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

MSc Computer Science (with Specialization - Data Science)

(2025-26 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 premier Data Science program that promotes innovation, research, and ethical data use, preparing students to become globally competent professionals who turn data into actionable knowledge for the betterment of society.

Mission

To deliver strong academic and research training in data science and machine learning, nurturing critical thinking, ethical values, and innovation for successful careers in industry, academia, and entrepreneurship.

Program Education Objectives

PEO1. Apply advanced knowledge in data science, machine learning, and computational methods to solve complex real-world problems.
PEO2. Pursue successful careers in industry, research, or academia by leveraging their technical, analytical, and communication skills.
PEO3. Demonstrate ethical responsibility, lifelong learning, and leadership in multidisciplinary environments.
PEO4. Engage in innovation and entrepreneurship to contribute to technological and societal advancements.

Program Outcomes

PO1. Core Competency in Data Science:
Apply advanced data science, statistical, and computational techniques to analyze and solve real-world problems.
PO2. Technical Proficiency:
Design and implement efficient data-driven solutions using modern programming languages, tools, and platforms.
PO3. Ethics and Communication:
Demonstrate ethical responsibility in data practices and communicate effectively in professional and interdisciplinary settings.
PO4. Lifelong Learning and Innovation:
Engage in continuous learning and apply emerging technologies to innovate and adapt in a rapidly evolving digital landscape.

Program Specific Outcomes

PSO1. Apply data analytics to process and analyze large-scale data for decision-making.
PSO2. Design and develop intelligent systems and predictive models using machine learning and deep learning techniques.
PSO3. Use domain-specific knowledge (e.g., health, finance, business) to create data-driven solutions tailored to real-world needs.

CATEGORY WISE CREDIT DISTRIBUTION

Category	Category Code	Credits
Open Elective Courses	OE	8
Audit Courses (without grade or credit)	AU	0
Professional Elective Courses	PE	16
Skill / Employment Enhancement Courses (Project/Internship/Seminar/Dissertation/Research)	EE	36
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-MSCCSD-101	DOAI250044	Probability and Statistical Techniques	3	1	0	4
PC-MSCCSD-102	DCSA250015	Data Structures using Python	3	0	2	4
PC-MSCCSD-103	DCSE250060	Database Management Systems	2	0	4	4
EE-MSCCSD-104	DASH250090	Professional Skill Development	3	1	0	4
PE-MSCCSD-105	Elective	Program Elective				4
AU-MSCCSD-106	Elective	Audit Elective				0

Semester 2						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
PC-MSCCSD-201	DOAI250040	Mathematics for Machine Learning	3	1	0	4
PC-MSCCSD-202	DOAI250015	Data Analytics and Visualization	2	0	4	4
PE-MSCCSD-203	Elective	Program Elective				4
PE-MSCCSD-204	Elective	Program Elective				4
OE-MSCCSD-205	Elective	Open Elective				4
AU-MSCCSD-206	Elective	Audit Elective				0

Semester 3						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-MSCCSD-301	EE	Internship				4
PE-MSCCSD-302	Elective	Program Elective				4
OE-MSCCSD-303	Elective	Open Elective				4
EE-MSCCSD-304	EE	Mini Project				8

Semester 4						
Semester Code	**Course Code**	**Course Title**	**L**	**T**	**P**	**Cr**
EE-MSCCSD-401	EE	Main Project				20

Elective Courses

Elective Courses for PE Category						
Course Code	**Course Title**	**L**	**T**	**P**	**Cr**	**For Sem./Code**
DCSE250008	Advanced Programming with Python	3	0	2	4	105
DOAI250046	R Programming	2	0	4	4	105
DCSA250029	Digital Image Processing	3	0	2	4	203
DOAI250003	Advanced Statistical Techniques	3	1	0	4	203
DCSA250003	Mobile Application Development using Android	2	1	2	4	204
DCSA250081	Web Development using Python and Django	2	0	4	4	204
DOAI250009	Big Data Analytics	3	0	2	4	302
DOAI250012	Business Intelligence	2	0	4	4	302
DCSA250074	Software Engineering for Data Scientists	3	0	2	4	303
DOAI250041	Natural Language Processing	3	0	2	4	303

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
DOAI250044	Probability and Statistical Techniques	3	1	0	4

Course Outcomes:

CO1: Summarize and represent raw data using graphical tools, central tendency, dispersion, and shape measures for effective data interpretation.
CO2: Understand probability concepts and apply probability functions to analyse discrete and continuous random variables.
CO3: Identify and apply standard probability distributions and generating functions to solve problems involving random events and statistical modelling.
CO4: Analyse joint distributions, apply transformation techniques, and interpret multivariate dependence using covariance, correlation, and conditional expectations.
CO5: Compute and interpret moments, covariance and correlation matrices, and understand Central Limit Theorem.

Module 1:

Descriptive Statistics & Data Representation
Variables and Raw Data
Graphical Representation: Plots, charts, histograms, frequency polygons
Frequency Distributions: Relative, cumulative, and frequency curves
Measures of Central Tendency: Mean, median, mode
Measures of Dispersion: Range, standard deviation, quartile deviation, mean/median absolute deviation
Measures of shape: Skewness and Kurtosis

Module 2:

Probability Fundamentals and Random Variables
Introduction to Probability: Sample space, events, axioms
Random Variables: Definition, types (discrete/continuous), independence
Probability Functions: PMF (Probability Mass Function), PDF (Probability Density Function), CDF (Cumulative Distribution Function)
Joint Probability Distributions: Marginal and conditional distributions
Expectation and Variance: Linearity of expectation, covariance, correlation

Module 3:

Probability Distributions and Generating Functions
Discrete Distributions: Binomial, Poisson (properties, applications)
Continuous Distributions: Normal, Uniform, Exponential, Gamma, Beta
Relationship between Binomial and Normal, Poisson and Normal distributions
Generating Functions: Probability generating functions (PGF), Moment generating functions (MGF), Characteristic functions (properties and applications)

Module 4:

Multivariate Distributions and Transformations
Bivariate Random Variables: Joint distributions, conditional distributions.
Transformation Techniques: Single variable , Multiple variables (Jacobian method)
Conditional Expectation and Variance

Covariance and Correlation.

Module 5:

Moments: Raw vs. central moments, interpretation of Moments.
Covariance Matrix: Definition, properties, and eigenvalues.
Correlation Matrix, Central Limit Theorem.

Labs / Practicals:

N/A

References:

1. Fundamentals of Mathematical Statistics by S.C. Gupta & V.K. Kapoor
2. Probability, Random Variables, and Stochastic Processes by Athanasios Papoulis and S. Unnikrishna Pillai (4th Ed.)
3. An Introduction to Probability and Statistics by Vijay K. Rohatgi & A.K. Md. Ehsanes Saleh (3rd Ed.)
4. Statistics for Management by Richard I. Levin & David S. Rubin.
5. Applied Multivariate Analysis by S. Dasgupta.

Course Code	Course Name	L	T	P	Cr
DCSA250015	Data Structures using Python	3	0	2	4

Course Outcomes:

CO1: Demonstrate an in-depth understanding of core data structures and their applications.
CO2: Develop and implement advanced data structures using Python.
CO3: Evaluate algorithm performance using time and space complexity analysis.
CO4: Design and implement efficient algorithms for solving computational problems.
CO5: Build end-to-end applications using appropriate data structures.

Module 1:

Introduction to Data Structures: Overview of Data Structures, Types of Data Structures, Abstract Data Types (ADT), Complexity Analysis (Best, Worst, and Average Case), Big O Notation, Recursion in Python, Python Data Structures Overview (Lists, Tuples, Sets, and Dictionaries), Basic Input-Output Operations, Python Libraries for Data Structure Implementation (NumPy, SciPy).

Module 2:

Linear Data Structures: Arrays and Lists in Python, Array Operations (Insertion, Deletion, Searching, Sorting), Stack and Queue Implementations using Arrays and Linked Lists, Circular Queue and Priority Queue, Implementing Deque using Python, Problem Solving with Linear Data Structures.

Module 3:

Non-Linear Data Structures: Introduction to Trees, Binary Trees, Binary Search Trees (BST), AVL Trees, B-Trees, Tree Traversal Techniques (Inorder, Preorder, Postorder), Implementing Trees using Python, Graphs - Representation using Adjacency Matrix and List, Graph Traversal Techniques (BFS, DFS), Graph Algorithms (Dijkstra's, Floyd-Warshall, Prim's, Kruskal's), Pathfinding Algorithms using Python.

Module 4:

Hashing and Advanced Searching Algorithms: Hashing Techniques, Hash Functions, Collision Resolution (Chaining, Open Addressing), Implementing Hash Tables in Python, Searching Algorithms (Linear Search, Binary Search, Interpolation Search), Search Optimization Techniques, Implementing Search Operations on Large Data Sets

Module 5:

Sorting and Real-World Applications: Advanced Sorting Algorithms (Heap Sort, Merge Sort, Quick Sort, Radix Sort, Bucket Sort), Comparative Analysis of Sorting Algorithms, Implementing Sorting Algorithms using Python, Real-World Problem Solving using Data Structures, Capstone Project: Design and Implement an Efficient Data Structure-Based Solution for a Real-Life Problem.

Labs / Practicals:

1. Write a Python program to demonstrate the use of built-in data structures: Lists, Tuples, Sets, and Dictionaries.
2. Implement a Python program to calculate time complexity (operation counting) for linear and binary search.
3. Write a recursive Python program to find the factorial of a number and trace its call stack.
4. Create a Python script using NumPy to perform array operations such as addition and slicing.
5. Develop a Python function that uses recursion to print Fibonacci series up to 'n' terms.
6. Write a Python program to implement an array with operations: insertion, deletion, and search.
7. Implement a Stack using Python list and perform Push, Pop, and Display operations.
8. Develop a Python class to implement a Queue using list and perform Enqueue and Dequeue operations.
9. Write a Python program to implement a Circular Queue using a list or class.
10. Implement a Singly Linked List in Python with methods to insert, delete, and display nodes.

11. Write a Python program to create a Binary Search Tree (BST) and perform Inorder, Preorder, and Postorder traversals.
12. Create a Python program to insert elements into an AVL Tree and show the balanced tree after rotations.
13. Represent a Graph in Python using Adjacency Matrix and Adjacency List.
14. Write a program to perform Breadth-First Search (BFS) and Depth-First Search (DFS) on a graph.
15. Implement Dijkstra's algorithm in Python to find the shortest path between two nodes.
16. Develop a Python program to implement Hashing with Linear Probing for collision resolution.
17. Write a Python program to create a Hash Table using a dictionary and allow Insert, Search, and Delete operations.
18. Implement Linear and Binary Search algorithms in Python and compare their number of comparisons.
19. Write a Python program to implement Quick Sort and Merge Sort. Compare their time complexity using input of different sizes.
20. Design and implement a capstone project in Python that uses appropriate data structures to manage student records (insert, search, update, delete).

References:

1. Harsh Basin (2023). Data Structures with Python. Bpb Publication.
2. Dr. Shriram K. Vasudevan, Abhishek S Nagarajan, Karthick Nanmaran, Data Structures using Python, OUP India, Oxford University Press.

Course Code	Course Name	L	T	P	Cr
DCSE250060	Database Management Systems	2	0	4	4

Course Outcomes:

CO1. Understand the basic database concepts, data models, schemas and instances.
CO2. Learn database design using Entity Relationship model and learn the use of constraints and relational algebra operations.
CO3. Gain proficiency in SQL and construct queries using SQL.
CO4. Understand the importance of normalization in database design.
CO5. Understand transaction processing, concurrency control and recovery

Module 1:

Introduction to Databases, File systems vs. Database systems, Characteristics of DBMS, Applications of DBMS, Advantages of using a DBMS, Database users and administrators, Three-level architecture of DBMS (ANSI/SPARC architecture), Data independence, Database schema, Instance, Generalization, Specialization, Structure of a DBMS, Overview of DBMS software (e.g., MySQL, SQL Server, Oracle, PostgreSQL)

Module 2:

Introduction to Data models: Hierarchical, Network, Relational, Object-oriented, Data Modeling using Entity Relationship (ER) model: entities, entity types, attributes, relationships, relationship types, E/R diagram notation, examples.
Relational data model: Concepts, schema, instances, keys, constraints, Relational algebra: Select, project, union, set difference, Cartesian product, joins, ER to Relational mapping, Relational algebra and relational calculus, Tuple Relational Calculus, Domain Relational Calculus

Module 3:

Structured Query Language (SQL): Basics of SQL, DDL, DML,DCL, structure – creation, alteration, defining constraints – Primary key, foreign key, unique, not null, check, IN operator, Functions - aggregate functions, Built-in functions – numeric, date, string functions, set operations, sub-queries, correlated sub-queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types. Transaction control commands – Commit, Rollback, Save point, Cursors, Indexes, stored procedures, Triggers

Module 4:

Normalization – Introduction, Non loss decomposition and functional dependencies, First, Second, and third normal forms – dependency preservation, Boyce/Codd normal form.
Higher Normal Forms - Introduction, Multi-valued dependencies and Fourth normal form, Join dependencies and Fifth normal form.

Module 5:

Transaction management and Concurrency control: Transaction processing and Error recovery - concepts of transaction processing, ACID properties, and serializability concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, and database recovery management. Error recovery and logging, undo, redo, undo-redo logging and recovery methods, Introduction to database security concepts.

Labs / Practicals:

1. Create a database, define tables using DDL, apply constraints (Primary, Foreign, Not Null, Unique).
2. Insert, update, delete, and retrieve records using SQL DML commands.
3. Write SQL queries using WHERE, ORDER BY, GROUP BY, HAVING, LIKE, etc.
4. Write SQL queries to demonstrate the use of INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, and SELF JOIN using appropriate sample tables.
5. Write nested subqueries with IN, EXISTS, and ANY/ALL.

6. Draw ER diagram for a sample application (e.g., Library or Hospital), map it to relational schema.
7. Normalize sample unstructured data up to 3NF or BCNF. Implement before-and-after schemas in SQL.
8. Create and execute stored procedures/functions for common operations.

References:

1. Silberschatz, Korth, Sudarshan – Database System Concepts, McGraw Hill Year: 2020 (7th edition).
2. Elmasri and Navathe – Fundamentals of Database Systems, Pearson Edition 2016
3. Raghu Ramakrishnan, Johannes Gehrke – Database Management Systems, McGraw Hill
4. Database Systems Concepts Author H.f.Korth and Silberschatz Publisher McGraw Hill
5. Database System Author Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom Publisher Pearson Edition 2nd Edition
6. C.J. Date – An Introduction to Database Systems, Pearson

Course Code	Course Name	L	T	P	Cr
DASH250090	Professional Skill Development	3	1	0	4

Course Outcomes:

CO1: Understand and apply professional communication skills (written, oral, and digital) in academic and industry settings.
CO2: Analyze personal strengths and behavioral patterns to enhance workplace readiness and interpersonal relationships.
CO3: Demonstrate time management, teamwork, and leadership skills required in data-driven and technical project-based environments.
CO4: Prepare industry-specific professional documents such as résumés, cover letters, and project proposals.
CO5: Evaluate workplace ethics, digital etiquette, and cultural sensitivity in alignment with current Indian and global practices.

Module 1:

Communication for Professionals
Principles of effective communication – verbal, non-verbal, written; communication models; barriers and overcoming techniques; professional writing – email, reports, memos and proposals; oral presentations using digital tools (Zoom, Google Meet, MS Teams).

Module 2:

Self-Development and Emotional Intelligence
SWOT analysis; Self-Confidence and emotional regulation; Goal setting using SMART principles; growth versus fixed mindset; understanding personality types (MBTI, Big Five); self-reflection tools; aligning behavior with academic and workplace goals.

Module 3:

Teamwork, Leadership, and Time Management
Team dynamics and roles (Belbin's framework); Leadership styles – autocratic, democratic, transformational; conflict resolution; time management frameworks – Eisenhower Matrix, Pomodoro Technique; collaborative work in technical and data science environments.

Module 4:

Employability and Career Readiness
Résumé writing, cover letter and Statement of Purpose (SOP) preparation; Mock Interview and group discussion strategies; LinkedIn profile building; networking and job search strategies; understanding Job Descriptions (JDs), KRA/KPIs, and career mapping in core and IT/data roles.

Module 5:

Professional Ethics, Etiquette, and Digital Presence
Workplace values – integrity, accountability, respect; etiquette in email, meetings, and telephonic communication; digital footprint and cybersecurity hygiene; ethics in technology (AI/ML bias, data privacy); Diversity, Equity, and Inclusion (DEI) in modern workplaces.

Labs / Practicals:

N/A

References:

1. Barun K. Mitra – Personality Development and Soft Skills, Oxford University Press
2. Meenakshi Raman & Sangeeta Sharma – Technical Communication: Principles and Practice, Oxford University Press
3. K. Alex – Soft Skills: Know Yourself and Know the World, S. Chand Publishing

4. Ashraf M. Rizvi – Effective Technical Communication, McGraw Hill India
5. R.C. Sharma & Krishna Mohan – Business Correspondence and Report Writing, Tata McGraw Hill

Course Code	Course Name	L	T	P	Cr
DOAI250040	Mathematics for Machine Learning	3	1	0	4

Course Outcomes:

CO1: Grasp core linear algebra and calculus concepts essential for machine learning.
CO2: Analyze and build simple machine learning models using mathematical principles.
CO3: Perform optimization of loss functions using gradient-based methods.
CO5: Apply regularization and matrix decomposition to improve model performance.

Module 1:

Foundations of Linear Algebra
Vectors and matrices: operations, transpose, dot product
Norms and projections
Linear transformations: rotation, scaling, reflection
Determinants, rank, and matrix inverse
Eigenvalues and eigenvectors

Module 2:

Vector Spaces and Systems of Equations
Systems of linear equations and Gaussian elimination
Representation in matrix form
Gaussian elimination and row echelon form
Linear independence

Module 3:

Differential Calculus and Optimization
Functions, limits, and continuity
Derivatives and partial derivatives
Gradient and chain rule
Backpropagation basics
First-order optimization: Gradient Descent
Second-order optimization: Newton’s method
Visual interpretation of function optimization

Module 4:

Linear Models and Regularization
Simple and multiple linear regression
Cost functions and analytical solutions using Normal Equation
Assumptions in linear models
Regularization: Ridge and Lasso Regression
Model evaluation: MSE, R^2 score

Module 5:

Mathematical Tools for Advanced ML Models
Introduction to Information Theory: Entropy, KL Divergence
Support Vector Machines (SVM): Geometry and math foundations
Mathematical foundations of neural networks: Activation functions, loss functions, weight updates

Labs / Practicals:

1.Compute vector projections and dot products using NumPy.
2.Visualize linear transformations on 2D vectors.
3.Find eigenvalues and eigenvectors of given matrices.
4.Use Gaussian elimination to solve a system of equations.
5.Plot 3D surfaces and gradient vectors of loss functions.
6.Implement linear regression.
7.Apply Ridge and Lasso regression to real-world datasets using Scikit-learn.
8.Calculate and compare MSE and R^2 for models with and without regularization
9.Compute entropy and KL divergence between distributions.
10.Visualize SVM decision boundaries in 2D datasets.
11.Manually implement a simple neural network with forward and backward pass using NumPy.

References:

1.Higher Engineering Mathematics, By B.S. Grewal, Khanna Publishers
2.Mathematics for Machine Learning, By Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong, Cambridge University Press
3.Linear Algebra and Learning from Data, By Gilbert Strang, Wellesley-Cambridge Pres
4.The Elements of Statistical Learning, By Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer
5.Introduction to Machine Learning, By E. Alpaydin , PHI
6.Pattern Recognition and Machine Learning, By Christopher M. Bishop, Springer
7.Deep Learning, By Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press

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
DCSE250008	Advanced Programming with Python	3	0	2	4

Course Outcomes:

CO1: Demonstrate understanding of Python programming fundamentals including data types, operators, control flow, and console-based input/output.
CO2: Apply Python's built-in data structures such as lists, tuples, and dictionaries to solve computational problems using structured and modular programming.
CO3: Develop Python programs using functions, file handling, and functional programming constructs for efficient data processing.
CO4: Utilize NumPy and Matplotlib to perform scientific computing and visualize data for analytical insights.
CO5: Implement object-oriented programming concepts and apply Python in image processing to create modular and media-rich applications.

Module 1:

Python Programming Fundamentals
Introduction to Python
Technical Strength of Python
Introduction to Python Interpreter and Program Execution
Using Comments, Literals, Constants
Built-in Data Types
Numbers (Integers, Floats, Complex Numbers, Real, Sets)
Strings (Slicing, Indexing, Concatenation, other operations)
Input and Output console operations
Operators, Expressions, and Statements
Assignment Statement, Arithmetic, Relational, Logical, Bitwise Operators
Operator Precedence
Conditional Statements
Control Flow – range(), Loops (for, while)
Loop Control: break, continue, pass, assert

Module 2:

Data Structures and Problem Solving
Sequence Data Types
Lists, Tuples, Dictionaries
Indexing, Slicing, Concatenation, Mutability
Problem-solving with Sequences
Finding Maximum, Minimum, Mean
Linear Search on List/Tuple
Frequency Counting using Dictionary
Using Lists as Stacks and Queues
List Comprehensions
Scope and Modules
Scope of Objects and Names, LEGB Rule
Module Basics, Import Model, Reloading Modules

Module 3:

Functions and File Handling
Functions
Top-down Approach, Modular Programming
Function Parameters, Return, Local & Global Variables

Default & Keyword Arguments, VarArgs Parameters
Docstrings, Recursion
Built-in Functions: input(), eval(), print()
Functional Programming: lambda(), map(), reduce(), filter(), zip()
String Functions & Operations
Membership, Pattern Matching
File Handling
File Concepts: Open, Close
Read/Write: read(), readline(), readlines(), write(), writelines()
File Pointer: tell(), seek()
Command Line Arguments

Module 4:

NumPy and Data Visualization
Introduction to ndarray, Data Types, Array Attributes
Array Creation, Arrays from Ranges or Existing Data
Indexing and Slicing
Linear Algebra on n-D Arrays
Data Visualization with Matplotlib
Introduction
Scatter Plot, Line Plot, Bar Chart
Histogram, Box Plot

Module 5:

OOP in Python and Image Processing
Object-Oriented Programming in Python
Classes and Objects, Constructors
Multiple Objects, Class vs Data Attributes
Encapsulation, Inheritance, Polymorphism
Image Processing Basics
Introduction to Digital Image Processing
Image Operations: Crop, Scale, Rotate, Flip
Adjusting Contrast, Brightness, Color
Edge Detection, Blur, Sharpening

Labs / Practicals:

To be included

References:

1. Fluent Python by Luciano Ramalho
2. Head-First Python by Paul Barry
3. Think Python by Allen B. Downey
4. Python Cookbook by David Beazley and Brian K. Jones

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
DCSA250029	Digital Image Processing	3	0	2	4

Course Outcomes:

CO1: Understand the fundamental concepts of digital image formation, representation, and pixel relationships in various image types.
CO2: Explore and apply spatial domain techniques such as intensity transformations, filtering, and enhancement methods.
CO3: Analyze frequency domain techniques using Fourier and Cosine transforms for image enhancement and filtering.
CO4: Investigate image restoration methods and compression techniques to improve image quality and storage efficiency.
CO5: Apply morphological processing and segmentation techniques to extract meaningful features and objects from images.

Module 1:

Introduction to Digital Image Processing
Fundamentals of Digital Images: Definition, Applications, Components
Brightness, Adaptation, and Discrimination
Light and the Electromagnetic Spectrum
Image Sampling and Quantization
Basic Relationships Between Pixels (4-, 8-connectivity, adjacency, distance)
Types of Images: Binary, Grayscale, Color, Multispectral

Module 2:

Spatial Domain Filtering
Intensity Transformation Functions: Log, Power-law, Piecewise Linear
Histogram Processing: Histogram Equalization and Specification
Spatial Filtering Basics: Correlation and Convolution
Smoothing Filters:
Averaging Filters (Box, Gaussian)
Order-Statistics Filters (Median, Max, Min)
Sharpening Filters:
Laplacian Operator
Gradient Operators (Sobel, Prewitt)

Module 3:

Frequency Domain Filtering
Discrete Fourier Transform (DFT) and its Properties
Filtering in Frequency Domain:
Ideal Low-Pass and High-Pass Filters
Butterworth Filters
Discrete Cosine Transform (DCT): 1D and 2D Applications

Module 4:

Image Restoration and Compression
Image Degradation and Restoration Model
Noise Models: Gaussian, Salt-and-Pepper, Speckle
Restoration Techniques:

Mean, Median, Wiener Filters
Fundamentals of Image Compression
Compression Techniques:
Huffman Coding
Run-Length Encoding (RLE)
JPEG Compression Standard

Module 5:

Morphological Processing and Segmentation
Morphological Operations:
Erosion, Dilation, Opening, Closing
Hit-or-Miss Transformation
Basic Morphological Algorithms: Boundary Extraction, Region Filling, Convex Hull
Image Segmentation:
Point, Line, and Edge Detection (Sobel, Canny)
Thresholding (Global and Adaptive)
Region-Based Segmentation (Region Growing, Splitting and Merging)

Labs / Practicals:

1. Implement histogram equalization and spatial filtering using OpenCV/Matlab
2. Apply DFT and DCT transforms on sample images
3. Noise addition and restoration using filters
4. Encode and decode images using Huffman and RLE techniques
5. Morphological operations and object extraction
6. Perform image segmentation using thresholding and edge detection techniques

References:

1. Rafael C. Gonzalez & Richard E. Woods, Digital Image Processing, Pearson Education, 4th Edition
2. Anil K. Jain, Fundamentals of Digital Image Processing, PHI
3. S. Jayaraman et al., Digital Image Processing, Tata McGraw Hill
4. Bernd Jähne, Digital Image Processing, Springer
5.OpenCV Documentation and Python Libraries (https://docs.opencv.org)

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
DCSA250003	Mobile Application Development using Android	2	1	2	4

Course Outcomes:

CO1: Understand the architecture, lifecycle, and development environment of Android applications.
CO2: Design and build responsive user interfaces using Android layout and widget components.
CO3: Develop Android applications using activities, intents, and persistent data storage.
CO4: Integrate multimedia, location, and device services such as sensors and GPS in Android apps.
CO5: Deploy Android applications and apply debugging, testing, and publishing techniques.

Module 1:

Introduction to Android and Development Environment
Overview of Mobile App Development Landscape
Introduction to Android OS: History, Features, and Architecture
Android Studio: Installation and Setup
Android Project Structure (Manifest, Java/Kotlin files, Res folder)
Activity Lifecycle
Hello World App
Emulator and Device Testing Basics

Module 2:

User Interface Design and Layout Management
UI Components: TextView, EditText, Button, ImageView, CheckBox, RadioButton, Spinner
Layouts: LinearLayout, RelativeLayout, ConstraintLayout, FrameLayout
Event Handling and Listeners
Toasts, Dialog Boxes, and Snackbars
RecyclerView and Adapters
Menus: Options Menu, Context Menu

Module 3:

Activities, Intents, and Data Persistence
Intents: Explicit and Implicit
Activity Communication (startActivityForResult)
Fragments: Introduction and Usage
Shared Preferences and Internal Storage
SQLite Database: CRUD Operations
Room Persistence Library (Basics)

Module 4:

Advanced Features and Device Integration
Multimedia: Audio, Video Playback
Camera Integration
Accessing Device Sensors: Accelerometer, Gyroscope
Location and Maps: GPS and Google Maps API (Basic)
Background Tasks: AsyncTask, WorkManager (Intro)
Permissions and Security

Module 5:

App Deployment, Debugging and Project Development
Debugging Tools in Android Studio
Unit Testing and UI Testing Basics
App Signing and Versioning
Publishing App to Google Play (Overview)
Mini Project / Capstone Project
Best Practices in Android Development

Labs / Practicals:

1. Create a “Hello World” Android app using Android Studio and test on emulator.
2. Design a login UI screen using different layout managers and widgets.
3. Create an app with multiple activities and pass data using intents.
4. Store user preferences using SharedPreferences and display them.
5. Develop a simple SQLite-based app for student record management.
6. Integrate Google Maps to display current location.
7. Capture image using device camera and display in the app.
8. Create a media player app that plays audio or video from the device.
9. Develop an app with sensor-based interaction, such as shake detection.
10. Final Mini Project: A complete Android app with at least one form of persistent data storage and sensor/location integration.

References:

Kushwaha, D. S., & Misra, R.
Android Application Development: A Beginner's Guide
McGraw Hill Education, 2018.

Mednieks, Z., Dornin, L., Meike, G., & Nakamura, M.
Programming Android
O’Reilly Media, 3rd Edition, 2012.

Donn Felker
Android Application Development for Dummies
Wiley Publishing, 3rd Edition, 2015.

Joseph Annuzzi Jr., Lauren Darcey, and Shane Conder
Android Wireless Application Development (Volume I & II)
Addison-Wesley, 2015.

Wei-Meng Lee
Beginning Android Programming with Android Studio
Wiley, 4th Edition, 2021.

Online Resources

Android Developers Official Documentation
https://developer.android.com

Google Codelabs – Android Development Tutorials
https://developer.android.com/codelabs

Android Jetpack (UI, Room, WorkManager)
https://developer.android.com/jetpack

Kotlin Language Documentation
https://kotlinlang.org/docs/home.html

Udacity Course: Developing Android Apps with Kotlin
https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012

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
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
DOAI250012	Business Intelligence	2	0	4	4

Course Outcomes:

CO1:Understand BI Fundamentals: Explain the purpose, structure, and role of Business Intelligence systems in data-driven decision-making, while distinguishing BI from data analytics.

CO2:Apply BI Tools and Techniques: Perform data analysis using spreadsheet software, manage data with MySQL, and execute SQL queries to support BI processes.

CO3:Create Visualizations with Power BI: Build and customize interactive dashboards and reports in Power BI to effectively communicate insights through data visualizations.

CO4:Integrate Machine Learning with BI: Identify how Machine Learning enhances BI through automation, customer segmentation, and pattern discovery, while recognizing associated challenges.
CO5:Implement Predictive Analytics: Apply supervised and unsupervised Machine Learning algorithms, such as linear regression, K-Nearest Neighbors, and Decision Trees, and evaluate model accuracy for BI applications.

Module 1:

Meaning and purpose of BI, Structure of BI systems: data sources, ETL (Extract, Transform, Load), data warehouses, and reporting, Importance of data-driven decision-making through BI, Similarities and differences between BI and data analytics

Module 2:

BI analysis using spreadsheet software (e.g., Microsoft Excel), Using MySQL for data management, Understanding SQL: writing and executing queries (SELECT, JOIN, GROUP BY, etc.)

Module 3:

Introduction to Power BI: features and capabilities, Building interactive dashboards and reports, Data modeling and DAX (Data Analysis Expressions) in Power BI, Enhancing BI capabilities through visualizations

Module 4:

Differences between ML and BI, Applications of ML in BI: automating tasks, customer segmentation, uncovering hidden patterns, handling unstructured data, Challenges of integrating ML into BI (e.g., data quality, complexity, interpretability)

Module 5:

Machine Learning for Predictive Analytics, Supervised vs. unsupervised learning, Linear regression and classification techniques, ML algorithms: K-Nearest Neighbors, Naïve Bayes, Decision Trees, Implementing ML models and evaluating accuracy (e.g., confusion matrix, precision, recall)

Labs / Practicals:

Lab 1: Data Analysis with Excel
Use Microsoft Excel to perform basic BI analysis, including pivot tables, charts, and conditional formatting on a sample sales dataset.

Lab 2: SQL Query Execution in MySQL
Set up a MySQL database, create tables for customer and order data, and write SQL queries (SELECT, JOIN, GROUP BY) to extract insights.

Lab 3: Creating a Power BI Dashboard
Import a dataset into Power BI, create visualizations (bar charts, pie charts), and build an interactive dashboard

to display sales performance.

Lab 4: Data Modeling with DAX in Power BI
Use Power BI to create calculated columns and measures using DAX to analyze profit margins and sales trends in a retail dataset.

Lab 5: Customer Segmentation with Excel
Apply clustering techniques in Excel (e.g., using filters and pivot tables) to segment customers based on purchase history and demographics.

Lab 6: Automating BI Tasks with Power BI
Automate data refresh and report generation in Power BI using a sample dataset to simulate a real-world BI workflow.

Lab 7: Linear Regression for Sales Prediction
Implement a linear regression model in Python (using scikit-learn) to predict future sales based on historical data and evaluate model accuracy.

Lab 8: Classification with Naïve Bayes
Use Python to apply a Naïve Bayes classifier on a customer churn dataset to predict which customers are likely to leave, and assess performance using a confusion matrix.

Lab 9: Decision Trees for Pattern Discovery
Build a decision tree model in Python to identify patterns in a marketing campaign dataset, visualizing the tree and calculating precision and recall.

Lab 10: K-Nearest Neighbors for Customer Segmentation
Implement a K-Nearest Neighbors algorithm in Python to segment customers based on purchasing behavior, and visualize clusters using a scatter plot.

References:

1. Business Intelligence Guidebook: From Data Integration to Analytics by Rick Sherman
2. Business Intelligence: A Managerial Perspective on Analytics by Ramesh Sharda, Dursun, Efraim
3. Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications by Larissa T. Moss and Shaku Atre
4. The State of AI in Business Intelligence (e-book)
5. The Definitive Guide to DAX: Business Intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel by Marco Russo and Alberto Ferrari
6. Introduction to Machine Learning with Python, O'reilly
7. Understanding Machine Learning: From Theory To Algorithms by Shai Shalev-Shwartz
8. Learning MySQL by Hugh E. Williams (O'reilly
9. Learning SQL: Generate, Manipulate, and Retrieve Data by Alan Beaulieu

Course Code	Course Name	L	T	P	Cr
DCSA250074	Software Engineering for Data Scientists	3	0	2	4

Course Outcomes:

CO1: Describe software engineering principles and their significance in data science projects and development workflows.
CO2: Design and build data pipelines using structured software engineering practices and data science techniques.
CO3: Write modular, well-documented Python code using PEP8 standards, exception handling, and version control.
CO4: Apply testing methods, cloud tools, and DevOps practices for deploying and managing data science applications.
CO5: Analyze and optimize code performance using profiling, and implement engineering practices in real-world data projects.

Module 1:

Foundations of Software Engineering for Data Science
Introduction to Software Engineering
Need for Software Engineering in Data Science
Software Development Life Cycle (SDLC)
Overview of Software Process Models (Waterfall, Spiral, V-Model, Iterative)
Agile Development for Data Science Projects
Software Requirement Engineering: Functional and Non-functional Requirements
Architectural Design and Real-time System Design Basics

Module 2:

Data Science Workflow and Software Design
Data Science Lifecycle: Data Ingestion, Preprocessing, Training, Evaluation, Prediction, Deployment, Monitoring
Components of a Data Pipeline
Deployment Models for ML Systems
Role of Software Engineering in Data Science Workflows
Scripting and Automation

Module 3:

Writing Quality Code for Data Science
Writing Robust and Maintainable Code
PEP8 Guidelines, Code Formatting, Naming Conventions
Modular Programming and Reusability
Code Summarization and Docstrings
Exception Handling
Version Control using Git and GitHub
Developing and Sharing Python Packages

Module 4:

Testing, Cloud Engineering, and DevOps
Unit Testing, Integration Testing in Python (pytest/unittest)
CI/CD Basics for Data Science Projects
Cloud Engineering Overview (AWS/GCP/Azure): Storage, Compute, APIs

Introduction to DevOps for ML: MLOps Basics
Deployment Strategies (Docker, Heroku, Streamlit, FastAPI)

Module 5:

Performance Optimization and Case Studies
Code Performance: Time and Memory Usage
Time Complexity and Profiling Tools
Vectorization and Use of NumPy over Loops
Choosing Efficient Data Structures
Case Studies:
Health Care: Predictive diagnostics
Banking: Fraud detection
Integration of Software Engineering Best Practices in Real-World Data Science Applications

Labs / Practicals:

1. Prepare a requirement document for a simple data science application (e.g., customer churn prediction)
2. Create a basic UML diagram for a data analysis pipeline
3. Implement a basic data pipeline using Python
4. Build a script to automate data loading and clening
5. Refactor a spaghetti Python script into a modular, well-documented version
6. Set up a GitHub repo with commit messages and documentation
7. Write unit tests for a data preprocessing function
8. Containerize a Python model using Docker and deploy it using a cloud-based service or local server
9. Profile and optimize a poorly performing Python script using timeit and memory_profiler
10. Mini-project: Apply software engineering practices to develop a case-study based data science solution

References:

1. Ian Sommerville – Software Engineering (10th Edition), Pearson
2. Joel Grus – Data Science from Scratch: First Principles with Python, O'Reilly
3. Emilien Dupont et al. – Engineering MLOps, Packt
4. Python Enhancement Proposals (PEP 8) – https://peps.python.org/pep-0008/
5. Jeffrey Elkner et al. – How to Think Like a Computer Scientist: Learning with Python
6. Zed A. Shaw – Learn Python the Hard Way

Course Code	Course Name	L	T	P	Cr
DOAI250041	Natural Language Processing	3	0	2	4

Course Outcomes:

CO1: Understand the history, stages, challenges, and applications of NLP in real-world systems.
CO2: Apply morphological analysis, finite state methods, stemming, and n-gram models for language processing.
CO3: Implement POS tagging, CFG parsing, and sentence-level syntactic analysis using statistical and ML-based methods.
CO4: Analyze word sense disambiguation techniques using machine learning and dictionary-based approaches.
CO5: Apply discourse analysis techniques including reference resolution, pronoun interpretation, and coherence modeling.

Module 1:

History of NLP; Generic NLP system; Levels of NLP; Knowledge in language processing problem; Ambiguity in natural language; Stages in NLP; Challenges of NLP; Role of machine learning; Brief history of the field; Applications of NLP: Machine translation, Question answering system, Information retrieval, Text categorization, text summarization & Sentiment analysis

Module 2:

Morphology analysis survey of English morphology, inflectional morphology & derivational morphology; Regular expressions; Finite automata; Finite state transducers (FST); Morphological parsing with FST; Lexicon free FST, Porter stemmer, N-Grams, N-gram language model, N-gram for spelling correction.

Module 3:

Part-of-Speech tagging (POS); Lexical syntax tag set for English (Penn Treebank); Rule based POS tagging; Stochastic POS tagging; Issues: Multiple tags & words, unknown words, class-based n-grams, HM Model ME, SVM, CRF; Context Free Grammar; Constituency; Context free rules & trees; Sentence level construction; Noun Phrase; Coordination; Agreement; Verb phrase & sub categorization

Module 4:

Attachment for fragment of English sentences, noun phrases, verb phrases, prepositional phrases; Relations among lexemes & their senses; Homonymy, Polysemy based disambiguation & limitations, Robust WSD; Machine learning approach and dictionary-based approach.

Module 5:

Discourse reference resolution; Reference phenomenon; Syntactic & semantic constraints on co reference; Preferences in pronoun interpretation; Algorithm for pronoun resolution; Text coherence; Discourse structure

Labs / Practicals:

Module 1: Introduction to NLP & Applications
1. Experiment 1: Demonstration of NLP Applications
o Implement basic NLP applications like sentiment analysis, text summarization, question answering, and information retrieval using pre-trained models or APIs.
2. Experiment 2: Text Preprocessing Techniques
o Perform tokenization, stop-word removal, stemming, and lemmatization on sample English texts.

Module 2: Morphology and Language Models
3. Experiment 3: Morphological Analysis using Stemming and Lemmatization
o Analyze inflectional and derivational forms of English words using stemmers and lemmatizers.
4. Experiment 4: Implementation of N-Gram Language Models

o Build unigram, bigram, and trigram models from a text corpus and compute probabilities of word sequences.
5. Experiment 5: Spelling Correction using N-Gram Probabilities
o Implement a simple spelling corrector using edit distance and N-gram based word probability estimation.

Module 3: POS Tagging and Grammar Parsing
6. Experiment 6: POS Tagging using Rule-Based and Statistical Methods
o Perform part-of-speech tagging using both rule-based and statistical approaches; evaluate accuracy on tagged corpora.
7. Experiment 7: Constituency Parsing using Context-Free Grammar (CFG)
o Define a CFG and parse given sentences to generate parse trees showing sentence structure.

Module 4: Word Sense Disambiguation
8. Experiment 8: Word Sense Disambiguation using Dictionary-Based Methods
o Use the Lesk algorithm to identify correct word senses in different sentence contexts.
9. Experiment 9: Word Sense Disambiguation using Supervised Learning
o Implement a supervised machine learning model to classify the correct sense of an ambiguous word using labeled data.

Module 5: Discourse and Reference Resolution
10. Experiment 10: Co-reference Resolution and Discourse Analysis
• Resolve pronouns and noun phrase references in a passage and analyze coherence and discourse structure.

References:

1. James Allen. Natural Language Understanding. The Benajmins/Cummings Publishing Company Inc. 1994. ISBN 0-8053-0334-0.
2. Tom Mitchell. Machine Learning. McGraw Hill, 1997. ISBN 0070428077.
3. Cover, T. M. and J. A. Thomas: Elements of Information Theory. Wiley. 1991. ISBN 0-471-06259-6.