MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

MK University, Patan
Faculty of Engineering Technology,
Department of Data Science Engineering (DSE)

B. TECH (DATA SCIENCE ENGINEERING) SEM-I									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE101	Engineering Mathematics-I	4	0	4	40	60	100
2	MAJOR	BTDSE102	Programming For DS(Python)	4	2	6	40	60	100
3	MAJOR	BTDSE103	Statistics for DS	4	2	6	40	60	100
4	MINOR	BTDSE104	Introduction To AI & ML	4	0	4	40	60	100
5	VAC	BTDSE105	Communication Skills in – I	2	0	2	0	50	50
TOTAL				18	4	22	200	350	550

B. TECH (DATA SCIENCE ENGINEERING) SEM-II									
SR NO.	COURSE TYPE	COURSE CODE	CUORSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE201	DATA STRUCTURES & ALGORITHMS	4	2	6	40	60	100
2	MAJOR	BTDSE202	DATA BASE MANAGEMENT SYSTEMS	4	2	6	40	60	100
3	MAJOR	BTDSE203	LINEAR ALGEBRA FOR DS	4	0	4	40	60	100
4	MINOR	BTDSE204	DATA VISUALIZATION	4	2	6	40	60	100
5	VAC	BTDSE205	ENVIRONMENTAL SCIENCE	2	0	2	0	50	50
TOTAL				18	6	24	200	350	550

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

B. TECH (DATA SCIENCE ENGINEERING) SEM-III									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE301	MACHINE LEARNING-I	4	2	6	90	60	150
2	MAJOR	BTDSE302	BIG DATA TECHNOLOGIES	4	2	6	90	60	150
3	MAJOR	BTDSE303	PROBABLITY & STOCHASTIC PROCESSES	4	0	4	40	60	100
4	MINOR	BTDSE304	BUSINESS INTELLIGENCE	4	0	4	40	60	100
5	SEC	BTDSE305	SQL & NOSQL DATABASE	0	2	2	00	50	50
TOTAL				16	6	22	250	300	550

B. TECH (DATA SCIENCE ENGINEERING) SEM-IV									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE401	MACHINE LEARNING-II	4	2	6	40	60	100
2	MAJOR	BTDSE402	DEEP LEARNING FUNDAMENTALS	4	0	4	40	60	100
3	MAJOR	BTDSE403	CLOUD COMPUTING FOR DS	4	0	4	40	60	100
4	MINOR	BTDSE404	NATURAL LANGUAGE PROCESSING	4	0	4	40	60	100
5	SEC	BTDSE405	DATA WRANGLING WITH PANDAS	0	2	2	40	60	100
6	VAC	BTDSE406	Indian Constitution	2	0	2	0	50	50
TOTAL				18	4	22	200	350	550

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

B. TECH (DATA SCIENCE ENGINEERING) SEM-V									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE501	ADVANCED DEEP LEARNING	4	2	6	90	60	150
2	MAJOR	BTDSE502	TIME SERIES ANALYSIS	4	2	6	40	60	100
3	MAJOR	BTDSE503	DATA ENGINEERING	4	2	6	90	60	150
4	MINOR	BTDSE504	STREAM PROCESSING (KAFKA/SPARK)	0	2	2	40	60	100
6	VAC	BTDSE505	MINI-PROJECT	0	4	4	50	0	50
TOTAL				12	12	24	250	300	350

B. TECH (DATA SCIENCE ENGINEERING) SEM-VI									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE601	REINFORCEMENT LEARNING	4	0	4	40	60	100
2	MAJOR	BTDSE602	MLOPS & MODEL DEVELOPMENT	4	2	6	90	60	150
3	MAJOR	BTDSE603	BIG DATA ANALYTICS	4	2	6	90	60	150
4	MINOR	BTDSE604	ETHICAL AI & RESPONSIBLE DS	4	0	4	40	60	100
5	SEC	BTDSE605	APTITUDE & CARRER SKILLS	0	2	2	50	0	50
TOTAL				16	6	22	250	300	550

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

B. TECH (DATA SCIENCE ENGINEERING) SEM-VII									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTDSE701	ADVANCED MACHINE LEARNING	4	2	6	90	60	150
2	MAJOR	BTDSE702	CLOUD NATIVE DATA SYSTEMS	4	2	6	90	60	150
3	MINOR	BTDSE703	IOT & SENSOR DS	4	0	4	40	60	100
4	SEC	BTDSE704	DEVOPS FOR DS	0	2	2	00	50	50
5	VAC	BTDSE705	Project Phase-I	0	4	4	100	00	100
TOTAL				12	10	22	220	230	550

B. TECH (DATA SCIENCE ENGINEERING) SEM-VIII									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERN AL	EXTERNAL	
1	MAJOR	BTDSE801	RESEARCH METHDOLOGY IN DS	4	0	4	40	60	100
2	MAJOR	BTDSE802	ENTERPRISE SOLUTIONS	4	2	6	90	60	150
3	MINOR	BTDSE803	PARALLEL COMPUTING	4	2	6	90	60	150
4	SEC	BTDSE804	Project Phase-II	0	10	10	100	100	200
TOTAL				12	14	26	320	280	600

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SUBJECT CODE: BTDSE101

SUBJECT NAME: ENGINEERING MATHEMATICS-I

Course Objective:

- The concept of rank of a matrix which is used to know the consistency of system of linear equations and also to find the eigen vectors of a given matrix.
- Finding maxima and minima of functions of several variables.
- Applications of first order ordinary differential equations. (Newton's law of cooling, Natural growth and decay)
- How to solve first order linear, nonlinear partial differential equations and also method of separation of variables technique to solve typical second order partial differential equations.
- Solving differential equations using Laplace Transforms.

Course Outcomes: At the end of the course students shall be able to

CO1	The concept of rank of a matrix which is used to know the consistency of system of linear equations and also to find the eigen vectors of a given matrix
CO2	Finding maxima and minima of functions of several variables
CO3	Applications of first order ordinary differential equations
CO4	How to solve first order linear, nonlinear partial differential equations and also method of separation of variables technique to solve typical second order partial differential equations

Unit	Content	Credit	Weightage
I	Matrices Introduction, types of matrices-symmetric, skew-symmetric, Hermitian, skew-Hermitian, orthogonal, unitary matrices. Rank of a matrix - echelon form, normal form, consistency of system of linear equations (Homogeneous and Non-Homogeneous). Eigen values and Eigen vectors and their properties (without proof), Cayley-Hamilton theorem (without proof), Diagonalization.	1	25%
II	Functions of Several Variables Limit continuity, partial derivatives and total derivative. Jacobian-Functional dependence and independence. Maxima and minima and saddle points, method of Lagrange multipliers, Taylor's theorem for two variables.	1	25%
III	Ordinary Differential Equations First order ordinary differential equations: Exact, equations reducible to exact form. Applications of first order differential equations - Newton's law of cooling, law of natural growth and decay. Linear differential equations of second and higher order with constant coefficients: Non-homogeneous term of the type f(x) = eax, sinax, cosax, xn , eax V and xn V. Method of variation of parameters.	1	25%

IV	Partial Differential Equations Introduction, formation of partial differential equation by elimination of arbitrary constants and arbitrary functions, solutions of first order Lagrange's linear equation and non-linear equations, Charpit's method, Method of separation of variables for second order equations and applications of PDE to one dimensional (Heat equation).	1	25%

TEXT BOOKS:

1. Higher Engineering Mathematics by B V Ramana ., Tata McGraw Hill.
2. Higher Engineering Mathematics by B.S. Grewal, Khanna Publishers.
3. Advanced Engineering Mathematics by Kreyszig, John Wiley & Sons.

REFERENCE BOOKS:

i)Advanced Engineering Mathematics by R.K Jain & S R K Iyenger, Narosa Publishers.
ii)Advanced Engineering Mathematics by Michael Green Berg, Pearson Publishers.
iii)Engineering Mathematics by N.P Bali and Manish Goyal.

SUBJECT CODE: BTDSE102

SUBJECT NAME: PROGRAMING FOR DS (PYTHON)

Course Objectives:

- To introduce Python as the primary programming language for data science.
- To develop proficiency in Python programming fundamentals, data structures, and libraries.
- To implement data manipulation, visualization, and basic analysis using Python.
- To prepare students for advanced data science topics through hands-on programming practice.

Course Outcomes: At the end of the course students shall be able to

CO1	Write, debug, and execute Python programs using core language constructs.
CO2	Use Python data structures (lists, dictionaries, sets, tuples) for efficient data handling.
C03	Perform data manipulation and analysis using NumPy and Pandas libraries.
C04	Create basic data visualizations using Matplotlib and Seaborn.

Unit	Content	Credit	Weightage
I	**Python Fundamentals** • **Introduction to Python:** History, features, installation, IDEs (Jupyter, VS Code). • **Basic Syntax:** Variables, data types, operators, input/output. • **Control Structures:** Conditional statements (if, elif, else), loops (for, while). • **Functions:** Definition, parameters, return values, lambda functions, scope. • **File Handling:** Reading/writing text and CSV files.	1	25%
II	**Data Structures in Python** • **Lists:** Creation, indexing, slicing, list	1	25%

	comprehensions, methods. • **Tuples and Sets:** Immutable sequences, set operations. • **Dictionaries:** Key-value pairs, methods, dictionary comprehensions. • **Strings:** String methods, formatting, regular expressions (regex basics). • **Error Handling:** Try-except blocks, custom exceptions.		
III	**Data Manipulation with NumPy and Pandas** • **NumPy:** Arrays creation, array operations, broadcasting, mathematical functions. • **Pandas Series and Data Frames:** Creation, indexing, data selection, filtering. • **Data Cleaning:** Handling missing values, duplicates, data transformation. • **Data Aggregation:** Group By operations, pivot tables, merging/joining datasets.	1	25%
IV	**Data Visualization and Basic Analysis** • **Matplotlib:** Line plots, bar charts, scatter plots, histograms, customization. • **Seaborn:** Statistical visualizations, heatmaps, pair plots, styling. • **Exploratory Data Analysis (EDA):** Descriptive statistics, correlation, outlier detection. • **Mini-Project:** End-to-end analysis of a real-world dataset.	1	25%

Textbooks:

- Python for Data Analysis by Wes McKinney (O'Reilly)
- Python Crash Course by Eric Matthes (No Starch Press)
- Data Science from Scratch by Joel Grus (O'Reilly)

Reference books:

- Fluent Python by Luciano Ramalho (O'Reilly)
- Python Data Science Handbook by Jake VanderPlas (O'Reilly)
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron

Online Platforms:

- Python Official Documentation: docs.python.org
- Kaggle Learn: Python and Pandas courses
- Coursera: "Python for Everybody" by University of Michigan
- Real Python: Tutorials and articles
- Stack Overflow: Q/A for programming issues

PRACTICAL LIST:

Module 1 Practical's:

- Lab 1: Python environment setup, basic I/O, and arithmetic operations.
- Lab 2: Control structures: Create a number guessing game.
- Lab 3: Functions: Write reusable code for factorial, Fibonacci, and prime checks.
- Lab 4: File handling: Read/write CSV, log file processing.

Module 2 Practical's:

- Lab 5: List operations: Sorting, searching, list comprehensions.

- Lab 6: Dictionary and set manipulations: Word frequency counter.
- Lab 7: String processing and regex: Email/phone number validation.
- Lab 8: Error handling: Robust input validation and file reading.

Module 3 Practical's:

- Lab 9: NumPy arrays: Matrix operations, statistical calculations.
- Lab 10: Pandas Data Frame: Data loading, filtering, and basic analysis.
- Lab 11: Data cleaning: Handling missing data, outliers, duplicates.
- Lab 12: Data aggregation: Group By and merging datasets.

Module 4 Practical's:

- Lab 13: Matplotlib: Create multiple plot types with customization.
- Lab 14: Seaborn: Advanced visualizations for categorical/numerical data.
- Lab 15: EDA on a dataset: Summary stats, correlation, visual insights.
- Lab 16: Mini-Project: Analyze a dataset (e.g., Titanic, Iris) and present findings.

SUBJECT CODE: BTDSE103

SUBJECT NAME: STATISTICS FOR DS

Course Objectives:

- To provide a strong foundation in statistical concepts essential for data analysis and machine learning.
- To develop skills in descriptive and inferential statistics for data-driven decision making.
- To apply probability theory and statistical methods to real-world data science problems.
- To prepare students for advanced analytical techniques through practical statistical reasoning.

Course Outcomes: At the end of the course students shall be able to

CO1	Apply descriptive statistics and probability distributions to summarize and model data.
CO2	Perform hypothesis testing and confidence interval estimation for inference.
C03	Conduct regression analysis and ANOVA for relationship modeling and comparison.
C04	Use statistical software/tools (Python/R) for data analysis and interpretation.

Unit	Content	Credit	Weightage
I	**Descriptive Statistics & Probability Fundamentals** • **Introduction to Statistics:** Role in data science, types of data, data collection methods. • **Descriptive Statistics:** Measures of central tendency (mean, median, mode), measures of dispersion (variance, standard deviation, range, IQR). • **Data Visualization:** Histograms, box plots, scatter plots, Q-Q plots. • **Probability Basics:** Sample space, events, axioms, conditional probability, Bayes' theorem. • **Random Variables:** Discrete and continuous, probability distributions.	1	25%
II	**Probability Distributions & Sampling Theory**	1	25%

	• **Discrete Distributions:** Binomial, Poisson, Geometric. • **Continuous Distributions:** Normal, Exponential, Uniform. • **Central Limit Theorem:** Concept, applications, sampling distribution of mean. • **Sampling Methods:** Random sampling, stratified, cluster, bootstrapping. • **Estimation Theory:** Point estimation, properties of estimators (bias, efficiency).		
III	**Inferential Statistics** • **Hypothesis Testing:** Null and alternative hypotheses, p-value, significance level, type I/II errors. • **Parametric Tests:** Z-test, t-test (one sample, two sample, paired), chi-square test. • **Non-parametric Tests:** Wilcoxon signed-rank test, Mann-Whitney U test. • **Confidence Intervals:** For means, proportions, difference between means. • **Power Analysis:** Sample size determination.	1	25%
IV	**Regression & Analysis of Variance** • **Simple Linear Regression:** Model formulation, least squares estimation, assumptions, R^2. • **Multiple Linear Regression:** Model building, multicollinearity, interpretation. • **ANOVA:** One-way and two-way ANOVA, F-test, post-hoc tests. • **Model Diagnostics:** Residual analysis, heteroscedasticity, normality checks. • **Introduction to Logistic Regression:** For binary classification.	1	25%

Textbooks:

- Statistics for Data Scientists by Maurizio Carpita and Marica Manisera (Springer)
- Practical Statistics for Data Scientists by Peter Bruce and Andrew Bruce (O'Reilly)
- Introduction to Statistical Learning by Gareth James et al. (Springer)

Reference books:

- The Elements of Statistical Learning by Trevor Hastie et al. (Springer)
- Naked Statistics by Charles Wheelan (Norton)
- Statistics in Plain English by Timothy C. Urdan

Online Platforms:

- Khan Academy: Statistics and probability modules
- Coursera: "Statistics with Python" (University of Michigan)
- edX: "Probability and Statistics in Data Science using Python" (UC San Diego)
- StatQuest with Josh Starmer (YouTube)
- Towards Data Science (Medium publication)
- RPubs & R-bloggers (for R-based statistical learning)
- Google Colab / Jupyter Notebooks for hands-on practice

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Data summarization using Python/R – mean, median, variance, SD.
- Lab 2: Data visualization – histograms, box plots, scatter plots using Matplotlib/Seaborn.
- Lab 3: Probability calculations – basic probability, conditional probability, Bayes' theorem.
- Lab 4: Random variable simulation and distribution plotting.

Module 2 Practicals:

- Lab 5: Discrete probability distributions – Binomial and Poisson simulations.
- Lab 6: Continuous distributions – Normal and Exponential PDF/CDF plots.
- Lab 7: Central Limit Theorem demonstration via simulation.
- Lab 8: Bootstrapping for confidence intervals.

Module 3 Practicals:

- Lab 9: Hypothesis testing – Z-test and t-test implementation.
- Lab 10: Chi-square test for independence and goodness-of-fit.
- Lab 11: Non-parametric tests – Wilcoxon and Mann-Whitney.
- Lab 12: Confidence intervals for mean and proportion.

Module 4 Practicals:

- Lab 13: Simple linear regression – model fitting and interpretation.
- Lab 14: Multiple regression – variable selection and diagnostics.
- Lab 15: ANOVA – one-way and two-way analysis with post-hoc tests.
- Lab 16: Logistic regression for binary classification.

SUBJECT CODE: BTDSE104

SUBJECT NAME: INTRODUCTION TO AI & ML

Course Objectives:

- To introduce fundamental concepts of Artificial Intelligence (AI) and Machine Learning (ML).
- To understand different types of machine learning: supervised, unsupervised, and reinforcement learning.
- To explore basic algorithms and real-world applications of AI and ML.
- To develop a conceptual foundation for advanced AI/ML topics in subsequent semesters.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain core AI concepts, intelligent agents, and problem-solving using search algorithms.
CO2	Differentiate between supervised, unsupervised, and reinforcement learning paradigms.
C03	Apply basic ML algorithms for classification, regression, and clustering tasks.
C04	Evaluate ethical implications and real-world applications of AI/ML systems.

Unit	Content	Credit	Weightage
I	**Foundations of Artificial Intelligence** • **Introduction to AI:** History, goals, and scope of AI. • **Intelligent Agents:** Types of agents, environments, rationality. • **Problem-Solving by Search:** Uninformed search (BFS, DFS), informed search (A*, greedy best-first). • **Adversarial Search:** Game theory, minimax algorithm, alpha-beta pruning.	1	25%

	• **Knowledge Representation:** Propositional and first-order logic basics.		
II	**Introduction to Machine Learning** • **ML Paradigms:** Supervised, unsupervised, reinforcement learning. • **Data Preprocessing:** Feature engineering, normalization, handling missing data. • **Model Evaluation:** Training-test split, cross-validation, performance metrics (accuracy, precision, recall, F1-score). • **Bias-Variance Tradeoff:** Underfitting vs. overfitting, regularization.	1	25%
III	**Supervised Learning Algorithms** • **Linear Regression:** Simple and multiple linear regression, gradient descent. • **Logistic Regression:** Binary classification, sigmoid function, decision boundary. • **Decision Trees:** Entropy, information gain, Gini impurity, tree pruning. • **k-Nearest Neighbors (k-NN):** Distance metrics, k-value selection. • **Support Vector Machines (SVM):** Linear SVM, kernel trick (introduction).	1	25%
IV	**Unsupervised Learning & AI Ethics** • **Clustering Algorithms:** k-Means clustering, hierarchical clustering. • **Dimensionality Reduction:** Principal Component Analysis (PCA) basics. • **Introduction to Reinforcement Learning:** Agent, environment, reward, policy. • **AI Ethics:** Bias in AI, fairness, accountability, transparency, societal impact. • **Real-World Applications:** Healthcare, finance, autonomous systems, NLP, computer vision (overview).	1	25%

Textbooks:

- Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig (4th Edition)
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron (3rd Edition)
- Pattern Recognition and Machine Learning by Christopher M. Bishop

Reference books:

- Machine Learning Yearning by Andrew Ng
- Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido
- The Hundred-Page Machine Learning Book by Andriy Burkov
- AI Ethics by Mark Coeckelbergh

Online Platforms:

- Coursera: “Machine Learning” by Andrew Ng (Stanford)
- edX: “Artificial Intelligence” by Columbia University
- Kaggle Learn: Intro to Machine Learning

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-II

SUBJECT CODE: BTDSE201

SUBJECT NAME: DATA STRUCTURES AND ALGORITHMS

Course Objectives:

- To understand fundamental and advanced data structures and their applications in data science.
- To develop algorithmic thinking and problem-solving skills using efficient data structures.
- To analyse time and space complexity of algorithms using asymptotic notations.
- To implement and apply data structures in Python for data manipulation and analysis tasks.

Course Outcomes: At the end of the course students shall be able to

CO1	Analyze algorithm complexity and apply appropriate data structures for given problems.
CO2	Implement and utilize linear data structures (arrays, linked lists, stacks, queues) for data processing.
C03	Design and apply non-linear data structures (trees, graphs, hash tables) for efficient data storage and retrieval.
C04	Solve real-world data science problems using algorithmic paradigms (searching, sorting, greedy, dynamic programming).

Unit	Content	Credit	Weightage
I	**Algorithm Analysis & Linear Data Structures** • **Algorithm Analysis:** Asymptotic notations (Big-O, Omega, Theta), time-space tradeoffs. • **Arrays & Lists:** Static vs dynamic arrays, Python lists, operations, memory allocation. • **Linked Lists:** Singly, doubly, circular linked lists, applications in data streaming. • **Stacks & Queues:** LIFO/FIFO principles, implementations, applications (parsing, scheduling). • **Python Collections:** Built-in data structures (list, tuple, set, dict) and their complexities.	1	25%
II	**Trees & Hierarchical Data Structures** • **Trees:** Terminology, binary trees, tree traversals (in order, preorder, post order). • **Binary Search Trees (BST):** Insertion, deletion, searching, balanced BST concepts. • **Heaps:** Min-heap, max-heap, heap operations, priority queues. • **Tries:** Structure, applications in autocomplete and dictionary implementations. • **Tree Applications:** Hierarchical clustering, decision trees (ML context), file systems.	1	25%
III	**Graphs & Hashing** • **Graphs:** Terminology, representations (adjacency list/matrix), BFS, DFS. • **Graph Algorithms:** Shortest path (Dijkstra), minimum spanning tree (Prim, Kruskal).	1	25%

	• **Hashing:** Hash functions, collision resolution (chaining, open addressing), load factor. • **Hash Tables:** Python dictionaries, sets, applications in data indexing and de-duplication. • **Graph Applications:** Social network analysis, recommendation systems, pathfinding.		
IV	**Algorithmic Paradigms & Optimization** • **Searching Algorithms:** Linear search, binary search, interpolation search. • **Sorting Algorithms:** Bubble, selection, insertion, merge, quick, heap sorts. • **Greedy Algorithms:** Activity selection, Huffman coding, coin change problem. • **Dynamic Programming:** Fibonacci, knapsack, longest common subsequence. • **Algorithmic Thinking for DS:** Space-time trade offs in big data, streaming algorithms overview.	1	25%

Textbooks:

- Data Structures and Algorithms in Python by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser
- Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
- Problem Solving with Algorithms and Data Structures Using Python by Brad Miller and David Ranum

Reference books:

- The Algorithm Design Manual by Steven S. Skiena
- Algorithms by Robert Sedgewick and Kevin Wayne
- Cracking the Coding Interview by Gayle Laakmann McDowell
- Python Algorithms by Magnus Lie Hetland

Online Platforms:

- Leet Code – For coding practice and interview preparation
- Hacker Rank – Algorithms and data structures challenges
- Geeks for Geeks – Tutorials and examples
- Visualgo – Algorithm visualizations
- Coursera: "Algorithms Specialization" (Stanford University)
- EdX: "Data Structures & Algorithms" (UC San Diego)
- YouTube: Abdul Bari (Algorithm tutorials), CS Dojo

PRACTICAL LIST

Module 1 Practicals:

- Lab 1: Algorithm complexity analysis using Python's time it module.
- Lab 2: Implementation of linked lists and operations (insert, delete, reverse).
- Lab 3: Stack applications – expression evaluation, parenthesis matching.
- Lab 4: Queue simulation – task scheduling using circular queue.

Module 2 Practicals:

- Lab 5: Binary Search Tree implementation and traversal.
- Lab 6: Heap implementation and priority queue for task prioritization.
- Lab 7: Trie implementation for autocomplete system.
- Lab 8: Application of trees in hierarchical data (JSON/XML parsing).

Module 3 Practicals:

- Lab 9: Graph representation and BFS/DFS traversal.
- Lab 10: Shortest path algorithm (Dijkstra) implementation.
- Lab 11: Hash table implementation with collision handling.
- Lab 12: Graph analysis on social network data (using NetworkX).

Module 4 Practicals:

- Lab 13: Sorting algorithm comparison and performance analysis.
- Lab 14: Greedy algorithm – activity selection problem.
- Lab 15: Dynamic programming – knapsack problem.
- Lab 16: Mini-project – Building a recommendation system using graph algorithms.

COURSE CODE: BTDSE202
COURSE NAME: DATA BASE MANAGEMENT SYSTEM (DBMS)
Course Objective

This course introduces fundamental concepts of database systems, covering data modeling, database design, SQL programming, transaction management, and system architecture. Emphasis is placed on relational database theory, normalization, and practical implementation using modern DBMS technologies. The course prepares students to design, implement, and manage database systems for real-world applications.

Course Outcomes (COs)

Upon successful completion, students will be able to:

CO1	Design database schemas using ER modeling and normalization
CO2	Implement and query databases using SQL and PL/SQL
CO3	Apply transaction management and concurrency control
CO4	Design efficient storage structures and indexing schemes

Unit	Content	Credit	Weightage

I	**Introduction to Database Systems**
	• **1.1 Database System Concepts**
	  ○ Data, information, and knowledge
	  ○ File systems vs database systems
	  ○ Three-schema architecture (internal, conceptual, external)
	  ○ Data independence (logical and physical)
	• **1.2 Database System Architecture**
	  ○ DBMS components: Query processor, storage manager
	  ○ Database users and administrators
	  ○ Database languages: DDL, DML, DCL
	• **1.3 Database Applications**
	  ○ Traditional applications (banking, airlines, universities)
		1	25%

	○ Modern applications (e-commerce, social media, IoT) ○ Emerging trends (big data, cloud databases)		
II	**Data Models and Database Design** • **2.1 Entity-Relationship Model** ○ Entities, attributes, relationships ○ ER diagram notation (Chen notation, Crow's foot) ○ Cardinality constraints (1:1, 1:N, M:N) ○ Weak entities, composite attributes, multivalued attributes • **2.2 Enhanced ER Modelling** ○ Specialization and generalization ○ Aggregation ○ Inheritance in EER • **2.3 Relational Model** ○ Relations, tuples, attributes, domains ○ Keys: Super, candidate, primary, foreign ○ Relational algebra operations ▪ Basic: select, project, union, set difference, Cartesian product ▪ Additional: rename, intersection, natural join, division ○ Relational calculus (tuple and domain)	1	25%
III	**Structured Query Language (SQL)** • **3.1 SQL Fundamentals** ○ Data types, schema definition ○ Basic queries: SELECT, FROM, WHERE ○ Aggregate functions, GROUP BY, HAVING ○ Set operations: UNION, INTERSECT, EXCEPT • **3.2 Advanced SQL** ○ Nested subqueries (correlated and non-correlated) ○ JOIN operations: INNER, LEFT, RIGHT, FULL OUTER ○ Views: creation, updating, materialized views ○ Integrity constraints: NOT NULL, UNIQUE, CHECK, DEFAULT • **3.3 SQL Programming** ○ PL/SQL or T-SQL basics ○ Stored procedures, functions, triggers	1	25%

	○ Cursors, exception handling ○ Embedded SQL, dynamic SQL		
IV	**Database Design Theory** • **4.1 Functional Dependencies** ○ Definition and properties ○ Armstrong's axioms ○ Closure of attribute sets ○ Canonical cover • **4.2 Normalization** ○ First Normal Form (1NF) ○ Second Normal Form (2NF) ○ Third Normal Form (3NF) ○ Boyce-Codd Normal Form (BCNF) ○ Higher normal forms (4NF, 5NF) overview • **4.3 Decomposition Algorithms** ○ Lossless join decomposition ○ Dependency preserving decomposition ○ Synthesis algorithm for 3NF ○ Decomposition algorithm for BCNF	1	25%

Textbooks:

- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe
- "Database Management Systems" by Raghu Ramakrishnan and Johannes Gehrke

Reference Books:

- "SQL and Relational Theory: How to Write Accurate SQL Code" by C.J. Date
- "Transaction Processing: Concepts and Techniques" by Jim Gray and Andreas Reuter
- "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" by Pramod J. Sadalage and Martin Fowler
- "Data Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems" by Martin Kleppmann

Online Resources:

- Stanford Online: Databases course by Jennifer Widom
- Coursera: Database series by University of Michigan
- NPTEL: Database Management Systems courses
- PostgreSQL Documentation
- MySQL Reference Manual
- MongoDB University (free courses)
- db-fiddle.com (online SQL playground)

Laboratory Experiments

- **Lab 1:** Installation and configuration of DBMS (MySQL/PostgreSQL)
- **Lab 2:** Basic SQL – DDL, DML commands
- **Lab 3:** SQL queries with aggregate functions and grouping
- **Lab 4:** Advanced SQL – Subqueries, joins, views
- **Lab 5:** PL/SQL programming – Stored procedures and functions

- **Lab 6:** Triggers and cursors implementation
- **Lab 7:** ER modeling and conversion to relational schema
- **Lab 8:** Normalization exercises
- **Lab 9:** Index creation and performance analysis
- **Lab 10:** Transaction management and concurrency control
- **Lab 11:** NoSQL database basics (MongoDB)
- **Lab 12:** Mini-project – Complete database application

SUBJECT CODE: BTDSE203

SUBJECT NAME: LINEAR ALGEBRA FOR DS

Course Objectives:

- To build a strong mathematical foundation in linear algebra essential for data science and machine learning.
- To understand vector spaces, matrices, transformations, and their applications in data representation and manipulation.
- To apply linear algebra concepts to dimensionality reduction, optimization, and machine learning algorithms.
- To develop computational skills using Python libraries for linear algebra operations.

Course Outcomes: At the end of the course students shall be able to

CO1	Perform vector and matrix operations and understand their geometric interpretations.
CO2	Solve systems of linear equations and analyze matrix decompositions (LU, QR, SVD).
C03	Apply eigenvalues, eigenvectors, and diagonalization in dimensionality reduction and PCA.
C04	Use linear algebra concepts in machine learning models such as regression, clustering, and NLP.

Unit	Content	Credit	Weightage
I	**Vectors, Matrices & Systems of Linear Equations** • **Vectors:** Definition, operations (addition, scalar multiplication), dot product, cross product, norms. • **Matrices:** Types of matrices, matrix operations, transpose, trace, determinant. • **Systems of Linear Equations:** Gaussian elimination, row reduction, rank, consistency. • **Matrix Inversion:** Properties, methods, and applications. • **Vector Spaces:** Subspaces, basis, dimension, linear independence.	1	25%
II	**Matrix Decompositions & Transformations** • **LU Decomposition:** Factorization, solving linear systems, computational efficiency. • **QR Decomposition:** Gram-Schmidt process, orthogonal matrices, least squares applications. • **Singular Value Decomposition (SVD):** Computation, interpretation, low-rank approximation.	1	25%

	• **Linear Transformations:** Matrix representation, kernel, image, rank-nullity theorem. • **Eigenvalues & Eigenvectors:** Characteristic polynomial, diagonalization, spectral theorem.		
III	**Applications in Machine Learning & Optimization** • **Principal Component Analysis (PCA):** Covariance matrix, eigenvalue decomposition, dimensionality reduction. • **Linear Regression:** Normal equations, least squares, geometric interpretation. • **Clustering & Distance Metrics:** Euclidean, Mahalanobis, cosine similarity. • **Matrix Calculus:** Gradient, Jacobian, Hessian, optimization in ML. • **Recommendation Systems:** Matrix factorization, collaborative filtering.	1	25%
IV	**Advanced Topics & Computational Linear Algebra** • **Positive Definite Matrices:** Properties, Cholesky decomposition. • **Tensor Basics:** Introduction to tensors, tensor operations, applications in deep learning. • **Sparse Matrices:** Storage formats (CSR, CSC), applications in large-scale data. • **Linear Algebra in NLP:** Word embeddings (Word2Vec, GloVe) as matrix factorization. • **Numerical Stability:** Condition number, regularization (Ridge/Lasso regression).	1	25%

Textbooks:

- Linear Algebra and Its Applications by Gilbert Strang (6th Edition)
- Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong
- Introduction to Linear Algebra by Gilbert Strang

Reference books:

- Linear Algebra Done Right by Sheldon Axler
- The Matrix Cookbook by Kaare Brandt Petersen and Michael Syskind Pedersen
- Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (Chapters on Linear Algebra)
- Numerical Linear Algebra by Lloyd N. Trefethen and David Bau III

Online Platforms:

- MIT Open Course Ware: Linear Algebra (Gilbert Strang)
- Khan Academy: Linear Algebra modules
- Coursera: “Mathematics for Machine Learning” (Imperial College London)
- YouTube: 3Blue1Brown – “Essence of Linear Algebra” series
- Brilliant.org: Interactive linear algebra courses
- Google Colab / Jupyter Notebooks: For hands-on coding with NumPy/SciPy
- Wolfram Alpha / Symbolab: For matrix calculations and visualizations

SUBJECT CODE: BTDSE204

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SUBJECT NAME: DATA VISUALIZATION

Course Objectives:

- To understand the principles and psychology of effective data visualization.
- To develop skills in creating static and interactive visualizations using Python and modern tools.
- To apply visualization techniques for exploratory data analysis (EDA) and storytelling.
- To design dashboards and reports for communicating insights to technical and non-technical audiences.

Course Outcomes: At the end of the course students shall be able to

CO1	Apply design principles and human perception theory to create effective visualizations.
CO2	Develop static and interactive visualizations using Python libraries (Matplotlib, Seaborn, Plotly).
C03	Design and implement dashboards using BI tools (Tableau, Power BI) and web frameworks (Dash, Streamlit).
C04	Communicate data-driven insights visually for decision-making in real-world scenarios.

Unit	Content	Credit	Weightage
I	**Foundations of Data Visualization** • **Introduction:** Importance of visualization in data science, history of data viz. • **Visual Perception:** Gestalt principles, color theory, cognitive load, pre-attentive attributes. • **Data Types & Chart Types:** Categorical vs. numerical, temporal, hierarchical, network data. • **Design Principles:** Tufte's principles, clutter reduction, chart junk, accessibility. • **Tools Overview:** Python vs. R vs. BI tools, open-source vs. proprietary.	1	25%
II	**Static Visualization with Python** • **Matplotlib:** Figure and axes, line plots, scatter plots, bar charts, histograms, customization. • **Seaborn:** Statistical plotting, distribution plots, categorical plots, heatmaps, pair plots. • **Advanced Plots:** Box plots, violin plots, swarm plots, kernel density estimation (KDE). • **Geospatial Visualization:** Introduction to GeoPandas, choropleth maps, basemaps. • **Customization:** Color palettes, annotations, themes, saving figures for publication.	1	25%
III	**Interactive Visualization & Dashboards** • **Plotly & Plotly Express:** Interactive line, scatter, bar, 3D plots, animations. • **Bokeh:** Interactive web-based visualizations, linking plots, hover tools. • **Dashboard Development:** Introduction to Dash (Plotly) and Streamlit. • **BI Tools:** Tableau Public (connecting data, building worksheets, dashboards).	1	25%

	• **Dashboard Design Principles:** Layout, interactivity, user experience (UX).		
IV	**Advanced Topics & Storytelling** • **Time Series Visualization:** Trend lines, seasonality, forecasting plots. • **Network Visualization:** Graph visualization with NetworkX and Gephi. • **Text Data Visualization:** Word clouds, sentiment visualization, topic modeling (LDAvis). • **Visual Storytelling:** Narrative visualization, annotation, sequencing, dashboard storytelling. • **Ethics in Visualization:** Misleading charts, bias, transparency, reproducibility.	1	25%

Textbooks:

- Storytelling with Data by Cole Nussbaumer Knaflic
- The Visual Display of Quantitative Information by Edward Tufte
- Fundamentals of Data Visualization by Claus O. Wilke

Reference books:

- Interactive Data Visualization for the Web by Scott Murray
- Data Visualization: A Practical Introduction by Kieran Healy
- Python Data Science Handbook by Jake VanderPlas (Visualization Chapters)
- Tableau Your Data! by Daniel G. Murray

Online Platforms:

- Data Visualization Society – Community and resources
- Flowing Data – Tutorials and articles
- Observable HQ – Interactive notebooks for visualization
- Kaggle Datasets & Notebooks – Real-world visualization examples
- Coursera: "Data Visualization with Python" (IBM)
- Udemy: "Tableau A-Z" and "Python for Data Visualization"
- YouTube: StatQuest, Data School, Plotly channel

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Exploratory Data Analysis (EDA) with basic plots using Matplotlib.
- Lab 2: Applying Gestalt principles and color theory to redesign a poor chart.
- Lab 3: Choosing the right chart type for different datasets (categorical, numerical, temporal).
- Lab 4: Creating a visual style guide for a mock project.

Module 2 Practicals:

- Lab 5: Advanced Seaborn visualizations – pair plots, heatmaps, and categorical plots.
- Lab 6: Customizing plots with annotations, themes, and multi-plot layouts.
- Lab 7: Geospatial visualization with GeoPandas – mapping COVID-19 or election data.
- Lab 8: Visualizing distributions – histograms, KDE, box plots, violin plots.

Module 3 Practicals:

- Lab 9: Interactive visualizations with Plotly – line, scatter, bar, and 3D plots.
- Lab 10: Building an interactive dashboard with Streamlit.
- Lab 11: Creating a Tableau dashboard from a CSV dataset.
- Lab 12: Bokeh for web-based interactive visualization.

Module 4 Practicals:

- Lab 13: Time series visualization – stock market or weather data.
- Lab 14: Network visualization with NetworkX – social network analysis.
- Lab 15: Text data visualization – word clouds, sentiment analysis charts.
- Lab 16: Final project – End-to-end visualization story on a real-world dataset (e.g., Airbnb, FIFA, COVID-19).

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-III

SUBJECT CODE: BTDSE301

SUBJECT NAME: MACHINE LEARNING-I

Course Objectives:

- To introduce fundamental concepts, algorithms, and mathematical foundations of machine learning.
- To develop skills in implementing supervised and unsupervised learning models.
- To understand model evaluation, validation, and optimization techniques.
- To apply ML algorithms to real-world datasets and interpret results.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain core ML concepts, learning paradigms, and the bias-variance tradeoff.
CO2	Implement and evaluate supervised learning models for regression and classification.
C03	Apply unsupervised learning techniques for clustering and dimensionality reduction.
C04	Perform model selection, hyperparameter tuning, and validation using real datasets.

Unit	Content	Credit	Weightage
I	**Foundations of Machine Learning** • **Introduction to ML:** Types of learning (supervised, unsupervised, reinforcement), applications. • **Data Preprocessing:** Feature scaling, encoding, handling missing values, train-test split. • **Model Evaluation Metrics:** Accuracy, precision, recall, F1-score, ROC-AUC, MSE, MAE, R^2. • **Bias-Variance Trade off:** Underfitting, overfitting, regularization (L1/L2), cross-validation. • **Mathematical Foundations:** Linear algebra review, probability basics, gradient descent.	1	25%
II	**Supervised Learning – Regression** • **Simple Linear Regression:** Model formulation, ordinary least squares, assumptions. • **Multiple Linear Regression:** Feature selection, multicollinearity, interpretation. • **Polynomial Regression:** Nonlinear relationships, degree selection. • **Regularized Regression:** Ridge, Lasso, Elastic Net. • **Model Diagnostics:** Residual analysis, homoscedasticity, normality tests.	1	25%
III	**Supervised Learning – Classification** • **Logistic Regression:** Binary and multiclass classification, sigmoid function, decision boundary. • **k-Nearest Neighbors (k-NN):** Distance metrics, k selection, pros/cons. • **Decision Trees:** Entropy, Gini impurity, splitting	1	25%

	criteria, pruning. • **Support Vector Machines (SVM):** Linear SVM, kernel trick (RBF, polynomial), margin maximization. • **Naïve Bayes:** Bayes theorem, Gaussian, Multinomial, Bernoulli variants.		
IV	**Unsupervised Learning & Ensemble Basics** • **Clustering Algorithms:** k-Means, hierarchical clustering, DBSCAN, silhouette score. • **Dimensionality Reduction:** Principal Component Analysis (PCA), t-SNE basics. • **Ensemble Methods Introduction:** Bagging vs. Boosting, Random Forest (concept). • **Model Selection & Hyperparameter Tuning:** Grid search, random search, pipeline creation. • **Case Study:** End-to-end ML project on a real dataset (e.g., Titanic, Iris, Housing).	1	25%

Textbooks:

- Introduction to Statistical Learning by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- Pattern Recognition and Machine Learning by Christopher M. Bishop
- Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow by Aurélien Géron

Reference books:

- The Hundred-Page Machine Learning Book by Andriy Burkov
- Machine Learning Yearning by Andrew Ng
- Python Machine Learning by Sebastian Raschka and Vahid Mirjalili

Online Platforms:

- Coursera: “Machine Learning” by Andrew Ng (Stanford)
- Kaggle Learn: “Intro to Machine Learning” and “Intermediate Machine Learning”
- edX: “Principles of Machine Learning” by Microsoft
- YouTube: StatQuest with Josh Starmer, Krish Naik
- GitHub: ML repositories and Jupyter notebooks
- Google Colab / Jupyter Notebooks for hands-on coding

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Data preprocessing with Pandas – cleaning, scaling, encoding.
- Lab 2: Implementing train-test split and cross-validation in Scikit-learn.
- Lab 3: Visualizing bias-variance tradeoff using polynomial regression.
- Lab 4: Model evaluation – calculating metrics and plotting ROC curves.

Module 2 Practicals:

- Lab 5: Simple and multiple linear regression from scratch (NumPy) and using Scikit-learn.
- Lab 6: Polynomial regression and feature engineering.
- Lab 7: Regularized regression – Ridge and Lasso implementation.
- Lab 8: Residual analysis and model diagnostics.

Module 3 Practicals:

- Lab 9: Logistic regression for binary classification (e.g., spam detection).
- Lab 10: k-NN classifier with different distance metrics.

- Lab 11: Decision tree classifier and visualization.
- Lab 12: SVM with linear and RBF kernels.

Module 4 Practicals:

- Lab 13: k-Means clustering and elbow method.
- Lab 14: Hierarchical clustering and dendrogram visualization.
- Lab 15: PCA for dimensionality reduction and visualization.
- Lab 16: End-to-end project – build, tune, and evaluate an ML model on a real dataset.

SUBJECT CODE: BTDSE302

SUBJECT NAME: BIG DATA TECHNOLOGIES

Course Objectives:

- To understand the fundamental concepts, challenges, and ecosystem of big data.
- To learn distributed storage systems (HDFS) and data processing frameworks (MapReduce, Spark).
- To develop skills in processing and analysing large-scale data using modern big data tools.
- To design and implement big data pipelines for real-world applications.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain big data characteristics, ecosystem components, and distributed storage principles.
CO2	Design and implement data processing workflows using Hadoop MapReduce and Apache Spark.
C03	Perform data ingestion, transformation, and analysis using big data tools like Hive, Pig, and Sqoop.
C04	Develop scalable big data applications and pipelines using cloud and on-premise platforms.

Unit	Content	Credit	Weightage
I	**Introduction to Big Data & Hadoop Ecosystem** • **Big Data Fundamentals:** 5 Vs (Volume, Velocity, Variety, Veracity, Value), challenges, use cases. • **Hadoop Ecosystem:** HDFS architecture, Name Node, Data Node, replication, fault tolerance. • **YARN:** Resource management, job scheduling. • **Data Serialization Formats:** Avro, Parquet, ORC. • **Cluster Setup:** Single-node and multi-node Hadoop setup (on-premise/cloud).	1	25%
II	**Distributed Processing with MapReduce & Spark** • **MapReduce Programming Model:** Mapper, Reducer, Combiner, Partitioner, job execution flow. • **Apache Spark:** RDDs, transformations, actions, lazy evaluation, Spark SQL, Data Frames. • **Spark Optimizations:** Caching, partitioning, broadcast variables, accumulators. • **Performance Tuning:** Memory management, serialization, speculative execution. • **Real-time vs Batch Processing:** Lambda and Kappa architectures.	1	25%

III	**Big Data Querying & Processing Tools** • **Apache Hive:** Architecture, HiveQL, partitioning, bucketing, UDFs. • **Apache Pig:** Pig Latin, data flow, UDFs, comparison with Hive. • **Data Ingestion Tools:** Sqoop (RDBMS to HDFS), Flume (log ingestion), Kafka basics. • **NoSQL Databases:** HBase architecture, data modeling, CRUD operations. • **Data Warehousing on Big Data:** OLAP vs OLTP, data lakes vs data warehouses.	1	25%
IV	**Advanced Big Data Analytics & Cloud Platforms** • **Stream Processing:** Apache Spark Streaming, structured streaming, window operations. • **Machine Learning on Big Data:** MLlib (Spark), distributed model training. • **Cloud Big Data Platforms:** AWS EMR, Google Data proc, Azure HDInsight. • **Data Pipelines & Orchestration:** Apache Airflow basics. • **Big Data Security & Governance:** Authentication, authorization, data lineage, GDPR compliance.	1	25%

Textbooks:

- Hadoop: The Definitive Guide by Tom White (5th Edition)
- Learning Spark: Lightning-Fast Data Analytics by Jules S. Damji et al.
- Big Data: Principles and Best Practices of Scalable Realtime Data Systems by Nathan Marz & James Warren

Reference books:

- Spark: The Definitive Guide by Bill Chambers & Matei Zaharia
- Designing Data-Intensive Applications by Martin Kleppmann
- Big Data Analytics with Hadoop 3 by Sridhar Alla

Online Platforms:

- Cloudera/Hortonworks Sandbox – Virtual Hadoop environments
- Databricks Community Edition – Free Spark environment
- AWS Educate / Google Cloud Training – Free cloud big data credits
- Coursera: “Big Data Specialization” (UC San Diego)
- edX: “Introduction to Apache Spark” (Databricks)
- YouTube: Hadoop and Spark tutorials by edureka!, Simplilearn

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: HDFS commands – file operations, replication, permissions.
- Lab 2: Setting up single-node Hadoop cluster (Docker/local).
- Lab 3: Working with Parquet and Avro file formats.
- Lab 4: YARN job submission and monitoring.

Module 2 Practicals:

- Lab 5: Word Count using MapReduce (Java/Python).
- Lab 6: Spark RDD operations – transformations and actions.
- Lab 7: Spark SQL – querying data using Data Frames.

- Lab 8: Spark optimization – caching, partitioning, and tuning.

Module 3 Practicals:

- Lab 9: Hive – creating tables, partitions, and running HiveQL queries.
- Lab 10: Pig Latin scripts for ETL operations.
- Lab 11: Sqoop – importing/exporting data between MySQL and HDFS.
- Lab 12: HBase – CRUD operations using shell and Java API.

Module 4 Practicals:

- Lab 13: Spark Streaming – real-time word count from socket stream.
- Lab 14: MLlib – building a classification model on large dataset.
- Lab 15: AWS EMR / Google Dataproc – running Spark job on cloud.
- Lab 16: End-to-end big data pipeline project (ingest → process → store → analyze).

SUBJECT CODE: BTDSE303

SUBJECT NAME: PROBABILITY AND STOCHASTIC PROCESSES

Course Objectives:

- To build a rigorous foundation in probability theory and stochastic processes for data science applications.
- To model uncertainty, randomness, and dynamic systems using probabilistic methods.
- To apply stochastic processes in time-series analysis, queuing theory, and machine learning.
- To develop analytical and simulation-based problem-solving skills for real-world stochastic systems.

Course Outcomes: At the end of the course students shall be able to

CO1	Apply probability axioms, random variables, and distributions to model uncertainty in data.
CO2	Analyze joint distributions, correlations, and limit theorems (CLT, LLN) for inference.
C03	Model and analyze discrete and continuous-time stochastic processes (Markov, Poisson, Brownian motion).
C04	Simulate stochastic systems and apply probabilistic models to real-world data science problems.

Unit	Content	Credit	Weightage
I	**Foundations of Probability Theory** • **Probability Basics:** Sample space, events, axioms, conditional probability, Bayes' theorem. • **Random Variables:** Discrete and continuous RVs, PDF, PMF, CDF, survival function. • **Mathematical Expectation:** Mean, variance, moments, moment-generating functions. • **Common Distributions:** Bernoulli, Binomial, Poisson, Geometric, Uniform, Exponential, Normal. • **Transformations of RVs:** Functions of random variables, Jacobian method.	1	25%
II	**Multivariate Distributions & Limit Theorems** • **Joint Distributions:** Joint PDF/PMF, marginal and conditional distributions, independence. • **Covariance & Correlation:** Pearson, Spearman, multivariate normal distribution.	1	25%

	• **Conditional Expectation:** Law of iterated expectations, prediction. • **Limit Theorems:** Law of Large Numbers (Weak & Strong), Central Limit Theorem. • **Convergence Concepts:** Convergence in probability, almost sure, distribution.		
III	**Discrete-Time Stochastic Processes** • **Introduction to Stochastic Processes:** Definition, classifications, real-world examples. • **Markov Chains:** Discrete-time Markov chains, transition matrix, Chapman-Kolmogorov equations. • **Properties:** States (transient, recurrent), periodicity, ergodicity, stationary distribution. • **Hidden Markov Models (HMMs):** Forward-backward algorithm, Viterbi algorithm (overview). • **Random Walks:** Simple, biased, applications in finance and physics.	1	25%
IV	**Continuous-Time Processes & Advanced Applications** • **Poisson Process:** Homogeneous and non-homogeneous, inter-arrival times, merging/splitting. • **Birth-Death Processes:** Queuing theory basics (M/M/1, M/M/c). • **Brownian Motion (Wiener Process):** Definition, properties, geometric Brownian motion. • **Stochastic Calculus Basics:** Ito's lemma (conceptual), stochastic differential equations (SDEs). • **Applications:** Time-series modeling, financial engineering, reinforcement learning, Monte Carlo methods.	1	25%

Textbooks:

- Introduction to Probability by Dimitri P. Bertsekas and John N. Tsitsiklis
- Stochastic Processes by Sheldon M. Ross
- Probability and Random Processes for Electrical and Computer Engineers by John A. Gubner

Reference books:

- A First Course in Stochastic Processes by Samuel Karlin and Howard M. Taylor
- Pattern Recognition and Machine Learning by Christopher M. Bishop (Chapters on Probability)
- Time Series Analysis by James D. Hamilton (Stochastic Processes Sections)
- Options, Futures, and Other Derivatives by John C. Hull (For Financial Applications)

Online Platforms:

- MIT Open Course Ware: "Introduction to Probability and Statistics" / "Stochastic Processes"
- Coursera: "Probability and Statistics for Data Science" (University of Colorado Boulder)
- Khan Academy: Probability and Statistics modules
- YouTube: 3Blue1Brown – "Probability" series, MIT Lectures on Stochastic Processes
- Brilliant.org: Interactive probability courses
- Wolfram Demonstrations Project: Stochastic process simulations

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SUBJECT CODE: BTDSE304

SUBJECT NAME: BUSINESS INTELLIGENCE

Course Objectives:

- To understand the role of Business Intelligence (BI) in data-driven decision-making and organizational strategy.
- To develop skills in designing and implementing BI solutions using modern tools and platforms.
- To apply data modelling, ETL, and visualization techniques for creating actionable business insights.
- To evaluate BI architectures, performance metrics, and governance frameworks for real-world applications.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain BI concepts, architectures, and their role in organizational decision-making.
CO2	Design and implement ETL pipelines, data warehouses, and OLAP models for BI.
C03	Develop interactive dashboards and reports using BI tools (Tableau, Power BI).
C04	Apply data analytics, KPIs, and performance management frameworks to solve business problems.

Unit	Content	Credit	Weightage
I	**Foundations of Business Intelligence** • **Introduction to BI:** Definition, evolution, and importance in modern organizations. • **BI Architecture:** Components (data sources, ETL, data warehouse, OLAP, reporting). • **Data Warehousing:** Concepts, schemas (star, snowflake, galaxy), dimensional modeling. • **BI vs Data Science vs Analytics:** Differences, overlaps, and complementary roles. • **BI Lifecycle:** From requirements gathering to deployment and maintenance.	1	25%
II	**Data Integration & ETL Processes** • **ETL Fundamentals:** Extract, Transform, Load processes, tools (Informatica, Talend, SSIS). • **Data Quality & Cleansing:** Handling missing values, duplicates, inconsistencies. • **Data Modeling for BI:** Fact tables, dimension tables, slowly changing dimensions (SCD). • **OLAP Operations:** Roll-up, drill-down, slice, dice, pivot. • **Real-time BI & Data Streaming:** Introduction to change data capture (CDC) and streaming ETL.	1	25%
III	**BI Visualization & Dashboarding** • **Dashboard Design Principles:** User-centric design, storytelling, interactivity. • **BI Tools Overview:** Tableau, Power BI, Qlik Sense, Looker. • **Visual Analytics:** Creating charts, maps, filters,	1	25%

	parameters, calculated fields. • **Advanced Visualizations:** Geospatial analytics, trend lines, forecasting visuals. • **Mobile BI & Self-Service BI:** Enabling business users for ad-hoc analysis.		
IV	**Performance Management & Advanced BI Topics** • **Key Performance Indicators (KPIs):** Defining, measuring, and visualizing KPIs. • **Scorecards & Balanced Scorecard:** Strategic performance management. • **Data Governance & Security:** Role-based access control (RBAC), data lineage, compliance. • **BI in the Cloud:** Cloud BI platforms (AWS Quick Sight, Google Looker, Microsoft Power BI Service). • **Emerging Trends:** Augmented analytics, natural language querying, AI-driven BI.	1	25%

Textbooks:

- Business Intelligence: Data Mining and Optimization for Decision Making by Carlo Vercellis
- The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling by Ralph Kimball and Margy Ross
- Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software by Daniel G. Murray

Reference books:

- Business Intelligence Guidebook: From Data Integration to Analytics by Rick Sherman
- Data Science for Business by Foster Provost and Tom Fawcett
- Microsoft Power BI Dashboards Step by Step by Erin O'Brien
- The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios by Steve Wexler, Jeffrey Shaffer, and Andy Cotgreave

Online Platforms:

- Tableau Public / Tableau eLearning – Free training resources
- Microsoft Learn – Power BI learning paths
- Coursera: "Business Intelligence and Visual Analytics" (University of Colorado Boulder)
- edX: "Data Science and Business Intelligence" (IBM)
- Udemy: "Tableau A-Z" and "Microsoft Power BI – The Practical Guide"
- YouTube: Andy Kriebel (Tableau), Guy in a Cube (Power BI)
- Kaggle & Mode Analytics – For practice datasets and SQL/BI challenges

SUBJECT CODE: BTDSE305

SUBJECT NAME: SQL AND NOSQL DATABASES

Course Objectives:

- To understand relational and non-relational database models and their use cases in data science.
- To develop proficiency in SQL for data querying, manipulation, and analysis.
- To learn NoSQL database concepts and practical usage for handling unstructured and big data.
- To design, implement, and manage hybrid database solutions for modern data science applications.

Course Outcomes: At the end of the course students shall be able to

CO1	Design and query relational databases using SQL for data

	extraction and transformation.
CO2	Normalize databases, write complex queries, and optimize SQL for performance.
C03	Implement and interact with NoSQL databases (document, key-value, column-family, graph).
C04	Integrate SQL and NoSQL databases in data pipelines for real-world data science workflows.

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Setting up MySQL/PostgreSQL and basic SQL commands.
- Lab 2: Creating and modifying tables, inserting/updating data.
- Lab 3: Writing SELECT queries with WHERE, ORDER BY, and LIMIT.
- Lab 4: Implementing INNER and LEFT JOINs on multiple tables.

Module 2 Practicals:

- Lab 5: Advanced queries using subqueries and set operations.
- Lab 6: Window functions for ranking and time-series analysis.
- Lab 7: Database normalization exercise (1NF to 3NF).
- Lab 8: Performance tuning – creating indexes and analyzing query plans.
- Lab 9: MongoDB CRUD operations and aggregation pipelines.
- Lab 10: Redis data structures and caching implementation.
- Lab 11: Connecting MongoDB and Redis with Python (PyMongo, redis-py).
- Lab 12: Schema design for a document-based application (e.g., blog, e-commerce).

Module 4 Practicals:

- Lab 13: Apache Cassandra – creating keyspaces, tables, and performing CQL queries.
- Lab 14: Neo4j – creating nodes, relationships, and Cypher queries for graph traversal.
- Lab 15: Building a recommendation system using Neo4j.
- Lab 16: Final project – Hybrid database solution (SQL + NoSQL) for a real-world dataset.

TOOLS & DATABASES:

- **SQL Databases:** MySQL, PostgreSQL, SQLite
- **NoSQL Databases:** MongoDB, Redis, Apache Cassandra, Neo4j
- **Cloud Services:** AWS RDS, DynamoDB, Google Cloud SQL, Firestore
- **Languages & Libraries:** SQL, Python (SQLAlchemy, PyMongo, Redis-Py, Cassandra Driver)

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-IV

SUBJECT CODE: BTDSE401

SUBJECT NAME: MACHINE LEARNING-II

Course Objectives:

- To explore advanced machine learning algorithms and ensemble methods for improved predictive performance.
- To understand and implement dimensionality reduction, clustering, and neural network fundamentals.
- To develop skills in model optimization, hyperparameter tuning, and evaluation for complex datasets.
- To apply advanced ML techniques to real-world problems in computer vision, NLP, and recommendation systems.

Course Outcomes: At the end of the course students shall be able to

CO1	Implement and evaluate ensemble learning methods for classification and regression.
CO2	Apply dimensionality reduction and advanced clustering techniques to unstructured data.
C03	Design and train basic neural networks and understand deep learning fundamentals.
C04	Optimize ML models using hyperparameter tuning, cross-validation, and performance metrics.

Unit	Content	Credit	Weightage
I	**Ensemble Learning & Model Optimization** • **Ensemble Methods:** Bagging, Boosting, Stacking, Voting. • **Random Forest:** Feature importance, OOB error, hyperparameters. • **Gradient Boosting Machines (GBM):** XGBoost, LightGBM, CatBoost. • **Hyperparameter Tuning:** Grid Search, Random Search, Bayesian Optimization. • **Model Interpretability:** SHAP, LIME, feature importance, partial dependence plots.	1	25%
II	**Dimensionality Reduction & Advanced Clustering** • **Dimensionality Reduction:** PCA, t-SNE, UMAP, LDA. • **Clustering Algorithms:** DBSCAN, OPTICS, Gaussian Mixture Models (GMM), Spectral Clustering. • **Anomaly Detection:** Isolation Forest, One-Class SVM, Local Outlier Factor (LOF). • **Evaluation of Clustering:** Silhouette score, Davies–Bouldin index, Calinski–Harabasz index. • **Applications:** Customer segmentation, image compression, feature engineering.	1	25%
III	**Introduction to Neural Networks & Deep Learning** • **Neural Network Basics:** Perceptron, activation functions (ReLU, Sigmoid, Tanh), loss functions.	1	25%

	• **Backpropagation & Optimization:** Gradient descent variants (SGD, Adam, RMSprop). • **Feedforward Neural Networks:** Architecture, weight initialization, regularization (Dropout, Batch Norm). • **Convolutional Neural Networks (CNN) Basics:** Convolution, pooling, CNN architecture (LeNet, AlexNet overview). • **Hands-on with Keras/TensorFlow:** Building and training simple neural networks.		
IV	**Advanced Topics & Real-World Applications** • **Recommendation Systems:** Collaborative filtering, matrix factorization, content-based filtering. • **Natural Language Processing (NLP) Basics:** TF-IDF, word embeddings (Word2Vec, GloVe), sentiment analysis. • **Time Series Forecasting:** ARIMA, LSTM basics, sequence prediction. • **Model Deployment Basics:** Introduction to MLflow, Docker for ML, Flask API. • **Ethical AI & Model Fairness:** Bias detection, fairness metrics, model auditing.	1	25%

Textbooks:

- The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Pattern Recognition and Machine Learning by Christopher M. Bishop
- Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville

Reference books:

- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron
- Applied Predictive Modeling by Max Kuhn and Kjell Johnson
- Machine Learning Yearning by Andrew Ng
- Interpretable Machine Learning by Christoph Molnar

Online Platforms:

- Kaggle: Competitions and datasets for advanced ML practice
- Coursera: “Advanced Machine Learning Specialization” (HSE)
- Fast.ai: Practical deep learning courses
- Google Colab / Jupyter Notebooks: For hands-on implementation
- YouTube: StatQuest, 3Blue1Brown (Neural Networks), sentdex
- GitHub: Open-source ML projects and notebooks

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Random Forest implementation and feature importance analysis.
- Lab 2: Gradient Boosting with XGBoost and LightGBM.
- Lab 3: Hyperparameter tuning using GridSearchCV and RandomizedSearchCV.
- Lab 4: Model interpretability with SHAP and LIME.

Module 2 Practicals:

- Lab 5: PCA and t-SNE for dimensionality reduction and visualization.
- Lab 6: DBSCAN and Gaussian Mixture Models for clustering.
- Lab 7: Anomaly detection using Isolation Forest and One-Class SVM.

- Lab 8: Customer segmentation project using clustering techniques.

Module 3 Practicals:

- Lab 9: Building a feedforward neural network with Keras/TensorFlow.
- Lab 10: Implementing CNN for image classification (CIFAR-10/MNIST).
- Lab 11: Regularization techniques: Dropout and Batch Normalization.
- Lab 12: Neural network hyperparameter tuning with Keras Tuner.

Module 4 Practicals:

- Lab 13: Building a recommendation system using collaborative filtering.
- Lab 14: Sentiment analysis with word embeddings (Word2Vec/GloVe).
- Lab 15: Time series forecasting with ARIMA and LSTM.
- Lab 16: End-to-end ML project: Model training, evaluation, and deployment using Flask.

SUBJECT CODE: BTDSE402

SUBJECT NAME: DEEP LEARNING FUNDAMENTALS

Course Objectives:

- To introduce the foundational concepts, architectures, and mathematical principles of deep learning.
- To develop proficiency in building, training, and evaluating deep neural networks.
- To understand and implement key deep learning architectures such as CNNs, RNNs, and basic Transformers.
- To apply deep learning to real-world problems in vision, text, and sequential data.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain the mathematical and algorithmic foundations of neural networks and backpropagation.
CO2	Design, implement, and train convolutional neural networks for image-based tasks.
C03	Build and apply recurrent neural networks and LSTMs for sequential data and time-series analysis.
C04	Utilize deep learning frameworks (TensorFlow/PyTorch) and apply transfer learning in real-world projects.

Unit	Content	Credit	Weightage
I	**Neural Networks & Training Fundamentals** • **Introduction to Deep Learning:** History, evolution, and applications. • **Neural Network Basics:** Perceptron, multi-layer perceptron (MLP), activation functions (ReLU, Sigmoid, Tanh). • **Training Neural Networks:** Loss functions, gradient descent, backpropagation, chain rule. • **Optimization Algorithms:** SGD, Momentum, RMSProp, Adam, learning rate scheduling. • **Regularization Techniques:** Dropout, Batch Normalization, L1/L2 regularization, early stopping.	1	25%
II	**Convolutional Neural Networks (CNNs)** • **CNN Fundamentals:** Convolution operation, filters, padding, stride, pooling.	1	25%

	• **CNN Architectures:** LeNet, AlexNet, VGG, ResNet (overview), Inception Net. • **Transfer Learning:** Fine-tuning pre-trained models (VGG16, ResNet, MobileNet). • **Object Detection Basics:** Introduction to R-CNN, YOLO, SSD. • **Visualization Techniques:** Feature maps, Grad-CAM, t-SNE of embeddings.		
III	**Recurrent Neural Networks (RNNs) & Sequence Modeling** •**RNN Basics:** Architecture, unfolding through time, vanishing/exploding gradients. •**Long Short-Term Memory (LSTM) & GRU:** Internal gates, memory cells, applications. •**Sequence-to-Sequence Models:** Encoder-decoder architecture, attention mechanism (basic). •**Time-Series Forecasting:** Using LSTMs for stock prediction, weather forecasting. •**Text Processing Basics:** Word embeddings (Word2Vec, GloVe), text classification with RNNs.	1	25%
IV	**Advanced Topics & Practical Deployment** • **Introduction to Transformers:** Self-attention mechanism, encoder-decoder in Transformers. • **Autoencoders & Generative Models:** Variational Autoencoders (VAEs), GANs (conceptual). • **Deep Learning Frameworks:** TensorFlow vs. PyTorch, building end-to-end pipelines. • **Model Deployment:** Converting models to TensorFlow Lite/ONNX, serving via Flask/FastAPI. • **Ethics in Deep Learning:** Bias in models, fairness, interpretability, and adversarial attacks.	1	25%

Textbooks:

- Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- Neural Networks and Deep Learning by Michael Nielsen
- Deep Learning with Python by François Chollet

Reference books:

- Pattern Recognition and Machine Learning by Christopher M. Bishop
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron
- Deep Learning for Computer Vision by Rajalingappaa Shanmugamani
- Natural Language Processing with Transformers by Lewis Tunstall, Leandro von Werra, and Thomas Wolf

Online Platforms:

- Practical deep learning courses
- Coursera: “Deep Learning Specialization” by Andrew Ng
- YouTube: 3Blue1Brown (Neural Networks), sentdex (TensorFlow/PyTorch tutorials), Two Minute Papers
- Kaggle: Deep learning competitions and notebooks
- TensorFlow Playground / CNN Explainer – Visualization tools
- Google Colab / Paperspace Gradient – Free GPU for practice

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Building a perceptron and MLP from scratch using NumPy.
- Lab 2: Implementing backpropagation and training a neural network on MNIST.
- Lab 3: Experimenting with optimizers (SGD, Adam) and learning rate schedules.
- Lab 4: Applying regularization techniques (Dropout, BatchNorm) to improve generalization.

Module 2 Practicals:

- Lab 5: Building a CNN from scratch for CIFAR-10 classification.
- Lab 6: Transfer learning with pre-trained CNN models (e.g., VGG16 on custom dataset).
- Lab 7: Implementing object detection using a pre-trained YOLO model.
- Lab 8: Visualizing CNN layers and feature maps using Grad-CAM.

Module 3 Practicals:

- Lab 9: Building an LSTM for time-series prediction (stock prices/weather).
- Lab 10: Text classification using word embeddings and RNNs/LSTMs.
- Lab 11: Implementing a simple Seq2Seq model for machine translation (short sentences).
- Lab 12: Sentiment analysis with attention-based RNNs.

Module 4 Practicals:

- Lab 13: Building a Transformer encoder for text classification (using Hugging Face).
- Lab 14: Implementing a simple autoencoder for image reconstruction.
- Lab 15: Deploying a trained model as a REST API using Flask.
- Lab 16: Final project – End-to-end deep learning application (e.g., facial expression recognition, news categorizer).

COURSE CODE: BTDSE403
COURSE NAME: CLOUD COMPUTING FOR DS
Course Objective

- Understand cloud computing concepts, models, and service architectures.
- Analyse virtualization technologies and cloud deployment models.
- Design and deploy applications using cloud platforms (AWS/Azure/GCP).
- Implement storage, compute, and networking solutions in cloud environments.
- Evaluate cloud security, compliance, and cost management strategies.
- Compare emerging cloud technologies and migration approaches.

Course Outcomes:

CO1	Explain cloud computing characteristics, service models, and deployment models.
CO2	Analyze virtualization techniques and containerization technologies
CO3	Deploy and manage applications using major cloud platforms
CO4	Implement cloud storage, networking, and compute services

Unit	Content	Credit	Weightage

I	Cloud Foundations & Virtualization, Introduction to Cloud Computing, Cloud Service Models, Cloud Deployment Models, Virtualization Fundamentals	1	25%
II	Cloud Architecture & Core Services, Cloud Reference Architecture, Compute Services, Storage Services, Networking in Cloud, Database Services	1	25%
III	Cloud Management & Security • Cloud Management & Monitoring, Cloud Security, Compliance & Governance, Cost Management	1	25%
IV	Advanced Cloud Services & Trends • Cloud-Native Development, Big Data & Analytics in Cloud, Multi-cloud & Hybrid Cloud, Emerging Trends	1	25%

Textbooks:

- "Cloud Computing: Concepts, Technology & Architecture" *Thomas Erl, Ricardo Puttini, Zaigham Mahmood Comprehensive coverage of cloud patterns and best practices*
- "Cloud Computing: A Practical Approach" *Anthony T. Velte, Toby J. Velte, Robert Elsenpeter Excellent for hands-on learning and implementation*
- "Architecting the Cloud: Design Decisions for Cloud Computing Service Models" *Michael J. Kavis Great for architectural decision-making*

Reference Books:

- "The Cloud Adoption Playbook" *Moe Abdula, Ingo Averdunk, et al. Practical guide for enterprise cloud adoption*
- "Site Reliability Engineering: How Google Runs Production Systems" *Betsy Beyer, et al. SRE principles in cloud context*
- "Cloud Native Patterns: Designing Change-tolerant Software" *Cornelia Davis Modern cloud-native application design*

Online Resources:

- AWS/Azure/GCP Documentation & Free Tier
- Cloud Academy, A Cloud Guru (Video courses)
- NPTEL: "Cloud Computing" by Prof. Soumya Kanti Ghosh
- Cloud Computing Specialization (Coursera - UIUC)
- CNCF (Cloud Native Computing Foundation) Resources

COURSE CODE: BTDSE404
COURSE NAME: NATURAL LANGUAGE PROCESSING
Course Objective

- Understand linguistic foundations and computational challenges in NLP.
- Implement text preprocessing, tokenization, and language modeling techniques.
- Design and evaluate classical and neural NLP models for various tasks.
- Apply sequence models, attention mechanisms, and transformers for NLP tasks.
- Analyse NLP applications in real-world scenarios and evaluate model performance.
- Compare different NLP architectures and their suitability for specific applications.

Course Outcomes:

CO1	Explain linguistic concepts and computational challenges in NLP.
CO2	Implement text preprocessing, tokenization, and language modeling techniques
CO3	Design and evaluate classical NLP models for text classification and sequence labeling.
CO4	Implement neural NLP models including RNNs, LSTMs, and attention mechanisms.

Unit	Content	Credit	Weightage
I	NLP Foundations & Text Processing • Introduction to NLP, Linguistic Fundamentals, Text Preprocessing, Statistical Foundations	1	25%
II	Classical NLP & Sequence Models • Text Representation, Text Classification, Sequence Labeling, Syntax and Parsing	1	25%
III	Neural NLP & Attention Models • Neural Network Basics for NLP, Recurrent Neural Networks, Sequence-to-Sequence Models, Attention Mechanisms, Word Embeddings Advanced	1	25%
IV	Transformers & Advanced Applications • Transformer Architecture, Pre-trained Language Models, Advanced NLP Tasks, Current Trends	1	25%

Textbooks:

- "Speech and Language Processing" *Daniel Jurafsky & James H. Martin* (3rd Edition Draft) *The "Bible" of NLP – Comprehensive and authoritative*
- "Natural Language Processing with Python" *Steven Bird, Ewan Klein, & Edward Loper Practical introduction using NLTK*

Reference Books:

- "Foundations of Statistical Natural Language Processing" *Christopher D. Manning & Hinrich Schütze Excellent for statistical foundations*
- "Neural Network Methods for Natural Language Processing" *Yoav Goldberg Great for neural approaches to NLP*
- "Deep Learning for Natural Language Processing" *Palash Goyal, Sumit Pandey, & Karan Jain Practical deep learning approaches*

Online Resources:

- Stanford CS224N: Natural Language Processing with Deep Learning
- NPTEL: "Natural Language Processing" by Prof. Pushpak Bhattacharyya

- Hugging Face Course (Free NLP course with transformers)
- spaCy documentation and tutorials
- Papers with Code (NLP leaderboards and implementations)
- AllenNLP library and tutorials

SUBJECT CODE: BTDSE405

SUBJECT NAME:DATA WRANGLING WITH PANDAS

Course Objectives:

- To master data manipulation, cleaning, and transformation techniques using the Pandas library in Python.
- To develop skills in handling real-world messy datasets including missing values, duplicates, and inconsistencies.
- To perform advanced data aggregation, merging, and reshaping for analysis-ready datasets.
- To automate data preprocessing pipelines and integrate Pandas with other data science tools.

Course Outcomes: At the end of the course students shall be able to

CO1	Import, inspect, and clean structured and semi-structured data using Pandas.
CO2	Handle missing data, outliers, duplicates, and data type conversions effectively.
C03	Perform advanced data operations including merging, grouping, pivoting, and time-series manipulation.
C04	Build reusable data preprocessing pipelines and integrate Pandas with SQL, Excel, and APIs.

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Creating Data Frames, basic Series operations, and data inspection.
- Lab 2: Importing data from CSV, Excel, and JSON; exporting cleaned data.
- Lab 3: Basic data selection and filtering using loc, iloc, and boolean indexing.
- Lab 4: Initial data cleaning: handling missing values and renaming columns.

Module 2 Practicals:

- Lab 5: Advanced missing data imputation and data type conversion.
- Lab 6: String cleaning and regex-based text extraction.
- Lab 7: Identifying and handling duplicates and outliers.
- Lab 8: Data normalization and standardization techniques.

Module 3 Practicals:

- Lab 9: Group By operations with multi-level aggregation.
- Lab 10: Creating pivot tables and cross-tabulations.
- Lab 11: Merging multiple datasets (inner, outer, left, right joins).
- Lab 12: Time-series manipulation: resampling, shifting, and rolling statistics.

Module 4 Practicals:

- Lab 13: Optimizing Pandas operations for large datasets (chunking, efficient dtypes).
- Lab 14: Integrating Pandas with SQL databases (reading/writing data).
- Lab 15: Building a reusable data wrangling pipeline with method chaining.
- Lab 16: Capstone Project: End-to-end data wrangling of a real-world messy dataset (e.g., COVID-19 data, e-commerce transactions, sensor data).

TOOLS & LIBRARIES:

- Primary Library: Pandas
- Supporting Libraries: NumPy, Matplotlib, Seaborn, openpyxl, sqlalchemy, requests
  Datasets: Kaggle datasets, UCI Repository, real-world CSV/Excel files

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-V

SUBJECT CODE: BTDSE501

SUBJECT NAME: ADVANCED DEEP LEARNING

Course Objectives:

- To explore state-of-the-art deep learning architectures including Transformers, GANs, and advanced CNNs.
- To develop skills in designing, training, and optimizing complex deep learning models.
- To implement deep learning solutions for complex domains such as NLP, computer vision, and generative AI.
- To apply advanced training techniques, interpretability methods, and deployment strategies for production-ready models.

Course Outcomes: At the end of the course students shall be able to

CO1	Design and implement advanced deep learning models including Transformers, GANs, and Autoencoders.
CO2	Apply transfer learning, meta-learning, and self-supervised learning techniques to complex tasks.
C03	Develop and deploy deep learning models for NLP, vision, and multimodal applications.
C04	Analyze model interpretability, fairness, and robustness in real-world deep learning systems.

Unit	Content	Credit	Weightage
I	**Advanced Neural Architectures & Optimization** • **Advanced CNN Architectures:** Inception, ResNet variants, EfficientNet, DenseNet. • **Attention Mechanisms:** Self-attention, multi-head attention, scaled dot-product attention. • **Optimization Advances:** AdamW, Lookahead, Ranger, learning rate schedulers (Cosine, OneCycle). • **Neural Architecture Search (NAS):** Concepts, AutoML frameworks (e.g., AutoKeras). • **Meta-Learning:** MAML, few-shot learning, and model-agnostic meta-learning basics.	1	25%
II	**Transformers & Advanced NLP** • **Transformer Architecture:** Encoder-decoder, positional encoding, layer normalization. • **BERT & Variants:** RoBERTa, DistilBERT, ALBERT, and their fine-tuning strategies. • **Large Language Models (LLMs):** GPT architecture, T5, and prompt engineering basics. • **Multimodal Models:** Vision-Language models (CLIP, ViLBERT), audio-visual learning. • **Efficient NLP:** Knowledge distillation, pruning, quantization for NLP models.	1	25%
III	**Generative Models & Unsupervised Learning** • **Generative Adversarial Networks (GANs):** DCGAN, StyleGAN, CycleGAN, Conditional GANs.	1	25%

	• **Variational Autoencoders (VAEs):** Reparameterization trick, beta-VAE, applications. • **Diffusion Models:** DDPM, Stable Diffusion basics, image generation. • **Self-Supervised Learning:** Contrastive learning (SimCLR, MoCo), pretext tasks. • **Anomaly Detection with Deep Learning:** Autoencoders for anomaly detection.		
IV	**Model Deployment, Ethics & Advanced Topics** • **Model Deployment:** ONNX, TensorRT, TensorFlow Serving, FastAPI with deep learning models. • **Model Interpretability:** LIME, SHAP for deep models, attention visualization. • **Robustness & Adversarial Attacks:** FGSM, PGD, adversarial training, certified robustness. • **Fairness in Deep Learning:** Bias detection, fairness metrics, debiasing techniques. • **Edge AI:** Model compression, quantization-aware training, TinyML basics.	1	25%

Textbooks:

- Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- Natural Language Processing with Transformers by Lewis Tunstall, Leandro von Werra, and Thomas Wolf
- Generative Deep Learning by David Foster

Reference books:

- Dive into Deep Learning by Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola
- Speech and Language Processing by Daniel Jurafsky and James H. Martin
- Interpretable Machine Learning by Christoph Molnar
- Machine Learning Engineering by Andriy Burkov

Online Platforms:

- Hugging Face Course – Free NLP and Transformers course
- Fast.ai – Practical deep learning and deployment tutorials
- Coursera: "Generative Adversarial Networks (GANs) Specialization" (deeplearning.ai)
- Papers With Code – Latest research papers and implementations
- YouTube: Yannic Kilcher, Two Minute Papers, Aladdin Persson
- Google Colab Pro / Kaggle Notebooks – GPU access for advanced models

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Implementing ResNet and EfficientNet from scratch using PyTorch/TensorFlow.
- Lab 2: Building a self-attention mechanism and multi-head attention layer.
- Lab 3: Advanced optimization: Implementing Ranger optimizer and Cosine annealing.
- Lab 4: Few-shot learning with MAML on Omniglot dataset.

Module 2 Practicals:

- Lab 5: Fine-tuning BERT for sentiment analysis or named entity recognition (NER).
- Lab 6: Building a text generation model with GPT-2 using Hugging Face.
- Lab 7: Implementing a multimodal model (CLIP) for image-text retrieval.

- Lab 8: Knowledge distillation: Compressing a large BERT model to a smaller one.

Module 3 Practicals:

- Lab 9: Training a DCGAN on CIFAR-10 for image generation.
- Lab 10: Building a VAE for image reconstruction and latent space exploration.
- Lab 11: Implementing a simple diffusion model for image synthesis.
- Lab 12: Self-supervised learning with SimCLR on image datasets.

Module 4 Practicals:

- Lab 13: Deploying a deep learning model as a REST API using FastAPI and Docker.
- Lab 14: Visualizing attention maps in Transformers for model interpretability.
- Lab 15: Adversarial attack simulation (FGSM) and defense on an image classifier.
- Lab 16: Capstone Project: End-to-end advanced DL pipeline (e.g., AI-powered caption generator, synthetic data generation with GANs).

SUBJECT CODE: BTDSE502

SUBJECT NAME: TIME SERIES ANALYSIS

- Course Objectives: To understand the fundamental concepts, characteristics, and components of time series data.
- To develop skills in modelling, forecasting, and analysing time series using statistical and machine learning methods.
- To apply time series decomposition, ARIMA models, and advanced techniques like LSTM and Prophet.
- To evaluate and compare forecasting models for real-world applications in finance, healthcare, IoT, and economics.

Course Outcomes: At the end of the course students shall be able to

CO1	Analyze and decompose time series data to identify trends, seasonality, and residuals.
CO2	Implement and evaluate statistical time series models such as ARIMA, SARIMA, and Exponential Smoothing.
C03	Apply machine learning and deep learning methods (LSTM, Prophet) for time series forecasting.
C04	Perform time series anomaly detection, multivariate forecasting, and model deployment.

Unit	Content	Credit	Weightage
I	**Foundations of Time Series Analysis** • **Introduction:** Definition, importance, and applications (finance, IoT, healthcare, climate). • **Time Series Components:** Trend, seasonality, cyclical patterns, noise. • **Descriptive Statistics:** Mean, variance, autocorrelation (ACF), partial autocorrelation (PACF). • **Stationarity:** Concept, tests (ADF, KPSS), differencing, transformations (log, Box-Cox). • **Decomposition Methods:** Additive vs. multiplicative, moving averages, STL decomposition.	1	25%

II	**Statistical Time Series Models** • **Exponential Smoothing:** Simple, double, triple (Holt-Winters). • **ARIMA Models:** AR, MA, ARMA, ARIMA model identification, parameter estimation. • **Seasonal ARIMA (SARIMA):** Modeling seasonality, parameter selection. • **Model Diagnostics:** Residual analysis, Ljung-Box test, forecast evaluation metrics (MAE, RMSE, MAPE). • **Vector Autoregression (VAR):** Multivariate time series modeling.	1	25%
III	**Machine Learning for Time Series** • **Feature Engineering for Time Series:** Lag features, rolling statistics, Fourier terms. • **Tree-based Models:** Random Forest, XGBoost for forecasting. • **Support Vector Regression (SVR):** Kernel methods for time series. • **Facebook Prophet:** Model components, holidays, changepoints, uncertainty intervals. • **Model Comparison:** Benchmarks, cross-validation for time series (time series split).	1	25%
IV	**Deep Learning & Advanced Topics** • **Recurrent Neural Networks (RNNs) & LSTMs:** Sequence modeling, time steps, forecasting with Keras/TensorFlow. • **Convolutional Neural Networks (CNNs) for Time Series:** 1D convolutions, WaveNet architecture. • **Attention & Transformer Models:** Time series transformers, Informer. • **Anomaly Detection:** Statistical methods (IQR, Z-score), isolation forest, autoencoders. • **Real-World Projects:** Stock prediction, energy demand forecasting, sensor data analysis.	1	25%

Textbooks:

- Forecasting: Principles and Practice by Rob J Hyndman and George Athanasopoulos (3rd Edition)
- Time Series Analysis and Its Applications by Robert H. Shumway and David S. Stoffer
- Introduction to Time Series and Forecasting by Peter J. Brockwell and Richard A. Davis

Reference books:

- Practical Time Series Analysis by Aileen Nielsen
- Deep Learning for Time Series Forecasting by Jason Brownlee
- Applied Time Series Analysis for the Social Sciences by Richard McCleary, David McDowall, and Bradley J. Bartos

Online Platforms:

- Kaggle: Time series competitions (e.g., M5 Forecasting, Web Traffic)
- Coursera: “Practical Time Series Analysis” (The State University of New York)
- YouTube: RitvikMath (Time Series tutorials), StatQuest with Josh Starmer

- GitHub: Time series notebooks and libraries (e.g., sktime, Darts)
- Google Colab / Kaggle Notebooks: For hands-on coding

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Loading and visualizing time series data (stock prices, temperature).
- Lab 2: Time series decomposition using moving averages and STL.
- Lab 3: Testing for stationarity (ADF, KPSS) and applying differencing.
- Lab 4: Autocorrelation (ACF) and partial autocorrelation (PACF) analysis.

Module 2 Practicals:

- Lab 5: Implementing exponential smoothing (Holt-Winters) for seasonal data.
- Lab 6: Building ARIMA models using auto_arima and manual parameter tuning.
- Lab 7: Seasonal ARIMA (SARIMA) for monthly sales data.
- Lab 8: Residual diagnostics and forecast evaluation.

Module 3 Practicals:

- Lab 9: Feature engineering for time series: lag features and rolling stats.
- Lab 10: Forecasting with XGBoost and Random Forest.
- Lab 11: Using Facebook Prophet for forecasting with holidays.
- Lab 12: Time series cross-validation and model comparison.

Module 4 Practicals:

- Lab 13: Building an LSTM model for multi-step forecasting.
- Lab 14: Anomaly detection in time series using autoencoders.
- Lab 15: Time series classification with 1D CNNs.
- Lab 16: Capstone Project: End-to-end time series pipeline (e.g., energy load forecasting, cryptocurrency price prediction).

SUBJECT CODE: BTDSE503

SUBJECT NAME: DATA ENGINEERING

Course Objectives:

- To understand the principles, architecture, and lifecycle of data engineering systems.
- To design and implement scalable data pipelines for ingestion, processing, storage, and orchestration.
- To develop skills in batch and stream processing using modern data engineering tools and cloud platforms.
- To apply data modelling, warehousing, and governance best practices in real-world data infrastructure.

Course Outcomes: At the end of the course students shall be able to

CO1	Design data architecture and pipelines for batch and stream processing.
CO2	Implement ETL/ELT workflows using tools like Apache Spark, Airflow, and cloud services.
C03	Build and optimize data warehouses and data lakes for analytical workloads.
C04	Deploy, monitor, and orchestrate data pipelines in cloud and on-premise environments.

Unit	Content	Credit	Weightage
I	**Data Engineering Foundations & Architecture** • **Introduction to Data Engineering:** Role, lifecycle,	1	25%

	and data engineering vs data science. • **Data Architecture Patterns:** Lambda, Kappa, data mesh, data lakehouse. • **Data Storage Solutions:** SQL vs NoSQL, data lakes, data warehouses, object storage (S3, GCS). • **Data Modeling:** Dimensional modeling (star/snowflake schemas), data vault, normalization/denormalization. • **Data Governance & Quality:** Metadata management, data lineage, data quality frameworks.		
II	**Batch Data Processing & ETL** • **ETL/ELT Fundamentals:** Batch vs streaming, ETL tools overview (Talend, Informatica, Air byte). • **Apache Spark for Batch Processing:** RDDs, Data Frames, Spark SQL, optimizations. • **Workflow Orchestration:** Apache Airflow (DAGs, operators, sensors), Prefect, Dagster. • **Data Pipeline Design:** Incremental loading, idempotency, fault tolerance, monitoring. • **Cloud Batch Services:** AWS Glue, Google Dataflow, Azure Data Factory.	1	25%
III	**Stream Processing & Real-time Data Pipelines** • **Stream Processing Concepts:** Event time vs processing time, windowing, watermarks. • **Apache Kafka:** Architecture, producers, consumers, connectors, schema registry. • **Stream Processing Frameworks:** Apache Flink, Spark Streaming, Kafka Streams. • **Real-time ETL:** Change Data Capture (CDC), streaming joins, stateful processing. • **Cloud Streaming Services:** AWS Kinesis, Google Pub/Sub, Azure Event Hubs.	1	25%
IV	**Modern Data Stack & Advanced Topics** • **Data Warehousing:** Snowflake, BigQuery, Redshift, Delta Lake, Iceberg. • **DataOps & MLOps Integration:** CI/CD for data pipelines, versioning (DVC), model deployment. • **Scalability & Performance:** Partitioning, indexing, caching, query optimization. • **Data Engineering in the Cloud:** Multi-cloud strategies, serverless data pipelines. • **Case Studies & Emerging Trends:** Data mesh, real-time analytics, ethical data engineering.	1	25%

Textbooks:

- Fundamentals of Data Engineering by Joe Reis and Matt Housley
- Designing Data-Intensive Applications by Martin Kleppmann
- Data Pipelines Pocket Reference by James Densmore

Reference books:

- The Data Warehouse Toolkit by Ralph Kimball and Margy Ross
- Kafka: The Definitive Guide by Neha Narkhede, Gwen Shapira, and Todd Palino

- Spark: The Definitive Guide by Bill Chambers and Matei Zaharia
- Data Engineering with AWS by Gareth Eagar

Online Platforms:

- Data Camp: Data Engineering with Python and Spark
- Coursera: "IBM Data Engineering Professional Certificate"
- edX: "Data Engineering on Google Cloud Platform"
- YouTube: Seattle Data Guy, Data Engineering Zoom camp, Confluent
- GitHub: Open-source data engineering projects and pipelines

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Designing a data architecture diagram for an e-commerce analytics platform.
- Lab 2: Implementing a star schema in PostgreSQL/MySQL.
- Lab 3: Building a data quality pipeline with Great Expectations or Deequ.
- Lab 4: Setting up a data lake on AWS S3 / Google Cloud Storage.

Module 2 Practicals:

- Lab 5: Creating an ETL pipeline with Apache Spark (batch processing of CSV/JSON data).
- Lab 6: Orchestrating pipelines with Apache Airflow (scheduled data ingestion).
- Lab 7: Incremental data loading using change data capture (CDC) with Debezium.
- Lab 8: Cloud ETL with AWS Glue / Google Dataflow.

Module 3 Practicals:

- Lab 9: Setting up Apache Kafka and producing/consuming messages.
- Lab 10: Real-time stream processing with Spark Streaming or Flink.
- Lab 11: Building a real-time dashboard with streamed data (Kafka + Streamlit).
- Lab 12: Implementing a CDC pipeline with Kafka Connect and PostgreSQL.

Module 4 Practicals:

- Lab 13: Building a cloud data warehouse with Snowflake/BigQuery.
- Lab 14: Creating a serverless data pipeline with AWS Lambda / Cloud Functions.
- Lab 15: Data pipeline monitoring with Prometheus and Grafana.
- Lab 16: Capstone Project – End-to-end data engineering solution (ingest → process → store → visualize).

SUBJECT CODE: BTDSE504

SUBJECT NAME: STREAM PROCESSING

Course Objectives:

- To understand the fundamentals of real-time data processing, event-driven architectures, and streaming data systems.
- To design and implement stream processing pipelines using frameworks like Apache Kafka, Spark Streaming, and Flink.
- To apply stream processing techniques for real-time analytics, monitoring, and decision-making.
- To develop scalable, fault-tolerant streaming applications for IoT, finance, social media, and other domains.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain the concepts of event time, processing time, windowing, and state management in stream processing.
CO2	Implement real-time data ingestion and processing pipelines using Apache Kafka and Kafka Streams.

C03	Build and optimize streaming applications with Apache Spark Streaming and Apache Flink.
C04	Design and deploy end-to-end stream processing systems for real-world use cases such as fraud detection, IoT monitoring, and real-time recommendations.

PRACTICAL LIST (12 Sessions)

Module 1 Practicals:

- Lab 1: Setting up a streaming data source (simulated sensor data/logs) and visualizing with Streamlit.
- Lab 2: Implementing tumbling and sliding windows on a simulated data stream.
- Lab 3: Watermark and late data handling in a custom Python stream processor.
- Lab 4: Stateful stream processing with a simple counter and session windows.

Module 2 Practicals:

- Lab 5: Setting up Kafka cluster (single-node/local), creating topics, and producing/consuming messages.
- Lab 6: Building a Kafka Connect pipeline to stream data from MySQL to Kafka.
- Lab 7: Kafka Streams application for real-time word count and filtering.
- Lab 8: Implementing exactly-once processing with Kafka Transactions.

Module 3 Practicals:

- Lab 9: Spark Structured Streaming – real-time aggregation and windowing on stock data.
- Lab 10: Apache Flink – event time processing and stateful operations on clickstream data.
- Lab 11: Joining streams in Flink/Spark (e.g., user clicks with metadata).
- Lab 12: Complex Event Processing (CEP) with Flink for pattern detection.

Module 4 Practicals:

- Lab 13: Building an end-to-end streaming ETL pipeline: Kafka → Spark/Flink → Database.
- Lab 14: Real-time dashboard for streaming data using Grafana or Streamlit.
- Lab 15: Cloud stream processing with AWS Kinesis or Google Pub/Sub.
- Lab 16: Capstone Project – Real-time fraud detection system or IoT alerting system.

TOOLS & TECHNOLOGIES:

- Streaming Frameworks: Apache Kafka, Kafka Streams, Apache Flink, Spark Structured Streaming
- Message Brokers: Apache Pulsar (optional), RabbitMQ (basics)
- Cloud Services: AWS Kinesis, Google Pub/Sub, Azure Event Hubs
- Databases for Streaming Sinks: PostgreSQL, Cassandra, Redis, Elasticsearch
- Languages: Java, Scala, Python (PySpark, Faust)
- Monitoring: Prometheus, Grafana, ELK Stack

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. **RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956**

SEMESTER-VI

SUBJECT CODE: BTDSE601

SUBJECT NAME: REINFORCEMENT LEARNING

Course Objectives:

- To introduce the mathematical foundations, algorithms, and applications of reinforcement learning (RL).
- To develop skills in implementing value-based, policy-based, and model-free RL methods.
- To apply RL to real-world problems in robotics, gaming, finance, and autonomous systems.
- To understand advanced topics such as deep reinforcement learning, multi-agent RL, and ethical considerations.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain the RL framework, Markov Decision Processes (MDPs), and the exploration-exploitation trade-off.
CO2	Implement and evaluate classic RL algorithms such as Q-Learning, SARSA, and Policy Gradient methods.
C03	Design and train deep reinforcement learning models using DQN, A2C, PPO, and DDPG.
C04	Apply RL techniques to solve practical problems in simulation environments and real-world systems.

Unit	Content	Credit	Weightage
I	**Foundations of Reinforcement Learning** • **Introduction to RL:** Agent-environment interaction, rewards, states, actions. • **Markov Decision Processes (MDPs):** State transition probabilities, policy, value functions. • **Bellman Equations:** State-value and action-value functions, optimality, Bellman optimality equation. • **Dynamic Programming:** Policy evaluation, policy improvement, policy and value iteration. • **Exploration vs Exploitation:** ε-greedy, optimistic initialization, UCB, Thompson sampling.	1	25%
II	**Model-Free RL & Temporal Difference Learning** • **Monte Carlo Methods:** Prediction and control, first-visit and every-visit MC. • **Temporal Difference Learning:** TD(0), TD(λ), eligibility traces. • **Q-Learning:** Off-policy TD control, convergence, and variants. • **SARSA:** On-policy TD control, expected SARSA. • **Function Approximation:** Linear approximation, tile coding, Fourier basis.	1	25%
III	**Deep Reinforcement Learning** • **Deep Q-Networks (DQN):** Experience replay, target networks, stability issues. • **Policy Gradient Methods:** REINFORCE, baseline methods, actor-critic architectures.	1	25%

	• **Advanced Algorithms:** A2C, A3C, PPO, TRPO, DDPG, TD3. • **Deep RL Libraries:** OpenAI Gym, Stable Baselines3, RLlib. • **Simulation Environments:** Atari, MuJoCo, CARLA, PyBullet.		
IV	**Advanced Topics & Real-World Applications** • **Multi-Agent RL:** Independent learners, centralized training decentralized execution (CTDE). • **Inverse Reinforcement Learning:** Learning reward functions from expert demonstrations. • **Hierarchical RL:** Options framework, MAXQ, feudal networks. • **Ethical & Safe RL:** Reward hacking, interpretability, fairness, and robustness. • **Applications:** Game playing (AlphaGo, Dota 2), robotics, finance, healthcare, recommendation systems.	1	25%

Textbooks:

- Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto (2nd Edition)
- Deep Reinforcement Learning Hands-On by Maxim Lapan (2nd Edition)
- Algorithms for Reinforcement Learning by Csaba Szepesvári

Reference books:

- Reinforcement Learning: State-of-the-Art by Marco Wiering and Martijn van Otterlo
- Foundations of Deep Reinforcement Learning by Laura Graesser and Wah Loon Keng
- Reinforcement Learning with TensorFlow by Sayon Dutta
- Probabilistic Machine Learning: An Introduction by Kevin P. Murphy (Chapters on RL)

Online Platforms:

- OpenAI Spinning Up – Educational content and code for deep RL
- Coursera: "Reinforcement Learning Specialization" by University of Alberta & Alberta Machine Intelligence Institute
- YouTube: DeepMind's RL lectures, Sergey Levine (UC Berkeley CS285), David Silver's lectures
- GitHub: RL implementations, OpenAI Baselines, Stable-Baselines3
- Papers With Code – Latest RL research and implementations

SUBJECT CODE: BTDSE602

SUBJECT NAME: MLOPS AND MODEL DEVELOPMENT

Course Objectives:

- To understand the principles, practices, and tools for operationalizing machine learning models.
- To design and implement end-to-end ML pipelines from development to deployment and monitoring.
- To develop skills in containerization, orchestration, CI/CD, and model serving in production environments.
- To apply MLOps best practices for scalability, reproducibility, and collaboration in real-world ML projects.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain the MLOps lifecycle, stages, and tools for managing

	ML workflows.
CO2	Design and build reproducible ML pipelines using version control, containerization, and orchestration.
C03	Deploy and serve ML models using REST APIs, microservices, and cloud platforms.
C04	Monitor, maintain, and scale ML systems in production with CI/CD, logging, and alerting.

Unit	Content	Credit	Weightage
I	**Foundations of MLOps & ML Workflow Management** • **Introduction to MLOps:** Definition, stages (experimentation, deployment, monitoring), benefits. • **ML Workflow:** Data collection, preprocessing, training, evaluation, deployment. • **Version Control for ML:** Git, DVC (Data Version Control), model versioning (ML flow, DVC). • **Experiment Tracking:** ML flow, Weights & Biases, • **Collaborative ML:** Team workflows, reproducibility, and project structure.	1	25%
II	**Building ML Pipelines & Orchestration** • **Pipeline Design:** Modular pipelines, dependency management, automation. • **Orchestration Tools:** Apache Airflow, Kubeflow Pipelines, Prefect, Dagster. • **Containerization:** Docker for ML, Docker files, multi-stage builds, best practices. • **Continuous Integration for ML (CI):** Automated testing (unit, integration), GitHub Actions, Jenkins. • **Feature Stores:** Introduction to Feast, Tecton, and feature management.	1	25%
III	**Model Deployment & Serving** • **Deployment Patterns:** Batch vs. real-time, online vs. offline inference. • **Model Serving Frameworks:** FastAPI, Flask, TensorFlow Serving, TorchServe. • **Container Orchestration:** Kubernetes basics, deploying ML models on K8s. • **Cloud ML Deployment:** AWS SageMaker, Google AI Platform, Azure ML. • **Serverless ML:** AWS Lambda, Google Cloud Functions, Azure Functions for ML.	1	25%
IV	**Monitoring, Maintenance & Advanced Topics** • **Model Monitoring:** Performance drift, data drift, concept drift, detection methods (Evidently, Why Logs). • **Logging & Alerting:** Structured logging, monitoring dashboards (Grafana, Prometheus). • **Model Retraining & A/B Testing:** Canary	1	25%

	deployments, shadow testing, automated retraining. • **Security & Compliance:** Model explainability (SHAP, LIME), bias detection, GDPR compliance. • **Case Studies & Emerging Trends:** MLOps in industry, edge ML, federated learning ops.		

Textbooks:

- Practical MLOps by Noah Gift and Alfredo Deza
- Introducing MLOps by Mark Treveil and the Dataiku Team
- Building Machine Learning Pipelines by Hannes Hapke and Catherine Nelson

Reference books:

- Machine Learning Engineering by Andriy Burkov
- Designing Machine Learning Systems by Chip Huyen
- Kubeflow for Machine Learning by Trevor Grant, Holden Karau, et al.
- The ML Engineering Book by Andriy Burkov

Online Platforms:

- Coursera: "Machine Learning Engineering for Production (MLOps)" by deeplearning.ai
- Udacity: "Machine Learning DevOps Engineer Nanodegree"
- YouTube: MLOps community talks, DevOps for Data Science channels
- GitHub: Open-source MLOps projects (ML flow, Kubeflow, Airflow examples)
- Medium: Towards Data Science, MLOps blogs

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Setting up Git and DVC for data and model versioning.
- Lab 2: Experiment tracking with ML flow: logging metrics, parameters, and artifacts.
- Lab 3: Creating a reproducible ML project structure (Cookie cutter, templates).
- Lab 4: Collaborative ML workflow with GitHub and CI integration.

Module 2 Practicals:

- Lab 5: Building a modular ML pipeline with Scikit-learn and joblib.
- Lab 6: Orchestrating ML pipelines with Apache Airflow.
- Lab 7: Containerizing an ML model with Docker and pushing to Docker Hub.
- Lab 8: CI/CD for ML with GitHub Actions (automated testing and model training).

Module 3 Practicals:

- Lab 9: Deploying a model as a REST API with FastAPI and Uvicorn.
- Lab 10: Serving a TensorFlow model with TensorFlow Serving.
- Lab 11: Deploying a model on Kubernetes (Minikube/local K8s cluster).
- Lab 12: Deploying a model on a cloud platform (AWS SageMaker / Google AI Platform).

Module 4 Practicals:

- Lab 13: Monitoring model performance and drift with Evidently AI.
- Lab 14: Implementing logging and alerting for a live ML service.
- Lab 15: Automated model retraining pipeline with Airflow and MLflow.
- Lab 16: Capstone Project: End-to-end MLOps pipeline (data → train → deploy → monitor).

COURSE CODE: BTDSE603

COURSE NAME: BIG DATA ANALYTICS

Course Objective

- Understand big data characteristics, challenges, and ecosystem components.
- Design and implement distributed storage solutions using Hadoop and HDFS.

- Develop data processing applications using MapReduce and Spark frameworks.
- Analyze big data using machine learning and stream processing techniques.
- Design data pipelines and implement NoSQL databases for big data.
- Evaluate big data architectures and deployment strategies for real-world applications.

Course Outcomes:

CO1	Explain big data characteristics, challenges, and ecosystem components.
CO2	Design and implement distributed storage solutions using Hadoop HDFS
CO3	Develop data processing applications using MapReduce and Apache Spark
CO4	Implement stream processing and machine learning on big data platforms

Unit	Content	Credit	Weightage
I	Big Data Foundations & Hadoop Ecosystem • Introduction to Big Data, Big Data Ecosystem, Hadoop Architecture, Big Data Storage Formats, Data Ingestion Tools	1	25%
II	Data Processing with MapReduce & Spark • MapReduce Programming Model, Apache Spark Fundamentals, Spark Programming	1	25%
III	NoSQL Databases & Stream Processing • NoSQL Databases, Apache HBase, Apache Cassandra, Stream Processing	1	25%
IV	Advanced Analytics & Big Data Applications • Machine Learning on Big Data, Graph Processing, Big Data Analytics, Big Data Security, Big Data DevOps, Real-world Case Studies, Emerging Trends	1	25%

Textbooks:

- "Hadoop: The Definitive Guide" *Tom White* (4th Edition) *Comprehensive Hadoop reference, industry standard*
- "Learning Spark: Lightning-Fast Data Analytics" *Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee* (2nd Edition) *Excellent Spark guide from Databricks creators*

Reference Books:

- "Big Data: Principles and Best Practices of Scalable Realtime Data Systems" *Nathan Marz & James Warren Lambda architecture and real-time systems*
- "Designing Data-Intensive Applications" *Martin Kleppmann*
  Modern perspective on distributed data systems

- "Spark: The Definitive Guide" *Bill Chambers & Matei Zaharia*
  Comprehensive Spark coverage

Online Resources:

- Hadoop & Spark Official Documentation
- Cloudera/Hortonworks Tutorials
- Databricks Academy (Free Spark courses)
- Google Cloud Big Data and Machine Learning Fundamentals (Coursera)
- AWS Big Data Specialty Certification Resources
- Apache Project Documentation (Kafka, Flink, Airflow)

Practical List:

Session 1: Hadoop cluster setup (single-node with Docker)
Session 2: HDFS operations and file management
Session 3: MapReduce Word Count and custom implementations
Session 4: Data ingestion with Sqoop and Flume
Session 5: Spark RDD operations and transformations
Session 6: Spark Data Frame operations and SQL queries
Session 7: Spark performance tuning and optimization
Session 8: Spark MLlib for machine learning
Session 9: MongoDB CRUD operations and aggregation
Session 10: HBase operations and data modeling
Session 11: Kafka producers and consumers

SUBJECT CODE: BTDSE601

SUBJECT NAME: REINFORCEMENT LEARNING

Course Objectives:

- To introduce the mathematical foundations, algorithms, and applications of reinforcement learning (RL).
- To develop skills in implementing value-based, policy-based, and model-free RL methods.
- To apply RL to real-world problems in robotics, gaming, finance, and autonomous systems.
- To understand advanced topics such as deep reinforcement learning, multi-agent RL, and ethical considerations.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain the RL framework, Markov Decision Processes (MDPs), and the exploration-exploitation trade-off.
CO2	Implement and evaluate classic RL algorithms such as Q-Learning, SARSA, and Policy Gradient methods.
C03	Design and train deep reinforcement learning models using DQN, A2C, PPO, and DDPG.
C04	Apply RL techniques to solve practical problems in simulation environments and real-world systems.

Unit	Content	Credit	Weightage
I	**Foundations of Reinforcement Learning** • **Introduction to RL:** Agent-environment interaction, rewards, states, actions.	1	25%

	• **Markov Decision Processes (MDPs):** State transition probabilities, policy, value functions. • **Bellman Equations:** State-value and action-value functions, optimality, Bellman optimality equation. • **Dynamic Programming:** Policy evaluation, policy improvement, policy and value iteration. • **Exploration vs Exploitation:** ε-greedy, optimistic initialization, UCB, Thompson sampling.		
II	**Model-Free RL & Temporal Difference Learning** • **Monte Carlo Methods:** Prediction and control, first-visit and every-visit MC. • **Temporal Difference Learning:** TD(0), TD(λ), eligibility traces. • **Q-Learning:** Off-policy TD control, convergence, and variants. • **SARSA:** On-policy TD control, expected SARSA. • **Function Approximation:** Linear approximation, tile coding, Fourier basis.	1	25%
III	**Deep Reinforcement Learning** • **Deep Q-Networks (DQN):** Experience replay, target networks, stability issues. • **Policy Gradient Methods:** REINFORCE, baseline methods, actor-critic architectures. • **Advanced Algorithms:** A2C, A3C, PPO, TRPO, DDPG, TD3. • **Deep RL Libraries:** OpenAI Gym, Stable Baselines3, RLlib. • **Simulation Environments:** Atari, MuJoCo, CARLA, PyBullet.	1	25%
IV	**Advanced Topics & Real-World Applications** • **Multi-Agent RL:** Independent learners, centralized training decentralized execution (CTDE). • **Inverse Reinforcement Learning:** Learning reward functions from expert demonstrations. • **Hierarchical RL:** Options framework, MAXQ, feudal networks. • **Ethical & Safe RL:** Reward hacking, interpretability, fairness, and robustness. • **Applications:** Game playing (AlphaGo, Dota 2), robotics, finance, healthcare, recommendation systems.	1	25%

Textbooks:

- "Reinforcement Learning: An Introduction" by Richard S. Sutton & Andrew G. Barto (2nd Edition)
- "Deep Reinforcement Learning Hands-On" by Maxim Lapan (2nd Edition)

Reference books:

- "Artificial Intelligence: A Modern Approach" by Russell & Norvig
- "Reinforcement Learning: State-of-the-Art" edited by Wiering & van Otterlo

Online Platforms:

- OpenAI Spinning Up – Educational content and code for deep RL
- Coursera: “Reinforcement Learning Specialization” by University of Alberta & Alberta

Machine Intelligence Institute
- YouTube: DeepMind's RL lectures, Sergey Levine (UC Berkeley CS285), David Silver's lectures
- GitHub: RL implementations, OpenAI Baselines, Stable-Baselines3
- Papers With Code – Latest RL research and implementation

SUBJECT CODE: BTDSE604

SUBJECT NAME: ETHICAL AI AND PRESPONSIBLE DS

Course Objectives:
- To understand the ethical, social, and legal implications of AI and data-driven technologies.
- To identify and mitigate biases, ensure fairness, and promote transparency in machine learning models.
- To apply ethical frameworks, guidelines, and responsible practices in real-world data science projects.
- To develop skills in AI governance, accountability, and human-cantered design for trustworthy systems.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain key ethical issues, biases, and societal impacts of AI and data science.
CO2	Apply fairness metrics, bias detection, and mitigation techniques to ML models.
C03	Implement transparency, interpretability, and accountability methods in AI systems.
C04	Design and evaluate AI systems following ethical guidelines, regulations, and human-centered principles.

Unit	Content	Credit	Weightage
I	**Foundations of AI Ethics & Responsible Data Science** • **Introduction to AI Ethics:** Historical context, importance, and societal impact. • **Ethical Theories & Frameworks:** Utilitarianism, deontology, virtue ethics, rights-based approaches. • **AI Principles & Guidelines:** IEEE, EU Ethics Guidelines, OECD AI Principles, UNESCO recommendations. • **Case Studies:** Algorithmic bias in hiring, criminal justice, credit scoring, and facial recognition. • **Stakeholder Analysis:** Identifying affected parties, power dynamics, and ethical trade-offs.	1	25%
II	**Fairness, Bias, & Discrimination in AI** • **Types of Bias:** Historical, representation, measurement, aggregation, evaluation bias. • **Fairness Definitions:** Group fairness (demographic parity, equalized odds), individual fairness. • **Bias Detection Tools:** AI Fairness 360 (AIF360), Fair learn, What-If Tool, IBM Watson Open Scale. • **Bias Mitigation Strategies:** Pre-processing (reweighting, sampling), in-processing (constraints, adversarial debiasing), post-processing (calibration).	1	25%

	• **Legal & Regulatory Aspects:** GDPR, CCPA, Algorithmic Accountability Act, EU AI Act.		
III	**Transparency, Interpretability, & Accountability** • **Interpretability vs. Explainability:** Importance, trade-offs, and use cases. • **Model Interpretability Methods:** LIME, SHAP, partial dependence plots, counterfactual explanations. • **Transparency by Design:** Documentation (Model Cards, Datasheets for Datasets), auditing, logging. • **Accountability Mechanisms:** Human-in-the-loop, redress processes, impact assessments. • **AI Governance:** Roles (ethics boards, AI auditors), policies, and organizational structures.	1	25%
IV	**Human-Centered AI & Emerging Ethical Challenges** • **Human-Centered Design:** Participatory design, co-creation, user consent, and agency. • **Privacy & Data Ethics:** Differential privacy, federated learning, data minimization, informed consent. • **AI Safety & Robustness:** Adversarial attacks, robustness testing, fail-safe mechanisms. • **Global & Cross-Cultural Ethics:** Ethical pluralism, cultural biases, and inclusive AI. • **Future Trends:** AI and climate change, autonomous weapons, AI in healthcare, and long-term societal impacts.	1	25%

Textbooks:

- Weapons of Math Destruction by Cathy O'Neil
- The Ethical Algorithm by Michael Kearns and Aaron Roth
- Atlas of AI by Kate Crawford

Reference books:

- Fairness and Machine Learning by Solon Barocas, Moritz Hardt, and Arvind Narayanan
- The Alignment Problem by Brian Christian
- Ethics of Artificial Intelligence and Robotics by Vincent C. Müller
- Data Feminism by Catherine D'Ignazio and Lauren F. Klein

Online Platforms:

- Coursera: “AI Ethics: Global Perspectives” (University of California, Davis)
- edX: “Data Science Ethics” (University of Michigan)
- MIT Moral Machine Platform – For exploring ethical dilemmas in autonomous systems
- Partnership on AI – Resources, case studies, and best practices
- AI Now Institute – Research reports and policy recommendations
- YouTube: TED Talks on AI ethics, lectures by Timnit Gebru, Joy Buolamwini

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-VII

SUBJECT CODE: BTDSE701

SUBJECT NAME: ADVANCED MACHINE LEARNING

Course Objectives:

- To explore cutting-edge machine learning algorithms, theories, and methodologies beyond foundational models.
- To develop expertise in probabilistic modelling, Bayesian inference, kernel methods, and advanced optimization.
- To implement and evaluate state-of-the-art ML techniques for complex, high-dimensional, and non-linear data.
- To apply advanced ML to real-world domains such as healthcare, finance, autonomous systems, and scientific discovery.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain and apply probabilistic graphical models, Bayesian methods, and Gaussian processes.
CO2	Design and implement kernel methods, support vector machines, and non-linear dimensionality reduction.
C03	Develop and optimize advanced ensemble methods, meta-learning, and automated machine learning (Auto ML).
C04	Apply advanced ML techniques to complex real-world problems and interpret model outcomes with theoretical rigor.

Unit	Content	Credit	Weightage
I	**Probabilistic Models & Bayesian Methods** • **Bayesian Inference:** Bayes theorem, conjugate priors, posterior approximation. • **Probabilistic Graphical Models:** Bayesian networks, Markov random fields, inference (exact & approximate). • **Hidden Markov Models (HMMs):** Forward-backward algorithm, Viterbi algorithm, applications. • **Gaussian Processes (GPs):** Kernel functions, GP regression and classification, hyperparameter tuning. • **Variational Inference & MCMC:** Gibbs sampling, Metropolis-Hastings, variational autoencoders (VAEs) as inference.	1	25%
II	**Kernel Methods & Advanced SVM** • **Kernel Trick:** Mercer's theorem, positive-definite kernels, kernel design (RBF, polynomial, string kernels). • **Support Vector Machines (SVMs):** Hard/soft margin, dual formulation, kernel SVM. • **Support Vector Regression (SVR):** ε-insensitive loss, kernel SVR. • **Multiple Kernel Learning (MKL):** Learning kernel combinations. • **Kernel PCA:** Non-linear dimensionality reduction.	1	25%

III	**Ensemble Methods & Meta-Learning** • **Advanced Ensemble Techniques:** Stacking, blending, super learners, Bayesian model averaging. • **Gradient Boosting Machines (GBMs):** XGBoost, Light GBM, Cat Boost—advanced tuning and interpretation. • **Random Forests & Extra Trees:** Out-of-bag estimates, feature importance, and uncertainty quantification. • **Meta-Learning:** Learning to learn, model-agnostic meta-learning (MAML), few-shot learning. • **Automated Machine Learning (AutoML):** Hyperparameter optimization (Hyperopt, Optuna), neural architecture search (NAS), Auto-sklearn.	1	25%
IV	**Advanced Topics & Real-World Systems** • **Multi-Task Learning:** Shared representations, applications in healthcare and NLP. • **Transfer Learning:** Domain adaptation, fine-tuning strategies, pre-trained models (BERT, ResNet). • **Causal Inference in ML:** Potential outcomes, counterfactuals, instrumental variables, Double ML. • **Scalable & Distributed ML:** Model parallelism, data parallelism, federated learning basics. • **ML in Production Systems:** Advanced MLOps, model monitoring, A/B testing, and ethical deployment.	1	25%

Textbooks:

- Pattern Recognition and Machine Learning by Christopher M. Bishop
- The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, and Jerome Friedman
- Bayesian Reasoning and Machine Learning by David Barber

Reference books:

- Gaussian Processes for Machine Learning by Carl Edward Rasmussen and Christopher K. I. Williams
- Understanding Machine Learning: From Theory to Algorithms by Shai Shalev-Shwartz and Shai Ben-David
- Automated Machine Learning: Methods, Systems, Challenges by Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren
- Causal Inference for Statistics, Social, and Biomedical Sciences by Guido W. Imbens and Donald B. Rubin

Online Platforms:

- Kaggle: Advanced ML competitions and kernels
- Coursera: “Probabilistic Graphical Models” (Stanford University)
- edX: “Bayesian Statistics: From Concept to Data Analysis” (UC Santa Cruz)
- YouTube: Mathematical Monk (PGMs), Alexander Amini (MIT Deep Learning)
- Papers With Code: Latest research in advanced ML with implementations

- GitHub: Open-source libraries (GPyTorch, XGBoost, Optuna, SHAP)

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Bayesian linear regression with PyMC3/Stan.
- Lab 2: Implementing Hidden Markov Models for sequence prediction.
- Lab 3: Gaussian Process regression with custom kernels.
- Lab 4: Variational inference for a simple probabilistic model.

Module 2 Practicals:

- Lab 5: Kernel SVM for non-linearly separable datasets.
- Lab 6: Kernel PCA for dimensionality reduction and visualization.
- Lab 7: Multiple Kernel Learning for heterogeneous data.
- Lab 8: Custom kernel design for structured data (e.g., graphs, strings).

Module 3 Practicals:

- Lab 9: Advanced ensemble: stacking with meta-learners.
- Lab 10: Hyperparameter optimization with Bayesian methods (Optuna, Hyperopt).
- Lab 11: Meta-learning with MAML on few-shot classification tasks.
- Lab 12: AutoML pipeline with Auto-sklearn or H2O.

Module 4 Practicals:

- Lab 13: Multi-task learning for related classification/regression problems.
- Lab 14: Causal inference using DoubleML and DoWhy.
- Lab 15: Distributed ML with Dask or Ray.
- Lab 16: Capstone project: End-to-end advanced ML pipeline on a complex dataset (e.g., healthcare diagnostics, financial forecasting).

SUBJECT CODE: BTDSE702

SUBJECT NAME: CLOUD NATIVE DATA SYSTEMS

Course Objectives:

- To understand cloud-native principles, architectures, and services for building scalable, resilient data systems.
- To design and implement cloud-native data pipelines, storage solutions, and analytics platforms.
- To apply serverless computing, containerization, and orchestration for data-intensive applications.
- To develop skills in deploying, monitoring, and optimizing data systems on major cloud platforms (AWS, GCP, Azure).

Course Outcomes: At the end of the course students shall be able to

CO1	Explain cloud-native architectures, microservices, and data mesh principles.
CO2	Design and implement cloud-native data pipelines using serverless and containerized services.
C03	Deploy and manage scalable data storage, processing, and analytics in the cloud.
C04	Monitor, secure, and optimize cloud-native data systems for performance and cost-efficiency.

Unit	Content	Credit	Weightage
I	**Foundations of Cloud-Native Data Systems**	1	25%

	• **Introduction to Cloud-Native Computing:** Principles, benefits, 12-factor apps. • **Cloud-Native Architectures:** Microservices, event-driven architecture, service mesh (Istio, Linkerd). • **Data Mesh:** Principles, domains, self-service data platforms. • **Cloud Data Services Overview:** AWS, GCP, Azure ecosystems for data. • **Infrastructure as Code (IaC):** Terraform, CloudFormation, Pulumi for data infrastructure.		
II	**Cloud-Native Data Storage & Processing** • **Cloud Storage Solutions:** Object storage (S3, GCS, Blob), block storage, file systems (EFS, FSx). • **Cloud-Native Databases:** Managed SQL (RDS, Cloud SQL, Azure SQL), NoSQL (DynamoDB, Cosmos DB, Firestore). • **Data Warehousing in Cloud:** Snowflake, BigQuery, Redshift, Azure Synapse. • **Streaming & Messaging Services:** Kafka on cloud (MSK, Confluent Cloud), Pub/Sub, Kinesis, Event Hubs. • **Batch & Stream Processing:** AWS Glue, Dataflow, Databricks on cloud, Spark on Kubernetes.	1	25%
III	**Serverless & Containerized Data Pipelines** • **Serverless Computing:** AWS Lambda, Google Cloud Functions, Azure Functions for data tasks. • **Containerization:** Docker for data applications, multi-container apps. • **Orchestration with Kubernetes:** Kubernetes basics, deploying data apps on K8s (Helm, Operators). • **Workflow Orchestration:** Managed Airflow (Cloud Composer, MWAA), Step Functions, Data Fusion. • **CI/CD for Data Systems:** GitOps, Jenkins/ArgoCD for data pipelines.	1	25%
IV	**Monitoring, Security, & Optimization** • **Observability & Monitoring:** CloudWatch, Stack driver, Azure Monitor, Prometheus/Grafana. • **Security & Compliance:** IAM, encryption (KMS, Cloud KMS), data governance, compliance (GDPR, HIPAA). • **Performance Tuning:** Auto-scaling, partitioning, caching (ElastiCache, Memorystore), query optimization. • **Cost Management:** Cloud cost optimization, spot instances, reserved capacity, FinOps. • **Emerging Trends:** Edge data systems, AI/ML in cloud-native data, Green cloud computing.	1	25%

Textbooks:

- Cloud Native Data Center Networking by Dinesh G. Dutt
- Data Mesh: Delivering Data-Driven Value at Scale by Zhamak Dehghani

- Cloud Native Transformation by Pini Reznik, Jamie Dobson, and Michelle Gienow

Reference books:

- Designing Data-Intensive Applications by Martin Kleppmann (Cloud-Native chapters)
- Kubernetes: Up and running by Brendan Burns, Joe Beda, and Kelsey Hightower
- AWS Certified Data Analytics Study Guide by Asif Abbasi
- Google Cloud Platform for Data Science by Anirban Das

Online Platforms:

- AWS Training & Certification – Free courses on data services
- Google Cloud Skills Boost – Data engineering and cloud-native tracks
- Microsoft Learn – Azure data engineer path
- Coursera: "Cloud Computing Specialization" (University of Illinois)
- edX: "Introduction to Cloud Infrastructure Technologies" (Linux Foundation)
- YouTube: Cloud Guru, TechWorld with Nana, IBM Cloud

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Setting up cloud accounts (AWS/GCP/Azure free tier) and CLI configuration.
- Lab 2: Infrastructure as Code: Deploying S3 bucket and RDS instance using Terraform.
- Lab 3: Building a simple microservice with REST API and Docker.
- Lab 4: Implementing a basic data mesh domain using cloud services.

Module 2 Practicals:

- Lab 5: Creating a cloud data warehouse (BigQuery/Redshift/Synapse) and loading data.
- Lab 6: Streaming data pipeline with Kafka (MSK/Confluent Cloud) and Kinesis.
- Lab 7: Serverless ETL with AWS Glue or Google Dataflow.
- Lab 8: Multi-model database setup with Cosmos DB or DynamoDB.

Module 3 Practicals:

- Lab 9: Serverless function (Lambda/Cloud Functions) for real-time data processing.
- Lab 10: Containerizing a data app with Docker and deploying on Kubernetes (EKS/GKE/AKS).
- Lab 11: Orchestrating a data pipeline with Managed Airflow (Cloud Composer/MWAA).
- Lab 12: CI/CD pipeline for data application using GitHub Actions and ArgoCD**.**

Module 4 Practicals:

- Lab 13: Monitoring and alerting with CloudWatch/Prometheus and Grafana.
- Lab 14: Implementing security: IAM roles, encryption, and VPC configurations.
- Lab 15: Performance tuning: Auto-scaling, caching, and partitioning strategies.
- Lab 16: Capstone Project: End-to-end cloud-native data system (ingest → store → process → visualize).

SUBJECT CODE: BTDSE703

SUBJECT NAME: IOT AND SENSOR DATA SCIENCE

Course Objectives:

- To understand IoT architecture, sensor technologies, and data acquisition methods for smart systems.
- To develop skills in processing, analysing, and visualizing time-series sensor data.
- To apply machine learning and deep learning techniques for anomaly detection, predictive maintenance, and real-time analytics.
- To design and implement end-to-end IoT data pipelines from edge to cloud.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain IoT architecture, sensor types, communication protocols, and edge computing concepts.
CO2	Preprocess, clean, and visualize time-series sensor data for analysis.
C03	Apply ML/DL models for sensor data tasks like classification, forecasting, and anomaly detection.
C04	Design and deploy scalable IoT data pipelines integrating edge devices, cloud platforms, and dashboards.

Unit	Content	Credit	Weightage
I	**IoT Fundamentals & Sensor Data Acquisition** • **Introduction to IoT:** Architecture (sensors, connectivity, edge, cloud), applications (smart cities, health, industry). • **Sensors & Actuators:** Types (temperature, motion, pressure, image), specifications, calibration. • **IoT Communication Protocols:** MQTT, CoAP, HTTP, Bluetooth Low Energy (BLE), LoRaWAN. • **Edge Computing:** Edge devices (Raspberry Pi, Arduino), preprocessing at edge, fog computing. • **Data Acquisition Systems:** Sampling, quantization, data logging, streaming from sensors.	1	25%
II	**Sensor Data Processing & Time-Series Analysis** • **Time-Series Data Characteristics:** Seasonality, trends, noise, stationarity. • **Preprocessing Sensor Data:** Handling missing values, outliers, smoothing (moving average, median filter). • **Feature Engineering:** Lag features, rolling statistics, Fourier transforms, wavelet transforms. • **Dimensionality Reduction:** PCA, t-SNE for high-dimensional sensor data. • **Visualization:** Time-series plots, heatmaps, spectrograms, dashboarding (Grafana, Plotly Dash).	1	25%
III	**Machine Learning for Sensor Data** • **Classification:** Activity recognition using accelerometer/gyroscope data. • **Regression:** Predictive maintenance (remaining useful life estimation). • **Clustering:** Sensor grouping, fault detection. • **Anomaly Detection:** Statistical methods (IQR, Z-score), ML methods (Isolation Forest, One-Class SVM). • **Deep Learning for Sensor Data:** 1D CNNs, RNNs/LSTMs for sequence modeling, attention mechanisms.	1	25%
IV	**IoT Data Systems & Deployment** • **IoT Cloud Platforms:** AWS IoT Core, Google Cloud IoT, Azure IoT Hub.	1	25%

	• **Stream Processing:** Real-time analytics with Kafka, Spark Streaming, Flink. • **Data Storage:** Time-series databases (Influx DB, Time scale DB), NoSQL (Cassandra), data lakes. • **End-to-End Pipeline:** Edge data collection → MQTT broker → cloud processing → visualization. • **Case Studies:** Smart agriculture, industrial IoT, wearable health monitoring, environmental sensing.		

Textbooks:

- IoT and Edge Computing for Architects by Perry Lea
- Sensor Data Analysis and Management by Akiyoshi Wachi
- Time Series Analysis and Its Applications by Robert H. Shumway and David S. Stoffer

Reference books:

- Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti
- Applied Sensor Data Analytics with IoT by K. G. Srinivasa and G. M. Siddesh
- Deep Learning for Time Series Forecasting by Jason Brownlee
- IoT System Design: Project-Based Approach by Alice James and Avinash Gupta

Online Platforms:

- Coursera: "IoT Sensors and Devices" (University of California, Irvine)
- edX: "IoT Programming and Big Data" (Curtin University)
- YouTube: Andreas Spiess (The Swiss Guy), IoT for Everyone
- Kaggle Datasets: Sensor time-series datasets (e.g., accelerometer, temperature)
- Google Colab / Jupyter: For data analysis and ML modeling
- Node-RED / Grafana: For IoT dashboarding

PRACTICAL LIST:

Module 1 Practicals:

- Lab 1: Setting up Raspberry Pi/Arduino with temperature/humidity sensor (DHT22).
- Lab 2: Publishing sensor data to cloud using MQTT (Mosquitto broker).
- Lab 3: Collecting and logging accelerometer data from a smartphone/IMU sensor.
- Lab 4: Simulating sensor data streams using Python.

Module 2 Practicals:

- Lab 5: Preprocessing sensor data: handling noise, missing values, and smoothing.
- Lab 6: Feature extraction from time-series sensor data (rolling stats, FFT).
- Lab 7: Visualizing multi-sensor data with Matplotlib and Plotly.
- Lab 8: Dimensionality reduction and clustering of sensor data

Module 3 Practicals:

- Lab 9: Activity recognition using accelerometer data and Random Forest/CNN.
- Lab 10: Predictive maintenance simulation with regression models.
- Lab 11: Anomaly detection in sensor data using Isolation Forest and Autoencoders.
- Lab 12: LSTM for sensor time-series forecasting (e.g., energy consumption).

Module 4 Practicals:

- Lab 13: Building an IoT data pipeline: Sensor → MQTT → AWS IoT → Lambda → S3.
- Lab 14: Real-time dashboard with Grafana for sensor monitoring.
- Lab 15: Storing sensor data in Influx DB/Time scale DB and querying.
- Lab 16: Capstone Project: Smart environment monitoring system (air quality, temperature) with alerting.

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SUBJECT CODE: BTDSE704

SUBJECT NAME: DEVOPS FOR DS

Course Objectives:

- To understand DevOps principles, practices, and tools tailored for data science and ML workflows.
- To design and implement CI/CD pipelines for automated model training, testing, and deployment.
- To develop skills in containerization, orchestration, and infrastructure management for reproducible ML systems.
- To apply monitoring, logging, and collaboration tools to streamline DS/ML operations in production.

Course Outcomes: At the end of the course students shall be able to

CO1	Explain DevOps fundamentals and their application to data science and ML lifecycle.
CO2	Build and automate CI/CD pipelines for ML models using Git, Jenkins, GitHub Actions, etc.
C03	Containerize ML applications with Docker and orchestrate using Kubernetes.
C04	Implement monitoring, logging, and collaborative practices for scalable, reproducible DS workflows.

PRACTICAL LIST (12 Sessions)

Module 1 Practicals:

- Lab 1: Setting up Git repository, branching, and collaborative workflow for an ML project.
- Lab 2: Infrastructure as Code: Deploying cloud resources (S3, EC2) using Terraform.
- Lab 3: Creating reproducible environments with Conda and Docker.
- Lab 4: Automating dependency installation and environment setup with scripts.

Module 2 Practicals:

- Lab 5: Building a CI pipeline with GitHub Actions: Automated testing of data and model code.
- Lab 6: Creating a CD pipeline: Auto-train and evaluate model on new data commits.
- Lab 7: Model versioning and registry with ML flow.
- Lab 8: Orchestrating ML pipeline with Apache Airflow**.**

Module 3 Practicals:

- Lab 9: Dockerizing an ML model: Create Dockerfile, build image, push to Docker Hub.
- Lab 10: Deploying containerized model on Kubernetes (Minikube/local cluster).
- Lab 11: Using Helm to deploy ML application on Kubernetes.
- Lab 12: Serverless ML deployment with AWS Lambda for model inference.

Module 4 Practicals:

- Lab 13: Implementing model monitoring with Evidently AI and logging with ELK stack.
- Lab 14: Setting up monitoring dashboards with Prometheus and Grafana.
- Lab 15: Securing pipelines: Environment variables, secret management (AWS Secrets Manager).
- Lab 16: Capstone Project: End-to-end DevOps pipeline for an ML application—from code commit to **production monitoring.**

TOOLS & TECHNOLOGIES:

- Version Control: Git, GitHub, GitLab
- CI/CD: Jenkins, GitHub Actions, GitLab CI, CircleCI
- Containerization: Docker, Docker Compose

- Orchestration: Kubernetes, Helm, Kubeflow
- MLOps Tools: MLflow, DVC, Seldon Core, Evidently
- Monitoring: Prometheus, Grafana, ELK Stack
- Cloud Platforms: AWS, GCP, Azure (optional for labs)

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

SEMESTER-VIII

SUBJECT CODE: BTDSE801

SUBJECT NAME: RESEARCH METHDOLOGY IN DS

Course Objectives:

- To introduce the principles, processes, and ethical considerations of scientific research in data science.
- To develop skills in formulating research problems, conducting literature reviews, and designing data-driven studies.
- To apply statistical and computational methods for hypothesis testing, experimental design, and empirical evaluation.
- To prepare and present research findings through technical writing, visualization, and publication.

Course Outcomes: At the end of the course students shall be able to

CO1	Formulate clear, researchable questions and design valid data science experiments.
CO2	Conduct systematic literature reviews, identify research gaps, and synthesize existing work.
C03	Apply statistical and computational methods for hypothesis testing and result validation.
C04	Communicate research findings effectively through reports, papers, and presentations following academic and ethical standards.

Unit	Content	Credit	Weightage
I	**Foundations of Research in Data Science** • **Introduction to Research:** Types of research (exploratory, descriptive, explanatory, applied, theoretical). • **Research Process:** Steps from problem identification to publication. • **Ethics in Data Science Research:** Informed consent, data privacy, bias, reproducibility, FAIR principles. • **Literature Review:** Searching databases (Google Scholar, IEEE Xplore, arXiv), systematic reviews, citation management (Zotero, Mendeley). • **Research Question Formulation:** SMART criteria, hypothesis development, variables (independent, dependent, control).	1	25%
II	**Research Design & Data Collection** • **Experimental Design:** Between-subjects, within-subjects, factorial designs, quasi-experiments. • **Sampling Methods:** Probability vs. non-probability sampling, sample size determination (power analysis). • **Data Collection Strategies:** Surveys, web scraping, APIs, public datasets, sensor data, simulations. • **Data Quality & Preprocessing:** Validity, reliability, handling missing data, bias mitigation.	1	25%

	• **Tools for Data Collection:** Python (Beautiful Soup, Scrapy), R, Qualtrics, Google Forms.		
III	**Data Analysis & Statistical Methods** • **Quantitative Analysis:** Descriptive statistics, inferential statistics (t-tests, ANOVA, chi-square). • **Multivariate Analysis:** Regression, PCA, factor analysis. • **Machine Learning for Research:** Model selection, cross-validation, performance metrics. • **Qualitative Methods:** Thematic analysis, content analysis, mixed-methods approaches. • **Statistical Software:** R, Python (SciPy, Statsmodels), SPSS (overview).	1	25%
IV	**Research Communication & Dissemination** • **Technical Writing:** Structure of research papers (IMRaD), abstracts, literature reviews, citations (APA, IEEE). • **Data Visualization for Research:** Effective charts, storytelling with data, tools (Matplotlib, ggplot2, Tableau). • **Presenting Research:** Conference presentations, posters, academic talks. • **Publishing & Peer Review:** Journal selection, submission process, responding to reviews, open access. • **Reproducible Research:** Version control, Jupyter notebooks, Docker, Binder, sharing code/data.	1	25%

Textbooks:

- Research Methods for Data Science by David R. Hardoon and Sandro Cavallari
- Doing Data Science by Cathy O'Neil and Rachel Schutt
- The Craft of Research by Wayne C. Booth, Gregory G. Colomb, and Joseph M. Williams

Reference books:

- Data Science for Business by Foster Provost and Tom Fawcett
- Research Design: Qualitative, Quantitative, and Mixed Methods Approaches by John W. Creswell
- Communicating Data with Tableau by Ben Jones
- Ethics and Data Science by Mike Loukides, Hilary Mason, and DJ Patil

Online Platforms:

- Coursera: “Data Science Methodology” (IBM)
- edX: “Principles, Statistical and Computational Tools for Reproducible Data Science” (Harvard)
- Google Scholar, arXiv, IEEE Xplore – For literature search
- Overleaf – For collaborative LaTeX writing
- GitHub – For reproducible research and code sharing
- YouTube: Research methodology lectures, conference presentations (NeurIPS, KDD)

MK UNIVERSITY

PATAN, GUJARAT

ESTABLISHED BY THE GUJARAT GOVT. RECOGNIZED BY UGC UNDER SECTION 2(f) OF UGC ACT,1956

COURSE CODE: BTDSE802

COURSE NAME: ENTERPRISE SOLUTIONS

Course Objective

- Understand enterprise solution architecture, integration patterns, and business process modelling.
- Design and implement Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Supply Chain Management (SCM) solutions.
- Analyse enterprise data management, business intelligence, and analytics solutions.
- Develop enterprise application integration using middleware and API management.
- Evaluate cloud-based enterprise solutions and digital transformation strategies.
- Design enterprise security, governance, and compliance frameworks.

Course Outcomes:

CO1	Analyze enterprise solution requirements and design integratedarchitectures.
CO2	Implement ERP, CRM, and SCM solutions for business processes
CO3	Design and develop enterprise data management and business intelligence solutions
CO4	Implement enterprise application integration using middleware platforms

Unit	Content	Credit	Weightage
I	Enterprise Solution Architecture & Integration • Introduction to Enterprise Solutions, Enterprise Architecture Frameworks, Business Process Management (BPM), Enterprise Integration Patterns, Service-Oriented Architecture (SOA), Microservices Architecture, Middleware Technologies	1	25%
II	Core Enterprise Solutions (ERP, CRM, SCM) • Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supply Chain Management (SCM), Human Capital Management (HCM), Enterprise Content Management (ECM)	1	25%
III	Enterprise Data & Analytics Solutions • Enterprise Data Management, Business Intelligence & Analytics, Big Data Solutions for Enterprises, Artificial Intelligence in Enterprises, Enterprise Search Solutions, Digital Experience Platforms (DXP)	1	25%
IV	Enterprise Solution Implementation & Management • Enterprise Solution Implementation, Cloud Enterprise Solutions, Enterprise Security & Governance,	1	25%

	Enterprise Solution Operations, Integration Platforms as a Service (iPaaS), Emerging Trends, Career Paths in Enterprise Solutions		

Textbooks:

- "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions" *Gregor Hohpe & Bobby Woolf Definitive guide to enterprise integration patterns*
- "The Architecture of Enterprise: Creating Business Value through Enterprise Architecture" *John Gøtze & Anders Jensen-Waud Modern enterprise architecture perspective*

Reference Books:

- "ERP: Making It Happen: The Implementers' Guide to Success with Enterprise Resource Planning" *Thomas F. Wallace & Michael H. Kremzar Practical ERP implementation guide*
- "Business Process Management: Concepts, Languages, Architectures" *Mathias Weske* (3rd Edition) *Comprehensive BPM reference*
- "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modelling" *Ralph Kimball & Margy Ross* (3rd Edition) *Essential data warehousing guide*

Online Resources:

- SAP Community Network
- Salesforce Trailhead (Free learning platform)
- Microsoft Learn (Dynamics 365, Azure)
- Oracle University
- Gartner Research (Enterprise technology trends)
- Forrester Wave Reports
- NPTEL: "Enterprise Systems" courses
- Coursera: "Enterprise Architecture" specializations

Practical List:

- Session 1: Enterprise architecture modelling with ArchiMate
- Session 2: Business process modelling with BPMN 2.0
- Session 3: Process automation with Camunda
- Session 4: Integration pattern implementation
- Session 5: ERP configuration and customization (Odoo/SAP)
- Session 6: CRM implementation (Salesforce/HubSpot)
- Session 7: SCM process implementation
- Session 8: Enterprise data warehouse design
- Session 9: Business Intelligence dashboard development
- Session 10: Enterprise application integration with MuleSoft
- Session 11: API management implementation
- Session 12: Cloud migration strategy development

COURSE CODE: BTDSE803
COURSE NAME: PARALLEL COMPUTING
Course Objective

- Understand parallel computing architectures, models, and performance metrics.
- Design and analyse parallel algorithms for shared and distributed memory systems.
- Implement parallel programs using MPI, OpenMP, and CUDA programming models.
- Optimize parallel applications for performance, scalability, and energy efficiency.

- Evaluate parallel algorithms using complexity measures and performance analysis tools.
- Design solutions for load balancing, synchronization, and communication in parallel systems

Course Outcomes:

CO1	Analyze parallel computing architectures and classify parallel systems using Flynn's taxonomy.
CO2	Design and analyze parallel algorithms using PRAM and work-span models
CO3	Implement shared memory parallel programs using OpenMP and P threads
CO4	Develop distributed memory parallel applications using MPI

Unit	Content	Credit	Weightage
I	Parallel Computing Foundations & Architectures • Introduction to Parallel Computing, Parallel Architectures, Performance Metrics and Laws, Parallel Algorithm Design Principles, Parallel Programming Models	1	25%
II	Shared Memory Parallel Programming • Thread-based Parallelism, OpenMP Programming Model, Task Parallelism in OpenMP, Advanced OpenMP Features, Performance Optimization for Shared Memory, Synchronization Primitives	1	25%
III	Distributed Memory Parallel Programming • Message Passing Interface (MPI), Advanced MPI Features, Parallel Algorithm Design for Distributed Memory, Performance Analysis and Optimization, Hybrid Programming	1	25%
IV	GPU Computing & Advanced Topics • GPU Architecture, CUDA Programming Model, CUDA Programming, CUDA Optimization Techniques, Advanced GPU Programming, Alternative GPU Programming Models, Parallel Patterns and Algorithms, Emerging Trends	1	25%

Textbooks:

- "Introduction to Parallel Computing" *Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar* (2nd Edition) *Comprehensive coverage of parallel computing concepts*
- "Programming Massively Parallel Processors: A Hands-on Approach" *David B. Kirk & Wen-mei W. Hwu* (4th Edition) *Excellent for GPU computing and CUDA programming*

Reference Books:

- "Parallel Programming in C with MPI and OpenMP" *Michael J. Quinn Practical approach to MPI and OpenMP programming*
- "Structured Parallel Programming: Patterns for Efficient Computation" *Michael McCool, James Reinders, Arch Robison Pattern-based approach to parallel programming*
- "CUDA by Example: An Introduction to General-Purpose GPU Programming" *Jason Sanders & Edward Kandrot Hands-on introduction to CUDA programming*

Online Resources:

- NVIDIA CUDA Documentation and Tutorials
- OpenMP Specifications and Examples
- MPI Forum Documentation
- LLNL Parallel Computing Tutorials
- NPTEL: "Parallel Computing" by Prof. Subodh Kumar
- MIT Open Course Ware: Parallel Computing courses
- Intel Parallel Studio Documentation

Practical List:

- Session 1: OpenMP setup and basic parallel constructs
- Session 2: OpenMP work-sharing and synchronization
- Session 3: OpenMP tasks and advanced features
- Session 4: Pthreads programming basics
- Session 5: Performance analysis and optimization of shared memory programs
- Session 6: MPI setup and point-to-point communication
- Session 7: MPI collective operations
- Session 8: Parallel matrix multiplication using MPI
- Session 9: Hybrid MPI+Open MP programming
- Session 10: CUDA setup and basic kernel programming
- Session 11: CUDA memory optimization and shared memory
- Session 12: Advanced CUDA features (streams, events)