MCA III Semester

10MC3BS01: OPERATIONS RESEARCH

UNIT –I

INTRODUCTION TO OR: Introduction, modeling in OR- Phases of OR study, Linear Programming: Formulation of LPP, Graphical solution of LPP, Simplex method, Artificial variable technique, Duality principle.

UNIT –II

TRANSPORTATION PROBLEM: Finding an initial basic feasible solution, North-West corner rule, Least cost Entry method, Vogel's Approximation Method, Degeneracy in Transportation Problem, Optimality test: Unbalanced Transportation Problem, Maximization case in Transportation Problem.

UNIT –III

ASSIGNMENT PROBLEM: Hungarian method of Assignment Problem, Maximization in Assignment Problem, Travelling salesman Problem and its restrictions.

UNIT –IV

SEQUENCING PROBLEM: Introduction, Optimal solution for processing n-jobs through two machines, n-jobs through three machines and n-jobs through k-machines, 2-jobs through k-machines.

UNIT –V

REPLACEMENT PROBLEM: Introduction, Replacement of items that deteriorate when money value is constant and variable, Replacement of items that fail suddenly, Individual Replacement policy, Group Replacement policy.

UNIT – VI


UNIT – VII

INVENTORY MODELS: Introduction, Reasons for maintaining inventories, types of inventory, Inventory costs, Deterministic inventory models: EOQ models with and without shortages, EOQ problems with one and two price breaks.
UNIT – VIII

PROJECT MANAGEMENT BY PERT/CPM: Basic steps in PERT/CPM technique, rules of drawing network diagrams, Fulkerson’s rule: Critical Path Method (CPM), Project Evaluation and Review Technique (PERT).

TEXT BOOKS:

REFERENCE BOOKS:
MCA III Semester
10MC30101: WEB APPLICATION DEVELOPMENT

UNIT –I

HTML COMMON TAGS: List, tables, images, forms, frames, Cascading Style Sheets. Introduction to Java Script, Objects in Java Script, Dynamic HTML with Java Script.

UNIT –II


UNIT –III

JDBC : Introduction to JDBC, JDBC and ODBC,Types of Drivers, Java SQL Package, Using a JDBC, Driver Manager, Connection Interface, Statement Interface, PreparedStatement Interface, CallableStatement Interface, ResultSet Interface.

UNIT –IV

SERVLETS: Introduction to Servlets, Servlet Technology, Handling HTTP GET Requests, Handling HTTP POST Requests, Session Tracking. Multi-tier Applications: Using JDBC from a Servlet

UNIT –V


UNIT –VI

JSP APPLICATION DEVELOPMENT: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging, Sharing Data between JSP pages, Requests, and Users Passing Control and Date between Pages, Sharing Session and Application Data, Accessing a Database from a JSP Page, Application.

UNIT – VII

STRUTS APPLICATION DEVELOPMENT: Introducing the Jakarta Struts Project and its Supporting Components, Getting started with Struts, Actions and ActionServlet, Advanced Action Classes: ForwardAction and Beyond, Include Action, Dispatch Action.
UNIT – VIII


TEXT BOOKS:

REFERENCE BOOKS:
UNIT-I

UNIT-II
**PHYSICAL LAYER**: Analog and Digital, Periodic Analog and Digital signals, Multiplexing, Guided transmission media, Wireless Transmission, Communication satellites, Public switched telephone Network.

UNIT-III

UNIT-IV

UNIT-V

UNIT-VI
**THE APPLICATION LAYER**: DNS(Domain Name System), Electronic Mail, World Wide Web, Multimedia.

UNIT-VII
**NETWORK SECURITY-1**: Cryptography: Substitution ciphers, Transposition ciphers, one time pads, Security services, Symmetric key Algorithms, Public key Algorithms, Digital signatures, Management of Public keys.

UNIT-VIII
TEXT BOOKS:

REFERENCE BOOKS:
MCA III Semester
10MC30103: DATA WAREHOUSING AND DATA MINING

UNIT-I
INTRODUCTION TO DATA WAREHOUSING: Need for Data Warehousing, Operational and Informational Data Stores, Data Warehouse Definition and Characteristics. Data Warehouse Architecture: Steps for the design and construction of Data Warehouse, Three-Tier Architecture. Data Warehouse Components: Sourcing, Acquisition, cleanup and transformation tools, Metadata and Data mart.

UNIT-II

UNIT-III
OLAP: Need for OLAP, OLAP Guidelines, OLAP Operations, From Data Warehousing to Data Mining. Introduction to Data Mining: Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives and Major issues in Data Mining.

UNIT-IV
DATA PREPROCESSING: Need for Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation, Further Development of Data Cube and OLAP Technology and Attribute-Oriented Induction.

UNIT-V
MINING FREQUENT PATTERNS, ASSOCIATIONS AND CORRELATIONS: Basic Concepts, Efficient and Scalable Frequent Itemset Mining Methods, Mining various kinds of Association Rules, From Association Mining to Correlation Analysis and Constraint-Based Association Mining.

UNIT-VI
CLASSIFICATION AND PREDICTION: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Backpropagation, Associative Classification, Prediction, Accuracy and Error measures, Evaluating the accuracy of a Classifier or a Predictor and Ensemble Methods.

SVEC10 - Master of Computer Applications
UNIT-VII

CLUSTER ANALYSIS INTRODUCTION: Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods. Mining streams and Time Data: Mining Data Streams and Time-Series Data.

UNIT-VIII


TEXT BOOKS:


REFERENCE BOOKS:

3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Introduction to Data Mining, Pearson Education.
MCA III Semester

10MC30104: SOFTWARE ENGINEERING

UNIT-I

UNIT-II

UNIT-III

UNIT-IV

UNIT-V

UNIT-VI
SOFTWARE TESTING STRATEGIES: A strategic approach to software testing, Test strategies (Unit testing and integration testing) for conventional software, Validation testing, System testing and the art of debugging.
UNIT-VII

TESTING CONVENTIONAL APPLICATIONS: Software testing fundamentals, White-Box testing, Basis path testing: flow graph notation, independent program paths, deriving test cases, graph matrices. Control structure testing: condition testing, data flow testing, loop testing, Black box testing: Graph based testing method, Equivalence partitioning and Boundary value analysis. Testing Object Oriented Applications: OO testing methods, testing methods applicable at class level and Interclass test case design.

UNIT-VIII

RISK MANAGEMENT: Reactive versus Proactive Risk strategies, software risks, risk identification, risk projection, risk refinement, risk mitigation, monitoring and management. Maintainanace and Reengineering: Software maintenance, software supportability, reengineering, software reengineering, reverse engineering and forward engineering.

TEXT BOOK:

REFERENCE BOOKS:
Week-1:
1. Develop static pages of an online Book Store by only using HTML (the pages should resemble: www.amazon.com). The website should consist the following pages.
   a. Home Page
   b. Registration and User Login
   c. Books Catalog

Week-2:
2. Design and develop a Feedback form using JavaScript.
3. Validate the Registration and User Login pages of program – 1.

Week-3:
4. Create a web page with all types of Cascading style sheets.
5. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

Week-4:
6. Programs using XML Schema, XSLT/XSL

Week-5:
7. Program using DOM / SAX.

Week-6:
8. a. Write a java program to get IP Address, Host Name and Port Numbers of a Local system
    b. Write a Java Program using Datagram Communication.

Week-7:
9. Write a basic Servlet program that must display information like
   a. Request method used by the client and
   b. Current system date

Week-8:
10. Generate a Servlet, that will accept form data from HTML file which includes Name of the user and favorite programming language described in terms of a LIST, and it must display user name and language selected by a particular Client.

Week-9:
11. Generate a Servlet page that showing the Session ID, Creation Time, Last Accessed Time and Max Inactive Interval. Along with page should display links like Reload this page and Invalidate the Session. When we press the Reload this page link the above information should be redisplayed. If we press Invalidate the Session link the page will display one more link like create new session along with it has to display message like session invalidated.
Week-10:
12. Generate JSP pages to
   a. Current system date
   b. Page should include two files (either html or jsp files) by using include directive.
   c. Page should include two files (either html or jsp files) by using include action.
   d. Any mathematical table by using Scriptlet Elements (Declaration, Expression and Scriptlet tags).
   e. Page must perform forward action.

Week-11:

Week-12:
14. Write a JSP program for finding total number of visitors in a site to keep track of active users at a given instance of time, and also display the user session starting time.

Week-13:
15. Write a JSP program that creates a cookie on username which is send from html file and display the cookie value as a response. The cookie must be active based on the maximum active interval time.

Week-14:
16. Develop java program for following SQL operations using JDBC.
   a. Create
   b. Insert
   c. Update and
   d. Delete
   Consider the following schema:
   Employee (EmpName, EmpNo Primary Key, Department, Salary)
17. Generate a JSP page that will retrieve the Employee information from the database. The page should display the employee records in a tabular format.

Week-15:
18. Convert static web pages of program – 1 to dynamic web pages, and make necessary database connections to user registration page, login page, and books catalog page. The books catalog should be dynamically loaded from the database.

Week-16:
19. Write programs in Java to create three-tier applications using JSP and Databases
   • For conducting on-line examination.
   • For displaying student mark list. Assume that student information is available in a Database which has been stored in a database server.
MCA III Semester

10MC30106: DATA WAREHOUSING AND DATA MINING LAB

L T P C
- - 3 2

Week 1:

Hospital Management System:
Creation of Data Warehouse Which consists Dimension Table and Fact Table.

When you create a definition for a hierarchy, Warehouse Builder creates an identifier key for each level of the hierarchy and a unique key constraint on the lowest level (Base Level).

Design a Hospital Management system data warehouse (TARGET) consists of Dimensions Patient, Medicine, Supplier, Time. Where measures are 'NO UNITS', UNIT PRICE.

Assume the Relational database (SOURCE) table schemas as follows:
TIME (day, month, year),
PATIENT (patient_name, Age, Address, etc.,)
MEDICINE (Medicine_Brand_name, Drug_name, Supplier, no_units, Uinit_Price, etc.,)
SUPPLIER :(Supplier_name, Medicine_Brand_name, Address, etc.,)

The following Dimension:
The dimension object (Dimension):
  _ Name
  _ Attributes (Levels), with one primary key
  _ Hierarchies

One time dimension is must.

Week 2:

Creation of various Levels of Hierarchies:
If each Dimension has 6 levels, decide the levels and hierarchies, Assume the level names suitably.

Dimension objects (dimension) consist of a set of levels and a set of hierarchies defined over those levels. The levels represent levels of aggregation. Hierarchies describe parent-child relationships among a set of levels.

For example, a typical calendar dimension could contain five levels. Two hierarchies can be defined on these levels:

H1: YearL > QuarterL > MonthL > WeekL > DayL
H2: YearL > WeekL > DayL

The hierarchies are described from parent to child, so that Year is the parent of Quarter, Quarter the parent of Month, and so forth.

Week 3:

Design the Hospital Management system data warehouse using all schemas.
**Week 4:**
Use an ETL tool to construct a data acquisition process (mapping) to perform the transformations from one data source to another data source as follows:
- Creation of Simple mapping transformations from source database to target database
- Creation of Filter transformations

**Week 5:**
- Design mapping of source and target data source with Expression transformation
- Construct Joiner Transformation for heterogeneous data sources
- Creation of Filter Transformation for various data sources
- Design mapping of source and target data source with Rank Transformation

**Week 6:**
- Construct Router Transformation for heterogeneous data sources
- Creation of Joiner Transformation for various data sources without source qualifier
- Design mapping of source and target data source with Aggregator Transformation

**Week 7:**
- Design mapping of source and target data source with sorter transformation
- Construct UNION Transformation for heterogeneous data sources
- Creation of Lookup Transformation for various data sources

**Week 8: Credit Risk Assessment:**

**Description:**
The business of banks is making loans. Assessing the creditworthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the bank's profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.
1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.

2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.

3. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.

4. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

Week 9:
The German Credit Data:
Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. credit dataset (original) Excel spreadsheet version of the German credit data.
In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer !)

A few notes on the German dataset
- DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
- owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
- foreign_worker. There are millions of these in Germany (many from Turkey). It is very hard to get German citizenship if you were not born of German parents.
- There are 20 attributes used in judging a loan applicant. The goal is to classify the applicant into one of two categories, good or bad.

Subtasks : (Turn in your answers to the following tasks)
- List all the categorical (or nominal) attributes and the real-valued attributes separately. (5 marks)
- What attributes do you think might be crucial in making the credit assessment ? Come up with some simple rules in plain English using your selected attributes. (5 marks)
- One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training. (10 marks)
- Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly ? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy ? (10 marks)
Week 10:
a) Is testing on the training set as you did above a good idea? Why or Why not? (10 marks)
b) One approach for solving the problem encountered in the previous question is using cross-validation. Describe what is cross-validation briefly. Train a Decision Tree again using cross-validation and report your results. Does your accuracy increase/decrease? Why? (10 marks)
c) Check to see if the data shows a bias against “foreign workers” (attribute 20), or “personal-status” (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka’s GUI Explorer. Did removing these attributes have any significant effect? Discuss. (10 marks)
d) Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.) (10 marks)

Week 11:
a) Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifications equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)? (10 marks)
b) Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees? How does the complexity of a Decision Tree relate to the bias of the model? (10 marks)
Week 12:
a) You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain. Also, report your accuracy using the pruned model. Does your accuracy increase? (10 marks)
b) (Extra Credit): How can you convert a Decision Trees into "if-then-else rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules.PART. Train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one! Can you predict what attribute that might be in this dataset? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a oneR classifier. Rank the performance of j48, PART and oneR. (10 marks)

Week 13:
- Mentor lecture on Decision Trees
  - Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)
  - Decision Trees (Source: Tan, MSU)
  - Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)
  - Weka resources:
    - Introduction to Weka (html version) (download ppt version)
    - Download Weka
    - Weka Tutorial
    - ARFF format
    - Using Weka from command line

Week 14:
Creation of dataset in Attribute Relation File Format (.ARFF) or Comma separator values (.csv)
Creation of dataset in Performing Preprocessing using ID3 classifier, Naïve Bayes Classifier, IBK classification based on cross validation and use training set text modes

Week 15:
Experiment on dataset using simple knowledge flow for normalizing dataset.

Week 16:
Simple knowledge flow for implementing and evaluating ID3 classifier based on training set and test set on gain ratio and ranker attribute selection method.