DR. D.Y. PATIL INSTITUTE OF TECHNOLOGY

Pimpri, Pune-411018

DEPARTMENT

of

COMPUTER ENGINEERING

LAB MANUAL

Laboratory Practice II

Vision of the Institute:

"Empowerment through Knowledge"

Mission of the Institute:

- ***Developing human potential to serve the Nation by***
- ***Dedicated efforts for quality education***
- ***Yearning to promote research and development***
- ***Persistent endeavor to imbibe moral and professional ethics***
- ***Inculcating the concept of emotional intelligence***
- ***Emphasizing extension work to reach out to the society***
- ***Treading the path to meet the future challenges***

Vision of the Department:

"To produce globally competitive computer professionals, enriched with knowledge and power of innovation"

Mission of the Department:

- ***Imparting quality education using state-of-art facilities to meet the global challenges.***
- ***Enhancing the potential of aspiring students and faculty for higher education and lifelong learning.***
- ***Imbibing ethical values and developing leadership skills those lead to professionals with strong commitments.***

Program Educational Objectives

PEO 1:

Have strong fundamental concepts in mathematics, science and engineering to address technological challenges.

PEO 2:

Possess knowledge and skills in the field of Computer Engineering for analyzing, designing and implementing novel software products in a dynamic environment for successful career and pursue higher studies.

PEO 3:

Demonstrate multidisciplinary approach and leadership skills that augment their professional competency.

PEO 4:

Exhibit commitment to ethical practices, societal contributions and lifelong learning.

Program Outcomes

a. An ability to apply knowledge of computing, mathematics, science and engineering fundamentals appropriate to Computer Engineering.

b. An ability to define the problems and provide solutions by designing and conducting experiments, interpreting and analyzing data.

c. An ability to design, implement and evaluate a system, process, component and programme to meet desired needs within realistic constraints.

d. An ability to investigate, formulate, analyze and provide appropriate solution to the engineering problems.

e. An ability to use modern engineering tools and technologies necessary for engineering practice.

f. An ability to analyze the local and global impact of computing on individuals, organizations and society.

g. An ability to understand the environmental issues and provide the sustainable system.

h. An ability to understand professional and ethical responsibility.

i. An ability to function effectively as an individual or as a team member to accomplish the goal.

j. An ability to communicate effectively at different levels.

k. An ability to keep abreast with contemporary technologies through lifelong learning.

l. An ability to understand engineering, management, financial aspects, performance, optimizations and time complexity necessary for professional practice.

Graduate Attributes and Program Outcomes

Graduate Attributes	Program Outcomes
1. Engineering Knowledge	**a.** An ability to apply knowledge of computing, mathematics, science and engineering fundamentals appropriate to Electronics and Telecommunication.
2. Problem Analysis	**b.** An ability to define the problems and provide solutions by designing and conducting experiments, interpreting and analysing data.
3. Design & Development of Solutions	**c.** An ability to design, implement and evaluate a system, process, component and program to meet desired needs within realistic constraints.
4. Investigation of Complex Problem	**d.** An ability to investigate, formulate, analyze and provide appropriate solution to the engineering problems.
5. Modern Tools Usage	**e.** An ability to use modern engineering tools and technologies necessary for engineering practices.
6. Engineer and Society	**f.** An ability to analyze the local and global impact of computing on individuals, organizations and society.
7. Environment & Sustainability	**g.** An ability to understand the environmental issues and provide the sustainable system.
8. Ethics	**h.** An ability to understand professional and ethical responsibility.
9. Individual & Team work	**i.** An ability to function effectively as an individual or as a team member to accomplish the goal.
10. Communication	**j.** An ability to communicate effectively at different levels.
11. Lifelong Learning	**k.** An ability to keep abreast with contemporary technologies through lifelong learning.
12. Project management & Finance	**l.** An ability to apply knowledge of principles of resource management and economics to provide better services in the field of Computer Engineering

INDEX

A.SYLLABUS STRUCTURE

TE (COMPUTER ENGINEERING)- 2015 COURSE STRUCTURE

Savitribai Phule University of Pune
Fourth Year Computer Engineering (2015 Course)
(with effect from 2018-19)

Semester I

Course Code	Course	Teaching Scheme Hours / Week		Examination Scheme and Marks						Credit	
		Theory	Practical	In-Sem	End-Sem	TW	PR	OR/ *PRE	Total	TH/ TUT	PR
410241	High Performance Computing	04	--	30	70	--	--	--	100	04	--
410242	Artificial Intelligence and Robotics	03	--	30	70	--	--	--	100	03	--
410243	Data Analytics	03	--	30	70	--	--	--	100	03	--
410244	Elective I	03	--	30	70	--	--	--	100	03	--
410245	Elective II	03	--	30	70	--	--	--	100	03	--
410246	Laboratory Practice I	--	04	--	--	50	50	--	100	--	02
410247	Laboratory Practice II	--	04	--	--	50	50	--	100	--	02
410248	Project Work Stage I	--	02	--	--	--	--	*50	50	--	02
									Total Credit	16	06
Total		16	10	150	350	100	100	50	750	22	
410249	Audit Course 5									Grade	

Elective I	Elective II
410244 (A)Digital Signal Processing	410245 (A) Distributed Systems
410244 (B)Software Architecture and Design Patterns	410245 (B)Software Testing and Quality Assurance
410244 (C)Pervasive and Ubiquitous Computing	410245 (C)Operations Research
410244 (D)Data Mining and Warehousing	410245 (D)Mobile Communication

Th: Theory TW: Term Work Pr: Practical Or: Oral Pre: Presentation

B. COURSE CONTENT

410247: Laboratory Practice II

Teaching Scheme

Practical: 4 hrs/week

Examination Scheme
Term Work: 50 Marks
Presentation: 50 Marks

Elective- I 410244(D): Data Mining and Warehousing

1 For an organization of your choice, choose a set of business processes. Design star / snow flake schemas for analyzing these processes. Create a fact constellation schema by combining them. Extract data from different data sources, apply suitable transformations and load into destination tables using an ETL tool. For Example: Business Origination: Sales, Order, and Marketing Process.

2 Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques (minimum 2). Visualize the clusters using suitable tool.

3 Apply a-priori algorithm to find frequently occurring items from given data and generate strong association rules using support and confidence thresholds. For Example: Market Basket Analysis

4 Consider a suitable text dataset. Remove stop words, apply stemming and feature selection techniques to represent documents as vectors. Classify documents and evaluate precision, recall.

5 Mini project on classification: Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix and compare these models. Also apply cross validation while preparing the training and testing datasets. For Example: Health Care Domain for predicting disease

Elective-II 410245(B): Software Testing and Quality Assurance

6 Mini-Project 1: Create a small application by selecting relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.

7	Mini-Project 2: Create a small web-based application by selecting relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.

C. COURSE OBJECTIVES:

To understand the fundamentals of Data Mining
To identify the appropriateness and need of mining the data
To learn the preprocessing, mining and post processing of the data
To understand various methods, techniques and algorithms in data mining

D. COURSE OUTCOMES:

- Analyze the output generated by the process of data mining
- Explore the hidden patterns in the data
- Optimize the mining process by choosing best data mining technique

- Apply recent automation tool for various software testing for testing software
- Apply different approaches of quality management, assurance, and quality standard to software system
- Apply and analyze effectiveness Software Quality Tools

E. PREREQUISITES:

- Database,Software Engineering,Software Modelling Design

F. ASSIGNMENT PLAN

Assignment Number	Assignment Name	References
Elective- I 410244(D): Data Mining and Warehousing		
1	For an organization of your choice, choose a set of business processes. Design star / snow flake schemas for analyzing these processes. Create a fact constellation schema by combining them. Extract data from different data sources, apply suitable transformations and load into destination tables using an ETL tool. For Example: Business Origination: Sales, Order, and Marketing Process.	T1,R2
2	Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques (minimum 2). Visualize the clusters using suitable tool.	T1,T2
3	Apply a-priori algorithm to find frequently occurring items from given data and generate strong association rules using support and confidence thresholds. For Example: Market Basket Analysis	R2
4	Consider a suitable text dataset. Remove stop words, apply stemming and feature selection techniques to represent documents as vectors. Classify documents and evaluate precision, recall.	R1,R2
5	Mini project on classification: Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix and compare these models. Also apply cross validation while preparing the training and testing datasets. For Example: Health Care Domain for predicting disease	T2,R1
Elective-II 410245(B): Software Testing and Quality Assurance		

6	Mini-Project 1: Create a small application by selecting relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.	T1,R1,R2
7	Mini-Project 2: Create a small web-based application by selecting relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.	T2,R2

Text:

Elective I

1. Han, Jiawei Kamber, Micheline Pei and Jian, “Data Mining: Concepts and Techniques”, Elsevier Publishers, ISBN:9780123814791, 9780123814807.

2. Parag Kulkarni, “Reinforcement and Systemic Machine Learning for Decision Making” by Wiley-IEEE Press, ISBN: 978-0-470-91999-6

Elective II

1. M G Limaye, “Software Testing Principles, Techniques and Tools”, Tata McGraw Hill, ISBN: 9780070139909 0070139903

2. Srinivasan Desikan, Gopalswamy Ramesh, “Software Testing Principles and Practices”, Pearson, ISBN-10: 817758121X

References:

Elective I

1. Matthew A. Russell, "Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More" , Shroff Publishers, 2nd Edition, ISBN: 9780596006068

2. Maksim Tsvetovat, Alexander Kouznetsov, "Social Network Analysis for Startups:Finding connections on the social web", Shroff Publishers , ISBN: 10: 1449306462

Elective II

1. Naresh Chauhan, “Software Testing Principles and Practices ", OXFORD, ISBN-10: 0198061846. ISBN-13: 9780198061847

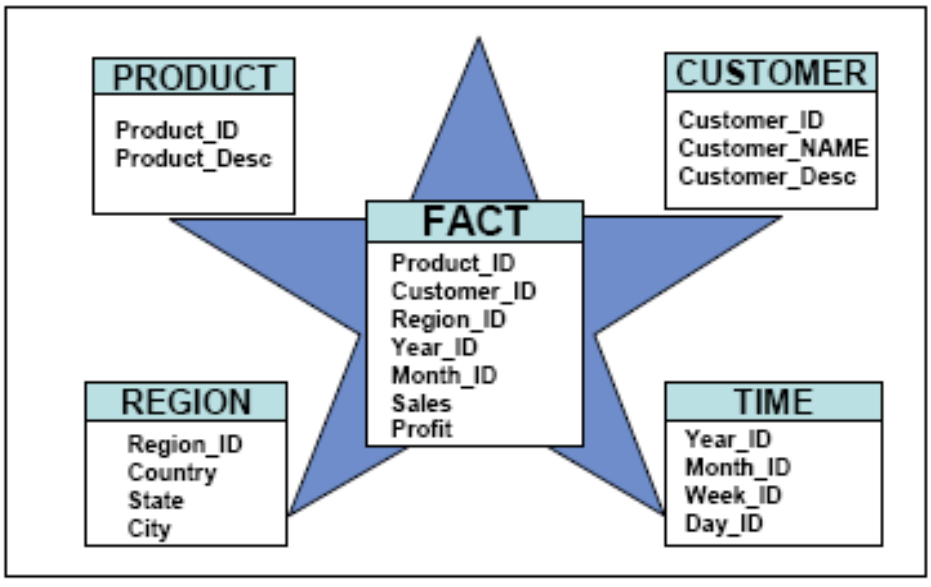
2. Stephen Kan, “Metrics and Models in Software Quality Engineering”, Pearson, ISBN-10: 0133988082; ISBN-13: 978-0133988086

ASSIGNMENT 1

Aim : For an organization of your choice, choose a set of business processes. Design star / snow flake schemas for analyzing these processes. Create a fact constellation schema by combining them. Extract data from different data sources, apply suitable transformations and load into destination tables using an ETL tool. For Example: Business Origination: Sales, Order, and Marketing Process.

Theory: To overcome performance issues for large queries in the data warehouse, we use dimensional models. The dimensional modeling approach provides a way to improve query performance for summary reports without affecting data integrity

Star Schema : A dimensional model is also commonly called a star schema. This type of model is very popular in data warehousing because it can provide much better query performance, especially on very large queries, than an E/R model. However, it also has the major benefit of being easier to understand. It consists, typically, of a large table of facts (known as a fact table), with a number of other tables surrounding it that contain descriptive data, called dimensions. When it is drawn, it resembles the shape of a star, therefore the name.



The dimensional model consists of two types of tables having different characteristics. They are:

_ Fact table

_ Dimension table

Fact table characteristics

_ The fact table contains numerical values of what you measure. For example, a fact value of 20 might mean that 20 widgets have been sold.

_ Each fact table contains the keys to associated dimension tables. These are called *foreign keys* in the fact table.

_ Fact tables typically contain a small number of columns.

_ Compared to dimension tables, fact tables have a large number of rows.

_ The information in a fact table has characteristics, such as:

– It is numerical and used to generate aggregates and summaries.

– Data values need to be additive, or semi-additive, to enable summarization of a large number of values.

Dimension table characteristics

_ Dimension tables contain the details about the facts. That, as an example, enables the business analysts to better understand the data and their reports.

_ The dimension tables contain descriptive information about the numerical values in the fact table. That is, they contain the attributes of the facts. For

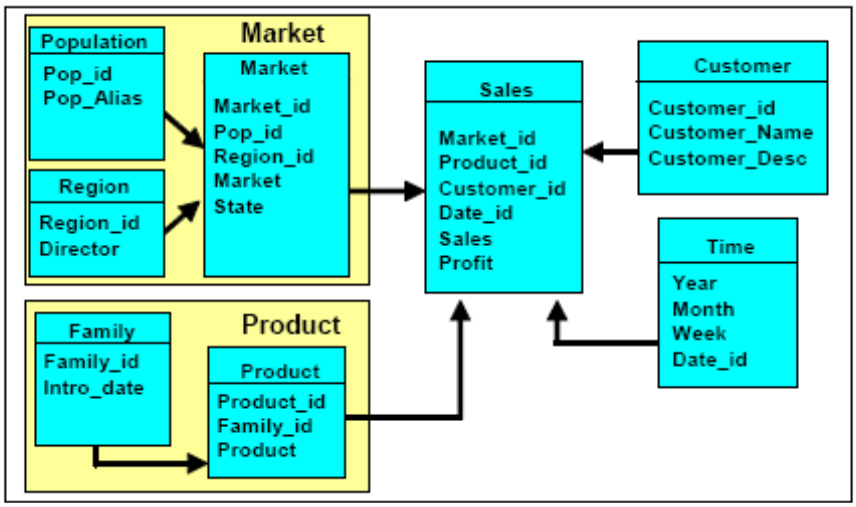
example, the dimension tables for a marketing analysis application might include attributes such as time period, marketing region, and product type.

_ Since the data in a dimension table is denormalized, it typically has a large number of columns.

_ The dimension tables typically contain significantly fewer rows of data than the fact table.

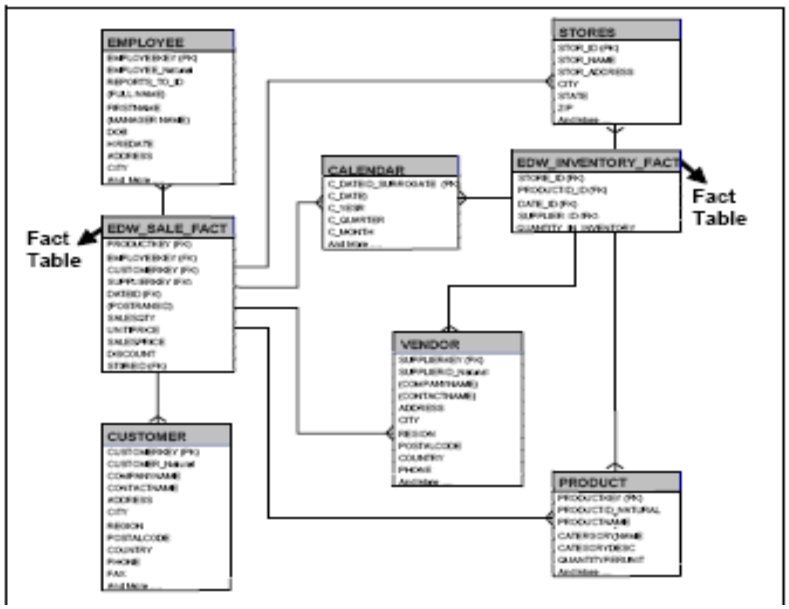
_ The attributes in a dimension table are typically used as row and column headings in a report or query results display. For example, the textual descriptions on a report come from dimension attributes.

Snowflake model: Further normalization and expansion of the dimension tables in a star schema result in the implementation of a snowflake design. A dimension is said to be snowflaked when the low-cardinality columns in the dimension have been removed to separate normalized tables that then link back into the original dimension table.



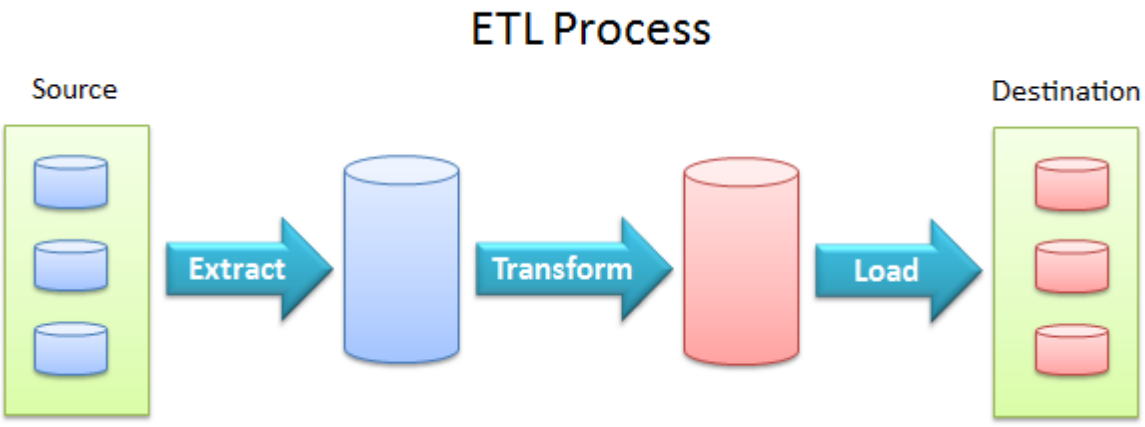
In this example, we expanded (snowflaked) the Product dimension by removing the low-cardinality elements pertaining to Family, and putting them in a separate Family dimension table. The Family table is linked to the Product dimension table by an index entry (Family_id) in both tables. From the Product dimension table, the Family attributes are extracted by, in this example, the Family Intro_date. The keys of the hierarchy (Family_Family_id) are also included in the Family table. In a similar fashion, the Market dimension was snowflaked.

Fact Constellation Schema: A multi-star model is a dimensional model that consists of multiple fact tables, joined together through dimensions. Following figure shows fact tables that were joined, which are EDW_Sales_Fact and EDW_Inventory_Fact.



ETL Tools:

ETL is short for extract, transform, load, three database functions that are combined into one tool to pull data out of one database and place it into another database. Extract is the process of reading data from a database. ... Transformation occurs by using rules or lookup tables or by combining the data with other data.



Following are some open access ETL Tools:

alend Open Source Data Integrator

Talend provides multiple solutions for data integration, both open source and commercial editions. Talend offers an Eclipse-based interface, drag-and-drop design flow, and broad connectivity with more than 400 pre-configured application connectors to bridge between databases, mainframes, file systems, web services, packaged enterprise applications, data warehouses, OLAP applications, Software-as-a-Service, Cloud-based applications, and more.

Scriptella

Scriptella is an open source ETL (Extract-Transform-Load) and script execution tool written in Java. Its primary focus is simplicity. You don't have to study yet another complex XML-based language - use SQL (or other scripting language suitable for the data source) to perform required transformations. Scriptella is licensed under the Apache License, Version 2.0

KETL

KETL is a premier, open source ETL tool. The data integration platform is built with portable, java-based architecture and open, XML-based configuration and job language. KETL features successfully compete with major commercial products available today. Highlights include:

Support for integration of security and data management tools

Proven scalability across multiple servers and CPU's and any volume of data

No additional need for third party schedule, dependency, and notification tools

Jaspersoft ETL

Jasper ETL is easy to deploy and out-performs many proprietary ETL software systems. It is used to extract data from your transactional system to create a consolidated data warehouse or data mart for reporting and analysis.

GeoKettle

GeoKettle is a powerful, metadata-driven Spatial ETL tool dedicated to the integration of different spatial data sources for building and updating geospatial data warehouses. GeoKettle enables the Extraction of data from data sources, the Transformation of data in order to correct errors, make some data cleansing, change the data structure, make them compliant to defined standards, and the Loading of transformed data

into a target DataBase Management System (DBMS) in OLTP or OLAP/SOLAP mode, GIS file or Geospatial Web Service.

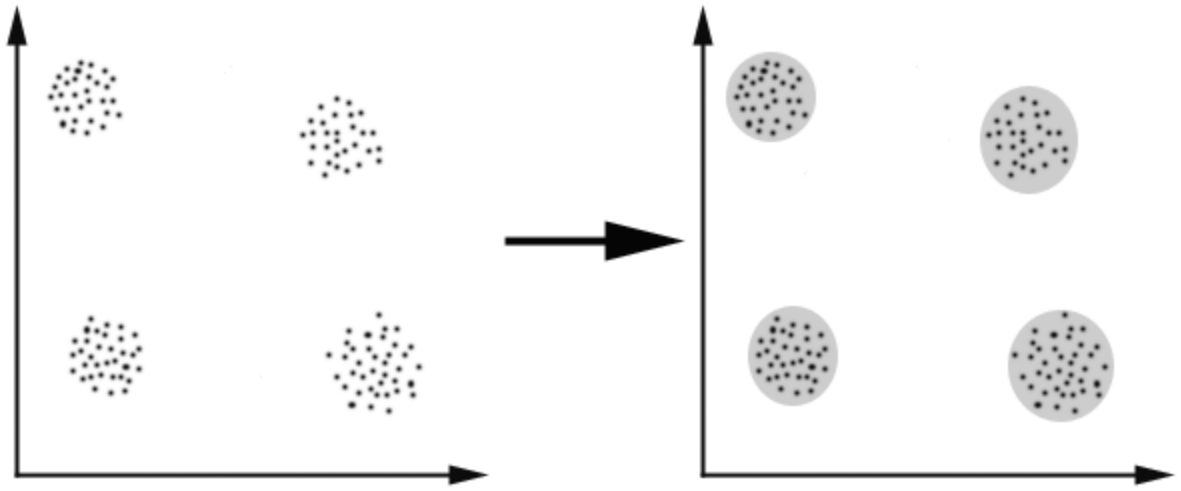
Conclusion: Thus we have designed sales business process using star schema. Fact constellation schema is used for design of combing sales and inventory process we desi and we learned the ETL tool application.

ASSIGNMENT 2

Aim: Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques. Visualize the clusters using suitable tool.

Theory:

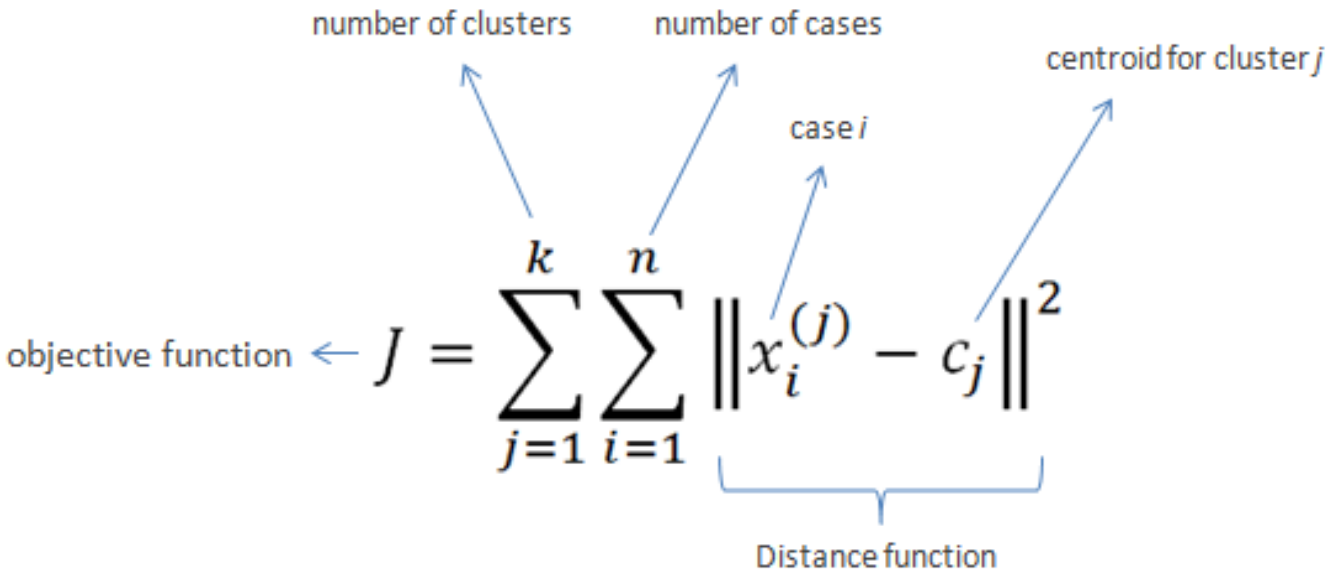
Clustering: It can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a *structure*in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are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



In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is *distance*: two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called *distance-based clustering*.
Another kind of clustering is *conceptual clustering*: two or more objects belong to the same cluster if this one defines a concept *common* to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

K-Means clustering

K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly k different clusters of greatest possible distinction. The best number of clusters k leading to the greatest separation (distance) is not known as a priori and must be computed from the data. The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function: 1) Mnemonic machine instructions – each one is translated to a single executable instruction.



Algorithm

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

Initial clusters:

Centroid (C1) = 16 [16] Centroid (C2) = 22 [22]

Iteration 1:

C1=15.33 [15,15,16] C2 = 36.25
[19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65]

Iteration 2:

C1=18.56[15,15,16,19,19,20,20,21,22] C2 = 45.90
[28,35,40,41,42,43,44,60,61,65]

Iteration 3:

C1=19.50[15,15,16,19,19,20,20,21,22,28] C2 = 47.89
[35,40,41,42,43,44,60,61,65]

Iteration 4:

C1=19.50[15,15,16,19,19,20,20,21,22,28] C2 = 47.89

[35,40,41,42,43,44,60,61,65]

No change between iterations 3 and 4 has been noted. By using clustering, 2 groups have been identified 15-28 and 35-65. The initial choice of centroids can affect the output clusters, so the algorithm is often run multiple times with different starting conditions in order to get a fair view of what the clusters should be.

Tools Used For Cluster visualization :KNIME

Conclusion : Thus we have successfully implemented clustering techniques using K-means algorithm

ASSIGNMENT 3

Aim: Apply a-priori algorithm to find frequently occurring items from given data and generate strong association rules using support and confidence thresholds. For Example:

Market Basket Analysis

Theory:

Apriori Algorithm:

General Process Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent item sets in a database.

2. Second, these frequent itemsets and the minimum con fidence constraint are used to form rules.

While the second step is straight forward, the firs t step needs more attention. Finding all frequent itemsets in a database is difficult since it involves searching all possible item sets (item combinations). The set of possible itemsets I s the power set over I and has size 2 n − 1 (excluding the empty set which is not a valid itemset). Although t he size of the powerset grows exponentially in the number of items n in I , efficient search is possible using the downward-closure property of support (also called anti-monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets m ust also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori) can find all f requent itemsets.

Apriori Algorithm Pseudocode:

```
procedure Apriori (T, minSupport )
{ //T is the database and minSupport is the minimum support
L1= {frequent items};
for (k= 2; L k-1 != ∅ ; k++)
{
C k = candidates generated from L k-1 //that is cartesian product L k-1 x L k-1 and eliminating any k-1 size itemset that is not
//frequent
for each transaction t in database do
{ #increment the count of all candidates in Ck that are contained in t
Lk= candidates in Ck with minSupport
}//end for each
}//end for
return UkLk;
```

}

Example:

Suppose you have records of large number of transactions at a shopping center as follows:

Transaction ID	Item1	Item2	Item3	Item4	Item 5	Item6
T1	Mnago	Onion	Jar	Key-chain	Eggs	Chocolates
T2	Nuts	Onion	Jar	Key-chain	Eggs	Chocolates
T3	Mnago	Apple	Key-chain	Eggs	-	-
T4	Mnago	Toothbrush	Corn	Key-chain	Chocolates	-
T5	Corn	Onion	Onion	Key-chain	Knife	Eggs

Organize the data items on a shelf means finding the items that are purchased together more frequently than others. Apriori is the classic and probably the most basic algorithm to do it.

Now, we follow a simple golden rule: we say an item/itemset is frequently bought if it is bought at least 60% of times(i.e Minimum Support=3). So for here it should be bought at least 3 times.

For simplicity M = Mango ,O = Onion , J=Jar, K= Key-chain, E=egg, C= Chocolate, Co=Corn, A=Apple Kn=Knife and so on… So the table becomes

Original table:

Transaction ID	Items Bought
T1	{M, O, J, K, E, C}
T2	{N, O, J, K, E, C}
T3	{M, A, K, E}
T4	{M, T, Co, K, C}
T5	{Co, O, O, K, Kn, E}

Step 1: Count the number of transactions in which each item occurs, Note „O=Onion‟ is bought 4 times in total, but, it occurs in just 3 transactions.

Candidate SetC1)	transactions(Support)
M	3

O	3
N	1
K	5
E	4
C	3
J	2
A	1
Kn	1
Co	1
T	1

Step 2: Now we said the item is said frequently bought if it is bought at least 3 times. So in this step we remove all the items that are bought less than 3 times from the above table and we are left with. This is the single items that are bought frequently. Now let‟s say we want to find a pair of items that are bought frequently. We continue from the above table (Table in step 2)

Item (Frequent Item Sets L1)	Number of transactions(Support)
M	3
O	3
K	5
E	4
C	3

Step 3: We start making pairs from the first item, like MO,MK,ME,MC and then we start with the second item like OK,OE,OC. We did not do OM because we already did MO when we were making pairs with M and buying a Mango and Onion together is same as buying Onion and Mango together. After making all the pairs we get,

Item pairs
MO

MK
ME
MC
OK
OE
OC
KE
KC
EC

Step 4: Now we count how many times each pair is bought together. For example M and O is just bought together in {M,O,N,K,E,C}

While M and K is bought together 3 times in { M, O, J, K, E, C }, { M, A, K, E } AND { M, T, Co, K, C } After doing that for all the pairs we get

Item Pairs (Candidate set C2)	**Number of transactions (Support)**
MO	1
MK	3
ME	2
MC	2
OK	3
OE	3
OC	2
KE	4
KC	3
EC	2

Step 5: Golden rule to the rescue. Remove all the item pairs with number of transactions less than three and we are left with

Item Pairs (Frequent Item Sets L2)	Number of transactions (Support)

MK	3
OK	3
OE	3
KE	4
KY	3

These are the pairs of items frequently bought together.

Now let‟s say we want to find a set of three items that are brought together. We use the above table (table in step 5) and make a set of 3 items.

Step 6: To make the set of three items we need one more rule (it‟s termed as self-join),

It simply means, from the Item pairs in the above table, we find two pairs with the same first Alphabet, so we get
It simply means, from the Item pairs in the above table, we find two pairs with the same first Alphabet, so we get

OK and OE, this gives OKE

KE and KC, this gives KEC

Then we find how many times O,K,E are bought together in the original table and same for K,E,Y and we get the following table

Item Set (Candidate Set C3)	Number of transactions (Support)
OKE	3
KEY	2

While we are on this, suppose you have sets of 3 items say ABC, ABD, ACD, ACE, BCD and you want to generate item sets of 4 items you look for two sets having the same first two alphabets.

ABC and ABD -> ABCD
ACD and ACE -> ACDE

And so on … In general you have to look for sets having just the last alphabet/item different.

Step 7: So we again apply the golden rule, that is, the item set must be bought together at least 3 times which leaves us with just OKE, Since KEY are bought together just two times.

Thus the set of three items that are bought together most frequently are : Frequent Item Set L3={O, K, E} .

CONCLUSION

Thus we have successfully implemented Apriori Approach for data mining to organize the data items on a shelf using following table of items purchased in a Mall.

ASSIGNMENT4

Aim: Consider a suitable text dataset. Remove stop words, apply stemming and feature selection techniques to represent documents as vectors. Classify documents and evaluate

precision, recall..

Theory:

'Preprocessing', is the most important subtask of text classification . The main objective of preprocessing is to obtain the key features or key terms from online news text documents and to enhance the relevancy between word and document and the relevancy between word and category. The goal behind preprocessing is to represent each document as a feature vector, that is, to separate the text into individual words.

Preprocessing steps:

Step 1:Data Collection

Step 2: Stop word removal

Step 3: Stemming

Step 4: Indexing

Step 5: Term weighting

Step 6: Feature Selection

Stop Word Removal

Stop-words are language-specific functional words, are frequent words that carry no information (i.e., pronouns, prepositions, conjunctions). In English language, there are about 400- 500 Stop words. Examples of such words include 'the', 'of', 'and', 'to'. The first step during preprocessing is to remove these Stop words, which has proven as very important .Many of the most frequently used words in English sentence are useless in Information Retrieval (IR) and text mining.

Stemming

Stemming techniques are used to find out the root/stem of a word. Stemming converts words to their stems, which incorporates a great deal of language-dependent linguistic knowledge. Behind stemming, the hypothesis is that words with the same stem or word root mostly describe same or relatively close concepts in text and so words can be conflated by using stems.

For example, the words, user, users, used, using all can be stemmed to the word 'USE'.

Term Weighting

In the vector space model, the documents are represented as vectors. Term weighting is an important concept which determines the success or failure of the classification system. Since different terms have different level of importance in a text, the term weight is associated with every term as an important indicator. The three main components that affect the importance of a term in a document are the Term Frequency (TF) factor, Inverse Document Frequency (IDF) factor and Document length normalization .Term frequency of each word in a document (TF) is a weight which depends on the distribution of each word in documents. It expresses the Importance of the word in the document. Inverse document frequency of each word in the document database (IDF) is a weight which depends on the distribution of each word in the document database. It expresses the importance of each word n the document database .TF/IDF is a technique which uses both TF and IDF to determine the weight a term. TF/IDF scheme is very popular in text classification field and almost all the other weighting schemes are variants of this scheme .In vector space model organization of document also affect the performance of system

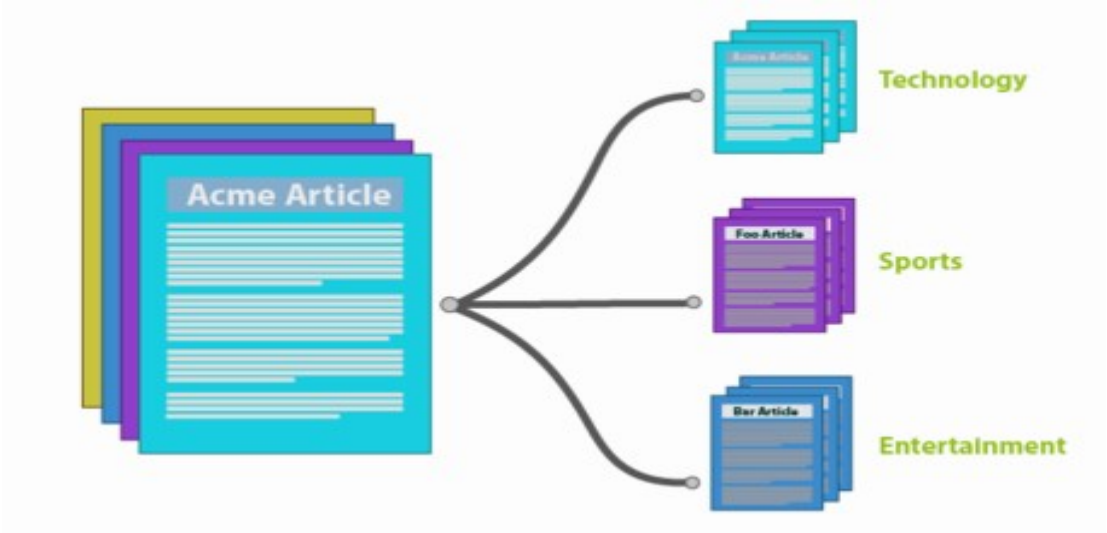
Feature Selection

The document term matrix contains the set of document as row and set of terms as columns. These terms are also known as features because there are used to uniquely identify the document. The sparsity of document term matrix represent the set of features that's frequency is zero. Higher the sparsity value lead to increase the set of feature and lower the sparsity value decrease the set of feature. Document frequency (DF) is the number of documents in which a term occurs. DF thresholding is the simplest technique for feature reduction. Stop word elimination explained previously, removes all high frequency words that are irrelevant to the classification task, while DF thresholding removes infrequent words. All words that occur in less than 'm' documents of the text collection are not considered as features, where 'm' is a pre-determined threshold. DF thresholding is based on the assumption that infrequent words are not informative for category prediction. DF thresholding easily scales to a very large corpora and has the advantage of easy implementation.

Document Classification

Document classification is an example of Machine Learning (ML) in the form of Natural Language Processing (NLP). By classifying text, we are aiming to assign one or more classes or categories to a document, making it easier to manage and sort. This is especially useful for publishers, news sites, blogs or anyone who deals with a lot of content.

Broadly speaking, there are two classes of ML techniques: supervised and unsupervised. In supervised methods, a model is created based on a training set. Categories are predefined and documents within the training dataset are manually tagged with one or more category labels. A classifier is then trained on the dataset which means it can predict a new document's category . Depending on the classification algorithm or strategy used, the classifier might also provide a confidence measure to indicate how confident it is that the classification



A simple Illustration of Document Classification

We can use the words within a document as "features" to help us predict the classification of a document. For example, we could have three very short, trivial documents in our training set as shown below:

Reference Document Class 1	Reference Document Class 2	Reference Document Class 3
Some tigers live in the zoo	Green is a color	Go to New York city

To classify these documents, we would start by taking all of the words in the three documents in our training set and creating a table or vector from these words.

(some,tigers,live,in,the,zoo,green,is,a,color,go,to,new,york,city)class

Then for each of the training documents, we would create a vector by assigning a 1 if the word exists in the training document and a 0 if it doesn't, tagging the document with the appropriate Class as follows:

some	tigers	live	in	the	zoo	green	is	a	color	go	to	new	york	city	
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	**class 1**
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	**class 2**
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	**class 3**

When a new untagged document arrives for classification and it contains the words "Orange is a color" we would create a word vector for it by marking the words which exist in our classification vector.

Untagged document arrives for classification and it contains the words "Orange is a color" we would create a word vector for it by marking the words which exist in our classification vector.

some	tigers	live	in	the	zoo	green	is	a	color	go	to	new	york	city	
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	**unknown class**

If we compare this vector for the document of unknown class, to the vectors representing our three document classes, we can see that it most closely resembles the vector for class 2 documents, as shown below:

< 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0 > Unknown class

< 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 > class 1 (6 matching terms)
< 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0 > class 2 (14 matching terms - winner!!)
< 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 > class 3 (7 matching terms)

It is then possible to label the new document as a class 2 document with an adequate degree of confidence.

CONCLUSION

Thus we have successfully classified a text document by removing stop words, applying stemming and feature selection techniques to represent documents as vectors.

ASSIGNMENT 5

Mini Project :Classification

Problem Statement: Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix and compare these models. Also apply cross validation while preparing the training and testing datasets. For Example: Health Care Domain for predicting disease

Theory:

Nowadays there is huge amount of data being collected and stored in databases everywhere across the globe. The tendency is to keep increasing year after year. It is not hard to find databases with Terabytes of data in enterprises and research facilities. That is over 1,099,511,627,776 bytes of data. There is invaluable information and knowledge "hidden" in such databases; and without automatic methods for extracting this information it is practically impossible to mine for them. Throughout the years many algorithms were created to extract what is called nuggets of knowledge from large sets of data. There are several different methodologies to approach this problem: classification, association rule, clustering, etc.

Classification consists of predicting a certain outcome based on a given input. In order to predict the outcome, the algorithm processes a training set containing a set of attributes and the respective outcome, usually called goal or prediction attribute. The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Next the algorithm is given a data set not seen before, called prediction set, which contains the same set of attributes, except for the prediction attribute – not yet known. The algorithm analyses the input and produces a prediction. The prediction accuracy defines how "good" the algorithm is. For example, in a medical database the training set would have relevant patient information recorded

previously, where the prediction attribute is whether or not the patient had a heart problem. Table 1 below illustrates the training and prediction sets of such database.

Training set

Age	Heart rate	Blood pressure	Heart problem
65	78	150/70	Yes
37	83	112/76	No
71	67	108/65	No

Prediction set

Age	Heart rate	Blood pressure	Heart problem
43	98	147/89	?
65	58	106/63	?
84	77	150/65	?

TABLE 1 – TRAINING AND PREDICTION SETS FOR MEDICAL DATABASE

Among several types of knowledge representation present in the literature, classification normally uses prediction rules to express knowledge. Prediction rules are expressed in the form of IF-THEN rules, where the antecedent (IF part) consists of a conjunction of conditions and the rule consequent (THEN part) predicts a certain predictions attribute value for an item that satisfies the antecedent. Using the example above, a rule predicting the first row in the training set may be represented as following: IF (Age=65 AND Heart rate>70) OR (Age>60 AND Blood pressure>140/70) THEN Heart problem=yes In most cases the prediction rule is immensely larger than the example above. Conjunction has a nice property for classification; each condition separated by OR's defines smaller rules that captures relations between attributes. Satisfying any of these smaller rules means that the consequent is the prediction. Each smaller rule is formed with AND's which facilitates narrowing down relations between attributes. How well predictions are done is measured in percentage of predictions hit against the total number of predictions. A decent rule ought to have a hit rate greater than the occurrence of the prediction attribute. In other words, if the algorithm is trying to predict rain in Seattle and it rains 80% of the time, the algorithm could easily have a hit rate of 80% by just predicting rain all the time. Therefore, 80% is the base prediction rate that any algorithm should achieve in this case. The optimal solution is a rule with 100% prediction hit rate, which is very hard, when not impossible, to achieve. Therefore, except for some very specific problems, classification by definition can only be solved by approximation algorithms

Decision Tree: The ID3 algorithm was originally developed by J. Ross Quinlan at the University of Sydney, and he first presented it in the 1975 book "Machine Learning". The ID3 algorithm induces classification models, or decision trees, from data. It is a supervised learning algorithm that is trained by examples for different classes. After being trained, the algorithm should be able to predict the class of a new item. ID3 identifies attributes that differentiate one class from another. All attributes must be known in advance, and must also be either continuous or selected from a set of known values. For instance, temperature (continuous), and country of citizenship (set of known values) are valid attributes. To determine which attributes are the most important, ID3 uses the statistical property of entropy. Entropy measures the amount of information in an attribute. This is how the decision tree, which will be used in testing future cases, is built.

One of the limitations of ID3 is that it is very sensitive to attributes with a large number of values (e.g. social security numbers). The entropy of such attributes is very low, and they don't help you in performing any type of prediction. The C4.5 algorithm overcomes this problem by using another statistical property known as information gain. Information gain measures how well a given attribute separates the training sets into the output classes. Therefore, the C4.5 algorithm extends the ID3 algorithm through the use of information gain to reduce the problem of artificially low entropy values for attributes such as social security number.

LABORATORY WORK

S.No	Name of the experiment	Type of task	Course Outcome Addressed	Program Outcome Addressed
1	For an organization of your choice, choose a set of business processes. Design star / snow flake schemas for analyzing these processes. Create a fact constellation schema by combining them. Extract data from different data sources, apply suitable transformations and load into destination tables using an ETL tool. For Example: Business Origination: Sales, Order, and Marketing Process.	DESIGN & PERFORMANCE	**CO2**	a,b,d,e,j,k,l
2	Consider a suitable dataset. For clustering of data instances in different groups, apply different clustering techniques (minimum 2). Visualize the clusters using suitable tool.	DESIGN & PERFORMANCE	**CO2**	a,b,d,e,j,k,l
3	Apply a-priori algorithm to find frequently occurring items from given data and generate strong association rules	DESIGN & PERFORMANCE	**CO2**	a,b,d,e,j,k,l

	using support and confidence thresholds. For Example: Market Basket Analysis			
4	Consider a suitable text dataset. Remove stop words, apply stemming and feature selection techniques to represent documents as vectors. Classify documents and evaluate precision, recall.	DESIGN & PERFORMANCE	**CO2**	a,b,d,e,j,k,l
5	Mini project on classification: Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix and compare these models. Also apply cross validation while preparing the training and testing datasets. For Example: Health Care Domain for predicting disease	DESIGN ,ANALYSIS & PERFORMANCE	**CO2**	a,b,d,e,j,k,l
6	Mini-Project 1: Create a small application by selecting relevant system	DESIGN ,ANALYSIS &	**CO2**	a,b,d,e,j,k,l

	environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.	PERFORMANCE		
7	Mini-Project 2: Create a small web-based application by selecting relevant system environment / platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing.	DESIGN ,ANALYSIS & PERFORMANCE	**CO2**	a,b,d,e,j,k,l

I. ORAL QUESTIONS

S.No	Question	Nature of Question	Course Outcome Addressed	Program Outcome Addressed
1		Logical & descriptive	CO1	abgijk
2		Logical & descriptive	CO1	abgijk
3		Logical & descriptive	CO1	abgijk
4		Logical & descriptive	CO1	abgijk
5		Logical & descriptive	CO1	abgijk
6		Logical & descriptive	CO1	abgijk
7		Logical & descriptive	CO1	abgijk
8		Logical & descriptive	CO1	abgijk
9		Logical & descriptive	CO2	abgijk
10		Logical & descriptive	CO2	abgijk
11		Logical & descriptive	CO2	abgijk
12		Logical &	CO2	abgijk

		descriptive		
13		Logical & descriptive	CO2	abgijk
14		Logical & descriptive	CO2	abgijk
15		Logical & descriptive	CO2	abgijk
16		Logical & descriptive	CO2	abgijk
17		Logical & descriptive	CO2	abgijk
18		Logical & descriptive	CO3	abgijk
19		Logical & descriptive	CO1	abgijk
20		Logical & descriptive	CO1	abgijk
21		Logical & descriptive	CO1	abgijk
22		Logical & descriptive	CO1	abgijk
23		Logical & descriptive	CO1	abgijk
24		Logical & descriptive	CO1	abgijk
25		Logical & descriptive	CO1	abgijk
26		Logical & descriptive	CO1	abgijk

27		**Logical & descriptive**	**CO1**	**abgijk**
28		**Logical & descriptive**	**CO1**	**abgijk**
29		**Logical & descriptive**	**CO1**	**abgijk**
30		**Logical & descriptive**	**CO1**	**abgijk**
31		**Logical & descriptive**	**CO1**	**abgijk**
32		**Logical & descriptive**	**CO2**	**abgijk**
33		**Logical & descriptive**	**CO2**	**abgijk**
34		**Logical & descriptive**	**CO2**	**abgijk**
35		**Logical & descriptive**	**CO2**	**abgijk**
36		**Logical & descriptive**	**CO2**	**abgijk**
37		**Logical & descriptive**	**CO2**	**abgijk**
38		**Logical & descriptive**	**CO2**	**abgijk**
39		**Logical & descriptive**	**CO2**	**abgijk**
40		**Logical & descriptive**	**CO2**	**abgijk**
41		**Logical & descriptive**	**CO2**	**abgijk**

42		**Logical & descriptive**	**CO2**	**abgijk**
43		**Logical & descriptive**	**CO2**	**abgijk**
44		**Logical & descriptive**	**CO2**	**abgijk**
45		**Logical & descriptive**	**CO2**	**abgijk**
46		**Logical & descriptive**	**CO2**	**abgijk**
47		**Logical & descriptive**	**CO2**	**abgijk**
48		**Logical & descriptive**	**CO2**	**abgijk**
49		**Logical & descriptive**	**CO2**	**abgijk**
50		**Logical & descriptive**	**CO2**	**abgijk**
51		**Logical & descriptive**	**CO2**	**abgijk**
52		**Logical & descriptive**	**CO2**	**abgijk**
53		**Logical & descriptive**	**CO2**	**abgijk**
54		**Logical & descriptive**	**CO2**	**abgijk**
55		**Logical & descriptive**	**CO2**	**abgijk**
56		**Logical & descriptive**	**CO2**	**abgijk**

57		**Logical & descriptive**	**CO2**	**abgijk**
58		**Logical & descriptive**	**CO2**	**abgijk**
59		**Logical & descriptive**	**CO2**	**abgijk**
60		**Logical & descriptive**	**CO2**	**abgijk**
61		**Logical & descriptive**	**CO2**	**abgijk**
62		**Logical & descriptive**	**CO2**	**abgijk**
63		**Logical & descriptive**	**CO2**	**abgijk**
64		**Logical & descriptive**	**CO2**	**abgijk**
65		**Logical & descriptive**	**CO2**	**abgijk**
66		**Logical & descriptive**	**CO2**	**abgijk**
67		**Logical & descriptive**	**CO2**	**abgijk**
68		**Logical & descriptive**	**CO2**	**abgijk**
69		**Logical & descriptive**	**CO2**	**abgijk**
70		**Logical & descriptive**	**CO2**	**abgijk**

J. PROGRESSIVE