1. Apply data cleaning techniques on any dataset (e,g, wine dataset). Techniques may include handling missing values, outliers, inconsistent values. A set of validation rules can be prepared based on the dataset and validations can be performed.
2. Apply data pre-processing techniques such as standardization / normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset
3. Run Apriori algorithm to find frequent itemsets and association rules on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns
   a) Use minimum support as 50% and minimum confidence as 75%
   b) Use minimum support as 60% and minimum confidence as 60 %

4. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations:
   I. a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3%
   II. Training set is chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained.

   Data is scaled to standard format.
5. Use Simple K-means algorithm for clustering on any dataset. Compare the performance of clusters by changing the parameters involved in the algorithm. Plot MSE computed after each iteration using a line plot for any set of parameters.

Project: Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Combinatorial Optimization

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Combinatorial Optimization	**4**	**3**	**1**	**0**	Pass in Class XII	NIL

Learning Objectives

This course is designed to introduce the fundamentals of combinatorial optimization to the students in terms of both theory and applications, so as to equip them to explore the more advanced areas of convex and non-convex optimizations.

Learning outcomes

On successful completion of the course, students will be able to:

- Model problems using linear and integer programs
- Apply polyhedral analysis to develop algorithms for optimization problems
- Use the concept of duality for design of algorithms

SYLLABUS OF DSE

Unit 1 (9 hours)

Introduction: Introduction to Combinatorial Optimization Problems, Linear and Integer Programs- LP Formulation, understanding integer programs, computational complexities of IP vs LP, using LP to find optimal or approximate integral solutions, concept of integrality gap.

Unit 2 (14 hours)

Theory of Linear Programming and Algorithmic Perspective to Simplex Method: standard vs. equational form, basic feasible solutions, convexity and convex polyhedra, correspondence between vertices and basic feasible solutions, geometry of Simplex algorithm, exception handling (unboundedness, degeneracy, infeasibility), Simplex algorithm, avoiding cycles.

Unit 3 (12 hours)

Primal-Dual Algorithms: interpretation of dual, optimality conditions for primal and dual, weak and strong duality, complementary slackness, primal-dual algorithm for the shortest path problem.

Unit 4 (10 hours)

Network Flows: linear programming formulations for network flows and bipartite matching, totally unimodular matrices.

Essential/recommended readings

2. Papadimitriou, C.H. & Steiglitz, K. *Combinatorial Optimization: Algorithms and complexity*, New edition, Dover Publications inc., 2000.

Additional References

(i) Bazaraa, M.S., Jarvis, J.J., & and Sherali, H.D. *Linear Programming and Network Flows*, 4th edition, Wiley, 2010.

(ii) Korte, B., & Vygen, J. *Combinatorial Optimization,* 6th edition, Springer, 2018.

Tutorials

Tutorials based on Theory