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.Sc (AI and ML)

(2026-27 Scheme)

रा.इ.सू.प्रौ.सं

NIELIT

Deemed to be University

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

Vision

To be a center of excellence in Artificial Intelligence and Machine Learning education, research, and innovation, producing globally competent professionals and thought leaders who drive transformative change across industries through intelligent systems and ethical AI practices.

Mission

The mission of the M.Sc. in Artificial Intelligence and Machine Learning program is to impart advanced theoretical knowledge and practical skills in AI and ML, enabling students to innovate, lead, and solve complex problems across interdisciplinary domains. The program seeks to foster a research-oriented mindset, encourage ethical AI development, and produce industry-ready professionals who can contribute to technological advancement, responsible innovation, and societal progress.

Program Education Objectives

PEO1: Acquire advanced knowledge in Artificial Intelligence and Machine Learning, enriched by both academic rigor and industry-driven skill sets.
PEO2: Excel in professional careers by developing AI-driven solutions to complex real-world problems across diverse domains such as healthcare, finance, cyber security, and automation.
PEO3: Demonstrate leadership and teamwork skills, actively contributing to innovation in AI-powered product development, research, and technology-driven industries.
PEO4: Exhibit adaptability and problem-solving capabilities in the rapidly evolving field of AI and ML, addressing technical, ethical, and managerial challenges effectively.

Program Outcomes

PO1: Ability to independently conduct research, investigations, and development work in Artificial Intelligence and Machine Learning to address real-world challenges.
PO2: Ability to write, present, and publish substantial technical reports and research documents in AI, ML, and data science domains.
PO3: Ability to demonstrate mastery in AI and ML, including deep learning, natural language processing etc as per program specialization.
PO4: Ability to apply modern AI tools and frameworks for model development, performance analysis, and system optimization in data-driven decision-making.
PO5: Ability to engage in continuous learning and adapt to evolving AI technologies, ethical considerations, and industrial advancements.

Program Specific Outcomes

PSO1: Apply AI and ML techniques to develop intelligent applications in domains such as healthcare, finance, cybersecurity, automation, and smart infrastructure.

PSO2: Design, develop, and optimize machine learning and deep learning models using advanced tools and frameworks such as TensorFlow, PyTorch, Scikit-learn, and Keras.

PSO3: Analyze and interpret large-scale datasets using statistical and computational methods to extract actionable insights for data-driven decision-making.

PSO4: Demonstrate understanding of ethical, legal, and societal implications of AI systems, and apply principles of responsible and explainable AI in real-world scenarios.
PSO5: Engage in research and innovation in emerging areas of AI, including generative AI, reinforcement learning, federated learning, and AI for cybersecurity and sustainability.

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-MSCAIM-101	DOAI250021	Data Science	3	0	2	4
PC-MSCAIM-102	DASH250095	Research Methodology and IPR	3	1	0	4
PC-MSCAIM-103	DASH250001	Mathematical Foundations for AI	3	1	0	4
PE-MSCAIM-104	Elective	Program Elective				4
OE-MSCAIM-105	Elective	Open Elective				4
AU-MSCAIM-106	Elective	Audit Elective				0

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-MSCAIM-201	DOAI250005	Artificial Intelligence and Machine Learning	2	0	4	4
PE-MSCAIM-202	Elective	Program Elective				4
PE-MSCAIM-203	Elective	Program Elective				4
PC-MSCAIM-204	DOAI250015	Data Analytics and Visualization	2	0	4	4
AU-MSCAIM-205	Elective	Audit Elective				0
EE-MSCAIM-206	EE	Mini Project with Seminar				4

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PE-MSCAIM-301	Elective	Program Elective				4
OE-MSCAIM-302	Elective	Open Elective				4
EE-MSCAIM-303	EE	Dissertation-I/ Industrial Project				12

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-MSCAIM-401	EE	Dissertation-I/ Industrial Project (Major)				20

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	
DCSE250056	Data Structures and Algorithms	3	0	2	4	104
DOAI260041	Natural Language Processing	2	0	4	4	104
DOEE250121	Internet of Things	3	0	2	4	104
DCSA250075	Software Project Management	3	1	0	4	202
DOAI250009	Big Data Analytics	3	0	2	4	202
DOAI250046	R Programming	2	0	4	4	202
DCSA250081	Web Development using Python and Django	2	0	4	4	203
DCSE250099	Object Oriented Analysis and Design	3	1	0	4	203
DOAI250003	Advanced Statistical Techniques	3	1	0	4	203
DOAI250018	Data Mining	3	0	2	4	301
DOAI250024	Deep Learning	2	0	4	4	301
DOAI250027	Generative AI	3	0	2	4	301

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
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
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
DASH250001	Mathematical Foundations for AI	3	1	0	4

Course Outcomes:

CO1 : Apply principles of linear algebra to understand AI models and representations.
CO2: Analyze and compute probabilistic models using random variables,distributions, and Bayes' theorem for AI/ML applications.
CO3: Evaluate datasets using statistical inference techniques including estimation, confidence intervals, and hypothesis testing to make data-driven decisions.
CO4 : Evaluate learning performance and entropy-based decision making using information theory.
CO5 : Model real-world AI problems using mathematical tools and numerical methods.

Module 1:

Linear Algebra for AI (9 Hours)
Vectors (vector spaces, norms, loop), matrices, matrix operations, rank, inverse, determinant, eigenvalues and eigenvectors, orthogonality, singular value decomposition (SVD), applications in Principal Component Analysis (PCA), word embeddings, image recognition.

Module 2:

Probability (9 Hours)
Basic rules and axioms events, sample space, dependent and independent events, conditional probability, Random variables- continuous and discrete, expectation, variance, distributions- joint and conditional, Bayes' Theorem, Popular distributions- binomial, Bernoulli, poisson, exponential, Gaussian

Module 3:

Statistics (8 Hours)
Fundamentals of Data: Collection, Summarization, and Visualization; Sampling and Sampling Distributions, Central Limit Theorem; Methods of Estimation, Unbiased estimators; Confidence Interval Estimation: Z-interval, t-interval; Hypothesis Testing, Types of Errors, Rejection Region Approach and p-value Approach.

Module 4:

Information Theory (8 Hours)
Measure of Information, Entropy, Source coding theorem - Shannon-Fano codes & Huffman codes, Discrete Memoryless channel, Mutual information, Channel Capacity, Shannon-Hartley theorem

Module 5:

Mathematical Tools for AI Applications (11 Hours)
Markov chains and Markov Decision Processes (MDPs), matrix factorization in recommender systems, basic game theory, numerical methods: interpolation, numerical integration, applications in Non-linear Programming (NLP), reinforcement learning, and generative models.

Labs / Practicals:

N/A

References:

1.S. Kumaresan – Linear Algebra: A Geometric Approach, PHI Learning.
2.B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers.
3.Pankaj Jalote – Mathematical Foundations of Computer Science, Wiley India.
4.K. D. Joshi – Foundations of Discrete Mathematics, New Age International.
5.Marc Deisenroth, A. Faisal, C. Ong – Mathematics for Machine Learning, Cambridge University Press.
6.Shai Shalev-Shwartz & Ben-David – Understanding Machine Learning: From Theory to Algorithms.

Course Code	Course Name	L	T	P	Cr
DOAI250005	Artificial Intelligence and Machine Learning	2	0	4	4

Course Outcomes:

CO1: Understand AI problem-solving techniques, heuristic search algorithms, and reinforcement learning.
CO2: Apply knowledge representation, logic-based reasoning, and probabilistic models in AI systems.
CO3: Develop and evaluate machine learning models using supervised and unsupervised learning techniques.
CO4: Design and implement deep learning architectures for vision, language, and sequential data applications.
CO5: Analyze advanced AI methods, ethical challenges, and emerging trends in artificial intelligence.

Module 1:

AI Techniques and Search Strategies
History of AI, problem-solving using search, uninformed vs. informed search, heuristic search techniques (Hill Climbing, Simulated Annealing, A* Algorithm), adversarial search (Minimax Algorithm, Alpha-Beta Pruning), AI-driven game strategies (Chess, Chinese Checkers), Reinforcement Learning fundamentals (Markov Decision Processes, Q-learning), heuristic optimization techniques (Genetic Algorithms, Particle Swarm Optimization).

Module 2:

Knowledge Representation, Reasoning, and Probabilistic Models
Propositional logic, inference mechanisms, forward and backward chaining, probabilistic reasoning techniques (Bayesian Networks, Markov Models, Probabilistic Graphical Models), logic-based AI vs. machine learning-driven reasoning, Natural Language Processing (NLP) fundamentals (tokenization, embeddings, language models), AI-driven text processing, machine translation.

Module 3:

Machine Learning Fundamentals and Neural Networks
Supervised and unsupervised learning paradigms, classification and regression techniques (decision trees, support vector machines, ensemble learning methods: Random Forest, XGBoost), artificial neural networks (activation functions, loss functions, back propagation, multilayer perceptions), overfitting, bias-variance tradeoff, hyperparameter tuning, performance evaluation metrics (precision-recall, ROC curves, F1-score).

Module 4:

Deep Learning and Advanced Neural Architectures
Convolutional Neural Networks (CNNs) for image classification and object detection (convolution layers, pooling, fully connected layers), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks for sequential data processing, Transformer models (BERT, GPT) for NLP applications, optimization techniques (batch normalization, dropout regularization, transfer learning), deep learning applications (medical imaging, self-driving vehicles, generative AI).

Module 5:

Unsupervised Learning, Generative Models, and Ethical AI
Clustering techniques (K-Means, Hierarchical Clustering), Autoencoders, Generative Adversarial Networks (GANs), self-supervised learning, foundation models in AI, AI applications in healthcare, finance, and autonomous systems, AI ethics (fairness, bias mitigation, adversarial attacks, explainable AI), advancements in AI (quantum computing, federated learning, AI governance frameworks).

Labs / Practicals:

1. Implement a basic uninformed search algorithm (e.g., Breadth-First Search, Depth-First Search).
2. Implement informed search algorithms using heuristics (e.g., A* Algorithm for pathfinding).
3. Implement the Hill Climbing algorithm for local search problems.
4. Develop a solution using Simulated Annealing for optimization problems.
5. Implement the Minimax algorithm to solve a simple game (e.g., Tic-Tac-Toe).
6. Implement Alpha-Beta Pruning for optimizing the Minimax algorithm.
7. Build a Reinforcement Learning agent using Q-learning to solve a basic maze problem.
8. Implement Genetic Algorithms to solve optimization problems (e.g., travelling salesman).
9. Build a Particle Swarm Optimization model for function optimization.
10. Implement propositional logic and inference mechanisms (e.g., forward and backward chaining).
11. Develop a basic Bayesian Network for probabilistic reasoning.

References:

1.Tom M. Mitchell: Machine Learning. McGraw Hill
2.Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig
3.Pattern Recognition and Machine Learning by Christopher M. Bishop
4.Machine Learning: A Probabilistic Perspective by Kevin P. Murphy
5.Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto
6.Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper
7.The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
8.Michael Nielsen: Neural Networks and Deep Learning (free online book)
9.Stuart Russel & Peter Norvig: Artificial Intelligence– A Modern Approach. Pearson, 4th edition, 2023
10.Deep Learning (Adaptive Computation and Machine Learning series) Illustrated Edition, by Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, London

Course Code	Course Name	L	T	P	Cr
DOAI250015	Data Analytics and Visualization	2	0	4	4

Course Outcomes:

CO1: Understand the fundamentals of data analytics, data preprocessing, and feature engineering techniques.
CO2: Perform efficient data manipulation, aggregation, and wrangling using Pandas and NumPy.
CO3: Implement numerical computing techniques for high-level mathematical operations and image data analysis.
CO4: Create effective data visualizations using Matplotlib, Seaborn, and Pandas for data-driven insights.
CO5: Utilize advanced visualization tools like Excel, Tableau, and Power BI for interactive and real-time data representation.

Module 1:

Fundamentals of Data Analytics and Preprocessing
Definition, importance, and applications of data analytics. Introduction to structured and unstructured data, data collection, data storage, and preprocessing techniques. Data ingestion, handling missing values, data cleaning, transformation, and feature engineering. Overview of NumPy and Pandas libraries for data analysis, including DataFrames, Series, indexing, slicing, and filtering.

Module 2:

Data Manipulation and Aggregation
Operations on Pandas DataFrames and Series, data wrangling techniques, handling categorical and numerical data, working with missing and duplicate values. Data aggregation, applying functions in Pandas, groupby operations, pivot tables, and cross-tabulations. Merging and concatenating datasets, time series analysis, and handling large datasets efficiently.

Module 3:

Numerical Computing with NumPy
Creating and manipulating multidimensional arrays, broadcasting, vectorized operations, and high-level mathematical functions. Array slicing, fancy indexing, filtering, and statistical analysis using NumPy. Working with structured arrays, handling image data, and performing linear algebra operations for data science applications.

Module 4:

Data Visualization Techniques
Importance of data visualization, choosing appropriate visualization methods, visualizing data insights and findings. Data visualization using Matplotlib: line plots, bar charts, scatter plots, histograms, and subplots. Advanced visualization using Seaborn: pair plots, violin plots, heatmaps, and regression plots. Using Pandas for quick data visualization and integrating visualization tools in data science workflows.

Module 5:

Tools and Advanced Visualization Techniques
Visualization using Excel: charts, pivot tables, and dashboards. Introduction to interactive visualization tools such as Tableau and Power BI. Creating real-time and dashboard-based visualizations, interactive plots using Plotly, and geographic data visualization with Folium. Case studies and real-world applications of data analytics and visualization.

Labs / Practicals:

1. Implement data wrangling techniques like reshaping and pivoting using Pandas.
2. Create Data Frames and Series in Pandas and perform basic data manipulation operations.
3. Load datasets from different file formats (CSV, Excel, JSON) into Pandas.
4. Perform basic exploratory data analysis (EDA) using summary statistics and visualizations.
5. Create data visualizations using Matplotlib (e.g., line plots, histograms, bar charts).
6. Create advanced visualizations using Seaborn (e.g., heatmaps, pair plots, violin plots).
7. Implement box plots, scatter plots, and correlation matrices for data analysis.
8. Perform data aggregation and group-by operations to analyze grouped data.
9. Merge multiple Data Frames using different join operations in Pandas.
10. Implement linear regression model and visualize the results using Matplotlib.
11. Create and customize subplots using Matplotlib to present multiple visualizations.
12. Perform data aggregation using pivot tables and cross-tabulation in Pandas.
13. Create interactive plots using Plotly for web-based visualization.
14. Create a word cloud from a textual dataset to visualize the frequency of words.
15. Perform principal component analysis (PCA) for dimensionality reduction and visualize the result.
16. Visualize time series data and trends using Matplotlib and Seaborn.
17. Use Excel for data visualization, such as creating charts, graphs, and dashboards.
18. Implement correlation analysis and create a heatmap for understanding relationships between variables

References:

1. Python – Data Visualisation by Samuel Burns
2. Storytelling with Data: A Data Visualization Guide for Business Professionals by Cole NussbaumerKnaflic
3. Data Science for Business by Foster Provost and Tom Fawcett:
4. Data Visualization: A Practical Introduction by Kieran Healy
5. Hands-On Data Visualization: Interactive Storytelling from Spreadsheets to Code by Jack Dougherty and IlyaIlyankou
6. Data Visualization with Excel Dashboards and Reports by Dick Kusleika
7. Beautiful Visualization: Looking at Data Through the Eyes of Experts edited by Julie Steele and Noah Iliinsky

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
DCSE250056	Data Structures and Algorithms	3	0	2	4

Course Outcomes:

1)Classify Data structures
2)Describe Linear Data Structures
3)Explain Non-Linear Data Structures
4)Explain basic algorithmic concepts and recursion
5)Apply different Sorting and Searching Algorithms

Module 1:

Introduction to Data Structures:Basic Terminology, Classification of Data Structures, Operations on Data Structures

Module 2:

Linear Data Structures: Arrays: Introduction to Arrays, Representation in Memory, Operations on an Array, Two Dimensional Arrays
Linked Lists: Singly Linked List, Representation in Memory, Operations on a Single Linked List, Circular Linked Lists, Doubly Linked Lists.
Stacks: Introduction to Stacks, Array Representation of Stacks, Operations on a Stack, Linked List Representation and Operations of Stack, Applications of Stacks-Infix-to-Postfix Transformation, evaluating Postfix Expressions.
Queues: Introduction to Queues, Array Representation of Queues, Operations on a Queue,
Linked List Representation and Operations of Queue, Types of Queues-DeQueue, Circular Queue, Applications of Queues-Round Robin Algorithm

Module 3:

Non Linear Data Structures:
Trees: Basic Terminologies, Definition and Concepts of Binary Trees, Representations of a Binary Tree using Arrays and Linked Lists, Operations on a Binary Tree-Insertion, Deletion, Traversals, Types of Binary Trees.
Graphs: Graph Terminologies, Representation of Graphs- Set, List, Matrix, Graph Traversals

Module 4:

Introduction to Algorithms:
Algorithms and flow charts, Time & Space complexity (definition only)
Recursion: Basic concepts and examples of recursion e.g. factorial problem, Fibonacci sequence.

Module 5:

Sorting & Searching Algorithms: Sorting Algorithms: Algorithms and their analysis (time and space) — Bubble sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap sort and Radix Sort
Searching Algorithms: Linear search — Binary search –Concept of Hashing.

Labs / Practicals:

1.Write a program to implement array
2.Write a program to add two matrices using two dimensional arrays
3.Write a program using recursive and non-recursive functions to perform search operation in a given list of integers using linear search technique
4.Write a program to implement search operation in a given list of integers using binary search technique

5.Write a program to implement insertion sorting for a given random data
6.Write a program to implement bubble sorting for a given random data
7.Write a program to implement quick sorting for a given random data
8.Write a program to implement selection sorting for a given random data.

References:

1)Data Structures, R.S. Salaria, Khanna Book Publishing, New Delhi
2)Data Structures Using C, Reema Thareja, Oxford University Press India.
3)Classic Data Structures, SamantaDebasis, Prentice Hall of India.
4)Fundamentals of Data Structure in C, Horowitz, Ellis, Sahni, Sartaj, AndersonFreed, Susan, University Press, India.
5)Data Structures: A Pseudo code approach with C, Richard F. Gilberg, Behrouz A. Forouzan, CENGAGE Learning, India.
6)Data Structures and Algorithms: Concepts, Techniques and Applications, G. A. V. Pai, McGraw- Hill Education, India.
7)Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R. L. Rivest, C. Stein, MIT Press

Course Code	Course Name	L	T	P	Cr
DOAI260041	Natural Language Processing	2	0	4	4

Course Outcomes:

CO1. Develop the ability to collect, preprocess, and analyze textual data using NLP techniques and Python libraries
CO2. Apply text representation techniques and machine learning algorithms to build, train, and evaluate NLP classification models using appropriate performance metrics.
CO3. Develop and apply advanced NLP techniques for information extraction, text summarization, question answering, sentiment analysis, and topic modelling in real-world applications.
CO4. Design, develop, and deploy intelligent chatbot systems using rule-based and neural network-based NLP techniques following the AI project development cycle.
CO5. Evaluate and apply modern NLP concepts including Large Language Models, prompt engineering, ethical AI practices, and model interpretability techniques for responsible deployment of NLP solutions

Module 1:

Introduction to NLP, Data Collection and Processing for NLP – definition and current applications of Natural Language Processing, exploration of text data, text preprocessing pipeline – sentence segmentation, tokenization, stop word removal, stemming, lemmatization, part-of-speech tagging, handling and cleaning raw text data in Python.

Module 2:

Classification for NLP – Bag of Words (BoW) representation of text data, TF-IDF (Term Frequency-Inverse Document Frequency) format, Introduction to word embeddings. Model building for text classification – Naive Bayes classifier, Logistic Regression classifier, training and evaluating the classification model, performance metrics – precision, recall, F1-score, confusion matrix. Roadmap from MLP to LSTM and LSTM to Transformers. Attention Mechanism, Transformer architecture (BERT and GPT). Transformer-based text representation using BERT.

Module 3:

Advanced NLP Tasks and Applications - Named Entity Recognition (NER) and Relation Extraction, Text Summarization: Extractive and Abstractive Approaches, Advanced Question Answering Systems (Open-Domain, Generative QA), Sentiment Analysis and Opinion Mining, Topic Modelling. Applications of BERT in NER, sentiment analysis, and question answering, Coreference resolution, Information extraction.

Module 4:

Chatbot architecture, cosine similarity for rule-based chatbots, neural network-based intent recognition, chatbot training, and building/testing chatbots in Python. Application of the AI Project Cycle for chatbot development, including problem scoping, Data Preprocessing, Classification Model Development, and Evaluation.

Module 5:

Current Topics, Ethics, and Deployment, Large Language Models (LLMs), Capabilities and Prompt Engineering, Bias, Fairness, and Ethical Considerations in NLP Models, Model Debugging, NLP Interpretability and Explainability.

Labs / Practicals:

Explore text data that has been downloaded and processed from a given corpus.
2. Process textual data by sentence segmentation, tokenization, and stop word removal using NLTK/spaCy in Python.
3. Express data in a bag of words format and use the TFIDF format to represent text features.
4. Construct a separate dataset for model training, including train-test split and cross-validation.
5. Construct and train a text classification model using Naive Bayes or Logistic Regression in Python.

6. Evaluate the classification model using precision, recall, F1-score, and confusion matrix.
7. Implement Named Entity Recognition and relation extraction for identifying entities and relationships in text.
8. Perform text summarization using extractive and abstractive techniques, and compare the generated summaries.
9. Develop NLP applications for sentiment analysis, topic modelling, and question answering using sample datasets.
10. Use cosine similarity to build and train a simple rule-based Chatbot.
11. Use a neural network to train a more advanced Chatbot capable of intent recognition.
12. Apply the AI Project Cycle for the Hotel Chatbot problem: data collection, pre-processing, model training, and deployment.
13. Apply AI Project Cycle to develop a complete AI solution for an NLP project of choice, documenting all stages from data collection to evaluation.
14. Explore LLM capabilities using prompt engineering for NLP tasks and analyze ethical issues such as bias and fairness.
15. Apply SHAP and LIME for NLP model interpretability, debugging, and ethical analysis.

References:

Intel AI for Future Workforce – Natural Language Processing Course Content, Intel Digital Readiness Program.
2. Steven Bird, Ewan Klein, and Edward Loper – Natural Language Processing with Python, O'Reilly Media.
3. Delip Rao and Brian McMahan – Natural Language Processing with PyTorch, O'Reilly Media.
4. Yoav Goldberg – Neural Network Methods for Natural Language Processing, Morgan and Claypool Publishers.
5. Jacob Devlin et al. – BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI.
6. Practical Natural Language Processing, Sowmya Vajjala et al. O'Reilly
7. "Attention Is All You Need", 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Course Code	Course Name	L	T	P	Cr
DOEE250121	Internet of Things	3	0	2	4

Course Outcomes:

CO 1 Explain the concept of IoT
CO 2 Networking basics for IoT application development
CO 3 Analyze various protocols for IoT
CO 4 Design a PoC of an IoT system using various hardware platforms
CO 5 Apply data analytics and use cloud offerings related to IoT
CO 6 Analyze applications of IoT in real time scenario

Module 1:

Overview of IoT

Introduction to the Internet of Things, IoT system architecture and standards, Networking Basics - TCP/IP, Networking Basics - IP addressing basics (IPV4 and IPV6), Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks.

Module 2:

IoT hardware Platforms

IoT Design Methodology – Embedded computing logic – Microcontroller-System on Chips, Hardware platforms for prototyping IoT node- Arduino, Raspberry Pi, NodeMCU, ESP32, ARM Cortex Microcontrollers, IoT mote hardware platforms, Swadeshi RISC V based solutions, Interfacing sensors and actuators with hardware platforms, Developing IoT applications using Raspberry Pi with Python Programming.

Module 3:

IoT connectivity & Protocols

IoT Access Technologies: WiFi, Zigbee, Zwave, Bluetooth,UWB, sub1GHz, LoRaWAN, Sigfox and NB-IoT, Topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN, IoT application level protocols: MQTT, CoAP, XMPP, HTTP/Rest Services, WebSockets.

Module 4:

Data analytics, Cloud and IoT Security

No SQL Databases Vs SQL Databases, Apache web servers, JSON, Open and commercial Cloud solutions for IoT, Python Web Application Frameworks for IoT, IoT data visualisation tools, IoT Security - Need for encryption, standard encryption protocol, lightweight cryptography, Trust models for IoT, ARM Cortex Microcontroller Security, Root Security Services (RSS).

Module 5:

IoT Case studies

Smart Lighting, Smart home, Smart Agriculture, Smart farming, IoT for health care & patient monitoring, Smart and Connected Cities, Building end-to-end smart applications with TinyML.

Labs / Practicals:

Hands-on Sessions on IoT Application Development using IoT Hardware Kits (ESP32/STM32/Rpi etc.)

- IPv4 and IPv6 Implementation
- 6LoWPAN Network Implementation
- Interfacing multiple sensors and actuators with Processor Hardware using different communication protocols and develop an integrated monitoring system.
- Implementation of different IoT connectivity technologies including WiFi, Bluetooth, and BLE for various application scenarios.
- Implementation of MQTT protocol for IoT device communication
- Implementation of CoAP for resource-constrained IoT devices
- Integration of IoT devices with cloud platforms for data analytics

References:

Reference Books

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, —IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, Cisco Press, 2017
2. Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner, “Enabling things to talk – Designing IoT solutions with the IoT Architecture Reference Model”, Springer Open, 2016
3. VijayMadisetti , ArshdeepBahga, Adrian McEwen (Author), Hakim Cassimally “Internet of Things: A Hands-on-Approach” ArshdeepBahga& Vijay Madisetti, 2014.
4. Gian Marco Iodice, TinyML Cookbook: Combine artificial intelligence and ultralow-power embedded devices to make the world smarter
5. Olivier Hersent, David Boswarthick, Omar Elloumi , —The Internet of Things – Key applications and Protocols, Wiley, 2012

Course Code	Course Name	L	T	P	Cr
DCSA250075	Software Project Management	3	1	0	4

Course Outcomes:

CO1: Understand and explain the fundamental principles of software project management, including the characteristics, challenges, life cycle models, and planning processes of software projects.
CO2: Apply various project planning and estimation techniques such as Function Point Analysis, COCOMO, and scheduling tools like Gantt and PERT charts for effective project planning and evaluation.
CO3: Analyze and develop optimized project schedules by identifying critical paths, managing resources, and implementing risk management strategies including risk planning and control.
CO4: Demonstrate the ability to manage software project teams effectively through leadership, communication, and understanding of organizational structures, while ensuring software quality through formal assurance and testing methods.
CO5: Evaluate and apply monitoring and control techniques, including Earned Value Analysis and Agile/PMBOK practices, to ensure successful project execution, progress tracking, and closure.

Module 1:

Foundations of Software Project Management:
Introduction to Software Project Management
Categorization of Software Projects & Common Challenges
Software Development Process & Life Cycle Models
Waterfall, Prototyping, RAD, Incremental, Spiral
Software Production Process & Process Tailoring
Understanding Stakeholders and Setting Project Objectives
Overview of Step-wise Project Planning

Module 2:

Project Planning and Estimation:
Software Project Evaluation and Selection of Project Objectives
Project Planning Approaches: Top-Down vs Bottom-Up
Activity Types and Task Structuring
Software Effort Estimation Techniques
Function Point Analysis, Object Points, COCOMO Models
Introduction to Scheduling Tools: Gantt Charts, PERT, CPM
Project Life Cycle & Scope Management

Module 3:

Scheduling, Resource, and Risk Management:
Sequencing and Scheduling Activities
Network Planning: AOA and AON
Identifying Critical Paths, Crashing, and Fast Tracking
Resource Allocation and Scheduling
Identifying resource needs, sequence scheduling, and publishing
Risk Management
Risk Identification, Categories, Planning, Control, and Evaluation
Earned Value Analysis & Schedule Monitoring

Module 4:

Project Execution, Communication & Quality Assurance:

Project Teams and Leadership
Organizational Structures for Software Projects
Effective Communication and Personality Traits in Teams
Software Quality Concepts: FURPS, Software Reliability
Software Quality Assurance (SQA)
Formal Technical Reviews, Cleanroom Methodology
Software Quality Assurance Plans
Testing & Verification
V&V, White Box and Black Box Testing, Unit & System Testing

Module 5:

Monitoring and Control:
Tracking and Escalation Meetings
Visualizing Project Progress and Recovery Plans
Agile Project Management Overview
Case Study: PMBOK Framework
Managing Small Projects
Tools and Techniques for Monitoring and Control
Final Product Delivery and Closure Activities

Labs / Practicals:

NA

References:

Mike Cotterell, Bob Hughes. Software Project Management, Fifth Edition, Tata McGraw-Hill; 2012.
Roger S. Pressman. Software Engineering – A Practioner's Approach, Eighth Edition, Tata McGraw- Hill publishers; 2014.
Jalote P. Software Project Management in practice, Second edition, Person Education; 2003.

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
DOAI250046	R Programming	2	0	4	4

Course Outcomes:

CO1. Understand the fundamentals of R programming and its environment.
CO2. Perform data manipulation and cleaning using R packages.
CO3. Visualize data effectively using R's graphical capabilities.
CO4. Apply statistical methods and data analysis techniques using R.
CO5. Implement real-world data science projects using R.

Module 1:

Introduction to R and Programming Basics
Overview of R, Installing and using packages
Basic syntax, variables, and data types
Operators, expressions, and control structures (if, else, loops)
Functions and user-defined functions

Module 2:

Working with Data Structures
Vectors, lists, matrices, arrays, and data frames
Factors and categorical data
Indexing and subsetting
Reading and writing data (CSV, Excel, JSON)
Data manipulation using dplyr, tidyr

Module 3:

Data Visualization with R
Base R plotting system
Advanced visualization with ggplot2
Customizing plots: labels, themes, and aesthetics
Creating bar charts, histograms, boxplots, scatter plots, etc.
Saving and exporting visualizations

Module 4:

Statistical Analysis
Descriptive statistics and probability distributions
Inferential statistics (t-tests, chi-square, ANOVA)
Linear regression and correlation
Handling missing values and outliers
Introduction to time series and clustering

Module 5:

Report and Interface Generation
Case studies: e.g., sales analysis, healthcare data analysis, social media trends
End-to-end data analysis workflow
Building a reproducible report using R Markdown
Introduction to Shiny for interactive dashboards

Labs / Practicals:

1. Create and assign variables of different data types: numeric, character, logical, complex.
2. Use arithmetic, relational, and logical operators.
3. Write conditional statements (if, if-else, switch) to classify numbers.
4. Write a function to compute factorial of a number using loops.
5. Generate a multiplication table using for loop.
6. Create and manipulate vectors, matrices, lists, arrays, and data frames.
7. Convert a vector into a factor and perform operations.
8. Index and subset rows/columns from a matrix and data frame.
9. Read a CSV file (e.g., students.csv) into a data frame and perform summary statistics.
10. Clean data: handle missing values using na.omit(), is.na() and replace().
11. Use dplyr functions like filter(), select(), mutate(), arrange() and summarise().
12. Plot line graphs, histograms, and scatter plots using base R.
13. Load mtcars dataset. Create a histogram of mpg and a barplot of cyl.
14. Use ggplot2 to:

a. Create a scatter plot of wt vs mpg.
b. Add color coding for different gear values.
c. Create a boxplot of mpg grouped by cyl.

15. Customize charts: axis labels, titles, legends, and themes.
16. Save your plots using ggsave().
17. Compute mean, median, mode, variance, and standard deviation for a dataset.
18. Perform a one-sample and two-sample t-test on simulated data.
19. Perform ANOVA on a built-in dataset (PlantGrowth).
20. Conduct linear regression: mpg ~ wt + hp on mtcars dataset.
21. Visualize the regression line and interpret the model summary.
22. Perform clustering (K-means) on iris dataset.
23. Choose a dataset (e.g., airquality, diamonds, or external CSV).
24. Perform the following steps:

a. Data import, cleaning, and transformation.
b. Exploratory data analysis and visualizations.
c. Statistical or predictive modeling.

25. Generate a report using R Markdown:

a. Include code, plots, and interpretations.
b. Export to HTML/PDF format.

26. Create a basic Shiny app to visualize your dataset interactively.

References:

1. R for Data Science by Hadley Wickham & Garrett Grolemund
2. The Art of R Programming by Norman Matloff
3. Hands-On Programming with R by Garrett Grolemund
4. Practical Statistics for Data Scientists by Peter Bruce & Andrew Bruce
5. CRAN Documentation and Vignettes (https://cran.r-project.org/)

Course Code	Course Name	L	T	P	Cr
DCSA250081	Web Development using Python and Django	2	0	4	4

Course Outcomes:

CO1: Develop basic Python programs utilizing variables, control structures, data types, and loops to solve problems.

CO2: Apply advanced Python concepts including data structures, file handling, functions, modules.

CO3: Demonstrate Object-Oriented Programming (OOP) skills by creating classes, methods.

CO4: Design and build basic back-end web applications using the Django framework, leveraging its MVT architecture, URL routing, forms, authentication, and the admin interface.

CO5: Develop and test RESTful APIs using Django REST Framework and Postman, and implement CRUD operations with proper validation and authentication.

Module 1:

Python functions and variables,Tuples, Dictionaries, Reading and Writing text files, loops, Custom functions, Exception handling, Importing files,OOP Concept

Module 2:

Client-Server Architecture, HTTP Methods (GET, POST, PUT, DELETE), Request-Response Cycle in Web Applications, IP Address and Port Numbers, URL Routing and Path Matching, HTTP Status Codes, CSRF

Module 3:

Introduction to Back-End Web Development (Django), HTTP protocol basics, MVC/MVT model, Virtual environment setup, Installing Django, Django's take on MVT (Model, View, Template), DRY programming principles, Core files (models.py, urls.py, views.py)

Module 4:

Working with database, Setting up database connections, use of .env file,Managing users and Django admin.

Module 5:

Advanced Django and REST Framework, Django URL patterns and views, Designing good URL schemes, Generic views, Django forms and validation, Authentication, Advanced form processing techniques, Creating CRUD applications in Django, Django REST Framework basics, Using Postman to test APIs.

Labs / Practicals:

1.Write a Python script that inputs two numbers and displays the sum, difference, product, and quotient using if-else statements.

2.Python Error Handling & Exceptions

3.Implement a program that reads a text file, counts the number of words and lines, and displays the result. Wrap this logic inside a reusable function.

4.Create a dictionary of student names and grades. Write a function to add new students, update grades, and list all students.

5.Define a Car class with attributes like make, model, year and a method to display car details. Instantiate the class and display the information.

6.Implement a base class Shape with an abstract method area() and two subclasses Rectangle and Circle. Compute the areas of shapes.

7.Create a new Django project and app. Implement a simple view that returns “Hello World” and map it using urls.py. Run the server and verify.

8.Crea a Django app to implement "To-Do List" app. Features like add task,view all tasks,delete tasks or any basic CRUD app

9.Design a Django model for a Product with fields like name, price, stock. Register the model in admin.py and add a few products via the Django admin interface.

10.Create a RESTful API for the Product model. Implement list and create operations, and test these endpoints using Postman.

References:

1.Python Crash Course — Eric Matthes
2.Django for Beginners — William S. Vincent
3.Django Web Development with Python — Samarth Shah
4.Django Web Development” — Vijay Joshi
5. The Definitive Guide to Django: Web Development Done Rightby Adrian Holovaty and Jacob Kaplan-Moss
6. Django 4 By Example: Build powerful and reliable Python web applications from scratchby Antonio Mele (Author), Bob Belderbos (Author)
7. Django for Beginners: Build websites with Python and Djangoby William S. Vincent
8. Beginning Django By Daniel Rubio

Course Code	Course Name	L	T	P	Cr
DCSE250099	Object Oriented Analysis and Design	3	1	0	4

Course Outcomes:

CO1: Understand and apply the principles of object-oriented modeling and system abstraction.
CO2: Design effective class, state, and interaction models using UML.
CO3: Apply systematic processes to system conception, domain analysis, and system design.
CO4: Develop class designs and model implementation while integrating legacy systems.
CO5: Identify and utilize appropriate design patterns and idioms for building scalable systems.

Module 1:

Introduction, Modeling Concepts & Class Modeling:
Object Orientation Concepts-What is OO development? OO themes and assumptions, OO development usefulness and history; Modeling as a Design Technique - Modeling and abstraction, Three models: class, state, interaction; Class Modeling- Objects and classes, links and associations, Generalization and inheritance, Sample class model and navigation

Module 2:

Advanced Class Modeling and State Modeling:
Advanced Class Modeling - N-ary associations, association ends, aggregation, Abstract classes, multiple inheritance, Reification, metadata, derived data, constraints, Packages and modeling tips; State Modeling - Events, states, transitions, and conditions, State diagrams and behaviour, State modeling practicals

Module 3:

Advanced State and Interaction Modeling:
Advanced State Modeling - Nested diagrams and concurrency, Signal generalization, sample models, Relationship between class and state models; Interaction Modeling - Use case models, sequence models, activity models, Use case relationships, Procedural models and special constructs in activity modeling

Module 4:

Process, System Conception & Analysis:
Process Overview - Life cycle and development stages; System Conception - System concept, elaboration, and problem statement; Domain Analysis - Domain class, state, and interaction models, Iterative analysis process; Application Analysis & System Design - Application interaction, class, and state models, Performance estimation, reuse planning, Subsystem design, concurrency, and resource management, Architectural styles and ATM system example

Module 5:

Class Design, Implementation & Patterns:
Class Design & Implementation Modeling - Bridging gaps, use case realization, Refactoring and recursion, Design optimization and inheritance adjustments, Realizing associations and testing; Legacy Systems - Reverse engineering - class, interaction, state models, Wrapping and maintenance; Design Patterns – I & II - What makes a pattern? Pattern categories and relations, Communication Patterns: Forwarder-Receiver, Publisher-Subscriber, Management Patterns: Command Processor, View Handler; Idioms - Introduction, usage, style, Example: Counted Pointer Idiom

Labs / Practicals:

NA

References:

1.Michael Blaha, James Rumbaugh: Object-Oriented Modelling and Design with UML, 2nd Edition, Pearson Education, 2005.
2.Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007.
3. Grady Booch et al: Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson Education, 2007.
4.Brahma Dathan, Sarnath Ramnath: Object-Oriented Analysis, Design, and Implementation, Universities Press, 2009.
5.Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, Wiley- Dreamtech India, 2004.
6.Simon Bennett, Steve McRobb and Ray Farmer: Object-Oriented Systems Analysis and Design Using UML, 2 Edition, Tata McGraw-Hill, 2002.

Course Code	Course Name	L	T	P	Cr
DOAI250003	Advanced Statistical Techniques	3	1	0	4

Course Outcomes:

CO1: Apply sampling techniques and estimation methods to analyze data and evaluate estimator properties like unbiasedness, consistency, efficiency, and sufficiency.
CO2: Apply the principle of maximum likelihood and method of least squares for parameter estimation, and construct confidence intervals for population parameters using Normal, t, and Chi-square distributions.
CO3: Formulate statistical hypotheses, differentiate between simple and composite hypotheses, apply tests of significance, identify critical regions, and analyze decision errors and significance levels in statistical inference and decision-making processes.
CO4: Conduct hypothesis tests using sampling distributions, including large sample tests, t-tests, chi-square tests, and F-tests to evaluate means, proportions, variances, and goodness of fit in statistical analysis.
CO5: Apply correlation and regression methods to analyze variable relationships, compute coefficients, and interpret regression lines and their properties effectively.

Module 1:

Sampling and Statistical Estimation Theory: Population and Sample, Random Samples, Sampling with and Without Replacement, Sampling Distributions, Sampling distributions of Mean, Proportion and Difference of Means, Standard Error, Parameter, Estimation of Parameters, Estimator, Properties of Estimators:(Unbiasedness, Consistency, Efficiency, Sufficiency), Point and Interval Estimates, Methods of point estimation, Method of Moments, Properties of Moment Method Estimators.

Module 2:

Principle of Maximum Likelihood: properties and applications, Method of least squares, Confidence Interval and confidence limits of Population Parameters Based on Normal, t and Chi-square Distributions.

Module 3:

Statistical Inference: Statistical Decisions, Statistical Hypothesis, Null and alternate hypothesis, simple & composite hypotheses, Tests of statistical Hypothesis and Significance, critical region, Types of errors, Level of significance.

Module 4:

Test of Sampling Distributions: Test of significance of Large sample, Test of single proportion and difference of proportions, Test of significance of Single Mean and difference of Means. Chi-square Test for population variance, Chi-square test of goodness of fit, Student t-test, t-test for single mean and difference of means, F-test for equality of population variances, F-test for equality of several means.

Module 5:

Correlation and Regression: Karl Pearson Coefficient of correlation, Limits of Correlation coefficient, Shortcut and Step deviation method, Correlation of Ranks. Partial Correlation coefficient, Multiple Correlation coefficient. Regression, Regression Coefficient, Properties of Regressions, Lines of Regression.

Labs / Practicals:

1. Solve the list of numeric problems in the specified topics.
2. Apply Advanced Statistical techniques to analyse any one real life dataset and submit the analysis results.

References:

References:
1. Gupta, S.C.andKapoor, V.K.(1997) Fundamentals of Mathematical Statistics. Sultan Chand and Sons, New

Delhi, 11.23-12.23.
2. Spiegel, M. R., Schiller, J. J., and Alu Srinivasan, R. (2013). Probability and Statistics, Fourth Edition, Schaum's Outline Series, McGraw Hill Companies, Inc
3. Manoj Kumar Srivastava and Namita Srivastava, Statistical Inference–Testing of Hypotheses, Prentice Hall of India, 2014.
4. Bansi lal, Sanjay Arora and Sudha Arora, Introducing Probability and Statistics, 2/e, Satya Prakash Publications, 2006.

Course Code	Course Name	L	T	P	Cr
DOAI250018	Data Mining	3	0	2	4

Course Outcomes:

CO1: Understand data warehousing concepts and apply frequent pattern, association, and sequential pattern mining techniques.
CO2: Perform classification, prediction, and clustering methods for different types of data.
CO3: Analyze time-series data using periodicity, trend, and similarity-based techniques.
CO4: Apply stream mining methodologies, address class imbalance, and perform graph and social network analysis.
CO5: Implement web mining techniques, distributed data mining, and explore recent trends in the field.

Module 1:

Introduction to Data Warehousing; Data Mining: Mining frequent
patterns,
association and correlations; Sequential Pattern Mining concepts, primitives,
scalable methods;

Module 2:

Unit 2:
Classification and prediction; Cluster Analysis – Types of Data in Cluster Analysis, Partitioning methods, Hierarchical Methods; Transactional Patterns and other
temporal based frequent patterns,

Module 3:

Mining Time series Data, Periodicity Analysis for time related sequence data,
Trend analysis, Similarity search in Time-series analysis;

Module 4:

Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams, Class Imbalance Problem; Graph Mining; Social Network Analysis;

Module 5:

Web Mining, Mining the web page layout structure, mining web link structure,
mining multimedia data on the web, Automatic classification of web documents

and web usage mining; Distributed Data Mining.
Recent trends in Distributed Warehousing and Data Mining, Class Imbalance Problem; Graph Mining; Social Network Analysis

Labs / Practicals:

t-I Build Data Warehouse and Explore WEKA
Page No

03
2
-Perform data preprocessing tasks and Demonstrate
performing association rule mining on data sets
-Demonstrate performing classification on data sets
-Demonstrate performing clustering on data sets
-Demonstrate performing Regression on data sets

References:

1. Jiawei Han and M Kamber, Data Mining Concepts and Techniques,, Second Edition, Elsevier Publication, 2011.

2. Vipin Kumar, Introduction to Data Mining - Pang-Ning Tan, Michael Steinbach, Addison Wesley, 2006.

3. G Dong and J Pei, Sequence Data Mining, Springer, 2007.

Course Code	Course Name	L	T	P	Cr
DOAI250024	Deep Learning	2	0	4	4

Course Outcomes:

CO1: Understand the key concepts and training techniques of neural networks.
CO2: Apply deep learning models like CNNs and RNNs to solve real-world problems.
CO3: Gain expertise in advanced models like transformers and generative models for diverse applications.

Module 1:

Introduction to Neural Networks: Basics of Artificial Intelligence and Machine Learning, Introduction to Neural Networks and Biological Inspiration, Perceptron Model and Multilayer Perceptrons (MLP), Activation Functions (Sigmoid, ReLU, Tanh, Softmax), Gradient Descent and Backpropagation

Module 2:

Deep Feedforward Networks: Deep Learning vs. Traditional Machine Learning, Architecture of Feedforward Neural Networks (FNN), Weight Initialization Techniques, Optimization Techniques (SGD, Adam, RMSprop), Vanishing and Exploding Gradient Problems
Convolutional Neural Networks (CNNs): Introduction to Convolutional Neural Networks, Convolutional Layers, Pooling, and Padding, Popular Architectures (LeNet, AlexNet), Applications of CNNs in Image Processing

Module 3:

Recurrent Neural Networks (RNNs) and LSTMs: Introduction to Sequential Data and RNNs, Backpropagation Through Time (BPTT), Vanishing Gradient Problem in RNNs, Long Short-Term Memory (LSTMs) and Gated Recurrent Units (GRUs).

Module 4:

Generative Models and Autoencoders: Introduction to Generative Models, Autoencoders and Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Applications in Image Generation and Data Augmentation

Module 5:

Generative Models and Autoencoders: Introduction to Attention Mechanisms, Transformer Architecture, Self-Attention and Multi-Head Attention, BERT, GPT, and Large Language Models, Applications in NLP and Computer Vision

Labs / Practicals:

Preparing

References:

1. Charu C. Aggarwal, “ Neural Networks and Deep Learning”, Springer
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press
3. Rowel Atienza , “Advanced Deep Learning with TensorFlow 2 and Keras”, Packt Publishing
4. Lewis Tunstall, Leandro von Werra, Thomas Wolf, "Natural Language Processing with Transformers"
5. "Deep Learning for Natural Language Processing" – Palash Goyal, Sumit Pandey, Karan Jain
6. "Fundamentals of Deep Learning" – Nikhil Buduma, Nicholas Locascio

Course Code	Course Name	L	T	P	Cr
DOAI250027	Generative AI	3	0	2	4

Course Outcomes:

CO1: Understand the scope, applications, and ethical challenges of Generative AI across domains.
CO2: Explain traditional and deep learning-based approaches to language modeling and LLM architectures.
CO3: Analyze transformer models, GPT architectures, and their real-world use cases.
CO4: Apply prompt engineering techniques to design effective prompts for generative models.
CO5: Evaluate ethical implications, address bias, and explore real-world applications and future trends in Generative AI.

Module 1:

- Definition and Scope of Generative AI
- Applications of Generative AI in Various Domains
- Importance of Generative AI in Business, Healthcare, Art, and Science
- Ethical Considerations and Challenges in Generative AI
- Introduction to Language Models and Their Role in AI

Module 2:

- Traditional Approaches to Language Modeling
- Deep Learning-Based Approaches in NLP
- Overview of Popular LLM Architectures: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) Networks, Autoencoders & Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs)

Module 3:

- Introduction to Transformers and Self-Attention Mechanism
- Understanding GPT and Its Significance in AI
- Pre-training and Fine-Tuning Processes in GPT
- Architecture and Working of GPT Models
- Overview of GPT Variants and Their Use Cases (GPT-3, GPT-4, ChatGPT, etc.)

Module 4:

- Understanding the Concept and Significance of Prompt Engineering
- Strategies for Designing Effective Prompts
- Best Practices for Prompt Engineering in Generative AI
- Experimenting with Different Prompting Techniques
- Real-World Examples and Applications of Prompt Engineering

Module 5:

- Ethical Implications of Generative Models
- Addressing Bias and Fairness in AI Systems
- Ensuring Responsible Use and Deployment of Generative Models
- Use Cases in NLP, Content Generation, and Creative Applications
- Case Studies Highlighting Successful Implementations
- Potential Future Applications and Emerging Trends in Generative AI

Labs / Practicals:

Practical 1: Exploring Generative AI Use Cases
In this practical, students will explore the diverse applications of Generative AI across domains such as healthcare, business, art, and science. They will select one domain and research real-world use cases where Generative AI has made significant contributions. Students will summarize their findings in a short report or

presentation, focusing on the problem solved, the type of generative model used, and the impact of the solution.

Practical 2: Write a Python program to implement a basic Recurrent Neural Network (RNN) using TensorFlow or PyTorch. Train the model on a small text dataset and generate simple text sequences to demonstrate how the model learns patterns in the input.

Practical 3: Use a Python-based tool like TensorFlow Hub or PyTorch to load a pre-trained GAN model and generate images from random noise. Display multiple generated images and experiment by modifying the latent input vectors.

Practical 4: Write a Python program to build an Autoencoder model for text reconstruction. Provide the model with noisy or incomplete input sentences and train it to reconstruct the original version, demonstrating its application in denoising.

Practical 5: Write a Python program to implement a Variational Autoencoder (VAE) for generating synthetic text or image data. Visualize how sampling from different parts of the latent space produces different outputs.

Practical 6: Write a Python program to demonstrate the self-attention mechanism used in Transformer models. Simulate a simple attention block and visualize the attention scores across different input tokens.

Practical 7: Using OpenAI GPT Models (GPT-3 / GPT-4 / ChatGPT)
In this practical, students will interact with large language models such as GPT-3 or GPT-4 via OpenAI Playground or API. They will experiment with generating content such as poems, emails, or stories by modifying parameters like temperature, top-p, and prompt structure. The goal is to understand how GPT models behave and how output quality can be influenced by prompt design.

Practical 8: Designing Effective Prompts for LLMs
Students will learn and apply the concepts of prompt engineering by crafting prompts for different tasks such as summarization, translation, storytelling, and question-answering. They will compare the results of various prompt designs and document the impact of changes in wording, context, and formatting on the model's response.

References:

1. Generative Deep Learning by David Foster, 2nd Edition, O'Reilly.
2. Neural Network Methods for Natural Language Processing by Goldberg, Morgan & Claypool Publishers.