



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

## **COURSE STRUCTURE AND SYLLABUS**

**For UG – R20**

**B. Tech - COMPUTER SCIENCE AND ENGINEERING with Specialization**

**ARTIFICIAL INTELLIGENCE**

*(Applicable for batches admitted from 2020-2021)*



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**

**KAKINADA - 533 003, Andhra Pradesh, India**



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B. Tech – I Semester</b>						
S.No	Course Code	Courses	Hours per week			Credits
			L	T	P	C
1	PC	Compiler Design	3	0	0	3
2	PC	Operating Systems	3	0	0	3
3	PC	Machine Learning	3	0	0	3
4	Open Elective/Job Oriented	<b>Open Elective-I</b> Open Electives offered by other departments/ Optimization in Operations Research(Job oriented course)	3	0	0	3
5	PE	<b>Professional Elective-I</b> 1. Software Engineering 2. Computer Vision 3. Data Visualization 4.DevOps	3	0	0	3
6	PC	Operating Systems & Compiler Design Lab	0	0	3	1.5
7	PC	Machine Learning Lab	0	0	3	1.5
8	SO	<b>Skill Oriented Course - III</b> Continuous Integration and Continuous Delivery using DevOps	0	0	4	2
9	MC	Employability Skills-I	2	0	0	0
10	PR	<b>Summer Internship 2 Months(Mandatory) after second year(to be evaluated during V semester</b>	0	0	0	1.5
<b>Total credits</b>						<b>21.5</b>
11	Minor	Machine Learning <sup>\$</sup>	3	0	2	<b>4</b>

\$- Integrated Course



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>COMPILER DESIGN</b>					

**Course Objectives:**

Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.

**Course Outcomes:** At the end of the course, student will be able to

- Demonstrate phases in the design of compiler
- Organize Syntax Analysis, Top Down and LL(1) grammars
- Design Bottom Up Parsing and Construction of LR parsers
- Analyze synthesized, inherited attributes and syntax directed translation schemes
- Determine algorithms to generate code for a target machine

**UNIT I:**

**Lexical Analysis:** Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Bootstrapping, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Finite Automata, Regular Expressions and Finite Automata, Design of a Lexical Analyzer Generator.

**UNIT II:**

**Syntax Analysis:** The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring, **Top Down Parsing:** Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing, Error Recovery in Predictive Parsing.

**UNIT III:**

**Bottom Up Parsing:** Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing, SLR Parsers, Construction of SLR Parsing Tables, More Powerful LR Parsers, Construction of CLR (1) and LALR Parsing Tables, Dangling Else Ambiguity, Error Recovery in LR Parsing, Handling Ambiguity Grammar with LR Parsers.

**UNIT III:**

**Syntax Directed Translation:** Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's. **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Intermediate Code for Procedures.

**UNIT IV:**

**Run Time Environments:** Storage Organization, Run Time Storage Allocation, Activation Records, Procedure Calls, Displays, **Code Optimization:** The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Structure Preserving Transformations, Flow Graphs,



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

Loop Optimization, Data-Flow Analysis, Peephole Optimization

**UNIT V: Code Generation:** Issues in the Design of a Code Generator, Object Code Forms, Code Generation Algorithm, Register Allocation and Assignment.

**Text Books:**

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson Publishers, 2007.

**Reference Books:**

1. Compiler Construction, Principles and Practice, Kenneth C Louden, Cengage Learning, 2006
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. Optimizing Compilers for Modern Architectures, Randy Allen, Ken Kennedy, Morgan Kaufmann, 2001.
4. Levine, J.R., T. Mason and D. Brown, Lex and Yacc, edition, O'Reilly & Associates, 1990



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>OPERATING SYSTEM</b>					

**Course Objectives:**

The objectives of this course is to

- Introduce to the internal operation of modern operating systems
- Define, explain, processes and threads, mutual exclusion, CPU scheduling, deadlock, memory management, and file systems
- Understand File Systems in Operating System like UNIX/Linux and Windows
- Understand Input Output Management and use of Device Driver and Secondary Storage (Disk) Mechanism
- Analyze Security and Protection Mechanism in Operating System

**Course Outcomes:**

After learning, the course the students should be able to:

- Describe various generations of Operating System and functions of Operating System
- Describe the concept of program, process and thread and analyze various CPU Scheduling Algorithms and compare their performance
- Solve Inter Process Communication problems using Mathematical Equations by various methods
- Compare various Memory Management Schemes especially paging and Segmentation in Operating System and apply various Page Replacement Techniques
- Outline File Systems in Operating System like UNIX/Linux and Windows

**UNIT I:**

Operating Systems Overview: Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems.

System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

**UNIT II:**

Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems.

Multithreaded Programming: Multithreading models, Thread libraries, Threading issues. Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling.

Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems - Dining philosophers problem, Readers and writers problem.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**UNIT III:**

Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation.

Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.

**UNIT IV:**

Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention.

File Systems: Files, Directories, File system implementation, management and optimization.

Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.

**UNIT V:**

System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights.

System Security: Introduction, Program threats, System and network threats, Cryptography for security, User authentication, implementing security defenses, Firewalling to protect systems and networks, Computer security classification.

Case Studies: Linux, Microsoft Windows.

**Text Books:**

1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9th edition, Wiley, 2013.
2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008. (for Interprocess Communication and File systems.)

**Reference Books:**

1. Dhamdhare D M, Operating Systems A Concept Based Approach, 3rd edition, Tata McGraw-Hill, 2012.
2. Stallings W, Operating Systems -Internals and Design Principles, 6th edition, Pearson Education, 2009
3. Nutt G, Operating Systems, 3rd edition, Pearson Education, 2004.

**e-Resources:**

- 1) <https://nptel.ac.in/courses/106/105/106105214/>



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>MACHINE LEARNING</b>					

**Course Objectives:**

- Identify problems that are amenable to solution by ANN methods, and which ML methods may be suited to solving a given problem.
- Formalize a given problem in the language/framework of different ANN methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, as a Markov decision process, etc).

**Course Outcomes:** After the completion of the course, student will be able to

- Explain the fundamental usage of the concept Machine Learning system
- Demonstrate on various regression Technique
- Analyze the Ensemble Learning Methods
- Illustrate the Clustering Techniques and Dimensionality Reduction Models in Machine Learning.
- Discuss the Neural Network Models and Fundamentals concepts of Deep Learning

**Unit I:**

**Introduction-** Artificial Intelligence, Machine Learning, Deep learning, Types of Machine Learning Systems, Main Challenges of Machine Learning.

**Statistical Learning:** Introduction, Supervised and Unsupervised Learning, Training and Test Loss, Tradeoffs in Statistical Learning, Estimating Risk Statistics, Sampling distribution of an estimator, Empirical Risk Minimization.

**Unit II:**

**Supervised Learning**(Regression/Classification):Basic Methods: Distance based Methods, Nearest Neighbours, Decision Trees, Naive Bayes, **Linear Models:** Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, **Binary Classification:** Multiclass/Structured outputs, MNIST, Ranking.

**Unit III:**

**Ensemble Learning and Random Forests:** Introduction, Voting Classifiers, Bagging and Pasting,Random Forests, Boosting, Stacking.

**Support Vector Machine:** Linear SVM Classification, Nonlinear SVM ClassificationSVM Regression, Naïve Bayes Classifiers.

**Unit IV:**

**Unsupervised Learning Techniques:**Clustering, K-Means, Limits of K-Means, Using Clustering for Image Segmentation, Using Clustering for Preprocessing, Using Clustering for Semi-Supervised Learning, DBSCAN, Gaussian Mixtures.

**Dimensionality Reduction:** The Curse of Dimensionality, Main Approaches for Dimensionality Reduction, PCA, Using Scikit-Learn, Randomized PCA, Kernel PCA.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**Unit V:**

**Neural Networks and Deep Learning:** Introduction to Artificial Neural Networks with Keras, Implementing MLPs with Keras, Installing TensorFlow 2, Loading and Preprocessing Data with TensorFlow.

**Text Books:**

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, O'Reilly Publications, 2019
2. Data Science and Machine Learning Mathematical and Statistical Methods, Dirk P. Kroese, Zdravko I. Botev, Thomas Taimre, Radislav Vaisman, 25th November 2020

**Reference Books:**

1. Machine Learning Probabilistic Approach, Kevin P. Murphy, MIT Press, 2012.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>OPTIMIZATION IN OPERATIONS RESEARCH</b>					
<b>(Job oriented course)</b>					

**Course Objectives:**

- To define an objective function and constraint functions in terms of design variables, and then state the optimization problem.
- To state single variable and multi variable optimization problems, without and with constraints.
- To explain linear programming technique to an optimization problem, define slack and surplus variables, by using Simplex method.
- To state transportation and assignment problem as a linear programming problem to determine Simplex method.
- To study and explain nonlinear programming techniques, unconstrained or constrained, and define exterior and interior penalty functions for optimization problems.

**Course Outcomes:** At the end of the course, student will be able to

- State and formulate the optimization problem, without and with constraints, by using design variables from an engineering design problem.
- Apply classical optimization techniques to minimize or maximize a multi-variable objective function, without or with constraints, and arrive at an optimal solution.
- Apply and Solve transportation and assignment problem by using Linear programming Simplex method.
- Apply gradient and non-gradient methods to nonlinear optimization problems and use interior or exterior penalty functions for the constraints to derive the optimal solutions
- Formulate and apply Dynamic programming technique to inventory control, production planning, engineering design problems etc. to reach a final optimal solution from the current optimal solution.

**UNIT I:**

**Introduction and Classical Optimization Techniques:** Statement of an Optimization problem, design vector, design constraints, constraint surface, objective function, objective function surfaces, classification of Optimization problems.

**Classical Optimization Techniques:** Single variable Optimization, multi variable Optimization without constraints, necessary and sufficient conditions for minimum/maximum, multivariable Optimization with equality constraints. Solution by method of Lagrange multipliers, multivariable Optimization with inequality constraints, Kuhn – Tucker conditions

**UNIT II:**

**Linear Programming :** Standard form of a linear programming problem, geometry of linear programming problems, definitions and theorems, solution of a system of linear simultaneous equations, pivotal reduction of a general system of equations, motivation to the simplex method, simplex algorithm, Duality in Linear Programming, Dual Simplex method.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**UNIT III:**

**Transportation Problem:** Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel’s approximation method, testing for optimality of balanced transportation problems, Special cases in transportation problem.

**UNIT IV:**

**Nonlinear Programming:** Unconstrained cases, One – dimensional minimization methods: Classification, Fibonacci method and Quadratic interpolation method, Univariate method, Powell’s method and steepest descent method.

Constrained cases– Characteristics of a constrained problem, Classification, Basic approach of Penalty Function method; Basic approaches of Interior and Exterior penalty function methods, Introduction to convex Programming Problem.

**UNIT V:**

**Dynamic Programming:** Dynamic programming multistage decision processes, types, concept of sub optimization and the principle of optimality, computational procedure in dynamic programming, examples illustrating the calculus method of solution, examples illustrating the tabular method of solution.

**Text Books:**

1. “Engineering optimization: Theory and practice”, S. S.Rao, New Age International (P) Limited, 3<sup>rd</sup> edition, 1998.
2. “Introductory Operations Research”, H.S. Kasene& K.D. Kumar, Springer (India), Pvt. LTd.

**Reference Books:**

1. “Optimization Methods in Operations Research and systems Analysis”, by K.V. Mital and C. Mohan, New Age International (P) Limited, Publishers, 3rd edition, 1996.
2. Operations Research, Dr. S.D.Sharma, Kedarnath, Ramnath& Co



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>SOFTWARE ENGINEERING</b> <b>(Professional Elective-I)</b>					

**Course Objectives:**

This course is designed to:

- Give exposure to phases of Software Development, common process models including Waterfall, and the Unified Process, and hands-on experience with elements of the agile process
- Give exposure to a variety of Software Engineering practices such as requirements analysis and specification, code analysis, code debugging, testing, traceability, and version control
- Give exposure to Software Design techniques

**Course Outcomes:**

Students taking this subject will gain software engineering skills in the following areas:

- Ability to transform an Object-Oriented Design into high quality, executable code
- Skills to design, implement, and execute test cases at the Unit and Integration level
- Compare conventional and agile software methods

**UNIT I:**

The Nature of Software, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths. A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology.

**UNIT II:**

Agility, Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, A Tool Set for the Agile Process, Software Engineering Knowledge, Core Principles, Principles That Guide Each Framework Activity, Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements.

**UNIT III:**

Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling, Requirements Modeling Strategies, Flow-Oriented Modeling, Creating a Behavioral Model, Patterns for Requirements Modelling, Requirements Modeling for WebApps.

**UNIT IV:**

Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model, Software Architecture, Architectural Genres, Architectural Styles  
 Assessing Alternative Architectural Designs, Architectural Mapping Using Data Flow, Components, Designing Class-Based Components, Conducting Component-Level Design,



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

Component-Level Design for WebApps, Designing Traditional Components, Component- Based Development.

**UNIT V**

The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebApp Interface Design, Design Evaluation, Elements of Software Quality Assurance, SQA Tasks, Goals & Metrics, Statistical SQA, Software Reliability, A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System Testing, The Art of Debugging, Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing.

**Text Books:**

1. Software Engineering a practitioner's approach, Roger S. Pressman, Seventh Edition, McGraw Hill Higher Education.
2. Software Engineering, Ian Sommerville, Ninth Edition, Pearson.

**Reference Books:**

1. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
2. Software Engineering, Ugrasen Suman, Cengage.

**e-Resources:**

- 1) <https://nptel.ac.in/courses/106/105/106105182/>



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>COMPUTER VISION (Professional Elective-I)</b>					

**Course Objectives:**

To introduce students the fundamentals of image formation; To introduce students the major ideas, methods, and techniques of computer vision and pattern recognition; To develop an appreciation for various issues in the design of computer vision and object recognition systems; and To provide the student with programming experience from implementing computer vision and object recognition applications.

**Course Outcomes:** After completing the course Student will be able to:

1. Identify basic concepts, terminology, theories, models and methods in the field of computer vision,
2. Describe known principles of feature detection and matching,
3. Describe basic methods of computer vision related to image stitching, photography like high dynamic range imaging and blur removal.
4. Suggest a design of a computer vision system for a 3D Reconstruction, Albedos, image based rendering views and depths.

**UNIT I:**

**Introduction:** Image Formation: Geometric Primitives and Transformation, Photometric Image Formation, Digital Camera, Image Processing: Point Operators, Linear Filtering, More Neighborhood Operators, Fourier Transforms, Pyramids and Wavelets, Geometric Transformations, Global Optimization.

**UNIT II:**

**Feature Detection and Matching:** Points and Patches, Edges, Lines, Segmentation: Active Contours, Split and Merge, Mean Shift and Mode Finding, Normalized Cuts, Feature-Based Alignment: 2D and 3D Feature-based Alignment, Pose Estimation, Geometric Intrinsic Calibration.

**UNIT III:**

**Structure and Motion:** Triangular, Two-frame Structure from Motion, Factorization, Bundle Adjustment, Constrained Structure and Motion, Dense Motion Estimation: Translation Alignment, Parametric Motion, Spline-based Motion, Optical Flow, Layered motion

**UNIT IV:**

**Image Stitching:** Motion Models, Global Alignment, Composing, Computational Photography: Photometric Calibration, High Dynamic Range Imaging, Super-Resolution and Blur Removal, image Matting and Compositing, Texture Analysis and Synthesis.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**UNIT V:**

**3D Reconstruction:** Shape From X, Active Range Finding, Surface Representation, Point-based Representation, Volumetric Representation, Model-based Reconstruction, Recovering Texture Maps and Albedos, Image- based Rendering: View Interpolation, Layered Depth Images, Light Fields and Lumigraphs, Environment Mattes, Video-based Rendering.

**Text Books:**

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited, 2011.
2. Simon J.D Prince, Computer Vision: Models, Learning and Inference, 1<sup>st</sup> Edition, 2012.

**Reference Books:**

1. Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall Robot Vision, by B. K. P. Horn, McGraw-Hill.
2. Haralick & Shapiro, “Computer and Robot Vision”, Vol II
3. G\_erard Medioni and Sing Bing Kang “Emerging topics in computer vision”95

**NPTEL LINK:** [https://onlinecourses.nptel.ac.in/noc22\\_ee48/preview](https://onlinecourses.nptel.ac.in/noc22_ee48/preview)



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>DATA VISUALIZATION</b> <b>(Professional Elective-I)</b>					

**Course Outcomes:**

Upon completion of this course, the students will be able to

- Understand basics of Data Visualization
- Implement visualization of distributions
- Write programs on visualization of time series, proportions & associations
- Apply visualization on Trends and uncertainty
- Explain principles of proportions

**UNIT I:**

**INTRODUCTION TO VISUALIZATION:** Visualizing Data-Mapping Data onto Aesthetics, Aesthetics and Types of Data, Scales Map Data Values onto Aesthetics, Coordinate Systems and Axes- Cartesian Coordinates, Nonlinear Axes, Coordinate Systems with Curved Axes, Color Scales-Color as a Tool to Distinguish, Color to Represent Data Values, Color as a Tool to Highlight, Directory of Visualizations- Amounts, Distributions, Proportions, x-y relationships, Geospatial Data

**UNIT II:**

**VISUALIZING DISTRIBUTIONS:** Visualizing Amounts-Bar Plots, Grouped and Stacked Bars, Dot Plots and Heatmaps, Visualizing Distributions: Histograms and Density Plots- Visualizing a Single Distribution, Visualizing Multiple Distributions at the Same Time, Visualizing Distributions: Empirical Cumulative Distribution Functions and Q-Q Plots-Empirical Cumulative Distribution Functions, Highly Skewed Distributions, Quantile Plots, Visualizing Many Distributions at Once-Visualizing Distributions Along the Vertical Axis, Visualizing Distributions Along the Horizontal Axis

**UNIT III:**

**VISUALIZING ASSOCIATIONS & TIME SERIES:** Visualizing Proportions-A Case for Pie Charts, A Case for Side-by-Side Bars, A Case for Stacked Bars and Stacked Densities, Visualizing Proportions Separately as Parts of the Total, Visualizing Nested Proportions- Nested Proportions Gone Wrong, Mosaic Plots and Treemaps, Nested Pies, Parallel Sets. Visualizing Associations Among Two or More Quantitative Variables-Scatterplots, Correlograms, Dimension Reduction, Paired Data. Visualizing Time Series and Other Functions of an Independent Variable-Individual Time Series, Multiple Time Series and Dose-Response Curves, Time Series of Two or More Response Variables

**UNIT IV:**

**VISUALIZING UNCERTAINTY:** Visualizing Trends-Smoothing, Showing Trends with a Defined Functional Form, Detrending and Time-Series Decomposition, Visualizing Geospatial Data-Projections, Layers, Choropleth Mapping, Cartograms, Visualizing Uncertainty-Framing



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

Probabilities as Frequencies, Visualizing the Uncertainty of Point Estimates, Visualizing the Uncertainty of Curve Fits, Hypothetical Outcome Plots

**UNIT V:**

**PRINCIPLE OF PROPORTIONAL INK:** The Principle of Proportional Ink-Visualizations Along Linear Axes, Visualizations Along Logarithmic Axes, Direct Area Visualizations, Handling Overlapping Points-Partial Transparency and Jittering, 2D Histograms, Contour Lines, Common Pitfalls of Color Use-Encoding Too Much or Irrelevant Information, Using Nonmonotonic Color Scales to Encode Data Values, Not Designing for Color-Vision Deficiency

**Text Books:**

1. Claus Wilke, “Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures”, 1st edition, O’Reilly Media Inc, 2019.
2. Ossama Embarak, Data Analysis and Visualization Using Python: Analyze Data to Create Visualizations for BI Systems, Apress, 2018

**Reference Books:**

1. Tony Fischetti, Brett Lantz, R: Data Analysis and Visualization, O’Reilly, 2016



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>
<b>DevOps (Professional Elective-I)</b>					

**Course Objectives:**

- Introduces the basic concepts of Information System.
- To understand The Management Control Framework and The Application Control Framework.

**Course Outcomes:** At the end of the course, student will be able to

- Enumerate the principles of continuous development and deployment, automation of configuration management, inter-team collaboration, and IT service agility.
- Describe DevOps & DevSecOps methodologies and their key concepts
- Illustrate the types of version control systems, continuous integration tools, continuous monitoring tools, and cloud models
- Set up complete private infrastructure using version control systems and CI/CD tools
- Acquire the knowledge of maturity model, Maturity Assessment

**UNIT I:**

Phases of Software Development Life Cycle, Values and principles of agile software development.

**UNIT II:**

Fundamentals of DevOps: Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps eco system.

**UNIT III:**

DevOps adoption in projects: Technology aspects, Agiling capabilities, Tool stack implementation, People aspect, processes

**UNIT IV:**

CI/CD: Introduction to Continuous Integration, Continuous Delivery and Deployment, Benefits of CI/CD, Metrics to track CICD practices

**UNIT V:**

Devops Maturity Model: Key factors of DevOps maturity model, stages of Devops maturity model, DevOps maturity Assessment

**Text Books:**

1. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim , John Willis , Patrick Debois , Jez Humb,1st Edition, O'Reilly publications, 2016.
2. What is Devops? Infrastructure as code, 1st Edition, Mike Loukides ,O'Reilly publications, 2012.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>OPERATING SYSTEMS &amp; COMPILER DESIGN LAB</b>					

**Course Objectives:**

The main objective of this course is to implement operating systems and compiler design concept

**Course Outcomes:**

By the end of the course, student will be able to

- Implement various scheduling, page replacement algorithms and algorithms related to deadlocks
- Design programs for shared memory management and semaphores
- Determine predictive parsing table for a CFG
- Apply Lex and Yacc tools
- Examine LR parser and generating SLR Parsing table

**List of Experiments:**

1. Simulate the following CPU scheduling algorithms:
  - a. Round Robin (b) SJF (c) FCFS (d) Priority
2. Simulate the following:
  - a) Multiprogramming with a fixed number of tasks (MFT)
  - b) Multiprogramming with a variable number of tasks (MVT)
3. Simulate the following page replacement algorithms:
  - a) FIFO b) LRU c) LFU
4. Write a C program that illustrates two processes communicating using shared memory
5. Write a C program to simulate producer and consumer problem using semaphores
6. Simulate Bankers Algorithm for Dead Lock Avoidance
7. Simulate Bankers Algorithm for Dead Lock Prevention.
8. Write a C program to identify different types of Tokens in a given Program.
9. Write a Lex Program to implement a Lexical Analyzer using Lex tool.
10. Write a C program to Simulate Lexical Analyzer to validating a given input String.
11. Write a C program to implement the Brute force technique of Top down Parsing.
12. Write a C program to implement a Recursive Descent Parser.
13. Write C program to compute the *First* and *Follow* Sets for the given Grammar.
14. Write a C program for eliminating the left recursion and left factoring of a given grammar
15. Write a C program to check the validity of input string using Predictive Parser.
16. Write a C program for implementation of LR parsing algorithm to accept a given input string.
17. Write a C program for implementation of a Shift Reduce Parser using Stack Data Structure to accept a given input string of a given grammar
18. Simulate the calculator using LEX and YACC tool.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>
<b>MACHINE LEARNING LAB</b>					

**Course Objectives:**

This course will enable students to learn and understand different Data sets in implementing the machine learning algorithms.

**Course Outcomes (Cos):** At the end of the course, student will be able to

- Implement procedures for the machine learning algorithms
- Design and Develop Python programs for various Learning algorithms
- Apply appropriate data sets to the Machine Learning algorithms
- Develop Machine Learning algorithms to solve real world problems

**Requirements:** Develop the following program using Anaconda/ Jupiter/ Spider and evaluate ML models.

**Experiment-1:**

Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

**Experiment-2:**

For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

**Experiment-3:**

Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

**Experiment-4:**

Exercises to solve the real-world problems using the following machine learning methods: a) Linear Regression b) Logistic Regression c) Binary Classifier

**Experiment-5:** Develop a program for Bias, Variance, Remove duplicates , Cross Validation

**Experiment-6:** Write a program to implement Categorical Encoding, One-hot Encoding

**Experiment-7:**

Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.

**Experiment-8:**

Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.

**Experiment-9:** Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**Experiment-10:**

Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

**Experiment-11:** Apply EM algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

**Experiment-12:** Exploratory Data Analysis for Classification using Pandas or Matplotlib.

**Experiment-13:**

Write a Python program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set

**Experiment-14:**

Write a program to Implement Support Vector Machines and Principle Component Analysis

**Experiment-15:**

Write a program to Implement Principle Component Analysis



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>
<b>CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY USING DevOps (Skill Oriented Course III)</b>					

**Course Outcomes:**

At the end of the Course, Student will be able to:

- Understand the why, what and how of DevOps adoption
- Attain literacy on Devops
- Align capabilities required in the team
- Create an automated CICD pipeline using a stack of tools

**List of Exercises:**

**Note:** There are online courses indicated in the reference links section. Learners need to go through the contents in order to perform the given exercises

**Exercise 1:**

Reference course name : [Software engineering and Agile software development](#)

Get an understanding of the stages in software development lifecycle, the process models, values and principles of agility and the need for agile software development. This will enable you to work in projects following an agile approach to software development.

Solve the questions [given in the reference course name](#) to gauge your understanding of the topic

**Exercise 2:**

Reference course name: [Development & Testing with Agile: Extreme Programming](#)

Get a working knowledge of using extreme automation through XP programming practices of test first development, refactoring and automating test case writing.

Solve the questions in the “Take test” module [given in the reference course name](#) to gauge your understanding of the topic

**Exercise 3:**

Module name :DevOps adoption in projects

It is important to comprehend the need to automate the software development lifecycle stages through DevOps. Gain an understanding of the capabilities required to implement DevOps, continuous integration and continuous delivery practices.

Solve the questions given in Quiz1, Quiz2, Quiz 3

**Exercise 4:**

Module name :Implementation of CICD with Java and open source stack

Configure the web application and Version control using Git using Git commands and version control operations.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**Exercise 5:**

Module Name: Implementation of CICD with Java and open source stack

Configure a static code analyzer which will perform static analysis of the web application code and identify the coding practices that are not appropriate. Configure the profiles and dashboard of the static code analysis tool.

**Exercise 6:**

Module Name: Implementation of CICD with Java and open source stack

Write a build script to build the application using a build automation tool like Maven. Create a folder structure that will run the build script and invoke the various software development build stages. This script should invoke the static analysis tool and unit test cases and deploy the application to a web application server like Tomcat.

**Exercise 7:**

Module Name: Implementation of CICD with Java and open source stack

Configure the Jenkins tool with the required paths, path variables, users and pipeline views.

**Exercise 8:**

Module name: Implementation of CICD with Java and open source stack

Configure the Jenkins pipeline to call the build script jobs and configure to run it whenever there is a change made to an application in the version control system. Make a change to the background color of the landing page of the web application and check if the configured pipeline runs.

**Exercise 9:**

Module name: Implementation of CICD with Java and open source stack

Create a pipeline view of the Jenkins pipeline used in Exercise 8. Configure it with user defined messages.

**Exercise 10 :**

Module name: Implementation of CICD with Java and open source stack

In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static analysis of code.

**Exercise 11:**

Module name :Implementation of CICD with Java and open source stack

In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static unit testing.

**Exercise 12:**

Module name :Course end assessment

In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for code coverage.

**Reference Books:**

1. Learning Continuous Integration with Jenkins: A beginner's guide to implementing Continuous Integration and Continuous Delivery using Jenkins - Nikhil Pathania ,Packt publication  
 [https://www.amazon.in/Learning-Continuous-Integration-Jenkins-Pathania/dp/1785284835]
2. Jenkins 2 – Up and Running: Evolve Your Deployment Pipeline for Next Generation Automation - Brent Laster, O'Reilly publication



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

[<https://www.amazon.in/Jenkins-2-Running-Brent-Laster/dp/1491979593>]

**Hardware and software configuration:**

1. Git [GitHub or Gitlab]
2. Sonarqube
3. Jenkins
4. JUnit
5. Eclipse
6. Tomcat server
7. Maven
8. Cobertura or JaCoCo
9. Java SDK
10. All necessary drivers and jar files for connecting the software
11. Windows machine with 16GB RAM

**Web Links:( Courses mapped to Infosys Springboard platform)**

1. [https://infyspringboard.onwingspan.com/en/app/toc/lex\\_auth\\_013382690411003904735\\_shared/overview](https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_013382690411003904735_shared/overview) [Software Engineering and Agile software development]
2. [https://infyspringboard.onwingspan.com/en/viewer/html/lex\\_auth\\_01350157819497676810467](https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_01350157819497676810467) [Development & Testing with Agile: Extreme Programming]
3. [https://infyspringboard.onwingspan.com/en/viewer/html/lex\\_auth\\_01353898917192499226\\_shared](https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_01353898917192499226_shared) [DevOps CICD]



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>EMPLOYABILITY SKILLS-I</b>					

**Course Objectives:**

The main objective of this course is to assist students in developing employability skills and personal qualities related to gaining and sustaining employment.

**Course Outcomes:** The end of the course student will be able to

- Understand the corporate etiquette.
- Make presentations effectively with appropriate body language
- Be composed with positive attitude
- Understand the core competencies to succeed in professional and personal life

**UNIT I:**

**Analytical Thinking & Listening Skills:** Self-Introduction, Shaping Young Minds - A Talk by Azim Premji (Listening Activity), Self – Analysis, Developing Positive Attitude, Perception.

**Communication Skills:** Verbal Communication; Non Verbal Communication (Body Language)

**UNIT II:**

**Self-Management Skills:** Anger Management, Stress Management, Time Management, Six Thinking Hats, Team Building, Leadership Qualities

**Etiquette:** Social Etiquette, Business Etiquette, Telephone Etiquette, Dining Etiquette

**UNIT III:**

**Standard Operation Methods:** Note Making, Note Taking, Minutes Preparation, Email & Letter Writing

**Verbal Ability:** Synonyms, Antonyms, One Word Substitutes-Correction of Sentences-Analogies, Spotting Errors, Sentence Completion, Course of Action -Sentences Assumptions, Sentence Arguments, Reading Comprehension, Practice work

**UNIT IV:**

**Job-Oriented Skills –I:** Group Discussion, Mock Group Discussions

**UNIT V:**

**Job-Oriented Skills –II:** Resume Preparation, Interview Skills, Mock Interviews

**Text Books and Reference Books:**

1. Barun K. Mitra, Personality Development and Soft Skills, Oxford University Press, 2011.
2. S.P. Dhanavel, English and Soft Skills, Orient Blackswan, 2010.
3. R.S. Aggarwal, A Modern Approach to Verbal & Non-Verbal Reasoning, S.Chand & Company Ltd., 2018.
4. Raman, Meenakshi & Sharma, Sangeeta, Technical Communication Principles and Practice, Oxford University Press, 2011.

**e-resources:**

1. [www.Indiabix.com](http://www.Indiabix.com)
2. [www.freshersworld.com](http://www.freshersworld.com)



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

<b>III B Tech I Sem</b>	<b>Minor</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>2</b>	<b>4</b>
<b>MACHINE LEARNING</b>					

**Course Objectives:**

This course will enable students to learn and understand different Data sets in implementing the machine learning algorithms.

**Course Outcomes:** At the end of the course, student will be able to

- Implement procedures for the machine learning algorithms
- Design and Develop Python programs for various Learning algorithms
- Apply appropriate data sets to the Machine Learning algorithms
- Develop Machine Learning algorithms to solve real world problems

**Unit I:**

**Introduction-** Artificial Intelligence, Machine Learning, Deep learning, Types of Machine Learning Systems, Main Challenges of Machine Learning.

**Statistical Learning:** Introduction, Supervised and Unsupervised Learning, Training and Test Loss, Tradeoffs in Statistical Learning, Estimating Risk Statistics, Sampling distribution of an estimator, Empirical Risk Minimization.

**Unit II:**

**Supervised Learning**(Regression/Classification):Basic Methods: Distance based Methods, Nearest Neighbours, Decision Trees, Naive Bayes, **Linear Models:** Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines

**Unit III:**

**Binary Classification:** Multiclass/Structured outputs, MNIST, Ranking.

**Ensemble Learning and Random Forests:** Introduction, Voting Classifiers, Bagging and Pasting, Random Forests, Boosting, Stacking.

**Unit IV:**

**Support Vector Machine:** Linear SVM Classification, Nonlinear SVM Classification SVM Regression, Naïve Bayes Classifiers.

**Unit V:**

**Unsupervised Learning Techniques:** Clustering, K-Means, Limits of K-Means, Using Clustering for Image Segmentation, Using Clustering for Preprocessing, Using Clustering for Semi-Supervised Learning, DBSCAN, Gaussian Mixtures.

Dimensionality Reduction: The Curse of Dimensionality, Main Approaches for Dimensionality Reduction, PCA, Using Scikit-Learn, Randomized PCA, Kernel PCA.

**Requirements:** Develop the following program using Anaconda/ Jupiter/ Spider and evaluate ML models.



**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA**  
**KAKINADA – 533 003, Andhra Pradesh, India**

**DEPARTMENT OF CSE - ARTIFICIAL INTELLIGENCE**

**List of Experiments:**

**Experiment-1:**

For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

**Experiment-2:**

Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

**Experiment-3:**

Exercises to solve the real-world problems using the following machine learning methods: a) Linear Regression b) Logistic Regression c) Binary Classifier

**Experiment-4:** Develop a program for Bias, Variance, Remove duplicates , Cross Validation

**Experiment-5:**

Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.

**Experiment-6:** Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

**Experiment-7:**

Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

**Experiment-8:**

Write a program to Implement Support Vector Machines and Principle Component Analysis

**Text Books:**

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, O'Reilly Publications, 2019
2. Data Science and Machine Learning Mathematical and Statistical Methods,Dirk P. Kroese, Zdravko I. Botev, Thomas Taimre, Radislav Vaisman,25th November 2020

**Reference Books:**

1. Machine Learning Probabilistic Approach, Kevin P. Murphy, MIT Press, 2012.