E7

TOTALTRVALUE: 1.8676312227E9

TIMESTAMP: 12-APR-2007

Plot : Master Plot

Instance: 646

SCRIPT: INDOWIND

EQ: Q

OPEN: 120.0

HIGH: 135.0

LOW: 120.0

CLOSE: 130.55

LAST: 131.5

PREVCLOSE: 113.65

TOTALTRQTY: 1.4840899E7

TOTALTRVALUE: 1.91967759895E9

TIMESTAMP: 17-SEP-2007

Plot : Master Plot

Instance: 673

SCRIPT: TRIVENI

EQ: EQ

OPEN: 111.0

HIGH: 142.2

LOW: 110.55

CLOSE: 136.15

LAST: 136.45

PREVCLOSE: 110.5

TOTALTRQTY: 1.9126004E7

TOTALTRVALUE: 2.4084266359E9

TIMESTAMP: 19-SEP-2007

Plot: Master Plot

Instance: 843

SCRIPT: TORNTPOWER

EQ: EQ

OPEN: 125.0

HIGH: 142.9

LOW: 113.2

CLOSE: 128.7

LAST: 125.5

PREVCLOSE: 121.85

TOTALTRQTY: 1.053272E7

TOTALTRVALUE: 1.3616478007E9

TIMESTAMP: 5-OCT-2007

Plot: Master Plot

Instance: 931

SCRIPT: IDBI

EQ: EQ

OPEN: 138.0

HIGH: 138.6

LOW: 118.0

CLOSE: 136.8

LAST: 136.0

PREVCLOSE: 141.85

TOTALTRQTY: 1.0390131E7

TOTALTRVALUE: 1.3876839619E9

TIMESTAMP: 17-OCT-2007

Plot : Master Plot

Instance: 937

SCRIPT: POWERGRID

EQ: EQ

OPEN: 114.0

HIGH: 140.0

LOW: 106.1

CLOSE: 136.8

LAST: 135.35

PREVCLOSE: 126.9

TOTALTRQTY: 1.15242925E8

TOTALTRVALUE: 1.47349284219E10

TIMESTAMP: 17-OCT-2007

Plot : Master Plot

Instance: 1213

SCRIPT: MRPL

EQ: EQ

OPEN: 107.0

HIGH: 131.6

LOW: 106.2

CLOSE: 128.05

LAST: 126.8

PREVCLOSE: 104.75

TOTALTRQTY: 4.9015305E7

TOTALTRVALUE: 6.03069952555E9

TIMESTAMP: 15-NOV-2007

Plot: Master Plot

Instance: 1236

SCRIPT: MRPL

EQ: EQ

OPEN: 120.0

HIGH: 148.7

LOW: 120.0

CLOSE: 130.4

LAST: 130.2

PREVCLOSE: 128.05

TOTALTRQTY: 4.4919996E7

TOTALTRVALUE: 6.1420333326E9

TIMESTAMP: 16-NOV-2007

Plot: Master Plot

Instance: 1901

SCRIPT: GMRINFRA

EQ: EQ

OPEN: 175.0

HIGH: 175.0

LOW: 111.1

CLOSE: 156.05

LAST: 154.0

PREVCLOSE: 169.65

TOTALTRQTY: 1.8947204E7

TOTALTRVALUE: 2.6731514413E9

TIMESTAMP: 22-JAN-2008

Plot: Master Plot

Instance: 1913

SCRIPT: RPL

EQ: EQ

OPEN: 165.0

HIGH: 165.0

LOW: 107.25

CLOSE: 147.0

LAST: 147.1

PREVCLOSE: 171.95

TOTALTRQTY: 4.494024E7

TOTALTRVALUE: 6.03048104445E9

TIMESTAMP: 22-JAN-2008

Following table shows the data of above instances generated by Weka.

SCRIPT	LOW	CLOSE	DATE	TRADING QTY
ORBITCORP	119.15	128.20	12-Apr-07	14577737.00
INDOWIND	120.00	130.55	17-Sep-07	14840899.00
TRIVENI	110.55	136.15	19-Sep-07	19126004.00
TORNTPOWER	113.20	128.70	5-Oct-07	10532720.00
IDBI	118.00	136.80	17-Oct-07	10390131.00
POWERGRID	106.20	136.80	17-Oct-07	115242925.00
MRPL	106.20	128.05	15-Nov-07	49015305.00
MRPL	120.00	130.40	16-Nov-07	44919996.00
GMRINFRA	111.10	156.05	22-Jan-08	18947204.00
RPL	107.25	147.00	22-Jan-08	44940240.00

We calculate Euclidean Distance between LOW and CLOSE rate in following table.

SCRIPT	LOW	CLOSE	TRADING QTY	Euclidean Distance of LOW and CLOSE
ORBITCORP	119.15	128.20	14577737.00	9.05
INDOWIND	120.00	130.55	14840899.00	10.55
TRIVENI	110.55	136.15	19126004.00	25.60
TORNTPOWER	113.20	128.70	10532720.00	15.50
IDBI	118.00	136.80	10390131.00	18.80
POWERGRID	106.20	136.80	115242925.00	30.60
MRPL	106.20	128.05	49015305.00	21.85
MRPL	120.00	130.40	44919996.00	10.40
GMRINFRA	111.10	156.05	18947204.00	44.95
RPL	107.25	147.00	44940240.00	39.75

Correlation Coefficient between Trading Quantity and Euclidean Distance between Low and Close is 0.292344. It means there is not much effect on fluctuation of Trading Quantity on Euclidean Distance between Low and Close. This can be said for different scripts for non-consecutive dates.

We can observe the data of 15th Nov. 2007 and 16th Nov. 2007 for the script MRPL. There is major difference in the Euclidean distance of both dates. When trading quantity is higher, Euclidean distance is also higher. So it can be concluded that when fluctuation in low and close rate is more, trading quantity is also increased. This can be said for single script for consecutive dates.

5.5.3 Analysis of Data using Weka

Scheme: weka.classifiers.rules.ZeroR

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

Test mode: 10-fold cross-validation

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

ZeroR predicts class value: 1.6170382486568458E7

Time taken to build model: 0.02 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	-0.0614
Mean absolute error	7627934.2008
Root mean squared error	8846610.1455
Relative absolute error	100 %
Root relative squared error	100 %
Total Number of Instances	2308

Scheme: weka.classifiers.rules.ZeroR

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: split 66% train, remainder test

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

ZeroR predicts class value: 1.6170382486568458E7

Time taken to build model: 0 seconds

=== Evaluation on test split ===

=== Summary ===

Correlation coefficient	0
Mean absolute error	7691482.3211

Root mean squared error	8906977.6853
Relative absolute error	100 %
Root relative squared error	100 %
Total Number of Instances	785

Scheme: weka.classifiers.rules.ZeroR

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

```

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

```

Test mode: Evaluate on training data

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

ZeroR predicts class value: 1.6170382486568458E7

Time taken to build model: 0 seconds

=== Evaluation on training set ===

=== Summary ===

Correlation coefficient	0
Mean absolute error	7623738.3568

Root mean squared error	8842698.919
Relative absolute error	100 %
Root relative squared error	100 %
Total Number of Instances	2308

Scheme: weka.classifiers.rules.DecisionTable-X1-S
"weka.attributeSelection.BestFirst -D 1 -N 5"

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: 10-fold cross-validation

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

Decision Table:

Number of training instances: 2308

Number of Rules: 241

Non matches covered by Majority class.

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 55

Merit of best subset found: 148379.077

Evaluation (for feature selection): CV (leave one out)

Feature set: 11,12

Time taken to build model: 4.97 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.9984
Mean absolute error	20223.2734
Root mean squared error	497863.4808
Relative absolute error	0.2651 %
Root relative squared error	5.6277 %
Total Number of Instances	2308

Scheme: weka.classifiers.rules.DecisionTable -X1 -S

"weka.attributeSelection.BestFirst -D 1 -N 5"

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: split 66% train, remainder test

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

Decision Table:

Number of training instances: 2308

Number of Rules: 241

Non matches covered by Majority class.

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 55

Merit of best subset found: 148379.077

Evaluation (for feature selection): CV (leave one out)

Feature set: 11,12

Time taken to build model: 4.33 seconds

=== Evaluation on test split ===

=== Summary ===

Correlation coefficient	0.9903
Mean absolute error	157944.0514
Root mean squared error	1240456.7238
Relative absolute error	2.0535 %
Root relative squared error	13.9268 %
Total Number of Instances	785

Scheme: weka.classifiers.rules.DecisionTable -X1 -S

"weka.attributeSelection.BestFirst -D 1 -N 5"

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: evaluate on training data

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

Decision Table:

Number of training instances: 2308

Number of Rules : 241

Non matches covered by Majority class.

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 55

Merit of best subset found: 148379.077

Evaluation (for feature selection): CV (leave one out)

Feature set: 11,12

Time taken to build model: 4.31 seconds

=== Evaluation on training set ===

=== Summary ===

Correlation coefficient	1
Mean absolute error	0
Root mean squared error	0
Relative absolute error	0 %
Root relative squared error	0 %
Total Number of Instances	2308

Evaluator: weka.attributeSelection.CfsSubsetEval

Search: weka.attributeSelection.BestFirst -D 1 -N 5

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

Evaluation mode: 10-fold cross-validation

=== Attribute selection 10 fold cross-validation seed: 1 ===

number of folds (%) attribute

0(0 %)	1	SCRIPT
8(80 %)	2	EQ
0(0 %)	3	OPEN
0(0 %)	4	HIGH
0(0 %)	5	LOW
0(0 %)	6	CLOSE
0(0 %)	7	LAST
0(0 %)	8	REVCLOSE
5(50 %)	9	TOTALTRQTY
10(100 %)	11	TIMESTAMP
10(100 %)	10	TOTALTRVALUE

Evaluator: weka.attributeSelection.CfsSubsetEval

Search: weka.attributeSelection.BestFirst -D 1 -N 5

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Evaluation mode: evaluate on all training data

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

Search Method:

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 56

Merit of best subset found: 0.063

Attribute Subset Evaluator (supervised, Class (numeric): 12 02042007):

CFS Subset Evaluator

Including locally predictive attributes

Selected attributes: 2,10,11 : 3

SERIES, TOTALTRQTY, TIMESTAMP

Scheme: weka.classifiers.trees.DecisionStump

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: evaluate on training data

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

Decision Stump

Classifications

TIMESTAMP = 2-JAN-2008 : 2012008.0

TIMESTAMP != 2-JAN-2008 : 1.6287905035823504E7

TIMESTAMP is missing : 1.6170382486568458E7

Time taken to build model: 0.05 seconds

=== Evaluation on training set ===

=== Summary ===

Correlation coefficient 0.1459

Mean absolute error 7508150.7542

Root mean squared error 8748108.1331

Relative absolute error 98.4838 %

Root relative squared error 98.9303 %

Total Number of Instances 2308

Scheme: weka.classifiers.trees.DecisionStump

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: 10-fold cross-validation

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

Decision Stump

Classifications

TIMESTAMP = 2-JAN-2008 : 2012008.0

TIMESTAMP != 2-JAN-2008 : 1.6287905035823504E7

TIMESTAMP is missing : 1.6170382486568458E7

Time taken to build model: 0.02 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient 0.08

Mean absolute error 7589317.3493

Root mean squared error 8814612.4836

Relative absolute error 99.4937 %

Root relative squared error 99.6383 %

Total Number of Instances 2308

Scheme: weka.classifiers.trees.DecisionStump

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT

EQ

OPEN

HIGH

LOW

CLOSE

LAST

PREVCLOSE

TOTALTRQTY

TOTALTRVALUE

TIMESTAMP

Test mode: split 50% train, remainder test

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

Decision Stump

Classifications

TIMESTAMP = 2-JAN-2008 : 2012008.0

TIMESTAMP != 2-JAN-2008 : 1.6287905035823504E7

TIMESTAMP is missing : 1.6170382486568458E7

Time taken to build model: 0.03 seconds

=== Evaluation on test split ===

Correlation coefficient 0.1114

Mean absolute error 7520473.843

Root mean squared error 8736409.9535

Relative absolute error 99.1484 %

Root relative squared error 99.3815 %

Total Number of Instances 1154

Scheme: weka.clusterers.SimpleKMeans -N 2 -S 10

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

Test mode: evaluate on training data

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

kMeans

Number of iterations: 5,

Within cluster sum of squared errors: 4285.924401509479

Cluster centroids:

Cluster 0

Mean/Mode:

Std Devs:

IFCI	N/A
EQ	N/A
118.16	124.8262
125.55	136.2908
114.0314	120.2781
120.8046	131.05
120.6893	131.6531
116.8906	124.5912
27083270.6135	25347721.5502
2786929905.418	3508408630.2691
15-NOV-2007	N/A
9712545.1422	6438585.8995

Cluster 1

Mean/Mode:

Std Devs:

RNRL

N/A

EQ

N/A

126.351

150.1867

133.2443

158.6329

120.0101

138.9289

126.8457

149.961

126.8116

149.701

125.2896

151.668

24550638.7622

21074331.8809

2812354196.4694

N/A

27-DEC-2007

6288118.0741

22020297.9009

Clustered Instances

0 1097 (48%)

1 1211 (52%)

Scheme: weka.classifiers.rules.M5Rules -M 4.0

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

Test mode: 10-fold cross-validation

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

M5 pruned model rules

(using smoothed linear models) :

Number of Rules : 1

Rule: 1

LOW =

0.5358 * SYMBOL	=IOB	,RENUKA
,AMBUJACEM	,POWERGRID	,PTC
,GAMMNINFRA	,SELMCL	,PFC
,KIRIDYES	,ORBITCORP	,YESBANK
,HCC	,MANAKSIA	,RPL
,IDFC	,ORIENTBANK	,SAIL
,NEYVELILIG	,HINDALCO	,MTNL
,KOLTEPATIL	,KSCL	,BAJAJHIND
		,RAJESHEXPO
		,HINDALCO
		,OMNITECH
		,ARIES
		,GTL
		,ITC
		,PRAJIND
		,SASKEN
		,AIRDECCAN

,CIPLA	,ESSAROIL	,HINDUNILVR	,NTPC	,CAIRN
,ORCHIDCHEM	,GMRINFRA	,SUZLON	,HCLTECH	
,JPASSOCIAT	,NICOLASPIR	,ROMAN	,ADSL	
,PURVA	,UNITECH	,MIC	,RELIGARE	,OMAXE
,IBREALEST	,EVERONN	,VRPRIME	,ZYLOG	
,NITINFIRE	,TIMETECHNO	,SATYAMCOMP	,INDIABULLS	
,VISHALRET	,MAYTASINFR	,TATASTEEL	,HDIL	
,DLF	,RCOM	,ICRA	,MUNDRAPORT	
,BHARTIARTL	,TCS	,REL	,HDFC	

- 66.6341 * SYMBOL =REL, HDFC

+ 0.7792 * OPEN

- 0.5548 * HIGH

+ 0.8092 * CLOSE

- 0.0279 * LAST

- 0.0294 * PREVCLOSE

- 0 * 84255952.00

- 0.8714 * TIMESTAMP =10-APR-2008 ,22-NOV-2007

,11-APR-2008 ,19-SEP-2007 ,12-MAY-2008 ,22-APR-2008

,8-OCT-2007 ,30-APR-2008 ,4-DEC-2007 ,1-JUN-2007

,1-JAN-2008 ,22-JAN-2008 ,16-AUG-2007 ,24-DEC-2007

,16-APR-2008 ,9-OCT-2007 ,31-DEC-2007 ,10-DEC-2007

,12-NOV-2007 ,6-JUN-2007 ,15-APR-2008 ,11-JAN-2008

,19-JUL-2007 ,5-OCT-2007 ,13-NOV-2007 ,26-NOV-2007

,23-NOV-2007 ,3-DEC-2007 ,21-NOV-2007 ,28-NOV-2007

,9-JUL-2007 ,8-APR-2008 ,29-OCT-2007 ,7-DEC-2007

,30-NOV-2007 ,10-OCT-2007 ,12-OCT-2007 ,30-AUG-2007

,12-DEC-2007 ,18-DEC-2007 ,27-JUL-2007 ,29-NOV-2007

,29-APR-2008 ,13-AUG-2007 ,4-JUL-2007 ,14-JAN-2008

,14-NOV-2007 ,3-APR-2008 ,3-JAN-2008 ,30-JAN-2008

,5-DEC-2007 ,25-JUL-2007 ,7-JAN-2008 ,17-JUL-2007

,21-JAN-2008 ,4-JAN-2008 ,18-JUL-2007 ,26-DEC-2007

,18-OCT-2007 ,24-JUL-2007 ,17-AUG-2007 ,18-JAN-2008

,6-JUL-2007	,16-JAN-2008	,13-JUN-2007	,8-JAN-2008
,16-OCT-2007	,27-NOV-2007	,23-OCT-2007	,15-MAY-2008
,13-JUL-2007	,24-APR-2007	,18-APR-2007	,5-JUL-2007
,15-JAN-2008	,1-NOV-2007	,31-OCT-2007	,7-MAY-2008
,25-OCT-2007	,10-APR-2007	,1-OCT-2007	,24-OCT-2007
,3-OCT-2007	,12-JUL-2007	,17-OCT-2007	,13-APR-2007
,17-APR-2007	,19-OCT-2007		

- 0 * 02042007

+ 1.5581 [2308/4.504%]

Time taken to build model: 13.77 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.9988
Mean absolute error	2.6281
Root mean squared error	6.5352
Relative absolute error	3.4221 %
Root relative squared error	5.0057 %
Total Number of Instances	2308

Scheme: weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

Test mode: evaluate on training data

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

Single conjunctive rule learner:

(OPEN <= 424.2) => LOW = 102.22239

Time taken to build model: 0.5 seconds

=== Evaluation on training set ===

=== Summary ===

Correlation coefficient	0.7428
Mean absolute error	62.2453
Root mean squared error	87.3115
Relative absolute error	81.1583 %
Root relative squared error	66.9694 %
Total Number of Instances	2308

Scheme: weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1

Relation: trdqty more than 100000000.txt

Instances: 2308

Attributes: 12

SCRIPT
EQ
OPEN
HIGH
LOW
CLOSE
LAST
PREVCLOSE
TOTALTRQTY
TOTALTRVALUE
TIMESTAMP

Test mode: split 50% train, remainder test

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

Single conjunctive rule learner:

(OPEN <= 424.2) => LOW = 102.22239

Time taken to build model: 0.59 seconds

=== Evaluation on test split ===

=== Summary ===

Correlation coefficient	0.7296
Mean absolute error	62.834
Root mean squared error	81.7491
Relative absolute error	82.409 %
Root relative squared error	68.9327 %
Total Number of Instances	1154

Decision Tree using Analysis services

A decision tree is a form of classification shown in a tree structure, in which a node in the tree structure represents each question used to further classify data.

The various methods used to create decision trees have been used widely for decades, and there is a large body of work describing these statistical techniques.

The algorithm builds a tree that will predict the value of a column based upon the remaining columns in the training set. Therefore, each node in the tree represents a particular case for a column. The decision on where to place this node is made by the algorithm, and a node at a different depth than its siblings may represent different cases of each column.

Data Cluster Using the Analysis services

Like decision trees, clustering is a well-documented data mining technique. Clustering is the classification of data into groups based on specific criteria. The topic discussing the Microsoft Clustering algorithm goes into greater detail regarding the details of clustering as a data mining technique. The Clustering algorithm is an expectation method that uses iterative refinement techniques to group records into neighborhoods (clusters) that exhibit similar, predictable characteristics. Often, these characteristics may be hidden or non-intuitive.

Analyzing the Visual Result

As shown in figures, by seeing you can understand the patterns in the data and also you can compare one subject marks with other subject marks in a graphical way and find the particular point (instance) information.

Weka Classifier Result

Here researcher has applied the different data-mining algorithm for classification using the different test mode.

The test modes are

- **Using training set:** The classifier is evaluated on how well it predicts the class of the instances it was trained on.
- **Supplied test set:** The classifier is evaluated on how well it predicts the class of a set of instances loaded from a file.
- **Cross-validation:** The classifier is evaluated by cross-validation, using the number of folds.
- **Percentage split:** The classifier is evaluated on how well it predicts a certain percentage of the data, which is held out for testing.

The result table shows the summary, a list of statistics summarizing how accurately the classifier was able to predict the true class of the instances under the chosen test mode.

Correlation coefficient shows the strength of relationship between variables. So if it is high then it will give a good result.

Absolute error shows difference between a measurement and its true value. So if this value is low, that indicate good algorithm. The low value of root mean square error also indicates the good result.

From the following table we can select the best algorithm based on the given parameter.

5.5.4 Comparison of Various Algorithms

Algorithm	Test Mode	Correlation Coefficient	Mean Absolute Error	Root Mean Square Error	Relative Absolute Error	Root Relative Square Error	Number of Instances
Zero R	10-fold cross-validation	-0.0614	7627934	8846610.1	100%	100%	2308
Zero R	split 66% train, remainder test	0	7691482	8906977.7	100%	100%	785
Zero R	Evaluate on training data	0	7623738	8842698.9	100%	100%	2308
Decision Tree -X1-S	10-fold cross-validation	0.9984	20223.27	497863.48	0.27%	5.63%	2308
Decision Tree -X16-S	split 66% train, remainder test	0.9903	157944.1	1240456.7	2.05%	13.93%	785
Decision Tree -X1-S	Evaluate on training	1	0	0	0%	0%	2308

	data						
DecisionStump	Evaluate on training data	0.1459	7508151	8748108.1	98.48%	98.93%	2308
DecisionStump	10-fold cross-validation	0.08	7589317	8814612.5	99.49%	99.64%	2308
DecisionStump	split 50 % train, remainder test	0.1114	7520474	8736410	99.15%	99.38%	1154
M5 Rules-M 4.0	10-fold cross-validation	0.9988	2.6281	6.5352	3.42%	5.01%	2308
ConjunctiveRule -N 3e -M 2.0 -P -1 -S 1	Evaluate on training data	0.7428	62.2453	87.3115	81.16%	66.97%	2308
ConjunctiveRule -N 35 -M 2.0 -P -1 -S 2	split 50 % train, remainder test	0.7296	62.834	81.7491	82.41%	68.93%	1154

Fig 5.31: Comparison of Algorithms

5.6 Data Mining and Data Warehousing Tools

5.6.1 SQL Server 2005 Data Mining Features

Analysis Services

Analysis Services delivers extensions to the scalability, manageability, reliability, availability, and programmability of data warehousing, business intelligence, and line-of-business solutions.

Data Transformation Services (DTS)

A complete redesign of the DTS architecture and tools provides developers and database administrators with increased flexibility and manageability.

Reporting Services

Reporting Services is a new report server and tool set for building, managing, and deploying enterprise reports.

Data Mining

Data mining is enhanced with four new algorithms as well as improved data modeling and manipulation tools.

Create an easy-to-use, extensible, accessible, and flexible business intelligence platform and take the next step in business intelligence with SQL Server data-mining capabilities. Explore your data, discover patterns, and uncover business data to reveal the hidden trends about your products, customer, market, and employees, and better analyze those components that are critical to your organization's success.

Integration

SQL Server Data Mining is part of a family of business intelligence technologies that can be used together to enhance and develop a new breed of intelligent applications. These technologies include the following:

- **SQL Server 2005 Integration Services.** Create a more powerful data pipeline by working with SQL Server 2005 Integration Services, allowing your organization to flag outliers, separate data, and fill in missing values based on the predictive analytics of the data-mining algorithms.
- **SQL Server 2005 Analysis Services.** Create a richer Unified Dimensional Model by adding data-mining dimensions that slice your data by the hidden patterns within.
- **SQL Server Reporting Services.** Create smarter, insightful reports based on data-mining queries that present the right information to the right audiences.

Architecture

Providing data mining to organizations of any size introduces new challenges. Deployment, scalability, manageability, and security all become important factors. SQL Server Data Mining is part of the SQL Server Analysis Services server that provides all the enterprise-class server features you would expect:

- **Deployment.** SQL Server Data Mining is based on client- server architecture, allowing you to access models from your local area network (LAN), wide area network (WAN), or the Internet. Standard application programming interfaces (APIs) provide access to your models regardless of location or client platform.
- **Scalability.** SQL Server Data Mining is designed from the ground up with a parallel architecture to scale to enterprise-class data sets and thousands of concurrent users, and can respond to millions of queries per day.
- **Manageability.** SQL Server Data Mining is integrated into the new SQL Server Management Studio, providing a one-stop tool for managing all your SQL Server family properties.
- **Security.** SQL Server Data Mining provides fine-grained, role- based security to ensure that your intellectual property will be further protected.

Extensibility

SQL Server Data Mining is fully extensible through Microsoft .NET– stored procedures and plug-in algorithms and viewers that embed seamlessly to take advantage of all the platform abilities and integration. Adopting SQL Server Data Mining as your platform means that you will never be limited by the inherent functionality of your data-mining system because it can always be extended to meet your needs.

5.6.2 DB2 Intelligent Miner

IBM's data mining capabilities help you detect fraud, segment your customers, and simplify market basket analysis. IBM's in-database mining capabilities integrate with your existing systems to provide scalable, high performing predictive analysis without moving your data into proprietary data mining platforms. Use SQL, Web Services, or Java to access DB2's data mining capabilities from your own applications or business intelligence tools from IBM's business partners.

Tools

- **DB2 Intelligent Miner Modeling**

Delivers DB2 Extenders for modeling operations

- **DB2 Intelligent Miner for Scoring**

Provides scoring technology as database extensions: DB2 Extenders and Oracle cartridges

- **DB2 Intelligent Miner Visualization**

Provides Java visualizes to Interact and graphically present the results of associations

- **DB2 Intelligent Miner for Data**

Provides new business insights and harvests valuable business intelligence from your enterprise data.

Tool Benefits

- **Using PMML**

Share data mining data using the industry standard PMML

- **Scalable Mining**

Intelligent Miner tools integrate seamlessly with DB2 UDB

5.6.3 Informatica PowerCenter Advanced Edition

One of the biggest challenges facing organizations today is the fragmentation of data across disparate IT systems. Unlocking and deriving business value from these strategic data assets — no matter where they reside — has become a top priority.

Companies are realizing that in order to support their business objectives—such as providing a single view of the customer, migrating away from legacy systems to new technology, or consolidating multiple instances of an ERP system—they must be able to effectively integrate, move and access their data, or its business value will be lost.

Informatica PowerCenter Advanced Edition, the leading independent data integration platform, addresses this need—delivering the industry's most comprehensive set of capabilities for enterprise-wide data integration. PowerCenter Advanced Edition supports the entire data integration lifecycle—allowing companies to access, discover, and integrate data from the widest variety of enterprise systems and deliver that data to other operational systems, applications, databases and to the business user for decision making.

5.6.4 Business Object

BusinessObjects XI Release 2 provides performance management, reporting, query and analysis, and data integration in one solution.

- Performance management - Match actions with strategy.
- Reporting - Access, format, and deliver data.
- Query and analysis - Self-serve analysis for users.
- BI platform - Manage BI tools, reports, and applications.
- Data integration - Access, transform, and integrate data.

5.6.5 Cognos 8 Business Intelligence

Cognos 8 Business Intelligence is the only BI product to deliver the complete range of BI capabilities: reporting, analysis, Scorecarding, dashboards, business event management as well as data integration, on a single, proven architecture.

Easy to integrate, deploy and use, Cognos 8 BI delivers a simplified BI environment that improves user adoption, enables better decision- making, and serves as an enterprise-scale foundation for performance management.

Reporting

Reporting is a key capability within Cognos 8 Business Intelligence, a single product that provides complete BI capabilities on a proven architecture.

Reporting gives you access to a complete list of self-serve report types, is adaptable to any data source, and operates from a single metadata layer for a variety of benefits such as multilingual reporting.

Analysis

Analysis is a key capability within Cognos 8 Business Intelligence, a single product that provides complete BI capabilities on a proven architecture.

Analysis enables the guided exploration and analysis of information that pertains to all dimensions of your business-regardless of where the data is stored. Analyze and report against online analytical processing (OLAP) and dimensionally aware relational sources.

Scorecarding

Scorecarding is a key capability within Cognos 8 Business Intelligence, a single product that provides complete BI capabilities on a proven architecture. Scorecarding helps you align your teams and tactics with strategy; communicate goals consistently, and monitor performance against targets.

Dashboards

Business dashboards communicate complex information quickly. They translate information from your various corporate systems and data into visually rich presentations using gauges, maps, charts, and other graphical elements to show multiple results together.

Business Event Management

Cognos 8 BI business event management tracks significant events that need attention. It monitors these events and uses decision- process and business-process automation to compress the time to action and resolution.

Data Integration

Data integration is a key component within Cognos 8 Business Intelligence, a single product that provides complete BI capabilities on a proven architecture.

Cognos data integration is an enterprise-wide ETL solution designed for high performance business intelligence. It optimizes data merging, extraction, transformation, and dimensional management to deliver data warehouses ready for business reporting and analysis.

5.6.6 Comparison of Data Mining Tools

Elder Research Inc. is a company engaged with Data Mining and pattern discovery. This group has evaluated and compared various data mining tools.

Tools Evaluated

Product	Company	URL	Version Tested	Our Experience
<i>Clementine</i>	Integral Solutions, Ltd.	http://www.isl.co.uk/clem.html	4	Moderate
<i>Darwin</i>	Thinking Machines, Corp.	http://www.think.com/html/products/products.htm	3.0.1	Moderate
<i>DataCruncher</i>	DataMind	http://www.datamindcorp.com	2.1.1	High
<i>Enterprise Miner</i>	SAS Institute	http://www.sas.com/software/components/miner.html	Beta	Moderate
<i>GainSmarts</i>	Urban Science	http://www.urbanscience.com/main/gainpage.htm	4.0.3	Low
<i>Intelligent Miner</i>	IBM	http://www.software.ibm.com/data/iminer/	2	Low
<i>MineSet</i>	Silicon Graphics, Inc.	http://www.sgi.com/Products/software/MineSet/	2.5	Low
<i>Model 1</i>	Group 1/Unica Technologies	http://www.unica-usa.com/modell.htm	3.1	Moderate
<i>ModelQuest</i>	AbTech Corp.	http://www.abtech.com	1	Moderate
<i>PRW</i>	Unica Technologies, Inc.	http://www.unica-usa.com/prodinfo.htm	2.1	High
<i>CART</i>	Salford Systems	http://www.salford-systems.com	3.5	Moderate
<i>NeuroShell</i>	Ward Systems Group, Inc.	http://www.wardsystems.com/neuroshe.htm	3	Moderate
<i>OLPARS</i>	PAR Government Systems	mailto://olpars@partech.com	8.1	High
<i>Scenario</i>	Cognos	http://www.cognos.com/busintell/products/index.html	2	Moderate
<i>See5</i>	RuleQuest Research	http://www.rulequest.com/see5-info.html	1.07	Moderate
<i>S-Plus</i>	MathSoft	http://www.mathsoft.com/splus/	4	High
<i>WizWhy</i>	WizSoft	http://www.wizsoft.com/why.html	1.1	Moderate

Fig 5.32: Comparison of Various Tools

6. Summary, Conclusion and Future Work

6.1 Summary of work

My research is focused on comparison of algorithms of statistical methods for data mining and computational modeling. There is extensive use of statistics in data mining. So first I studied the classification of the data, there are two types of variables that are qualitative and quantitative, then I took overview of data warehouse, data webhouse and data mart. KDD (Knowledge Discovery in Databases) is process of extracting basics of data mining process to extract the data from very large database. I also studied about the reasons for growth of data mining research, there is requirement of applied, multidisciplinary and interdisciplinary research in data mining and knowledge discovery.

Chapter 1:

There is an operational and informational system, it is very much interesting to understand difference between them. The data warehouse is an informational environment; it is Subject-Oriented, Integrated, Non-Volatile, and Time variant collection of data. The architecture of data warehouse may be two-tiered or three-tiered architecture, it is depending on application. Three-tiered architecture is complete centralized data warehouse, while to build Two-Tiered architecture is to build the data marts without building the centralized data warehouse. But these data marts do not depend on the existence of a consolidated data warehouse, so it can be referred as independent data mart. Both types of architectures have their own advantages and disadvantages. The data warehouse stores its information in a form that is called application generic. A data mart is a powerful and natural extension of a data warehouse to a specific functional usage. The detailed data is found in the enterprise data warehouse, while very little detailed data is found in the data mart, because enterprise data warehouse is the source of data inside the data mart.

Chapter 2:

OLAP is Online Analytical Processing. There are mainly two different types: Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). Hybrid OLAP (HOLAP) refers to technologies that combine MOLAP and ROLAP. Both the types of OLAP have their own advantages and disadvantages.

ETL means Extraction, Transformation and Loading of the data. A recent development in ETL software is the implementation of parallel processing. This has enabled a number of methods to improve overall performance of ETL processes when dealing with large volumes of data. There are three main types of parallelisms as implemented in ETL applications: Data, Pipeline and Component.

Metadata (data about data) describes the details about the data in a data warehouse or in a data mart. The metadata of a data mart is created and updated from the load programs that move data from data warehouse to data mart. The linkages and relationships between metadata of data warehouse and metadata of data mart have to be well established or well understood by the analyst using the metadata. There are two basic architectures for metadata in the DSS data warehouse environment. Those architectures are a centralized architecture and a distributed architecture.

Chapter 3:

Data mining techniques are the result of a long process of research and product development. Data mining derives its name from the similarities between searching for valuable business information in a large database. Large amount of data generated by organizations worldwide is mostly unorganized. If data is organized one can generate/extract meaningful and useful information to convert unorganized data into organized data. Data mining is the technique of abstracting meaningful information from large and unorganized databanks. It involves the process of performing automated abstraction and generating predictive information from large databanks. The abstraction of meaningful large databanks can also be known as knowledge discovery. The data mining process uses a variety of analysis tools to determine the relationship between

data and the databank and to use the same to make valid prediction. Data mining techniques are a result of integration of various techniques forms multiple disciplines such as statistic, machine learning, pattern recognition, neural networks, image processing, etc.

Data mining is an iterative process that typically involves the Problem Definition, Data Understanding, Data Preparation, Creation of database for data mining, Exploring the database, Preparation for creating a data mining model, Building a data mining model, Evaluation of data mining model and Deployment of the data mining model.

There are many different techniques used to perform data mining tasks. These techniques not only require specific types of data structure, but also imply certain types of algorithmic approaches.

1. Statistics: Many statistical concepts that are the basis for data mining techniques, these are Point Estimation, Model Based Summarization, Bayes Theorem, Hypothesis Testing, Regression and Correlation,
2. Machine Learning: The concept of machine learning is implemented by way of computing software system that act as human being who learns from experience, analyses the observations made and self-improves providing increased efficiency and effectiveness. *Machine learning* is the area of Artificial Intelligence (AI) that examines how to write programs that can learn. In data mining, machine learning is often used for prediction or classification. When machine learning is applied to data mining tasks, a model is used to represent the data.
3. Decision Trees: Decision trees are one of the most popular methods of predictive modeling for data mining purposes because they provide interpretable rules and logic statements that enable more intelligent decision-making. Decision tree is a tree-shaped structure, which represents a predictive model used in classification, clustering, and prediction tasks. In this topic decision tree is represented with diagram. Solution of problem of decision tree is also explained with data set.

4. **Neural Networks:** The neural networks approach, like decision trees, requires that a graphical structure be built to represent the model and then that the structure be applied to the data. The neural networks can be viewed as a directed graph with source (*input*), sink (*output*), and internal (*hidden*) nodes. The input nodes exist in an *input layer*, while the output nodes exist in an *output layer*. The hidden nodes exist over one or more *hidden layer*. A Neural network (NN) model is a computational model consisting of three parts: (a) Neural network graph (b) Learning algorithm (c) Recall techniques

5. **Genetic Algorithm:** When using genetic algorithms to solve a problem, the first thing, and perhaps the most difficult task, that must be determined is how to model the problem as a set of individuals. A genetic algorithm (GA) is a computational model consisting of parts: (a) Starting set of individuals (b) Crossover technique (c) Mutation algorithm (d) Fitness function

6. **Association Rules:** Association rules provide information of this type in the form of "if-then" statements. These rules are computed from the data and, unlike the if-then rules of logic, association rules are probabilistic in nature. Basic algorithms for association rules are (a) Apriori Algorithm (b) Sampling Algorithm (c) Partitioning (d) Pincer-Search Algorithm (e) FP-Tree Growth Algorithm
 There are also advanced association rules techniques like (a) Generalized Association Rules (b) Multiple-Level Association Rules.

7. **Clustering:** Clustering is similar to classification in that data are grouped. However, unlike classification, the groups are not predefined. Instead, the grouping is accomplished by finding similarities between data according to characteristics found in the actual data. The groups are called clusters, some researchers view clustering as a special type of classification. In this thesis we followed a more conventional view in that the two are different. We studied many definitions for cluster. Most popular clustering algorithms are (a) Hierarchical Algorithms (b) Agglomerative Algorithm

- (c) Single-Linkage Agglomerative Algorithm (d) Complete-Linkage Agglomerative Algorithm (e) Average-Linkage Agglomerative Algorithm
- (f) Divisive Clustering (g) Partitional Algorithm (h) *K* – Means Clustering
- (i) Nearest Neighbor Algorithm

BIRCHES, DBSCAN, CURE are the algorithms for clustering of large database.

Chapter 4:

This web interface constructs the bridges between Oracle server and workstation on which one wants to use oracle database. It makes easy to access Oracle database without much prior knowledge of Oracle.

Architecture of web based data access interface is explained. Architecture of this tool includes Web Browser, IIS web server, HTML documents, CSS, JavaScript, AJAX, ASP and Oracle database.

This software also provides web session security. In which if any user is idle for 30 seconds, then automatically session is destroyed, and then user have to log in again. This way this tool provides database security and web security.

This tool provides also DBA level functionality like create profile, role, new users, etc at any client machine. For this it is not compulsory for DBA to log in at server machine.

Chapter 5:

ETL is the main concept in data mining. In chapter 2, ETL is described theoretically, In this chapter ETL tools are developed and implemented practically.

Data Extraction tool is developed to extract the data having different format from different environment. Model of Data Extraction tool is developed and it is implemented to extract the various data from NSE server. This tool has its own advantages and disadvantages.

After extracting the data it is very much necessary to transform, we developed a model for Data Transformation and practically it is implemented to transform the extracted data in our database in our desired format. This tool also indicates the cleansing of the data if necessary.

The charting tool performs loading of the data. This tool loads the data from our database and prepares the various charts. From the chart we have extracted some patterns. Using charts we also applied statistical techniques to understand the nature of historical data, then the data is analyzed.

In this thesis we have used two freeware software for analytical study of the historical data. One is Analysis Service of Microsoft's SQL Server 2005. This software provides algorithms for the data mining. We applied our data set and analyzed the data. Another is Java Bases software that is Weka, this software is developed by University of Waikato, New Zealand. This is open source software and very much useful for researchers and academicians. This software also provides charting of the data and algorithms for the data mining. We have developed the charts using Weka and found some hidden knowledge in the historical data. Data set also applied for the different algorithms in Weka and found interesting comparison of various statistical components generated by different algorithms using different test mode.

6.2 Conclusion

This research work established that Data mining and applied statistical methods are the appropriate tools to extract knowledge from historical data.

Following is the complete model of the research work.

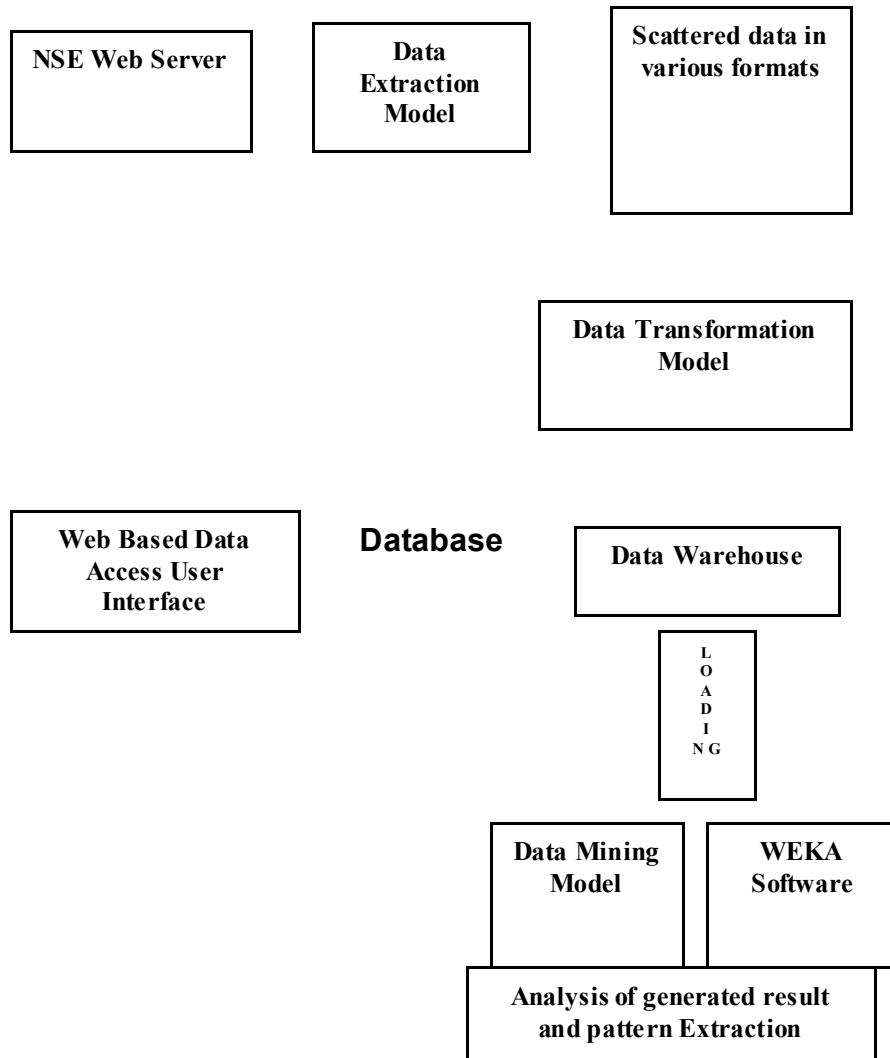


Fig 6.1: Complete Model of Research

The research work provides efficient model to be implemented for the creation and handling the database without installing client software, that is Web Based Data Access User Interface (WDAUI), so this model will be very much useful for the researchers and academicians. Furthermore, This tool provides web session security. This tool provides also DBA level functionality like create profile, role, new users, etc at any client machine. For this it is not compulsory for

DBA to log in at server machine. **We experienced that Implementation of this tool is very much easy and user friendly for the users at the client machine. It can be concluded that this model removes the existing complexity for the users of databases and provides more security.**

The research work also provides ETL (Extraction, Transformation and Loading) model, it is a way to automate the process of data extraction from web server, transformation from distributed database mode to data warehouse mode. This suggested model transfers the scattered data into the selected desired database. **We conclude that this ETL model is very fast and efficient to extract the data from web server and after gathering transforms into database, then data becomes ready to load for other tools for the data mining tools.**

The suggested computational data-mining model in this research is web-based model; it automatically loads the data from database and computes various results in visual format and numerical format. This tool is functioning very fast and millions of the data can be mined within fraction of seconds. We experienced to analyze the mined data from this computational model. We found following interesting results and extracted many important patterns for used data set.

- **The remarkable movement is there in the rate of the selected script or not?**
- **Maximum and minimum fluctuations based on variance of rate compare to other selected scripts.**
- **Dates on which rate of the selected script is fall down.**
- **From the correlation coefficient between Euclidean distance and Trading Quantity, it is extracted and analyzed that what is effect on trading quantity when rate of the script is fall down.**

- From the correlation coefficient between Euclidean distance and Trading Quantity, it is extracted and analyzed that rise in the rate of script is affected by trading volume or not?
- Fluctuation in the script based on variance of volume (trading quantity) for the selected range of dates.
- Correlation coefficient between open, high, low, close rate with volume gives the pattern that remarkable change in high rate of the script is depending on volume. It is up to 60% true for the selected script for the selected range of dates.
- It is also seen that, when correlation coefficient is remarkable positive for high rate compare to other categories, variance of the high rate is remarkable low compare to the rate of other categories.
- It is extracted from Pie Chart, for the specified range of date for the script having low variance has the highest volume. The script having maximum variance having lower volume. It means variance of the script is increases, it affect negative on the volume of that script.
- It can be concluded from the Numeric view of the scripts, total trading volume does not affect to the fluctuation of any category.

For the research purpose we have used Weka software, which is Java based open source software developed at Waikato University, New Zealand. We experienced that this software is very much useful for researchers and academicians. We have used this software for analytical study of various data mining algorithms and various statistical methods using in data mining. We found interesting results and many important patterns are extracted for our data set.

Finally, it can be concluded that use of the statistical methods is depending on the nature of the problem and the data set is selected. We implemented our suggested model efficiently during the research study. Further it can also be concluded that, the experts can analyze the outcome and result generated by any Data Mining application manually. We experienced throughout the research study that there is need of automation for this.

6.3 Future Work

Data mining is one step at the core of the knowledge discovery process, dealing with the extraction of patterns and relationships from large amounts of data. The demand is increasing rapidly for better decision support is answered by an extending availability of knowledge discovery and data mining products, in the form of research prototypes developed at various universities as well as software products from commercial vendors.

However, the development of improved data mining applications still remains a tedious process, and KDD is an emerging as well as booming field today. The following is a (naturally incomplete) list of issues that are unexplored or at least not satisfactorily solved yet:

6.3.1 Selection of different techniques

During the research study we experienced that the best techniques for data mining cannot be assumed before we study the problem. The issue is not that therefore which technique is better than another, but expert knowledge is required to decide which technique is suitable for the problem at hand. Currently available tools deploy either a single technique or a limited set of techniques to carry out data analysis. A truly useful tool has to provide a wide range of different techniques for the solution of different problems. The future work possibility in this area is to develop prototype for automation to understand the problem and selection of statistical techniques for the data mining.

6.3.2 Managing changing data:

In many applications, including the vast variety of nearly all business problems, the data is not stationary, but rather changing and evolving. This changing data may make previously discovered patterns invalid and hold new ones instead. Currently, the only solution to this problem is to repeat the same analysis process in periodic time intervals. There is clearly a need for incremental methods that are able to update changing models, and for strategies to identify and manage patterns of temporal change in knowledge bases.

6.3.3 Non-standard data types:

Numbers, strings and date are standard and basic data types. To reach growing need of information technology, latest databases provide facility to store many types of non-standard data types like, multimedia, free-form text, audio, image, video data etc. It is very much difficult to handle these types of data by the standard analysis model. To get rid of this problem there is requirement of research for the special and domain-specific methods and algorithms.

6.4 Extension work in Research

The current research has a possibility of extension in the areas specified here under:

6.4.1 Web-Based Data Access User Interface (WBDAUI) Model

This research suggests the three-tiered model of Web-Based Data Access User Interface. This model is practically implemented to create and handle Oracle database at client side. Based on this model practical work can be extended for the creation and handling of other databases like, SQL Server 2005, MS-Access, My SQL etc at the client machine. It can be extended to create more than one database, in which user can select any one database software. This extension will be very much useful for users of any database software using three-tiered application.

6.4.2 Data Extraction Model

This research suggests the Data Extraction Model. This model is practically implemented to extract, fetch and pull the data from web server to local machine. To extract the data from web server we are using Internet connection. The fetching and pulling of the data is depending on speed of Internet connection. We observed that sometime speed of the Internet is too low, it downloads incomplete file. During the extraction process, if connection is lost accidentally it does not display that how many files are downloaded. We have to see manually, which files are saved and which files are remaining.

Extension work can be performed practically for this model is to get rid from the speed of the Internet connection. Most important part can be implemented to display the name and number of the files to be saved on our machine whenever then tool is terminated anyway.

6.4.3 Data Transformation Model

This research suggests the Data Transformation Model. This model is practically implemented to transform the data from legacy system to our database. In our study it transforms the data extracted by Data Extraction Model. The extracted operational data is resides in scatted files as CSV or DBF format. This tool transforms the data and transformed data is collected and stored in MS-Access database.

Based on this suggested model, this work can be extended practically to transform multiple types and formats of the data (not only CSV or DBF). It also can be extended to store the transformed data into the database created in the software other than MS-Access also, like SQL Server 2005, My SQL, Oracle etc.

6.4.4 Data Mining Model

This suggested model is Data Mining model for the data mining. This model is practically implemented to mine the data extracted and transformed in our data warehouse. Based on this data-mining model, a web-based tool is developed for the practical study. This tool generates the charting and numerical outcomes. We observed practically, it is very much interesting to study the results generated using this tool.

Based on this model, more statistical techniques for the data mining can be implemented. An intelligent tool can be developed and implemented, if work can be extended for the automation of the selection of statistical techniques based on the nature of the data and selected problem.

We studied and applied the Weka software for our research, it satisfies our need up to some limit.

6.4.5 Weka Environment

One can enhance the result of WEKA using the followings in research study.

- Use more data sets:

The use of a large number of data sets would allow an increase in size of the data sets generated for classifying analysis.

- Use more data set sources:

The data sets used in this thesis came from the one stock exchange data collection. The use of a larger variety of real data sets from different stock exchanges may allow the formation of decision tree, which reveal patterns between different stock exchanges and data mining algorithms.

- Use small to very big data sets:

In the data mining industry the size of the data to be analyzed can be very large. The maximum size of the data sets used in this thesis was 2308. It would be useful to see what kind of performance is obtained and what types of decision tree are formed when large data sets are used.

- Use optimal parameter values by fine-tuning the settings of each algorithm
- Use of more characteristics of data sets:

Only 'number of instances' and 'number of attributes' was used in this thesis. Characteristics of the data sets such as whether or not the data sets contain numeric, symbolic or mixed values and missing values could be useful.

- Use visualization tools to analyze the generated data set: Visualization of the generated data set may provide important information and may allow better analysis of the decision tree formed.

6.5 Bibliography

Books:

- Data Mining Methods and Models
 - Wiley Publication, Daniel T Larose
- Data Mining Techniques
 - Wiley Publication, Michael J A Berry and Gordon S Linoff
- Data Mining - Practical Learning Tools and Techniques
 - ELSEVIER, Ian H Witten and Eibe Frank
- Data Mining - Concept and Techniques
 - ELSEVIER, Jiawei Han and Micheline Kamber
- Data Mining – Introduction and Advanced Topics
 - Pearson Education, Margaret H. Dunoham and S. Sridhar
- Pattern Recognition - Statistical Structural and Neural Approaches
 - Wiley Publication, Robert Schalkoff
- Pattern Recognition - Techniques and Applications
 - Oxford Publication, Rajjan Shinghal
- Mining the Web – Transforming Customer Data into Customer Value
 - Wiley Publication, Gordon S. Linoff and Michael J A Berry
- Principals of Data Mining
 - PHI Publication, David Hand, Heikki Mannila and Padhraic Smyth
- Data Warehousing – Concepts, Techniques, Products and Applications
 - PHI Publication, C S R Prabhu
- Clickstream Data Warehousing
 - Wiley Publication, Mark Sweiger, Mark R Madsen, Jimmy Langston and Howard Lombard
- Data Mining – Next Generation Challenges and Future Diretions
 - PHI Publication, Hillol Kargupta, Anupam Joshi, Krishnamoorthy Sivakumar and Yelena Yesha
- Data Warehousing Fundamentals
 - Wiley Publication, Paulraj Ponniah
- Decision Support and Data Warehouse Systems
 - TATA McGraw- Hill Publication, Efrem G Mallach
- Neural Networks for Pattern Recognition

- Oxford Publication, Christopher M Bishop
- Data Mining Explained
 - Digital Press, Rhonda Delmater and Monte Hancock
- Elements of Artificial Neural Networks
 - Penram International Publishing (India), Kishan Mehrotra, Chilukuri K Mohan, Sanjay Ranka
- Insight into Data Mining – Theory and Practice
 - PHI Publication, K P Soman, Shyam Diwakar and V Ajay
- Decision Support Systems in the 21st Century
 - PHI Publication, George M Marakas

Internet Sites:

- <http://www.kenorrinst.com/dwpaper.html>
- <http://www.inmoncif.com>
- <http://www.thearling.com/text/dmwhite/dmwhite.htm>
- <http://intelligent-web.org/wsm/overview/>
- http://en.wikipedia.org/wiki/Text_mining
- <http://www.microsoft.com/sql/prodinfo/features/features-at-a-glance.aspx>
- <http://www.microsoft.com/sql/technologies/dm/default.aspx>
- <http://www-306.ibm.com/software/data/iminer/>
- <https://informatica-news.com/>
- <http://www.businessobjects.com/products/businessobjectsxidefault.asp>
- <http://www.oracle.com/applications/peoplesoft-enterprise.html>
- <http://www.cognos.com/>
- http://www.microstrategy.com/Software/Products/Service_Modules/DataMining_Services/
- <http://dev.hyperion.com/products/intelligence/>
- <http://www.cs.waikato.ac.nz/~ml/index.html>
- <http://www.jcp.org/en/jsr/detail?id=247>
- <http://www.oracle.com/technology/products/bi/odm/index.html>
- http://www.spss.com/data_mining/index.htm
- <http://www.sas.com/technologies/analytics/datamining/>
- <http://www.datamininglab.com>