NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY

DEEMED TO BE UNIVERSITY UNDER DISTINCT CATEGORY

(AN AUTONOMOUS INSTITUTION UNDER MINISTRY OF ELECTRONICS & IT, GOVT. OF INDIA)

Curriculum and Syllabi for

M.Tech - Data Engineering

(2026-27 Scheme)

Main Campus at Ropar and Constituent Units at Aizawl, Agartala, Aurangabad, Calicut, Gorakhpur, Imphal, Itanagar, Kekri, Kohima, Patna and Srinagar

Vision

To be a globally recognized center of excellence in Data Engineering education and research, empowering students to become innovative leaders and ethical practitioners capable of shaping the future of data-driven technologies for societal and industrial advancement.

Mission

The mission of the M.Tech in Data Engineering program at NIELIT University is to deliver a rigorous, industry-aligned curriculum that blends theoretical foundations with hands-on skills, fostering innovation, critical thinking, and ethical practices. By promoting interdisciplinary collaboration and experiential learning through projects and laboratories, the program aims to equip students with the expertise to design data-driven solutions for real-world challenges, preparing them to excel and lead in the evolving landscape of data science and engineering.

Program Education Objectives

PEO1: Graduates leverage foundational and advanced concepts in data engineering to excel in diverse professional roles, fostering innovation and critical problem-solving.
PEO2: Graduates demonstrate proficiency in advanced data engineering techniques, algorithms, and tools, addressing industry challenges and contributing to research and development.
PEO3: Graduates exhibit effective communication, teamwork, and ethical values, positioning them as responsible leaders and collaborators in the data engineering domain.

Program Outcomes

PO1: Apply mathematical foundations and data engineering principles to solve complex computing challenges, both independently and collaboratively.
PO2: Design, analyze, and implement advanced data structures, algorithms, and computational solutions, demonstrating strong competence in data engineering.
PO3: Critically evaluate and select appropriate data engineering methodologies and tools, adapting to emerging technologies and industry trends.
PO4: Analyze and interpret data effectively, applying machine learning techniques to advance data preparation, analysis, and storage practices.
PO5: Demonstrate expertise in data warehousing, data mining, and data security to ensure efficient management, retrieval, and protection of large-scale datasets.

Program Specific Outcomes

PSO1: Graduates will be able to design and develop scalable data pipelines and architectures to process, store, and retrieve massive data streams effectively in diverse domains.
PSO2: Graduates will apply data engineering techniques, including big data frameworks and cloud-based solutions, to build efficient, reliable, and secure data-driven systems tailored to industry needs.
PSO3: Graduates will integrate ethical practices and professional standards while addressing societal challenges through innovative data engineering solutions.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	36
Audit Courses (without grade or credit)	AU	0
Open Elective Courses	OE	8
Professional Elective Courses	PE	16
Program Core Courses	PC	20
Total Credits (Common track)		**80**

EVALUATION AND ASSESSMENT

1. Performance of a student in a semester shall be evaluated through continuous Class assessment, Tutorial/Lab assessment, Mid-Semester Examination (MSE) and End-Semester Examination (ESE). Both the MSE and ESE shall be the University examination and will be conducted as notified by the Controller of Examinations (CoE) of the University.

2. The continuous assessment shall be based on assignments, tutorials, paper presentation / quizzes/ viva-voce / flipped classes, lab work / projects / fieldwork and attendance, etc.

3. The MSE/ESE shall be comprising of written papers.

4. The overall assessment of the students will be done as per the following scheme:

S. No.	Assessment Type	Marks Weightage
1.	Mid Semester Examination	20
2.	End Semester Examination	40
3.	Continuous Assessment - Practical /Lab/ Tutorial	25
4.	Continuous Assessment - Theory	15
Total		**100**

For more details, please refer the applicable ordinance for this programme and Assessment SOPs.

CURRICULUM

Semester 1

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-MTEDEN-101	DASH250075	Mathematics for Data Science	3	1	0	4
PC-MTEDEN-102	DCSE250004	Advanced Data Structures and Algorithms	3	0	2	4
PE-MTEDEN-103	Elective	Program Elective				4
OE-MTEDEN-104	Elective	Open Elective				4
PC-MTEDEN-105	DASH250095	Research Methodology and IPR	3	1	0	4
AU-MTEDEN-106	Elective	Audit Elective				0

Semester 2

Semester Code	Course Code	Course Title	L	T	P	Cr
PC-MTEDEN-201	DOAI250004	Artificial Intelligence	3	1	0	4
PC-MTEDEN-202	DCSA250068	Soft Computing	3	0	2	4
PE-MTEDEN-203	Elective	Program Elective				4
PE-MTEDEN-204	Elective	Program Elective				4
AU-MTEDEN-205	Elective	Audit Elective				0
EE-MTEDEN-206	EE	Mini Project				4

Semester 3

Semester Code	Course Code	Course Title	L	T	P	Cr
PE-MTEDEN-301	Elective	Program Elective				4
OE-MTEDEN-302	Elective	Open Elective				4
EE-MTEDEN-303	EE	Dissertation-I/ Industrial project				14

Semester 4

Semester Code	Course Code	Course Title	L	T	P	Cr
EE-MTEDEN-401	EE	Dissertation-II				18

Elective Courses

Elective Courses for PE Category

Course Code	Course Title	L	T	P	Cr	For Sem./Code
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4	
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4	
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4	
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4	
DCSE250020	Cloud Computing	3	0	2	4	
DCSE250050	Data Security and Access Control	3	0	2	4	
DCSE250051	Data Storage Technologies and Networks	3	0	2	4	
DCSE250069	Distributed Databases	3	0	2	4	
DCSE250070	Distributed Systems	3	1	0	4	
DCSE250084	GPU Computing	3	1	0	4	
DCSE250124	Web Analytics and Development	3	0	2	4	
DOAI250001	Advanced Data Analytics	3	0	2	4	
DOAI250009	Big Data Analytics	3	0	2	4	
DOAI250011	Business Analytics	3	0	2	4	
DOAI250019	Data Mining and Warehousing	3	0	2	4	
DOAI250021	Data Science	3	0	2	4	
DOAI250023	Data Visualization	2	0	4	4	
DOAI250032	Machine Learning	3	0	2	4	
DOAI250047	Recommender Systems	3	1	0	4	

Elective Courses for OE Category

Students may opt courses under MOOC, NPTEL, SWAYAM, NSQF/NCVET aligned courses or other relevant technical subjects not included in their core curriculum. The exercise of option shall be subject to the Course Schedule of the respective Constituent Unit, the credit requirements and availability of seats. Guidelines for choosing MOOC/Online courses are given separately and shall have to be adhered to.

Elective Courses for AU (Audit) Category

Course Code	Course Title	L	T	P	Cr
DASH240027	Disaster Management	2	0	0	0
DASH240047	English for Research Paper Writing	2	0	0	0
DASH240052	Environmental Science	3	0	0	0
DASH240085	Pedagogy Studies	2	0	0	0
DASH240086	Personality Development through Life Enlightenment Skills	2	0	0	0
DASH240097	Sanskrit for Technical Knowledge	2	0	0	0
DASH240103	Stress Management by Yoga	2	0	0	0
DASH240104	Technical Presentation and Report Writing	0	0	2	0
DASH240106	Value Education	2	0	0	0

DASH240124	Constitution of India - AU	2	0	0	0
DASH250023	Constitution of India	2	1	0	0
DASH250027	Disaster Management	3	0	0	0
DASH250034	Energy and Environmental Engineering	3	0	0	0
DASH250047	English for Research Paper Writing	2	0	0	0
DASH250054	Essence of Indian Knowledge and Tradition	2	0	0	0
DASH250085	Pedagogy Studies	2	0	0	0
DASH250086	Personality Development	2	0	0	0
DASH250097	Sanskrit for Technical Knowledge	3	0	0	0
DASH250103	Stress Management by Yoga	3	0	0	0
DASH250106	Value Education	2	0	0	0
DASH250117	Innovative Thinking and Entrepreneurship Skills	1	0	0	0
DASH250118	Intellectual Property Rights	2	0	0	0
DCSA240080	Training on Computer Networking	0	0	2	0
DCSA240304	Lab Based on Statistical Package	0	0	2	0
Any other Scheduled Course as per the Course Schedule of the respective Constituent Unit can also be chosen, subject to availability of seats. However, there shall be no assessment and grading for the course chosen as Audit Course.					

The choice of all type of Electives available shall be as per the Course Schedule of the respective Constituent Unit.

Guidelines for Massive Open Online Courses (MOOC) / approved Online Courses

1. Relevant Swayam, MOOC, NPTEL, NIELIT NSQF and any other online courses approved by UGC/AICTE shall also be allowed as Open Electives(OE).

2. One credit of MOOC course should be minimum of 30 hours.

3. A student can choose any approved online course as Open Elective, subject to minimum credit requirements.

4. The students willing to opt OE under MOOC in their next semester, will submit their request to the MOOC Coordinator at their respective campus.

5. Once approved, the campus shall display the list of accepted courses.

6. The student has to submit certificate issued by the institution offering the MOOCs course, along with the number of credits and grades, to get credits transferred into his/her marks certificate issued by NDU.

7. The transfer of credits shall be as adopted by NDU.

Detailed Syllabus

Course Code	Course Name	L	T	P	Cr
DASH250075	Mathematics for Data Science	3	1	0	4

Course Outcomes:

CO1: Apply the concepts of vectors, matrices, eigenvalues, eigenvectors, and Singular Value Decomposition (SVD) to solve data science problems such as dimensionality reduction and data compression.
CO2: Demonstrate understanding of probability distributions, sampling, Bayesian concepts, and hypothesis testing to model uncertainty and make statistical inferences in data-driven applications.
CO3: Utilize calculus and optimization techniques, including gradient descent, stochastic gradient descent, and Lagrange multipliers, to solve constrained and unconstrained optimization problems in machine learning.
CO4: Analyze information-theoretic measures such as entropy, cross-entropy, KL divergence, and mutual information to evaluate data representation, feature selection, and model performance.
CO5: Apply mathematical tools such as numerical methods, matrix factorizations, least squares regression, and Markov chains for effective data modeling, signal analysis, and real-world computational applications.

Module 1:

Linear Algebra for Data Science (9 Hours)
Vectors, matrices, vector spaces, rank, null space, systems of linear equations, eigenvalues and eigenvectors, diagonalization, Singular Value Decomposition (SVD), principal component analysis (PCA); applications in dimensionality reduction and data compression.

Module 2:

Probability and Statistics (9 Hours)
Random variables, expectations, variance, covariance, distributions (Gaussian, Bernoulli, Poisson), law of large numbers, central limit theorem, Bayesian concepts, conditional independence, sampling techniques, hypothesis testing.

Module 3:

Calculus and Optimization (9 Hours)
Differentiation, partial derivatives, chain rule, multivariate Taylor series, Jacobian, Hessian Matrix, gradient descent, stochastic gradient descent (SGD), Newton's method, convex optimization, constraints and Lagrange multipliers.

Module 4:

Information Theory and Entropy (9 Hours)
Entropy, cross-entropy, Kullback–Leibler (KL) divergence, mutual information, Shannon's theory, applications in decision trees, feature selection, and deep learning loss functions, basics of MDL (Minimum Description Length) and compression theory.

Module 5:

Mathematical Tools for Data Modeling (9 Hours)
Numerical methods: interpolation, numerical differentiation and integration, matrix factorization (LU, QR, Cholesky), least squares regression, Markov chains, Fourier and Laplace transforms for data filtering and signal analysis.

Labs / Practicals:

N/A

References:

1. P. N. Gupta & S. C. Gupta – Mathematics for Data Science, Khanna Publishers.
2. B. S. Grewal – Higher Engineering Mathematics, Khanna Publishers.
3. V. Sundaram & S. Jayaraman – Probability and Random Processes, Tata McGraw Hill.
4. Kanti B. Datta – Matrix and Linear Algebra, PHI Learning.
5. S. Kumaresan – Linear Algebra: A Geometric Approach, Prentice-Hall of India.
6. Marc Deisenroth et al. – Mathematics for Machine Learning, Cambridge University Press.
7. Sheldon Ross – Introduction to Probability and Statistics for Engineers and Scientists.
8. Shai Shalev-Shwartz and Shai Ben-David – Understanding Machine Learning.

Course Code	Course Name	L	T	P	Cr
DCSE250004	Advanced Data Structures and Algorithms	3	0	2	4

Course Outcomes:

CO1: Analyze the performance of basic data structures and sorting/searching techniques.
CO2: Implement and apply advanced trees, heaps, and hash tables to solve complex problems.
CO3: Solve problems using graph algorithms and greedy strategies.
CO4: Design algorithms using divide and conquer, backtracking, and dynamic programming.
CO5: Develop efficient and scalable solutions using appropriate algorithmic techniques.

Module 1:

Review of Fundamental Concepts
Pointers and Dynamic Memory Allocation, Arrays, Dynamically Allocated Arrays, Structures, Algorithm Specification, Data Abstraction, Performance Analysis, Performance Measurement, Time and space complexity

Module 2:

Basic Data Structures and Advanced Trees
Stacks, Queues, Linked Lists, Hash Tables, Searching and Sorting Techniques: Linear Search, Bubble Sort, Selection Sort, Insertion Sort, Binary Trees, Binary Tree Traversals, Applications of Binary Trees, Threaded Binary Trees, Binary Search Trees, Advanced Trees: AVL Trees: Rotations, insertion and deletion, Red-Black Trees, Splay Trees, B Trees, B+ Trees, Binary Heap: Min/Max heap operations, Priority Queue applications.

Module 3:

Divide and Conquer, Backtracking, String Matching algorithms
Merge Sort, Quick Sort, Binary Search, Backtracking: N-Queens, Graph Coloring, String matching algorithms: KMP, Rabin-Karp

Module 4:

Graph Algorithms
Graph Representations: Adjacency list, Adjacency Matrix, BFS, DFS, Shortest Path: Dijkstra's, Greedy algorithms: Activity Selection, Huffman Coding Minimum Spanning Tree: Prim's and Kruskal's Algorithms

Module 5:

Dynamic Programming
Basic concepts: Overlapping subproblems, optimal substructure, 0/1 Knapsack, Longest Common Subsequence, Matrix Chain Multiplication

Labs / Practicals:

Preparing

References:

1. "Introduction to Algorithms (Fourth Edition)" by Thomas H. Cormen, Charles Eric Leiserson, Ronald Linn Rivest, Clifford Seth Stein
https://mitpress.mit.edu/9780262046305/introduction-to-algorithms

2. "A Second Course in Algorithms" by Timothy Avelin Roughgarden
https://timroughgarden.org/w16/l

3. "Algorithm Design" by Jon Michael Kleinberg and Eva Tardos
https://dl.acm.org/doi/10.5555/1051910

4. "Computational Intractability: A Guide to Algorithmic Lower Bounds" by Erik D. Demaine, William Ian Gasarch, and Mohammad Taghi Hajiaghayi
https://hardness.mit.edu/drafts/2025-02-02.pdf

5. "The Design and Analysis of Algorithms" by Dexter Campbell Kozen
https://www.cs.cornell.edu/kozen/Papers/daa.pdf

6. "The Art of Computer Programming" by Donald Ervin Knuth
https://www-cs-faculty.stanford.edu/~knuth/taocp.html

Course Code	Course Name	L	T	P	Cr
DASH250095	Research Methodology and IPR	3	1	0	4

Course Outcomes:

CO1: Understand the principles and process of research methodology.
CO2: Apply appropriate research design and data collection methods in computing projects.
CO3: Analyze and interpret qualitative and quantitative data using statistical tools.
CO4: Develop ethically sound, well-documented research or project reports conforming to academic and industrial standards.
CO5: Understand various forms of intellectual property and apply processes for protection and patent filing.

Module 1:

Introduction to Research and Research Ethics
Definition and objectives of research, types of research – fundamental, applied, exploratory, empirical, significance of research in computer applications, research ethics and integrity, plagiarism, citation styles (APA, IEEE), tools for plagiarism check.

Module 2:

Research Design and Data Collection
Research problem identification and formulation, hypothesis generation, literature review techniques, types of research design: experimental, descriptive, analytical, sampling techniques, primary and secondary data sources, survey design, case study design.

Module 3:

Data Analysis and Interpretation
Quantitative and qualitative data analysis, measures of central tendency and dispersion, t-test, ANOVA, correlation and regression, chi-square test, software tools: Excel, R, SPSS (demo-level), data visualization and result interpretation.

Module 4:

Technical Writing and Project Report Preparation
Structure of research papers and reports, abstract, introduction, literature review, methodology, results, and conclusions, referencing tools (Mendeley/Zotero), IEEE/ACM/SCI journal guidelines, thesis and dissertation formatting, proposal writing basics.

Module 5:

Intellectual Property Rights and Patent Process
Introduction to IPR: patents, copyrights, trademarks, industrial design, Indian and international patent systems (WIPO, TRIPS), criteria for patentability, patent filing process in India (IPINDIA portal), open-source licenses, patent drafting basics, IPR in software and AI innovations.

Labs / Practicals:

N/A

References:

1. C.R. Kothari & Gaurav Garg – Research Methodology: Methods and Techniques, New Age International.
2. Ranjit Kumar – Research Methodology – A Step-by-Step Guide, Sage Publications India.
3. P. Narayan – Intellectual Property Law, Eastern Book Company.
4. Neeraj Pandey & Khushdeep Dharni – Intellectual Property Rights, PHI Learning.
5. Yogesh Kumar Singh – Fundamentals of Research Methodology and Statistics, New Age International.
6. Bharat Bhushan – Research Methodology in Computer Science, Himalaya Publishing.
7. WIPO and IPIndia Manuals – Guidelines for Filing Patents in India.

Course Code	Course Name	L	T	P	Cr
DOAI250004	Artificial Intelligence	3	1	0	4

Course Outcomes:

CO1: Understand AI foundations, intelligent agents, and problem-solving formulations.
CO2: Apply uninformed and heuristic search strategies, and adversarial game-playing techniques.
CO3: Represent knowledge using logic, rules, and reasoning methods.
CO4: Analyze different learning approaches including neural networks and connectionist models.
CO5: Design and evaluate expert systems for domain-specific knowledge representation and reasoning.

Module 1:

Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

Module 2:

Searching: Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Greedy best first search, A* search Game Playing: Adversial search, Games, minimax, algorithm, optimal decisions in multiplayer games, Alpha-Beta pruning, Evaluation functions, cutting of search.

Module 3:

Knowledge Representation: Using Predicate logic, representing facts in logic, functions and predicates, Conversion to clause form, Resolution in propositional logic, Resolution in predicate logic, Unification.
Representing Knowledge Using Rules: Procedural Versus Declarative knowledge, Logic Programming, Forward versus Backward Reasoning

Module 4:

Learning: What is learning, Rote learning, Learning by Taking Advice, Learning in Problem-solving,
Learning from example: induction, Explanation-based learning.
Connectionist Models: Hopfield Networks, Learning in Neural Networks, Applications of Neural Networks, Recurrent Networks. Connectionist AI and Symbolic AI.

Module 5:

Expert System: Representing and using Domain Knowledge, Reasoning with knowledge, Expert System Shells, Support for explanation examples, Knowledge acquisition-examples.

Labs / Practicals:

NA

References:

1. Artificial Intelligence – A Modern Approach. Second Edition, Stuart Russel, Peter Norvig, PHI/ Pearson Education.
2. Artificial Intelligence, Kevin Knight, Elaine Rich, B. Shivashankar Nair, 3rd Edition,2008
3. Artificial Neural Networks B. Yagna Narayana, PHI.
4. Artificial Intelligence, 2nd Edition, E.Rich and K.Knight (TMH)
5. Artificial Intelligence and Expert Systems – Patterson PHI.
6. Expert Systems: Principles and Programming- Fourth Edn, Giarrantana/ Riley, Thomson.
7. PROLOG Programming for Artificial Intelligence. Ivan Bratka- Third Edition – Pearson Education.
8. Neural Networks Simon Haykin PHI.
9. Artificial Intelligence, 3rd Edition, Patrick Henry Winston., Pearson Edition.

Course Code	Course Name	L	T	P	Cr
DCSA250068	Soft Computing	3	0	2	4

Course Outcomes:

After completion of course, students would be able to:
CO1: Identify and describe soft computing techniques and their roles in building intelligent machines.
CO2: Apply genetic algorithms to combinatorial optimization problems.
CO3: Explain the basic concepts, components, and paradigms of neural networks, and compare them with other information processing methods.
CO4: Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems.
CO5: Evaluate and compare solutions by various soft computing approaches for a given problem.

Module 1:

Introduction: What is computational intelligence?- Biological basis for neural networks- Biological versus Artificial neural networks- Biological basis for evolutionary computation- Behavioral motivations for fuzzy logic, Myths about computational intelligence- Computational intelligence application areas, Evolutionary computation, computational intelligence-Adoption, Types, self-organization and evolution, Historical views of computational intelligence, Computational intelligence and Soft computing versus Artificial intelligence and Hard computing.

Module 2:

Evolutionary computation concepts and paradigms- History of evolutionary computation & overview, Genetic algorithms, Evolutionary programming &strategies, Genetic programming, Particle swarm optimization, Evolutionary computation implementations-Implementation issues, Genetic algorithm implementation, Particle swarm optimization implementation.

Module 3:

Neural Network Concepts and Paradigms- What Neural Networks are? Why they are useful, Neural network components and terminology- Topologies - Adaptation, Comparing neural networks and other information Processing methods- Stochastic- Kalman filters - Linear and Nonlinear regression - Correlation - Bayes classification -Vector quantization -Radial basis functions -Preprocessing - Post processing.

Module 4:

Fuzzy Systems concepts and paradigms - Fuzzy sets and Fuzzy logic - Approximate reasoning, Developing a fuzzy controller - Fuzzy rule system implementation.

Module 5:

Performance Metrics- General issues- Partitioning the patterns for training, testing, and validation-Cross validation - Fitness and fitness functions - Parametric and nonparametric statistics, Evolutionary algorithm effectiveness metrics, Receiver operating characteristic curves, Computational intelligence tools for explanation facilities, Case Studies for implementation of practical applications in computational intelligence.

Labs / Practicals:

preparing

References:

1. David B. Fogel and Charles J. Robinson; Computational Intelligence: The experts speak. Wiley - Intersciece 2003.
2. Neuro Fuzzy & Soft Computing - J.-S.R.Jang, C.-T.Sun, E.mizutani, Pearson Education
3. Digital Neural Network - S.Y.Kung, Prentice Hall International Inc.
4. Spiking Neural Networks - Wulfram Gerstner, Wenner Kristler, Cambridge University Press.
5. Neural Networks and Fuzzy Systems: Dynamical Systems Application to Machine Intelligence - Bart Kosko,

Prentice Hall, 1992.

Course Code	Course Name	L	T	P	Cr
DOAI260001	AI (Industry)-Applied Ethics for AI	3	0	2	4

Course Outcomes:

CO1. Understand and be able to define and explain the terms 'AI for good' and 'responsible AI'. Describe the significance of responsible AI and recognize the potential benefits of AI in society.
CO2. Discuss and examine the impact of ethics on AI decision-making processes. Identify common ethical pitfalls in AI development and deployment. Explore methods for ethical AI practices.
CO3. Understand to apply an ethics-first approach to AI project planning and development. Compare different approaches to ethical AI design which include design of rules based on ethics and virtues. Identify factors influencing ethical decisions in AI development.
CO4. Apply Utilitarian and Kantian ethical principles to AI case studies, scenarios, and dilemmas. Explain the purpose and benefits of AI ethics codes. Develop a code of ethics for AI within an organization.
CO5. Identify the importance of interpretability in AI models. Evaluate the strengths and weaknesses of three different explainable AI tools: SHAP, LIME, and Tree Interpreter.
CO6. Conduct impact assessments for AI solutions. Apply ethical considerations in designing an AI system or solution. Implement measures to mitigate ethical risks and challenges post deployment.

Module 1:

Introduction to AI Ethics – meaning and importance, Role of Ethics in Responsible AI – defining 'AI for good' and 'responsible AI', significance of responsible AI in society, Common Ethical Pitfalls in AI – bias, fairness, accountability, transparency issues in AI development and deployment, Principles of Ethical AI – core principles guiding ethical AI, methods for ethical AI practices in development and deployment.

Module 2:

Ethics-First Approach in the AI Project Cycle – applying ethics at every stage of AI project planning and development, Approaches for Designing Ethical AI – rules-based ethics, values-based design, Introduction to Ethical Frameworks – overview of major ethical frameworks applied to AI, Virtue-Based Ethics – virtue ethics principles and the 4-box method for analysing ethical dilemmas, Applying Bio-Ethics Framework in AI – bioethical principles applied to AI in healthcare and biotechnology.

Module 3:

Utilitarianism for Ethical AI – utilitarian principles applied to AI case studies and dilemmas, Kantianism for Ethical AI – Kantian ethical principles applied to AI scenarios, Code of AI Ethics for Organizations – purpose, benefits and development of an organizational AI code of ethics, Data Protection and AI – risks of data breaches, privacy violations, responsible data handling.

Module 4:

Explainable AI – importance of interpretability in AI models, Intel XAI Toolkit, impact of explainability on decision-making and trust, Explainable AI in Practice – SHAP, LIME and Tree Interpreter tools: strengths, weaknesses and methods for applying them to analyze and interpret AI applications.

Module 5:

Impact Assessment of AI Solutions – conducting impact assessments for AI systems, Mitigating Ethical Issues Post Deployment – strategies to identify and resolve ethical risks after deployment, Project Creation – creating an AI startup with data card, model card, code of ethics, explainability report, impact self-assessment and transparency report.

Labs / Practicals:

1. Develop comprehensive Data Cards for datasets, as outlined in the Data Cards Playbook by Google, to ensure transparency and accountability in AI training data.
2. Design Responsible AI Best Practices.

3. Apply Human-Centered AI Canvas to design AI systems that address user needs and ethical considerations.
4. Learn to use The Box to embed AI principles into organizational workflows and operations.
5. Apply bio-ethics principles to AI case studies, scenarios, and dilemmas; evaluate the impact of virtue-based ethics on AI systems.
6. Develop AI ethics cards for given AI scenarios. Analyse a variety of ethical dilemmas by following the 4-box method from Virtue Ethics.
7. Analyse a variety of ethical dilemmas by walking through the ethical frameworks: Virtue Ethics, Bioethics, Utilitarian, Kantian, and Moral System Agnostic.
8. Develop an AI Code of Ethics for a fictitious company.
9. Use the UXAI Toolkit to brainstorm and plan different explainable interfaces for AI applications. Use the explainable AI library SHAP to create an explainability report for income prediction on the standard adult census income dataset.
10. Conduct impact assessments for AI solutions; create your own AI startup, develop business documents, write documentation (data card, model card, code of ethics, explainability report, impact self-assessment) and create and analyze transparency report.

References:

1. Intel AI for Future Workforce – Applied Ethics for AI Course Content, Intel Digital Readiness Program.
2. Virginia Eubanks – Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor, St. Martin's Press.
3. Cathy O'Neil – Weapons of Math Destruction, Crown Publishers.
4. Kate Crawford – Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence, Yale University Press.
5. Google Data Cards Playbook – https://pair-code.github.io/datacardsplaybook/

Course Code	Course Name	L	T	P	Cr
DOAI260002	AI (Industry)-Introduction to Artificial Intelligence	3	0	2	4

Course Outcomes:

CO1. Describe what AI is and what it is not. Appreciate the history of AI and its evolution over time.
CO2. Identify and analyze current trends in AI by correlating them with other technologies like IoT, Big Data and 5G.
CO3. Identify 3 common domains of AI – Natural Language Processing, Computer Vision and Statistical Data – based on the type of underlying data.
CO4. Examine and appreciate typical steps involved in an AI Project through the AI Project Cycle.
CO5. Examine the immediate impacts of AI practices on society and appreciate the ethical concerns of AI applications.
CO6. Classify Supervised Learning, Unsupervised Learning and Reinforcement Learning with the help of examples. Discuss the roles played by different key stakeholders in a typical AI team.

Module 1:

Demystification of AI – what AI is and what it is not, History and Evolution of AI, Current trends in AI, AI and Technology convergence, Industry 4.0 and Digitalization, Role of IoT, Big Data and 5G in AI, Domains of AI – Natural Language Processing, Computer Vision and Statistical Data.

Module 2:

AI Project Cycle-I – Problem Scoping and Data Acquisition, AI Project Cycle-II – Modelling and Evaluation, Societal Impact of AI – ethical concerns and responsible practices, Elements of ML – Supervised Learning, Unsupervised Learning and Reinforcement Learning with examples.

Module 3:

Elements of AI – types and characteristics, Data as Fuel for AI – structured and unstructured data, methods of data mining and data storage, AI/Data Science Team – roles and responsibilities of key stakeholders.

Module 4:

Tools to implement AI – No-Code tools (Teachable Machine, Orange, Roboflow, Chatfuel) and Code-based tools, Introduction to Neural Networks – working of simple Neural Networks, difference between common Deep Learning models with examples and applications.

Module 5:

AI Project-I – Natural Language Processing use case using no-code tool Chatfuel, AI Project-II – Statistical Data use case using no-code tool Orange Data Mining, AI Project-III – Computer Vision use case using no-code tool Roboflow, Future Possibilities of AI – emerging software and hardware technology trends and their direct impact on development of AI.

Labs / Practicals:

1. Familiarize with AI tools and techniques, including statistical data analysis using raw graphs, computer vision applications like Vision API, and generative AI tools such as DALL·E, Stable Diffusion, Gemini, and ChatGPT.
2. Apply the AI Project Cycle to real-world scenarios by identifying problems, collecting data, developing models, and evaluating their effectiveness to propose impactful AI-based solutions.
3. Deconstruct the working behind simple Neural Networks and outline the difference between some common DL models with the help of examples and applications.
4. Explore and utilize no-code tools, such as Teachable Machine, to build AI projects.
5. Project Building using No-code tool – Orange Data Mining for Statistical Data Use cases.
6. Project Building using No-code tool – Roboflow for Computer Vision Use cases.
7. Project Building using No-code tool – Chatfuel for Natural Language Processing Use cases.
8. Classify the type of data into structured and unstructured and evaluate data quality using real datasets.

9. Explore various AI domains by running Vision API experiments and comparing outcomes from NLP, CV, and Statistical Data use cases.
10. Demonstrate the AI Project Cycle on a chosen real-world problem from an industrial or social impact perspective and present findings.

References:

1. Intel AI for Future Workforce – Introduction to AI Course Content, Intel Digital Readiness Program.
2. Stuart Russell and Peter Norvig – Artificial Intelligence: A Modern Approach, Pearson Education.
3. Andreas Mueller and Sarah Guido – Introduction to Machine Learning with Python, O'Reilly Media.
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Deep Learning, MIT Press.
5. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.

Course Code	Course Name	L	T	P	Cr
DOAI260003	AI (Industry)-Introduction to Generative AI	3	0	2	4

Course Outcomes:

CO1. Define Gen AI and distinguish its role in the broader context of AI. Describe how the basic model of Gen AI works. Discuss the wider societal impact of Gen AI.
CO2. Understand how NLP has evolved over time and the shift from Traditional AI to Gen AI. Understand Prompt Engineering and its impact on the output of NLP tasks.
CO3. Learn the limitations of current AI generation tools for video and audio. Recognize the challenges in ethical considerations and responsibilities when using generative AI tools.
CO4. Describe the stages of a generative AI project and learn to select and use appropriate AI tools for development.
CO5. Understand the basic structure of neural networks, architecture of transformer models, workings of diffusion models, and the pre-training process for generative AI models.
CO6. Learn to fine-tune generative AI models, understand RLHF, prototype Gen AI applications, and apply principles of responsible and ethical use of LLMs.

Module 1:

Introduction to Gen AI – definition and scope, role of Gen AI in the broader context of AI, how the basic model of Gen AI works, societal impact of Gen AI, Demystifying Gen AI Models – types of generative models (GANs, VAEs, Diffusion Models, LLMs), evolution of NLP to Gen AI, shift from Traditional AI to Gen AI.

Module 2:

Prompt Engineering-I – introduction to prompt engineering, basic applications and impact on NLP task outputs, text generation and conversation initiation, Prompt Engineering-II – advanced prompt engineering techniques for diverse NLP applications, real-world examples such as Mint Mobile Commercial, Basics of Video and Audio for Generation using Gen AI for Productivity-I – introduction to video generation tools (Runway ML, Genmo), text-to-video and image-to-video generation, limitations of current AI generation tools for video and audio.

Module 3:

Basics of Video and Audio for Generation using Gen AI for Productivity-II – audio and music generation using AIVA, code generation using Codeium, text generation using ChatGPT, Gemini and Perplexity, Generative AI-Ethics – ethical considerations, responsibilities and challenges when using generative AI tools, Generative AI Project Lifecycle – stages of a Gen AI project, selection of appropriate AI tools, Gen AI Tools for Development – open-source tools such as Stable Diffusion XL, proprietary tools such as Bing Image Creator powered by DALLE3.

Module 4:

Neural Network Fundamentals – basic structure and functioning of neural networks, how neural networks process data to make predictions, Transformers and the Evaluation of LLMs – architecture of transformer models, methods to evaluate large language models for performance and reliability, walkthrough of OpenAI Playground and parameter exploration, Diffusion Models and the Evaluation of Text to Image Generation – workings of diffusion models, evaluation of quality of text-to-image generation, Pre-Training Gen AI Models-I – pre-training process for generative AI models, importance of pre-training in building effective AI systems, modifying classification labels of Cohere datasets and its impact on fine-tuning and model confidence.

Module 5:

Fine-Tuning Gen AI Models – techniques for fine-tuning generative AI models to adapt them for specific tasks and applications, RLHF – Evaluating and Optimizing Gen AI Models: reinforcement learning with human feedback, evaluation and improvement of performance of generative AI models, Prototyping Gen AI Applications – skills to design, develop and test prototypes of generative AI applications, LangChain components – LLM, Agents, Prompts, Chains, Document loaders and utils with code-based implementation, LLM Application and Responsible use of AI – practical applications of large language models, ethical and responsible use of LLMs, building an AI

Tutor using open-source models such as Dolly 3B and Intel Neural Chat 7B.

Labs / Practicals:

1. Apply prompt engineering skills in basic NLP tasks, such as text generation, conversation initiation, and simple problem-solving.
2. Develop advanced prompt engineering skills for diverse NLP applications. Examine real-world prompt engineering examples, such as the Mint Mobile Commercial, to discern its role in marketing.
3. Explore cutting-edge Gen AI tools, their features, and capabilities for content generation including video and audio.
4. Explore open-source models such as Stable Diffusion XL and proprietary tools such as Bing Image Creator (powered by DALLE3) for image generation.
5. Text-to-video generation and image-to-video generation using tools like Runway ML and Genmo. Generate audio and music files using the tool AIVA.
6. Generate code by prompts using the tool Codeium. Generate text using tools like ChatGPT, Gemini, and Perplexity.
7. Walkthrough of OpenAI Playground. Explore how changing parameters can affect responses generated by different models.
8. Explore how modifying classification labels of Cohere datasets impacts fine-tuning and model confidence.
9. Explore basic components of LangChain like LLM, Agents, Prompts, Chains, and Document loaders and utils, along with a code-based implementation.
10. Utilize open-source text generation models such as Dolly 3B and Intel Neural Chat 7B for building an AI Tutor application.

References:

1. Intel AI for Future Workforce – Introduction to Generative AI Course Content, Intel Digital Readiness Program.
2. Antony Hagiwara – Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media.
3. Ashish Vaswani et al. – Attention is All You Need, Advances in Neural Information Processing Systems.
4. OpenAI – GPT-4 Technical Report, OpenAI.
5. Harrison Chase – LangChain Documentation – https://docs.langchain.com/

Course Code	Course Name	L	T	P	Cr
DOAI260004	AI (Industry)-Introduction to Machine Learning	3	0	2	4

Course Outcomes:

CO1. Describe Machine Learning and how Deep Learning is a subset of the wider field of Machine Learning.
CO2. Understand the importance of dashboards and discuss various applications of dashboards across different industries.
CO3. Explain the mathematical working behind different ML models. Implement different classification and regression ML models using Python.
CO4. Outline the difference between supervised, unsupervised and reinforcement learning algorithms. Examine the working of different reinforcement learning models.
CO5. Describe the working of Neural Networks and contrast it with biological Neurons. Describe various applications of neural networks. Outline different methods to overcome variance and bias in DL models.
CO6. Discuss and interpret the future of ML based on current and upcoming trends. Develop Python-based AI Projects incorporating Supervised Learning, Unsupervised Learning and Deep Neural Networks.

Module 1:

Introduction to Machine Learning – definition, types and applications, Relationship between AI, ML and Deep Learning, Tools to Implement AI (Python & Mathematics) – I: Python basics for ML, NumPy, Pandas, data manipulation.

Module 2:

Tools to Implement AI (Python & Mathematics) – II: mathematical foundations – linear algebra, statistics and probability, No-Code tool for Data Exploration – Tableau dashboard creation, data visualization techniques, statistical concepts implementation using no-code visualization tools.

Module 3:

AI (Modeling) Supervised Learning-I – Linear Regression, Logistic Regression, mathematical working behind models, classification and regression using Python, AI (Modeling) Supervised Learning-II – SVM, Decision Tree, industrial applications of ML models, AI (Modeling) Unsupervised Learning-I – K-Means Clustering, mathematics of clustering algorithms, AI (Modeling) Unsupervised Learning-II – Recommendation Systems, mathematics behind recommendation system, applications of recommendation systems.

Module 4:

Reinforcement Learning – definition, difference from supervised and unsupervised learning, working of reinforcement learning models with applications, Neural Network and Deep Learning-I – biological vs artificial Neurons, working of Neural Networks, applications of neural networks, Deep Learning-II – Convolutional Neural Networks, Recurrent Neural Networks, methods to overcome variance and bias in DL models, implementation using Python and deep learning frameworks, exploring OpenAI's Gym for reinforcement learning.

Module 5:

ML Project – Supervised Learning AI models: end-to-end Python project using classification/regression, ML Project – Unsupervised Learning AI models: end-to-end Python project using clustering, ML Project – Deep Neural Networks: building and training DNN in Python, Future of Machine Learning and Deep Learning – current and upcoming trends in ML and DL.

Labs / Practicals:

1. Interpret and understand the basic tools required for building ML Projects through a selected list of topics from Mathematics and Python Programming.
2. Develop and create a simple dashboard for visualizing data through Tableau.
3. Implement statistical concepts using the no-code visualization tool and create your own dashboard with the help of a no-code visualization tool.

4. Implement supervised machine learning algorithms, such as SVM and Decision Tree, using Python to develop a deeper understanding of their practical applications.
5. Implement unsupervised machine learning algorithms, such as K-Means, using Python to develop a deeper understanding of their practical applications.
6. Implement neural networks and create simple generative AI models using Python and deep learning frameworks, including exploring OpenAI's Gym for hands-on experience with AI and reinforcement learning.
7. Develop Python-based use cases and AI Projects incorporating Supervised Learning methods.
8. Develop Python-based use cases and AI Projects incorporating Unsupervised Learning methods.
9. Develop Python-based use cases and AI Projects incorporating Deep Neural Networks.
10. Implement a recommendation system using Python and evaluate its performance on a real-world dataset.

References:

1. Intel AI for Future Workforce – Introduction to Machine Learning Course Content, Intel Digital Readiness Program.
2. Andreas Mueller and Sarah Guido – Introduction to Machine Learning with Python, O'Reilly Media.
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville – Deep Learning, MIT Press.
4. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
5. Christopher Bishop – Pattern Recognition and Machine Learning, Springer.

Course Code	Course Name	L	T	P	Cr
DCSE250020	Cloud Computing	3	0	2	4

Course Outcomes:

CO1: Explain the evolution, architecture, and key characteristics of cloud computing, including service models (IaaS, PaaS, SaaS) and deployment types (public, private, hybrid).

CO2: Describe various types and implementation levels of virtualization, including CPU, memory, storage, and network virtualization in cloud environments.

CO3: Analyze layered cloud architecture and address design challenges related to inter-cloud resource management and platform deployment.

CO4: Apply distributed and parallel programming paradigms such as MapReduce and Hadoop for cloud-based application development and understand various cloud platforms like AWS, OpenStack, and Google App Engine.

CO5: Evaluate cloud security challenges and solutions, including data and application security, VM security, and risk management strategies in cloud environments.

Module 1:

Introduction to Cloud Computing

Evolution of Cloud Computing, System Models for Distributed and Cloud Computing, NIST Cloud Computing Reference Architecture, Features of Cloud Computing, Cloud Services – IaaS, PaaS, SaaS, Cloud service Providers – Public , Private and Hybrid Clouds.

Module 2:

Introduction to component virtualization

Basics of Virtualization, Types of Virtualization, Implementation Levels of Virtualization, Virtualization of CPU, Memory, I/O Devices, Desktop Virtualization, Server Virtualization, Storage Virtualization, Network Virtualization.

Module 3:

Architectural Design of Compute and Storage Clouds

Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources.

Module 4:

Parallel and Distributed Programming Paradigms

Map Reduce, Twister and Iterative MapReduce, Hadoop Library from Apache, Mapping Applications, Programming Support, Google App Engine, Amazon AWS, Cloud Software Environments - Eucalyptus, Open Nebula, OpenStack.

Module 5:

Security Overview

Cloud Security Challenges, Software-as-a-Service Security, Security Governance, Risk Management, Security Monitoring, Security Architecture Design, Data Security, Application Security, Virtual Machine Security.

Labs / Practicals:

1. Exploring Cloud Deployment and Service Models
2. Simulating Public, Private, and Hybrid Clouds
3. Virtualization of CPU and Memory
4. Network and Storage Virtualization
5. Deployment of a Layered Cloud Architecture
6. Resource Provisioning and Scaling
7. Deploying a Web Application
8. Simulating Security Attacks and Defenses in a Cloud VM
9. Implementing Access Control and Encryption on Cloud Storage

References:

[1] R. Buyya, C. Vecchiola, and T. Selvi, Mastering Cloud Computing: Foundations and Applications Programming. New Delhi, India: McGraw-Hill Education, 2013.
[2] A. T. Velte, T. J. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach. New Delhi, India: McGraw-Hill Education, 2010.
[3] K. Hwang, G. C. Fox, and J. J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington, MA, USA: Morgan Kaufmann, 2012.
[4] T. Erl, R. Puttini, and Z. Mahmood, Cloud Computing: Concepts, Technology & Architecture. Boston, MA, USA: Pearson Education, 2013.
[5] G. Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA, USA: O'Reilly Media, 2009.

Course Code	Course Name	L	T	P	Cr
DCSE250050	Data Security and Access Control	3	0	2	4

Course Outcomes:

CO1: Explain core concepts of confidentiality, integrity, and availability in data security.

CO2: Apply cryptographic techniques for securing communication and stored data.

CO3: Design and implement authentication and access control mechanisms.

CO4: Analyze security threats and apply appropriate mitigation strategies in networked and cloud environments.

CO5: Evaluate and enforce access control policies using modern tools and technologies.

Module 1:

Foundations of Data Security (9 Hours)
CIA triad: confidentiality, integrity, availability, Threats, vulnerabilities, attacks, and risk, Security goals and policies, Legal and ethical issues in data security, Security standards and compliance (ISO 27001, HIPAA, GDPR), Access control concepts: subjects, objects, permissions, security labels, Identification, authentication, and authorization

Module 2:

Cryptographic Foundations (9 Hours)
Symmetric key cryptography: DES, AES, modes of operation, Public key cryptography: RSA, ECC, Diffie-Hellman, Cryptographic hash functions: MD5, SHA-2, HMAC, Digital signatures and certificates, Public Key Infrastructure (PKI), Cryptanalysis and brute-force attacks, Secure key management and storage

Module 3:

Authentication, Authorization & Access Control Models (9 Hours)
User authentication methods: passwords, biometrics, OTPs, multi-factor authentication, Directory-based authentication: LDAP, Kerberos, Access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Access Control Lists (ACLs) and capabilities, Single Sign-On (SSO) and Federated Identity Management (OAuth, SAML, OpenID)

Module 4:

Secure Systems and Data Protection (9 Hours)
Security in operating systems and databases, File system and database access controls, Trusted Computing Base (TCB) and secure boot, Secure audit trails and logging, Malware types and protection: viruses, worms, ransomware, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Secure software development practices (SDL), Data leakage prevention (DLP), Cloud data protection techniques

Module 5:

Network and Cloud Security (9 Hours)
Network-level security: firewalls, VPNs, TLS, Secure network protocols: IPsec, HTTPS, DNSSEC, Web security issues: XSS, CSRF, SQL injection, clickjacking, Email security: PGP, S/MIME, DMARC, SPF, DKIM, Security in virtualized and multi-tenant environments, Cloud security controls and shared responsibility model, Zero Trust

Architecture (ZTA) and Identity-centric security, Real-world case studies of data breaches

Labs / Practicals:

Lab 1: Exploring file-level access control in Linux/Windows
Lab 2: Implementing symmetric encryption and decryption using AES
Lab 3: RSA encryption, key generation, and digital signatures
Lab 4: Password strength analysis and hashing with salting (bcrypt/sha256)
Lab 5: Configuring and testing LDAP or Kerberos authentication
Lab 6: Defining and enforcing RBAC/ABAC in a simulated environment
Lab 7: Setting up a basic VPN and securing communication using OpenSSL
Lab 8: Simulating SQL injection and XSS attacks with prevention techniques
Lab 9: Monitoring logs and detecting threats using Snort/Wireshark
Lab 10: Mini-project: Building a secure file-sharing or identity management system

Tools Used: OpenSSL, GPG, Wireshark, Snort, LDAP/Kerberos servers, OpenVPN, Burp Suite, Metasploit, Linux ACL, bcrypt/Hashcat, AWS IAM/GCP IAM, MySQL/MongoDB security features

References:

1. William Stallings – Cryptography and Network Security, Pearson.

2. Matt Bishop – Computer Security: Art and Science, Addison-Wesley.

3. Charles P. Pfleeger & Shari Lawrence Pfleeger – Security in Computing, Pearson.

4. Nina Godbole & Sunit Belapure – Cyber Security, Wiley India.

5. NIST Publications – SP 800 series for access control, cryptography, and cloud security.

6. OWASP – Web Security Testing Guide and Top 10 vulnerabilities.

7. RFCs and ISO Standards – TLS, IPsec, X.509, PKI, ISO 27001.

8. NPTEL – Information Security Courses by IIT Bombay/IIT Madras.

Course Code	Course Name	L	T	P	Cr
DCSE250051	Data Storage Technologies and Networks	3	0	2	4

Course Outcomes:

CO1. Describe storage system architecture, its elements, and characteristics.
CO2. Compare the intelligent storage systems and select one for a storage application.
CO3. Demonstrate storage virtualization using Xen or KVM
CO4. Demonstrate the functioning of SAN and NAS using open-source simulators.
CO5. Describe the mechanisms to secure storage infrastructure.

Module 1:

Introduction to Storage Systems and Data Centers: Introduction to Information Storage,Data, Types of Data,Big Data, Information, and Storage,Evolution of Storage Architecture, Data Center Infrastructure,Core Elements and Key Characteristics,Data Center Management,Data Center Environment,Application, DBMS, Host (Compute), Connectivity,Storage and Disk Drive Components,Disk Drive Performance, Host Access to Data, Direct-Attached Storage (DAS), Application-based Storage Design.
Disk Native Command Queuing (NCQ)
Introduction to Flash Drives

Module 2:

Intelligent Storage Systems and RAID Technologies: RAID (Redundant Array of Independent Disks),Implementation Methods, RAID Array Components,RAID Techniques and Levels,Performance Impacts and Comparisons,Intelligent Storage Systems,Components: Front End, Cache, Back End, Physical Disk,Storage Provisioning,Traditional vs Virtual Provisioning,Types of Intelligent Storage Systems,High-End and Midrange Storage Systems

Module 3:

Storage Virtualization and I/O Architecture: Server and I/O Architectures, Storage Hierarchy and Disk Fundamentals, Initiators and Targets,Reading and Writing to Storage Devices,Storage Sharing vs Data Sharing,Types of Storage, I/O Connectivity,Networking Fundamentals,Introduction to IT Clouds,Virtualization: Servers, Storage, and Networking,Virtualization and Storage Services,Data and Storage Access

Module 4:

Storage Networking Technologies – SAN, iSCSI, NAS, and Object-Based Storage: Fibre Channel SAN Overview, Evolution, and Components,FC Connectivity and Switched Fabric Ports, FC Architecture and Protocol Stack,FC Addressing, WWNs, FC Frame Structure, Flow Control, Classes of Service, Zoning, Topologies,IP SAN,iSCSI: Components, Host Connectivity, Topologies, Protocol Stack, FCIP: Protocol Stack, FCoE: I/O Consolidation and Components,NAS and Object-Based Storage: Network-Attached Storage (NAS),Introduction, Benefits, File Systems,Components and I/O Operation, Implementations: Unified NAS, Gateway NAS, Scale-Out NAS,NAS Protocols: NFS, CIFS,Object-Based and Unified Storage,Architecture and Components,Object Retrieval and Addressing, Use Cases and Content-Addressed Storage (CAS)

Module 5:

NAS and Object-Based Storage: Network-Attached Storage (NAS),Introduction, Benefits, File Systems, Components and I/O Operation, Implementations: Unified NAS, Gateway NAS, Scale-Out NAS,NAS Protocols: NFS, CIFS, Object-Based and Unified Storage,Architecture and Components,Object Retrieval and Addressing, Use Cases and Content-Addressed Storage (CAS)

Labs / Practicals:

Data Types & Storage Formats – Identify and store structured/unstructured data.

Data Center Design – Simulate components using diagrams/tools.
Disk Performance Test – Measure read/write speed using tools like iostat or CrystalDisk.
RAID Configuration – Set up RAID 0, 1, 5 using FreeNAS or mdadm.
Storage Provisioning – Simulate traditional vs virtual provisioning.
Cache Performance – Analyze speed with/without cache using FIO or DiskMark.
VM Storage Setup – Attach and manage virtual disks in VirtualBox/VMware.
Shared Storage Access – Use NFS or CIFS for storage sharing between VMs.
I/O Monitoring – Analyze disk I/O using iotop or Resource Monitor.
FC SAN Topology (Simulated) – Design and explain FC SAN components.
iSCSI Setup – Create iSCSI target/initiator and mount storage remotely.
NAS Configuration – Setup shared folders with FreeNAS.
Object Storage with MinIO – Upload/download files using REST API.
User Access Control – Create users and assign access in FreeNAS.
Storage Network Security – Simulate security for FC SAN/NAS/iSCSI.

References:

Books & Other Resources:
Textbooks:
1. "Information storage and management", EMC Education Services, 2nd edition, SAGE Publication
2. "Cloud and Virtual Data Storage Networking", Greg Schulz, CRC Press
Reference Books:
1. "Storage Networks: The Complete Reference, Robert Spalding", Publisher: McGraw-Hill Osborne Media ISBN: 0072224762, 9780072224764
2. "Storage area network essentials", Richard Barker, Paul Massiglia, Wiley

Course Code	Course Name	L	T	P	Cr
DCSE250069	Distributed Databases	3	0	2	4

Course Outcomes:

CO1. Understand principles of distributed databases, transparency levels, reference architectures, data fragmentation, constraints, and distributed database design approaches.
CO2. Apply query translation, equivalence transformations, and optimization strategies to process global queries efficiently in distributed environments.
CO3. Analyze distributed transaction management, ensuring atomicity and concurrency using timestamp ordering, deadlock handling, and optimistic control methods.
CO4. Evaluate reliability and security mechanisms in distributed databases, including commitment protocols, consistency, checkpoints, catalog management, and authorization techniques.
CO5. Explain distributed object database architectures, object management techniques, query processing, and transaction management in object-oriented distributed systems.

Module 1:

INTRODUCTION: Features of Distributed versus Centralized Databases, Principles of Distributed Databases, Levels Of
Distribution Transparency, Reference Architecture for Distributed Databases, Types of Data Fragmentation, Integrity Constraints in Distributed Databases, Distributed Database Design

Module 2:

QUERY PROCESSING: Translation of Global Queries to Fragment Queries, Equivalence transformations for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parametric Queries. Optimization of Access Strategies, A Framework for Query Optimization, Join Queries, General Queries

Module 3:

TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL: The Management of Distributed Transactions, A Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions Concurrency Control, Foundation of Distributed Concurrency Control, Distributed Deadlocks, Concurrency Control based on Timestamps, Optimistic Methods for Distributed Concurrency Control.

Module 4:

RELIABILITY AND SECURITY IN THE DISTRIBUTED DATABASES: Reliability, Basic Concepts, Non blocking Commitment Protocols, Reliability and concurrency Control, Determining a Consistent View of the Network, Detection and Resolution of Inconsistency, Checkpoints and Cold Restart, Distributed Database Administration, Catalog Management in Distributed Databases,
Authorization and Protection

Module 5:

DISTRIBUTED OBJECT DATABASE MANAGEMENT SYSTEMS: Architectural Issues, Alternative Client/Server Architectures, Cache Consistency, Object Management,
Object Identifier Management, Pointer Swizzling, Object Migration, Distributed Object Storage, Object Query Processing, Object Query Processor Architectures, Query Processing Issues, Query Execution, Transaction Management, Transaction Management in Object DBMSs, Transactions as Objects

Labs / Practicals:

Lab 1: Centralized vs Distributed Database Setup
Objective: Set up a simple centralized database and a simulated distributed database (using multiple schemas or databases).
Tools: PostgreSQL/MySQL with multiple instances or schema-based simulation.
Lab 2: Data Fragmentation and Allocation
Objective: Implement horizontal and vertical fragmentation manually.
Task: Simulate fragmentation on student or employee datasets and allocate fragments to different nodes.
Lab 3: Designing Distributed Database Schema
Objective: Design a normalized schema with fragmentation and distribution.
Task: Document and simulate fragment allocation and distribution strategies.
Lab 4: Translating Global Queries to Fragment Queries
Objective: Write SQL queries on a global schema and manually transform them for fragments.
Task: Implement on horizontally and vertically fragmented tables.
Lab 5: Distributed Join and Aggregation
Objective: Perform joins and aggregate functions on distributed data.
Task: Simulate joins using fragments and aggregate results from different nodes.
Lab 6: Query Optimization Techniques
Objective: Implement simple cost-based query optimization techniques.
Task: Compare performance of join queries using different join orders.
Lab 7: Simulating Distributed Transactions
Objective: Simulate a transaction that involves multiple fragments on different nodes.
Task: Implement commit and rollback across nodes.
Lab 8: Concurrency Control Mechanisms
Objective: Simulate two-phase locking and timestamp ordering protocols.
Task: Demonstrate read-write conflicts and resolution strategies.
Lab 9: Deadlock Detection in Distributed Transactions
Objective: Implement a simple wait-for graph to detect distributed deadlocks.
Task: Create and resolve deadlock scenarios with sample transactions.
Lab 10: Non-blocking Commit Protocol (2PC/3PC) Simulation
Objective: Simulate Two-Phase and Three-Phase Commit protocols.
Task: Use pseudocode or programming language like Python/Java to demonstrate commit process.
Lab 11: Security in Distributed Databases
Objective: Implement basic user-based access control.
Task: Create roles, users, and access privileges on distributed fragments.
Lab 12: Backup and Recovery in DDBMS
Objective: Simulate a checkpoint and recovery operation.
Task: Use tools like pg_dump or MySQLdump for manual checkpoint and cold start.
Lab 13: Object Database Creation
Objective: Use an object-oriented database like db4o or MongoDB (document-based) to simulate object storage

References:

REFERENCE BOOKS:
1. Data Base Management System; Leon & Leon; Vikas Publications
2. Introduction to Database Systems; Bipin C Desai; Galgotia
WEB REFERENCES:
1. https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_databases.htm
2. https://www.geeksforgeeks.org/distributed-database-system/
TEXT BOOKS:
1. Distributed Databases - Principles and Systems; Stefano Ceri; Guiseppe Pelagatti; Tata McGraw Hill; 1985.
2. Fundamental of Database Systems; Elmasri & Navathe; Pearson Education; Asia Database System Concepts; Korth & Sudarshan; TMH
3. Principles of Distributed Database Systems; M. Tamer Özsu; and Patrick Valduriez Prentice Hall

Course Code	Course Name	L	T	P	Cr
DCSE250070	Distributed Systems	3	1	0	4

Course Outcomes:

CO1: Understand the architecture, characteristics, and goals of distributed systems.

CO2: Implement communication, synchronization, and coordination algorithms in distributed environments.

CO3: Analyze consistency models, replication strategies, and fault tolerance mechanisms.

CO4: Apply distributed system principles in real-world contexts like cloud platforms, distributed file systems, and blockchains.

CO5: Design and evaluate distributed applications using appropriate programming paradigms, tools, and frameworks.

Module 1:

Introduction to Distributed Systems (9 Hours)

Definition and goals of distributed systems, Characteristics of distributed systems (concurrency, no global clock, independent failure), Architectural models: client-server, peer-to-peer, layered architectures, Middleware and its role in distributed systems, Examples of distributed systems: WWW, distributed databases, cloud platforms, Scalability, transparency, and openness in system design, Challenges in distributed computing

Module 2:

Communication and Coordination (9 Hours)

Interprocess communication: message passing, sockets, RPC (Remote Procedure Calls), RMI (Remote Method Invocation), RESTful services, Protocols: TCP vs UDP, Group communication and multicast, Logical clocks and vector clocks, Clock synchronization algorithms: Christian's algorithm, Berkeley algorithm, Lamport timestamps, Mutual exclusion: centralized, distributed, and token ring algorithms, Election algorithms: Bully and Ring algorithms

Module 3:

Consistency, Replication, and Fault Tolerance (9 Hours)

Consistency models: strict, linearizability, sequential, causal, eventual, Replication strategies: active vs passive replication, primary-backup, quorum-based protocols, Fault tolerance: failure models, reliable communication, agreement in faulty systems, Byzantine failures and Paxos algorithm, Distributed commit protocols: two-phase and three-phase commit, Checkpointing and rollback recovery, Gossip protocols

Module 4:

Distributed File Systems, Cloud, and Edge Computing (9 Hours)

Distributed File Systems: Google File System (GFS), Hadoop Distributed File System (HDFS), Distributed hash tables (DHT), Cloud computing concepts: virtualization, resource provisioning, public/private/hybrid clouds, Edge computing and fog computing overview, Case studies: Amazon AWS, Google Cloud, Kubernetes, Cloud-native microservices and service discovery, Containers and orchestration: Docker and Kubernetes basics

Module 5:

Blockchain and Real-World Distributed Applications (9 Hours)

Blockchain fundamentals: distributed ledger, transactions, consensus mechanisms, Proof of Work, Proof of Stake, and Byzantine Fault Tolerance (BFT), Smart contracts and distributed apps (DApps), Peer-to-peer file sharing systems (BitTorrent, IPFS), Real-world applications: distributed databases (Cassandra, MongoDB), distributed pub-sub systems (Apache Kafka), Design of scalable real-time chat/messaging systems, Case study: Ethereum, Hyperledger Fabric

Labs / Practicals:

Lab 1: Socket programming and message passing in Python/Java
Lab 2: Implementing RPC/RMI with JSON or gRPC
Lab 3: Logical and vector clock simulation
Lab 4: Distributed mutual exclusion algorithm implementation
Lab 5: Leader election using Bully and Ring algorithms
Lab 6: Simulating replication with primary-backup protocol
Lab 7: Implementing a simplified two-phase commit protocol
Lab 8: Hands-on with HDFS and Hadoop file operations
Lab 9: Building a basic blockchain prototype with Python
Lab 10: Mini-project: Distributed chat system or file sync system

Tools Used: Python, Java, gRPC, Apache Hadoop, Docker, Kubernetes (Minikube), MongoDB, Kafka, Ethereum (Ganache), VS Code, Postman

References:

1. Andrew S. Tanenbaum & Maarten van Steen – Distributed Systems: Principles and Paradigms, Pearson Education.

2. George Coulouris et al. – Distributed Systems: Concepts and Design, Pearson.

3. Kai Hwang – Distributed and Cloud Computing, Morgan Kaufmann.

4. Martin Kleppmann – Designing Data-Intensive Applications, O'Reilly.

5. Gaurav Vaish – Getting Started with NoSQL, Packt.

6. NPTEL – Distributed Systems by IIT Bombay / IIT Madras

7. Ethereum Docs – Ethereum Whitepaper and Developer Guide

8. Apache Docs – Kafka, Hadoop, HDFS Official Documentation

Course Code	Course Name	L	T	P	Cr
DCSE250084	GPU Computing	3	1	0	4

Course Outcomes:

CO1: Understand GPU architecture, accelerators, and parallel programming models for high-performance computing.

CO2: Design and implement GPU kernels using CUDA/OpenCL with appropriate thread hierarchies and launch configurations.

CO3: Apply GPU memory hierarchy concepts to optimize performance in large-scale computing tasks.

CO4: Develop parallel algorithms with synchronization and atomic operations using libraries and functions.

CO5: Analyze, debug, and optimize GPU programs using profiling tools and asynchronous streams.

Module 1:

Introduction to GPU Computing (13 Hours)

History and evolution of GPU computing, GPU architecture: cores, streaming multiprocessors, and warp scheduling, Clock speeds and performance metrics, Comparison between CPU and GPU architectures, Heterogeneous computing and hardware accelerators, Introduction to parallel programming paradigms: CUDA, OpenCL, OpenACC, Kernel programming and launch configurations, Thread hierarchy: threads, thread blocks, grids, Warps and wavefronts, Streaming multiprocessors (SMs), 1D, 2D, and 3D thread mapping, Device properties and capability querying, Writing, compiling, and running basic GPU programs

Module 2:

GPU Memory Architecture and Programming (7 Hours)

Global (DRAM), shared (local), private (registers), constant, and texture memory, Memory coalescing and access patterns, Parameter passing and pointer usage in device functions, Static and dynamic memory allocation, Arrays and multidimensional arrays in device code, Memory copying between host and device, Matrix operations using different memory types, Performance optimization based on memory access

Module 3:

Synchronization and GPU Functions (10 Hours)

Memory consistency and synchronization models, Local and global barriers (__syncthreads() and beyond), Memory fences and atomic operations, Parallel reduction and prefix sum, Worklists, linked lists, and other concurrent data structures on GPU, Thread divergence and control flow, Kernel functions, device functions, host functions, Developing and using libraries like Thrust, cuBLAS, and cuDNN

Module 4:

Profiling, Debugging, and Asynchronous GPU Programming (7 Hours)

Debugging GPU programs using cuda-gdb and cuda-memcheck, Profiling and performance tools: Nsight Systems, Nsight Compute, nvprof, Identifying and fixing performance bottlenecks, CUDA streams and concurrency, Asynchronous kernel execution, Data transfer overlapping and task parallelism, Event handling and event-based synchronization, Pitfalls in asynchronous and concurrent programming, Best practices for performance tuning

Module 5:

Advanced Concepts and Applications (7 Hours)

Multi-GPU programming and peer-to-peer memory access, Unified memory and memory management enhancements, Cooperative groups and persistent threads, Task scheduling and grid-stride loops, Image processing, Deep learning acceleration, Simulation and scientific computing, Reinforcement learning and game engines, Case studies of GPU-accelerated systems, Future trends: tensor cores, GPU virtualization, and edge GPU computing

Labs / Practicals:

Lab 1 : Setting up the CUDA environment and writing the first GPU program
Lab 2 : Matrix addition and multiplication using thread hierarchy
Lab 3 : Implementing memory coalescing and shared memory optimization
Lab 4 : Image filtering using convolution kernels
Lab 5 : Writing prefix sum and reduction kernels
Lab 6 : Implementing atomic operations and synchronization
Lab 7 : Profiling and debugging GPU kernels
Lab 8 : Using streams to overlap memory transfer and execution
Lab 9 : GPU-accelerated real-world task (e.g., neural network inference)
Lab 10 : Final mini-project presentation and optimization report

Tools Used: CUDA Toolkit, Nsight Systems, Nsight Compute, cuBLAS, cuDNN, OpenCL SDKs, Visual Studio Code with CUDA extensions

References:

David B. Kirk & Wen-mei W. Hwu – Programming Massively Parallel Processors, Morgan Kaufmann.

Shane Cook – CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, Elsevier.

Benedict Gaster et al. – Heterogeneous Computing with OpenCL, Morgan Kaufmann.

NVIDIA Developer Documentation – CUDA C Programming Guide, official guide.

NPTEL – Parallel Programming in CUDA, by IIT Bombay / IIT Madras.

Nathan Whitehead & Robert G. – CUDA by Example: An Introduction to General-Purpose GPU Programming.

Jeffrey L. Vetter (Ed.) – Contemporary High Performance Computing: From Petascale toward Exascale, CRC Press.

Official Profiling Tools – Nsight Systems, Nsight Compute, CUDA-GDB, nvprof.

Course Code	Course Name	L	T	P	Cr
DCSE250124	Web Analytics and Development	3	0	2	4

Course Outcomes:

CO1: Understand web analytics fundamentals, including definitions, processes, key terms, categories, platforms, evolution, need, advantages, and limitations.
CO2: Explain and compare data collection methods, and apply surveys, heuristic evaluations, and site visits to analyze user behavior.
CO3: Interpret key web metrics and KPIs, optimize website performance, and apply Google Analytics for reporting and campaign improvement.
CO4: Understand Web Analytics 2.0 concepts and apply competitive intelligence techniques to analyze site traffic trends and business opportunities.
CO5: Use Google Analytics for traffic analysis, evaluate privacy and performance issues, and understand supporting technologies like HTTP, cookies, and logs.

Module 1:

Introduction: Definition, Process, Key terms: Site references, Keywords and Key phrases; building block terms: Visit characterization terms, Content characterization terms, Conversion metrics; Categories: Offsite web, On site web; Web analytics platform, Web analytics evolution, Need for web analytics, Advantages, Limitations.

Module 2:

Data Collection: Clickstream Data: Web logs, Web Beacons, JavaScript tags, Packet Sniffing; Outcomes Data: E-commerce, Lead generation, Brand/Advocacy and Support; Research data: Mindset, Organizational structure, Timing; Competitive Data: Panel-Based measurement, ISP-based measurement, Search Engine data.
Qualitative Analysis: Heuristic evaluations: Conducting a heuristic evaluation, Benefits of heuristic evaluations; Site Visits: Conducting a site visit, Benefits of site visits; Surveys: Website surveys, Post-visit surveys, Creating and running a survey, Benefits of surveys.

Module 3:

Web Analytic fundamentals: Capturing data: Web logs or JavaScripts tags, Separate data serving and data capture, Type and size of data, Innovation, Integration, Selecting optimal web analytic tool, Understanding clickstream data quality, Identifying unique page definition, Using cookies, Link coding issues.
Web Metrics: Common metrics: Hits, Page views, Visits, Unique visitors, Unique page views, Bounce, Bounce rate, Page/visit, Average time on site, New visits; Optimization(e-commerce, non e-commerce sites): Improving bounce rates, Optimizing adwords campaigns; Real time report, Audience report, Traffic source report, Custom campaigns, Content report, Google analytics, Introduction to KPI, characteristics, Need for KPI, Perspective of KPI, Uses of KPI.

Module 4:

Web analytics 2.0: Web analytics 1.0, Limitations of web analytics 1.0, Introduction to analytic 2.0, Competitive intelligence analysis : CI data sources, Toolbar data, Panel data ,ISP data, Search engine data, Hybrid data, Website traffic analysis: Comparing long term traffic trends, Analyzing competitive site overlap and opportunities.

Module 5:

Google Analytics: Brief introduction and working, Adwords, Benchmarking, Categories of traffic: Organic traffic, Paid traffic; Google website optimizer, Implementation technology, Limitations, Performance concerns, Privacy issues.
Relevant technologies:Internet & TCP/IP, Client / Server Computing, HTTP (HyperText Transfer Protocol),

Server Log Files & Cookies, Web Bugs.

Labs / Practicals:

Laboratory Work:to analyzing the web for various functionalities given in the subject and using various tools and technologies to do the experimentation. It also involves installation and working on tools and technologies in this domain.

References:

1. Clifton B., Advanced Web Metrics with Google Analytics, Wiley Publishing, Inc. (2010), 2nd ed.
2. Kaushik A., Web Analytics 2.0 The Art of Online Accountability and Science of Customer Centricity, Wiley Publishing, Inc. (2010),1st ed.
3. Sterne J., Web Metrics:Proven methods for measuring web site success, John Wiley and Sons (2002),1sted

Course Code	Course Name	L	T	P	Cr
DOAI250001	Advanced Data Analytics	3	0	2	4

Course Outcomes:

CO1: Understand fundamental concepts and methodologies in data analytics, including data preprocessing, exploratory data analysis (EDA), dimensionality reduction, and predictive modeling.

CO2: Apply machine learning algorithms for supervised, unsupervised, and reinforcement learning tasks in the context of data analytics.

CO3: Demonstrate proficiency in programming languages and tools commonly used in data analytics, including Python, R, and associated libraries.

CO4: Analyze large-scale datasets using distributed big data frameworks, including Apache Hadoop and Apache Spark.

CO5: Interpret results through appropriate visualization techniques and apply insights in practical domains such as social media analytics, fraud detection, and recommender systems.

Module 1:

Module 1: Introduction to Data Analysis

Introduction to Data Analysis: Concepts, Analytics Process Models, Analytical Model Requirements, Data Analytics Lifecycle Overview, Data Collection Methods, Sampling Techniques, Data Preprocessing, Handling Missing Data, Dimensionality Reduction Techniques (PCA, t-SNE, UMAP), Introduction to Feature Engineering.

Module 2:

Module 2: Predictive and Descriptive Analytics

Predictive Analytics: Linear Regression, Decision Trees, Neural Networks, Dimensionality Reduction through PCA.
Descriptive Analytics: Association Rule Mining (Apriori, FP-Growth), Market Basket Analysis, Sequential Pattern Mining, Clustering Techniques (K-Means, Hierarchical Clustering, DBSCAN).
Introduction to Reinforcement Learning Concepts for Analytics.

Module 3:

Big Data Framework and Analytics

Overview of Big Data Technologies, Challenges in Big Data Acquisition, Management, and Processing.
Big Data Architectures and Frameworks: Hadoop Ecosystem, HDFS, YARN.
Spark Ecosystem: RDDs, DataFrames, Spark SQL.
NoSQL Databases (MongoDB, Cassandra): Data Models and Applications.
Big Data Trends: Streaming, IoT Data, Real-Time Analytics.

Module 4:

Data Analysis Using R

Introduction to R: Data Import, Data Cleaning, Data Exploration, Visualization with ggplot2.
Descriptive Statistics in R, Statistical Methods for Evaluation, Exploratory Data Analysis (EDA), Handling Dirty Data, Visualization Techniques for Single and Multiple Variables, Data Presentation versus Data Exploration.

Module 5:

Big Data Techniques and Applications

MapReduce Paradigm, Hadoop System and Data Processing, Advanced Analytics with Spark Streaming, Real-Time Processing Use Cases.
Applications and Case Studies: Social Media Analytics, Recommender Systems, Fraud Detection—Introduction to Time Series Analytics: ARIMA Models.

Labs / Practicals:

1. Implement Principal Component Analysis (PCA) for dimensionality reduction and visualization.

2. Build a Linear Regression model for predictive analytics (e.g., housing price prediction).

3. Develop Decision Tree models for classification problems (Iris dataset or similar).

4. Apply Association Rule Mining (Apriori) on transactional data.

5. Implement K-Means Clustering for unsupervised grouping and visualize clusters.

6. Setup and operate Hadoop for data processing using MapReduce.

7. Data analysis and visualization in R with real datasets (EDA, plotting).

8. Perform exploratory data analysis (EDA) using R for multivariate datasets.

9. Analyze social media data using Apache Spark and perform sentiment analysis.

10. Implement Time Series Forecasting using ARIMA.

References:

1. Bart Baesens, Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends, John Wiley & Sons, 2014.

2. David Dietrich, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, John Wiley & Sons, 2015.

3. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann / Elsevier, 2011.

4. Margaret H. Dunham, Data Mining: Introductory and Advanced Topics, Pearson, 2011.

5. Wes McKinney, Python for Data Analysis, 2nd Edition, O'Reilly Media, 2018.

6. Hadley Wickham, Garrett Grolemund, R for Data Science, O'Reilly Media, 2017.

7. Official Documentation:

Apache Hadoop: https://hadoop.apache.org/

Apache Spark: https://spark.apache.org/docs/latest/

R Language: https://cran.r-project.org/

Course Code	Course Name	L	T	P	Cr
DOAI250009	Big Data Analytics	3	0	2	4

Course Outcomes:

CO1: Describe big data characteristics, challenges, and applications across industries and domains.

CO2: Evaluate and apply NoSQL databases and distributed storage architectures for big data management.

CO3: Install, configure, and operate Hadoop and HDFS for batch-oriented big data processing.

CO4: Develop and implement data analysis workflows using Hadoop ecosystem tools including Hive, Pig, and HBase.

CO5: Design and implement real-time data processing and analytics using Spark Streaming and PySpark.

CO6: Apply visualization and regression techniques for analytical insights from large-scale data.

Module 1:

Introduction to Big Data Ecosystem

Big Data Characteristics: Volume, Variety, Velocity, Veracity, and Value, Business applications and case studies of big data analytics, Challenges in conventional systems, Big Data Analytics Lifecycle, Tools and Technologies in Big Data, Hadoop vs. Traditional Systems, NoSQL Overview (Key-Value, Document, Columnar, Graph databases), Basics of MongoDB and Cassandra for big data storage.

Module 2:

Hadoop Framework for Big Data

History and Evolution of Hadoop, Hadoop Distributed File System (HDFS): Architecture, Internals, and Operations, Block-level Storage, Fault Tolerance, MapReduce Paradigm: Concepts, Anatomy of a MapReduce Job, Failures, Scheduling, Shuffle, Sort, and Task Execution, Input / Output Formats in MapReduce, Hadoop Streaming and Integrations, Developing MapReduce Applications with Java and Python APIs.

Module 3:

Data Analysis with Hadoop Ecosystem

Hive: Data Modeling, Data Types, File Formats, HiveQL (DDL, DML, DQL), Querying Structured Data, Hive Services and Optimization.
Pig: Data Flow Language, Pig Latin Programming, Grunt Shell, Data Models, Operators, Joins, Developing and Testing Pig Scripts.
HBase: Architecture, Data Model, Integration with Hadoop, Use Cases.
Introduction to Data Lakes using Hadoop.
Hands-on with Hadoop Ecosystem for ETL pipelines.

Module 4:

Real-Time Big Data Analytics and Streaming

Stream Processing Fundamentals, Stream Data Models, Stream Architecture, Concepts of Sampling, Filtering, Counting in Streams, Real-time Sentiment Analysis, Stock Market Analysis using Streams, Introduction to Apache Kafka for Streaming Data Ingestion, Spark Basics: RDD, DataFrames, Spark SQL, In-Memory Computing, PySpark APIs, Spark Streaming for Real-Time Processing, Real-Time Analytics Applications, Case

Studies.

Module 5:

Analytical Models and Visualization for Big Data

Regression Techniques: Simple Linear Regression, Multiple Linear Regression, Model Interpretation for Large-Scale Data, Feature Engineering for Big Data, Visualization Techniques: Interactive Dashboards, Graphical Tools for Big Data (Tableau / PowerBI concepts), Visual Analytics: Trends, Patterns, Correlations, Interaction Models, Real-World Applications of Big Data Visualizations.

Labs / Practicals:

1. Setting up Hadoop, HDFS, and YARN on local and cloud-based environments (AWS / Azure / GCP).

2. Developing basic MapReduce jobs using Java and Python.

3. Implementing ETL pipelines using Hive and Pig.

4. Performing analytics with HBase and Hive.

5. Real-time data streaming with Apache Kafka and Spark Streaming (using PySpark).

6. Implementing basic ML pipelines over Spark (MLlib).

7. Visualizing Big Data using interactive tools (Matplotlib, Plotly, Tableau / PowerBI).

8. Group Project: Building and deploying an end-to-end Big Data Analytics pipeline on a real-world dataset.

References:

1. Tom White, Hadoop: The Definitive Guide, 4th Edition, O'Reilly Media, 2015.

2. Anand Rajaraman, Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 3rd Edition, 2020.

3. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & Sons, 2012.

4. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, 3rd Edition, Morgan Kaufmann / Elsevier, 2011.

5. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, Learning Spark: Lightning-Fast Big Data Analysis, 2nd Edition, O'Reilly Media, 2020.

6. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Harness the Power of Big Data – The IBM Big Data Platform, Tata McGraw Hill, 2012.

7. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

8. Da Ruan, Guoqing Chen, Etienne E. Kerre, Geert Wets, Intelligent Data Mining, Springer, 2007.

Official Documentation (Highly Recommended for Labs / Practical Work):

9. Official documentation for Hadoop: https://hadoop.apache.org/

10. Official documentation for Spark: https://spark.apache.org/docs/latest/

11. Official documentation for Hive: https://cwiki.apache.org/confluence/display/Hive/Home

12. Official documentation for Pig: https://pig.apache.org/

13. Official documentation for HBase: https://hbase.apache.org/

14. Official documentation for Kafka: https://kafka.apache.org/documentation/

15. Official documentation for PySpark: https://spark.apache.org/docs/latest/api/python/

Course Code	Course Name	L	T	P	Cr
DOAI250011	Business Analytics	3	0	2	4

Course Outcomes:

CO1: Students will demonstrate foundational knowledge of business analytics, including the four primary methods: descriptive, diagnostic, predictive, and prescriptive analytics.
CO2: Students will be able to collect, validate, manage, and present large-scale structured and unstructured data, applying statistical inference and big data visualization techniques.
CO3: Students will demonstrate the ability to apply analytical and modeling techniques—including optimization, forecasting, count data regression, and survival analysis—for solving real-world business problems.
CO4: Students will develop technical skills in machine learning, deep learning, and R programming for predictive and prescriptive decision-making.
CO5: Students will effectively analyze and translate domain-specific datasets (e.g., healthcare, retail, finance, social media) into actionable business insights using appropriate tools and case-based applications.

Module 1:

Business analytics: Four primary methods of business analysis-Descriptive, Diagnostic, Predictive, Prescriptive. Business Analytics & Data Science. Growing Role of Business Analytics. Data Analytics as a Predictive Tool. Data Analytics as an Evaluative Tool. Data Analytics as an Evaluative Tool.

Module 2:

Essentials of Business Analytics: Introduction to the Methodology-Data Collection, Challenges in Data Collection, Data Collection Validation and Presentation. Big Data ManagementElements of Big Data, Characteristics of Big Data. Cloud Computing for Big Data. Data Visualization-Six Meta-Rules for Data Visualization. Statistical Methods for Basic Inferences.

Module 3:

Modelling Methods: Decision Making Under Uncertainty, Methods in Optimization, Forecasting Analytics, Count Data Regression, Survival Analysis, Machine Learning- Supervised &Unsupervised. Deep Learning. Introduction to R.

Module 4:

Decision Analysis: Formulating Decision Problems, Decision Strategies with the without Outcome Probabilities, Decision Trees, The Value of Information, Utility and Decision Making.

Module 5:

Applications: Retail Analytics, Marketing Analytics, Financial Analytics, Social media and Web Analytics, Healthcare Analytics, Supply Chain Analytics. Case Study-Health Insurance Industry, Airline Corp, Info Media Solutions.

Labs / Practicals:

Tools: Excel / Python / R
Lab 1: Data Exploration with Descriptive & Diagnostic Analytics
Perform summary statistics (mean, median, variance) on a sales dataset
Create basic dashboards in Excel or Python (Pandas + Matplotlib)
Lab 2: Business Use Case Mapping
Identify and classify real-world business problems into: Descriptive, Predictive, Prescriptive, Diagnostic
Tools: Python, R, Excel, Tableau/Power BI
Lab 3: Data Collection and Cleaning
Import raw data (CSV/JSON/API)

Handle missing values, duplicates, outliers using Python/R
Lab 4: Data Visualization Principles
Use Tableau or Power BI to build dashboards with filters and KPIs
Apply “Six Meta-Rules” in creating clean, interpretable visualizations
Lab 5: Basic Inferential Statistics
Perform t-tests, chi-square tests, correlation analysis in R or Python
Interpret business implications from the results
Tools: R, Python (Statsmodels, Scikit-learn)
Lab 6: Decision Making Under Uncertainty
Use expected value or payoff matrices for decision trees
Lab 7: Optimization Problem Solving
Solve linear programming problems using PuLP in Python or lpSolve in R
Lab 8: Forecasting & Time Series Analysis
Use ARIMA model on time series data (e.g., stock prices, sales)
Forecast future values and validate with RMSE
Lab 9: Count Data & Survival Analysis
Apply Poisson/Negative Binomial Regression for count data
Use Kaplan-Meier estimators for survival analysis
Tools: Python (Scikit-learn, TensorFlow/Keras), R
Lab 10: Supervised Learning
Implement Classification (Logistic Regression, Decision Tree) on customer churn dataset
Lab 11: Unsupervised Learning
Apply K-Means Clustering on a customer segmentation dataset
Lab 12: Deep Learning Introduction
Build a basic neural network for regression/classification using Keras
Lab 13: R Programming
Data manipulation using dplyr
Visualization using ggplot2
Tools: R / Python / Tableau / Power BI
Lab 14: Web & Social Media Analytics
Extract Twitter/Facebook data via APIs
Perform sentiment analysis on user comments
Lab 15: Healthcare or Insurance Case Study
Use survival models or clustering on patient/insurance data
Interpret policy recommendations

References:

1. Business analytics Principles, Concepts, and Applications by Marc J. Schniederjans, Dara G. Schniederjans, Christopher M. Starkey, Pearson FT Press.
2. Business Analytics by James Evan
3. EMC Education Services.(2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Wiley.
4. Michael Minelli, Michele Chambers, Ambiga Dhiraj. (2013). Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses. Wiley.
5. Bart Baesens. (2014). Analytics in a Big Data World”, The Essential Guide to Data Science and its Applications, Wiley, First edition.

Course Code	Course Name	L	T	P	Cr
DOAI250019	Data Mining and Warehousing	3	0	2	4

Course Outcomes:

CO1: Understand and apply data warehousing concepts and architectures.

CO2: Perform clustering, classification, and prediction for pattern discovery.

CO3: Address issues related to data streams and implement stream mining techniques.

CO4: Analyze and apply web mining methodologies for real-world applications.

CO5: Explore current trends and advancements in distributed warehousing and pattern mining.

Module 1:

Data warehousing components, Building a data warehouse, Data warehouse architecture, DBMS schemas for decision support, Data extraction, cleanup, and transformation tools, Metadata, Reporting, Query tools and applications, Online Analytical Processing (OLAP), OLAP and multidimensional data analysis.

Module 2:

Data mining functionalities, Data preprocessing: cleaning, integration, transformation, reduction, discretization, Concept hierarchy generation, Architecture of typical data mining systems, Classification of data mining systems, Efficient and scalable frequent itemset mining methods, Mining various kinds of association rules, From association mining to correlation analysis, Constraint-based association mining.

Module 3:

Issues in classification and prediction, Classification methods: Decision Trees, Bayesian classification, Rule-based classification, Neural networks: backpropagation, Support Vector Machines (SVM), Associative classification, Lazy learners, Other classification methods, Prediction techniques, Accuracy and error measures, Evaluating classifiers/predictors, Ensemble methods, Model selection.

Module 4:

Types of data in clustering, Categorization of major clustering methods, Partitioning methods, Hierarchical methods, Density-based methods, Grid-based methods, Model-based clustering methods, Clustering high-dimensional data, Constraint-based clustering, Outlier analysis.

Module 5:

Mining objects, Spatial data mining, Multimedia data mining, Text mining, Mining the World Wide Web, Multidimensional analysis of complex data, Descriptive mining of complex data objects.

Labs / Practicals:

1. Design a Data Warehouse Schema

Create Star Schema and Snowflake Schema for business data (sales, e-commerce, etc.).

2. ETL Process Implementation

Perform Extract, Transform, Load (ETL) operations on structured data using SQL-based tools (MySQL / PostgreSQL).

3. Association Rule Mining

Apply the Apriori Algorithm for mining frequent itemsets and generating association rules from transactional datasets.

4. Classification Using Decision Trees

Build, visualize, and evaluate Decision Tree Classifiers on datasets like Iris or Titanic.

5. Clustering Using K-Means

Apply K-Means Clustering on multi-dimensional datasets and visualize the results.

6. Data Preprocessing & Dimensionality Reduction

Handle missing values, normalize data, and apply Principal Component Analysis (PCA).

7. OLAP Operations and Cube Construction

Perform OLAP operations (Roll-up, Drill-down, Slice, Dice) using sample datasets.

8. Web Mining Basics

Analyze web log files for pattern discovery (frequent user paths, sessions).

References:

1. Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, Third Edition, Elsevier, 2011.

2. Alex Berson, Stephen J. Smith, Data Warehousing, Data Mining & OLAP, Tata McGraw-Hill, 10th Reprint, 2007.

3. K.P. Soman, Shyam Diwakar, V. Ajay, Insight into Data Mining: Theory and Practice, PHI, 2006.

4. G.K. Gupta, Introduction to Data Mining with Case Studies, PHI, 2006.

5. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education, 2007.

Course Code	Course Name	L	T	P	Cr
DOAI250021	Data Science	3	0	2	4

Course Outcomes:

CO1: Describe the data science life cycle, key concepts, and the role of Python in data acquisition, preprocessing, and visualization.
CO2: Apply core Python programming constructs and data structures (lists, dictionaries, tuples, stacks, queues, data frames) for effective data manipulation and analysis.
CO3: Develop and utilize user-defined functions and classes to implement modular and object-oriented programming techniques in data science workflows.
CO4: Analyze and handle data using file operations, pandas data frames, and essential machine learning libraries such as NumPy, Scikit-learn, and Matplotlib.
CO5: Implement and evaluate machine learning models using Scikit-learn for classification, regression, clustering, and present insights through case studies like spam detection and customer segmentation.

Module 1:

Data Science Overview: Data Science life cycle; data acquisition, public data repositories, pre-processing, data visualization, Machine Learning algorithms, pattern evaluation, knowledge representation, analytical strategies; Python fundamentals: data types, mutable & immutable types, operators & expressions, standard library modules, the flow of control, etc.

Module 2:

Python data structures: String manipulation: string operators, slices, methods; List manipulation: creating & accessing the list, list operations, true copy of lists, nested lists; Tuples: creation, accessing, modifying tuples, nested tuples, methods & functions, tuple slicing; Dictionaries: key-value pairs, creation, accessing, modifying dictionaries, Stack, Queue, data frame, etc.

Module 3:

User-defined functions & Classes: modules in Python, importing python modules, name resolution, module aliasing, package/library, locating modules, using Python's standard library, user-defined functions: parameters in functions, passing arrays/tuples/dictionary, and lists to functions, function call, the scope of variables, dynamic typing and dynamic binding, Creating and instantiating classes with attributes and methods, public v/s non-public members, Inheritance, Polymorphism, etc.

Module 4:

Data Handling: data file operations: opening & closing a file, reading/writing/appending a file, relative and absolute paths, standard file streams, binary file operations, CSV file handling & pickling, concatenating multiple CSV files/text files, panada's DataFrame, convert pandas data frames to numpy arrays/CSV, etc; introduction to various ML libraries: NumPy, Pandas, Scikit-learn, SciPy, Statsmodels, MatplotLib, Seaborn, Sympy, etc. and their use cases.

Module 5:

Scikit-learn library: using sci-kit-learn modules for classification e.g. Support Vector Classifier & Regressor, CART, clustering (k-means), data reduction (PCA), data visualization (box plot), etc. Web Scraping with Beautiful Soup, use cases and case studies: spam detection, drug response, stock price prediction, customer segmentation, etc.

Labs / Practicals:

1. Drawing different patterns e.g. Pascal triangle, Pyramid of '*', etc. using loop and branching statements
2. Creating a Python class, instantiate 10 objects, save them in csv file, re-open and load them in Python workspace
3. Sorting a student database based on student identification numbers by reading it from csv file

4. Problem related to classification using CART, SVC, K-NN, etc
5. Problem e.g. Market-Basket analysis, related to association mining using Apriori, FPGrowth, etc.
6. Problem related to clustering e.g. image segmentation using K-means,
7. Problem related to regression analysis e.g. electricity load prediction during peak hours using SVR, CART, etc.
8. Problems related to anomaly detection e.g.
9. Training a Machine learning model for a real-life dataset with good practices such as Train/Test split, 10-fold cross validation, reporting performance with RoC plots etc. over a separate test dataset
10. Using all plot libraries to do data pre-processing & visualization e.g. Box plot, Bar chart, probability density function plot, etc.
11. Understanding data processing, reading, pre-processing, cleaning data set, generating Train/Test/Validation datasets, label encoding, one-hot encoding, etc. using Sklearn
12. Importing & using various ML libraries: NumPy, Pandas, Scikit-learn, SciPy, Statsmodels, MatplotLib, Seaborn, Sympy, etc.
13. Creating and using NumPy Array and converting it into other types
14. Creation and usage of panda's data frame and converting it into other types
15. Using various data visualization techniques such as probability density function, box plot, etc.
16. CSV file handling & pickling, concatenating multiple CSV files/text files
17. Implementing a regression algorithm for rain-fall prediction, stock market prediction, wine quality, etc.
18. Implementing a classification algorithm for image classification, breast cancer, etc.
19. Implementing an unsupervised algorithm for grouping data points into meaningful clusters.
20. Implementing a clustering algorithm for data reduction

References:

1. Cormen, Leiserson, Rivest and Stein, Introduction to algorithms (Main textbook)
2. Kleinberg and Tardos , Algorithm Design
3. Mark Weiss, Data structures and algorithm analysis in C++ (Java)
4. Aho, Hopcroft and Ullman, Data structures and algorithms
5. S. Sahni, Data Structures, Algorithms, and Applications in C++, Silicon Press

Course Code	Course Name	L	T	P	Cr
DOAI250023	Data Visualization	2	0	4	4

Course Outcomes:

CO1: To introduce the fundamental concepts and methodologies of data visualization.
CO2: To understand various techniques for visualizing different types of data, including time series, networks, and hierarchies.
CO3: To explore the process of acquiring, parsing, and handling data for visualization.
CO4: To develop interactive data visualizations using frameworks like D3.js, Tableau, and Matplotlib.
CO5: To analyze and interpret security-related data through visualization techniques.

Module 1:

INTRODUCTION
Context of data visualization – Definition, Methodology, Visualization design objectives. Key Factors – Purpose, visualization function and tone, visualization design options – Data representation, Data Presentation, Seven stages of data visualization, widgets, data visualization tools.

Module 2:

VISUALIZING DATA METHODS
Mapping - Time series - Connections and correlations - Scatterplot maps - Trees, Hierarchies and Recursion - Networks and Graphs, Info graphics

Module 3:

VISUALIZING DATA PROCESS
Acquiring data, - Where to Find Data, Tools for Acquiring Data from the Internet, Locating Files for Use with Processing, Loading Text Data, Dealing with Files and Folders, Listing Files in a Folder, Asynchronous Image Downloads, Advanced Web Techniques, Using a Database, Dealing with a Large Number of Files. Parsing data - Levels of Effort, Tools for Gathering Clues, Text Is Best, Text Markup Languages, Regular Expressions (regexps), Grammars and BNF Notation, Compressed Data, Vectors and Geometry, Binary Data Formats, Advanced Detective Work.

Module 4:

INTERACTIVE DATA VISUALIZATION
Drawing with data – Scales – Axes – Updates, Transition and Motion – Interactivity - Layouts – Geomapping – Exporting, Framework – D3.js, tableau, Matplotlibsns.

Module 5:

SECURITY DATA VISUALIZATION
Port scan visualization - Vulnerability assessment and exploitation - Firewall log visualization -Intrusion detection log visualization -Attacking and defending visualization systems - Creating security visualization system.- Visual Data Analysis Techniques, Interaction Techniques - Systems and Applications.

Labs / Practicals:

1. Graphical Integrity & Perception
Create accurate vs misleading visuals; understand scale, axis distortion.

2. Data-Ink Ratio & Redesign
Apply Tufte's principles to redesign charts by reducing non-data ink.

3. Bar & Dot Plots
Visualize amounts using grouped bars, stacked bars, dot plots, and heatmaps.

4. Distribution Visualization
Use histograms, density plots, Q-Q plots to explore data distribution.

5. Pie & Proportional Charts
Visualize proportions using pie charts and donut charts; compare with bar plots.

6. Time Series Visualization
Plot individual and multiple time series with different response variables.

7. Geospatial Visualization
Create scatterplot maps, choropleth maps using geographic data.

8. Multivariate Data Visualization
Use scatterplot matrices, bubble charts, parallel coordinates for multidimensional data.

9. Network & Hierarchical Visualization
Build tree maps, sunburst charts, and network graphs for relationships.

10. Visual Analytics Dashboard
Create interactive dashboards in Tableau with filters, KPIs, and insights.

11. Python Library Comparison
Compare matplotlib, seaborn, plotly, and bokeh using the same dataset.

References:

1. Edward R. Tufte, The Visual Display of Quantitative Information, 2nd edition, Graphics Press, 2007
2. Claus O. Wilke, Fundamentals of Data Visualization, O'Reilly, 2019
3. Ben Fry, Visualizing Data: Exploring and Explaining Data with the Processing Environment, O'reilly, 1st edition, 2008
4. A. Kerren, J. T. Stasko, J. Fekete, C. North, Information Visualization, Springer, 2008
5. Alex Campbell, Data Visualization: Ultimate Guide to Data Mining and Visualization, 2020
6. A. Loth, Visual Analytics with Tableau, Willey, 2019
7. Kieran Healy, Data Visualization: A Practical Introduction, Princeton University Press, 2018
8. Alexandru C. Telea, Data Visualization: Principles and Practice, Second Edition, CRC Press, 2015

Course Code	Course Name	L	T	P	Cr
DOAI250032	Machine Learning	3	0	2	4

Course Outcomes:

CO1: Identify and extract relevant features from raw IoT data suitable for applying appropriate Machine Learning (ML) techniques.
CO2: Evaluate and compare the strengths and limitations of various ML algorithms across diverse application domains, particularly in IoT.
CO3: Select and implement appropriate ML models based on data characteristics and application requirements.
CO4: Mathematically formulate and analyze ML models and algorithms, including their underlying assumptions and performance metrics.
CO5: Design and validate machine learning-based solutions for real-world problems in IoT, demonstrating practical understanding and technical proficiency.

Module 1:

Module 1
Introduction: Introduction to Machine Learning: Introduction. Different types of learning, Hypothesis space and inductive bias, Evaluation. Training and test sets, cross validation, Concept of over fitting, under fitting, Bias and Variance.
Linear Regression: Introduction, Linear regression, Simple and Multiple Linear regression, Polynomial regression, evaluating regression fit.

Module 2:

Module 2
Decision tree learning: Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning, Python exercise on Decision Tree.
Instance based Learning: K nearest neighbor, the Curse of Dimensionality, Feature Selection: forward search, backward search, univariate, multivariate feature selection approach, Feature reduction (Principal Component Analysis), Python exercise on KNN and PCA. Recommender System: Content based system, Collaborative filtering based.

Module 3:

Probability and Bayes Learning: Bayesian Learning, Naïve Bayes, Python exercise on Naïve Bayes, Logistic Regression.
Support Vector Machine: Introduction, the Dual formulation, Maximum margin with noise, nonlinear SVM and Kernel function, solution to dual problem.

Module 4:

Artificial Neural Networks: Introduction, Biological motivation, ANN representation, appropriate problem for ANN learning, Perceptron, multilayer networks and the back propagation algorithm.

Module 5:

Ensembles: Introduction, Bagging and boosting, Random forest, Discussion on some research papers.
Clustering: Introduction, K-mean clustering, agglomerative hierarchical clustering, Python exercise on k-mean clustering.

Labs / Practicals:

1 Write a Python program that imports a dataset of your choice, displays the first five rows, and prints the data type of the DataFrame as a whole. Then, generate and display the statistical summary (mean, median, standard deviation, etc.) for each column and provide complete information about the DataFrame.

2 Write a Python program to import (or create) a dataset with missing values, display the count of missing values per column, and handle them using both imputation and dropping methods. Then, compare the dataset before and after handling missing values, and display the final cleaned dataset.
3 Write a Python program to load (or create) a dataset, split it into training and testing sets, and standardize the features using StandardScaler from sklearn. preprocessing. Finally, display the dataset before and after standardization to observe the changes.
4 Write a Python program to load (or create) a dataset, separate it into predictor variables (X) and target variable (y), and then split the dataset into training and testing sets. Finally, print both the training and testing datasets.
5 Apply EM algorithm to cluster a set of data stored in a .CSV file. 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 Python ML library classes in the program.
6 Write a Python program to load (or create) a dataset with missing values, count missing values in each column, and handle them by replacing with NaN, the column mean, and the most frequent value. Finally, display the dataset before and after performing the replacements.
7 Write a Python program to load (or create) a dataset, split it into training and testing sets, and apply feature scaling to the data. Finally, display the dataset before and after scaling to observe the changes.
8 Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
9 Implement the working principle of Artificial Neural Network with feed forward and feed backward principle.
10 Implement and demonstrate the working of SVM algorithm for classification.

References:

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
2. C. M. Bishop. Pattern Recognition and Machine Learning. First Edition. Springer, 2006. (Second Indian Reprint, 2015).
3. Tom M. Mitchell, “Machine Learning”, McGraw-Hill, 2010
4. P. Flach. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. First Edition, Cambridge University Press, 2012.

Course Code	Course Name	L	T	P	Cr
DOAI250047	Recommender Systems	3	1	0	4

Course Outcomes:

CO1: Understand the fundamental concepts, types, and applications of recommender systems.
CO2: Analyze and differentiate between various recommendation approaches, including content-based, collaborative filtering, and hybrid methods.
CO3: Evaluate the performance and effectiveness of recommender systems using appropriate metrics and evaluation techniques.
CO4: Design and implement basic recommender system models for real-world datasets and application domains.

CO5: Apply ethical and practical considerations in the deployment and personalization aspects of recommender systems.

Module 1:

Foundations of Recommender Systems

Linear Algebra Notation: Matrix addition, multiplication, transposition, and inverses; covariance matrices, Introduction to Recommender Systems, Functions and applications, Challenges and limitations, Types of Recommender Systems, Content-based, collaborative, knowledge-based, and hybrid

Module 2:

Content-Based Filtering

High-Level Architecture of Content-Based Systems, Item representation techniques, User profile learning methods, Similarity computation (cosine, Jaccard), Advantages and drawbacks of content-based filtering, Case study: Book/movie recommender using TF-IDF

Module 3:

Collaborative Filtering Techniques

User-based nearest neighbor recommendation, Item-based nearest neighbor recommendation, Model-based collaborative filtering: Matrix Factorization (SVD, NMF), Preprocessing-based techniques, Attacks on collaborative filtering: types and countermeasures, Distributed and privacy-aware recommendation

Module 4:

Knowledge-Based and Hybrid Recommender Systems

Knowledge-Based Recommendation, Constraint-based and case-based reasoning, Rule-based recommendation systems, Hybrid Recommender Systems, Need and motivations for hybridization, Hybridization strategies: Monolithic, Parallel, Pipelined, Real-world hybrid systems (e.g., Netflix, Amazon)

Module 5:

Evaluation and Emerging Trends

Evaluation of Recommender Systems, Accuracy metrics: Precision, Recall, F1-score, MAE, RMSE, Evaluation methodologies: offline, online, user studies, Community and Social-Based Recommendations, Social tagging and folksonomies, Case study: BibSonomy and HeyStaks, Trust-based Recommender Systems, Trust modeling and integration in recommendation, Computational trust and robustness, Future Trends: Explainability, fairness, and deep learning in recommendations

Labs / Practicals:

Tutorial Topics (Problem Solving / Case Discussions):

1. Matrix Algebra for Recommender Systems.

2. Cosine & Jaccard Similarity Problem Solving.

3. User Profile and Item Profile Creation.

4. Comparative Case Studies: Content-Based vs Collaborative Filtering.

5. Hands-on Matrix Factorization Exercises (SVD, NMF).

6. Evaluation Metrics: Problems on Precision, Recall, F1, MAE, RMSE.

7. Diagrammatic Explanation of Hybrid Systems.

8. Emerging Issues: Explainability, Fairness, and Ethics Discussions.

References:

1. Jannach D., Zanker M. and FelFering A., Recommender Systems: An Introduction, Cambridge University Press (2011), 1st ed
https://www.researchgate.net/publication/235910467_Recommender_Systems

2. Ricci F., Rokach L., Shapira D., Kantor B.P., Recommender Systems Handbook, Springer (2011), 1st ed.
https://www.researchgate.net/publication/227268858_Recommender_Systems_Handbook

3. Erwin Kreyzig, Advanced Engineering Mathematics, 10th Edition, Wiley, 2011.
http://www.uop.edu.pk/ocontents/AdvancedEngineeringMathematics.pdf

4. Aggarwal, C. C. Recommender Systems: The Textbook. Springer 2016
https://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal_c_c_recommender_systems_the_textbook.pdf