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 Artificial Intelligence (AI)

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-I									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI101	ENGINEERING MATHEMATICS-I	4	0	4	40	60	100
2	MAJOR	BTAI102	PROGRAMING FOR AI (PYTHON)	4	2	6	90	60	150
3	MAJOR	BTAI103	FUNDAMENTALS OF AI	4	2	6	90	60	150
4	MINOR	BTAI104	DIGITAL LOGIC AND SYSTEMS	4	0	4	40	60	100
5	VAC	BTAI105	COMMUNICATION SKILLS-I	2	0	2	0	50	50
TOTAL				18	4	22	260	290	550

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-II									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI201	DATA STRUCTURES & ALGORITHMS	4	2	6	90	60	150
2	MAJOR	BTAI202	PROBABLITY AND STATICS FOR AI	4	0	4	40	60	100
3	MAJOR	BTAI203	LINEAR ALGEBRA FOR AI	4	0	4	40	60	100
4	MINOR	BTAI204	DATA BASE MANAGEMENT SYSTEMS	4	2	6	90	60	150
5	VAC	BTAI205	ENVIRONMENTAL SCIENCE	2	0	2	0	50	50
TOTAL				18	4	22	260	290	550

MK UNIVERSITY

PATAN, GUJARAT

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

B. TECH ARTIFICIAL INTELLIGENCE) SEM-III									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI301	MACHINE LEARNING	4	2	6	90	60	150
2	MAJOR	BTAI302	BIG DATA TECHNOLOGIES	4	2	6	90	60	150
3	MAJOR	BTAI303	COMPUTER VISION FUNDAMENTALS	4	0	4	40	60	100
4	MINOR	BTAI304	BUSINESS INTELLIGENCE	4	0	4	40	60	100
5	SEC	BTAI305	SQL & NOSQL DATABASE	0	2	2	00	50	50
TOTAL				16	6	22	260	290	550

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-IV									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI401	NATURAL LANGUAGE PROCESSING	4	0	4	40	60	100
2	MAJOR	BTAI402	DEEP LEARNING FUNDAMENTALS	4	2	6	90	60	150
3	MAJOR	BTAI403	CLOUD COMPUTING	4	0	4	40	60	100
4	MINOR	BTAI404	WEB TECHNOLOGIES	4	0	4	40	60	100
5	SEC	BTAI405	DATA WRANGLING WITH PANDAS	0	2	2	00	50	50
6	VAC	BTAI406	Indian Constitution	2	0	2	0	50	50
TOTAL				18	4	22	210	340	550

MK UNIVERSITY

PATAN, GUJARAT

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

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-V									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI501	AI ETHICS & RESPONSIBLE AI	4	2	6	90	60	150
2	MAJOR	BTAI502	TIME SERIES ANALYSIS	4	2	6	90	60	150
3	MAJOR	BTAI503	REINFORCEMENT LEARNING	4	0	4	40	60	100
4	MINOR	BTAI504	STREAM PROCESSING (KAFKA/SPARK)	4	0	4	40	60	100
6	VAC	BTAI505	MINI-PROJECT	0	4	4	50	0	50
TOTAL				16	8	24	310	240	550

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-VI									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI601	ADVANCED DEEP LEARNING	4	2	6	90	60	150
2	MAJOR	BTAI602	MLOPS & MODEL DEVELOPMENT	4	2	6	90	60	150
3	MAJOR	BTAI603	AI SYSTEMS DESIGN	4	0	4	40	60	100
4	MINOR	BTAI604	IOT & SENSOR	4	0	4	40	60	100
5	SEC	BTAI605	APTITUDE & CARRER SKILLS	0	2	2	50	0	50
TOTAL				16	6	22	310	240	550

MK UNIVERSITY

PATAN, GUJARAT

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

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-VII									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERNAL	EXTERNAL	
1	MAJOR	BTAI701	AI IN ROBOTICS	4	2	6	90	60	150
2	MAJOR	BTAI702	CLOUD NATIVE DATA SYSTEMS	4	2	6	90	60	150
3	MINOR	BTAI703	AI DEPLOYMENT & MLOPS	4	0	4	40	60	100
4	SEC	BTAI704	DEVOPS FOR AI	0	2	2	00	50	50
5	VAC	BTAI705	Project Phase-I	0	4	4	100	00	100
TOTAL				12	10	22	320	230	550

B. TECH (ARTIFICIAL INTELLIGENCE) SEM-VIII									
SR NO.	COURSE TYPE	COURSE CODE	COURSE NAME	LECTURE (HRS.)/ WEEK	PRACTICAL (HRS.)/ WEEK	CREDITS	EXAMINATION		TOTAL MARKS
							INTERN AL	EXTERN AL	
1	MAJOR	BTAI801	RESEARCH METHDOLOGY IN AI	4	0	4	40	60	100
2	MAJOR	BTAI802	ENTERPRISE SOLUTIONS	4	2	6	90	60	150
3	MINOR	BTAI803	PARALLEL COMPUTING	4	2	6	90	60	150
4	SEC	BTAI804	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: BTAI101

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: BTAI102

SUBJECT NAME: PROGRAMING FOR AI (PYTHON)

Course Objectives:

- To introduce Python as the primary programming language for AI.
- 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 AI 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: BTAI103

SUBJECT NAME: FUNDAMENTALS OF AI

Course Objectives:

- To introduce the core concepts, history, and scope of Artificial Intelligence.
- To understand the architecture of intelligent agents and their environments.
- To explore problem-solving strategies using search algorithms and game theory.
- To build a foundation in knowledge representation, reasoning, and AI ethics.
- To prepare students for advanced AI topics such as machine learning, NLP, and robotics.

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

CO1	Explain the history, goals, and taxonomy of AI and intelligent agents.
CO2	Apply search algorithms (uninformed, informed, adversarial) to solve AI problems.
C03	Represent knowledge using logic and reason under uncertainty.
C04	Discuss ethical, societal, and future implications of AI systems.

Unit	Content	Credit	Weightage
I	**Introduction to AI & Intelligent Agents** • **Introduction to AI:** Definition, history, Turing Test, AI categories (Weak vs Strong AI). • **Goals of AI:** Rationality, optimality, and human-like behaviour. • **Intelligent Agents:** Agent architecture, PEAS framework, types of agents (simple reflex, model-based, goal-based, utility-based). • **Agent Environments:** Fully vs partially observable, deterministic vs stochastic, static vs dynamic, discrete vs continuous. • **AI Applications:** Overview of AI in healthcare, finance, gaming, robotics, and NLP.	1	25%
II	**Problem-Solving & Search Algorithms** • **Problem Formulation:** State space, initial state,	1	25%

	goal state, actions, path cost. • **Uninformed Search:** BFS, DFS, Uniform Cost Search, Depth-Limited Search, Iterative Deepening. • **Informed Search:** Greedy Best-First Search, A* Algorithm, heuristic functions. • **Adversarial Search:** Game theory, Minimax algorithm, Alpha-Beta pruning. • **Constraint Satisfaction Problems (CSP):** Backtracking, forward checking, arc consistency.		
III	**Knowledge Representation & Reasoning** • **Logic-Based Representation:** Propositional logic, First-Order Logic (FOL), syntax, semantics. • **Inference in Logic:** Forward chaining, backward chaining, resolution. • **Uncertainty & Probability:** Bayesian networks, probabilistic reasoning. • **Rule-Based Systems:** Expert systems, production rules, inference engines. • **Introduction to Planning:** STRIPS representation, planning algorithms.	1	25%
IV	**AI Ethics & Future Trends** • **AI Ethics:** Bias in AI, fairness, transparency, accountability. • **Societal Impact:** Job displacement, privacy, security, autonomous weapons. • **AI Regulations & Guidelines:** GDPR, AI ethics frameworks (IEEE, EU). • **Future of AI:** AGI vs Narrow AI, singularity, AI in sustainability. • **Case Studies:** Real-world AI successes and failures.	1	25%

Textbooks:

- Artificial Intelligence: A Modern Approach *Stuart Russell & Peter Norvig (4th Edition, Pearson)*
- Artificial Intelligence: Foundations of Computational Agents *David L. Poole & Alan K. Mackworth (Cambridge University Press)*
- AI: The Very Idea *John Haugeland (MIT Press)*

Reference books:

- Artificial Intelligence: Structures and Strategies for Complex Problem-Solving *George F. Luger (Pearson)*
- The Quest for Artificial Intelligence *Nils J. Nilsson (Cambridge University Press)*
- Life 3.0: Being Human in the Age of Artificial Intelligence *Max Tegmark (Penguin Books)*
- Human Compatible: Artificial Intelligence and the Problem of Control *Stuart Russell (Penguin Books)*

Online Platforms:

- Coursera: *AI For Everyone* (Andrew Ng)
- edX: *Introduction to Artificial Intelligence* (UC Berkeley)
- NPTEL: *Artificial Intelligence* (IIT Madras)

PRACTICAL LIST:

Module I Practicals:

- Lab 1: Setting up Python AI environment (Anaconda, Jupyter, libraries).
- Lab 2: Implementing a simple reflex agent (e.g., vacuum cleaner world).
- Lab 3: PEAS description for real-world agents (e.g., self-driving car, chatbot).
- Lab 4: Case study presentation on AI applications in industry.

Module II Practicals:

- Lab 5: BFS and DFS for maze solving.
- Lab 6: A* algorithm implementation for pathfinding (using grids).
- Lab 7: Minimax algorithm for Tic-Tac-Toe.
- Lab 8: Constraint Satisfaction Problem – Map coloring.

Module III Practicals:

- Lab 9: Propositional logic inference using truth tables.
- Lab 10: First-Order Logic representation in Python (using sympy).
- Lab 11: Building a simple rule-based expert system (medical diagnosis).
- Lab 12: Bayesian network simulation with pgmpy.

Module IV Practicals:

- Lab 13: Bias detection in a dataset using AI Fairness 360.
- Lab 14: Ethical analysis of an AI case study (e.g., facial recognition).
- Lab 15: Debate session – "AI Ethics: Regulation vs Innovation".
- Lab 16: Future AI trends research and presentation.

SUBJECT CODE: BTAI104

SUBJECT NAME: DIGITAL LOGIC AND SYSTEMS

Course Objectives:

To provide a comprehensive understanding of digital systems design fundamentals, from basic logic gates to complex sequential circuits and memory elements. The course aims to develop the ability to analyze, design, and implement digital circuits that form the foundation of computer architecture, embedded systems, and digital hardware.

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

CO1	**Apply** number systems, codes, and binary arithmetic to solve digital computation problems.
CO2	**Analyze** and **simplify** Boolean functions using algebraic methods and K-maps.
C03	**Design** and **implement** combinational circuits for arithmetic operations, data routing, and code conversion.
C04	**Design** sequential circuits including counters, registers, and finite state machines.

Unit	Content	Credit	Weightage
I	**Number Systems and Codes** • **Number Systems:** Binary, Octal, Hexadecimal, conversions • **Complements:** 1's, 2's, 9's, 10's complements • **Binary Arithmetic:** Addition, subtraction, multiplication, division • **Binary Codes:** BCD, Gray code, Excess-3, ASCII,	1	25%

	Unicode • **Error Detection & Correction:** Parity bits, Hamming codes		
II	**Boolean Algebra and Logic Gates** • **Boolean Algebra:** Postulates, theorems, De Morgan's laws • **Logic Gates:** AND, OR, NOT, NAND, NOR, XOR, XNOR - symbols, truth tables • **Gate-level Implementation:** SOP, POS forms • **Universal Gates:** Implementation using NAND/NOR only • **Integrated Circuits:** SSI, MSI, LSI, VLSI concepts, TTL/CMOS families	1	25%
III	**Combinational Logic Design** • **Minimization Techniques:** o Karnaugh Maps (2-5 variables) o Quine-McCluskey method o Don't care conditions • **Combinational Circuits:** o Adders (Half, Full, Ripple carry, Carry look-ahead) o Subtractors o Comparators o Multiplexers (MUX) and Demultiplexers (DEMUX) o Encoders and Decoders (Binary, BCD to 7-segment) • **Code Converters:** Binary to Gray, BCD to Excess-3, etc. **Hazards:** Static, dynamic hazards and elimination	1	25%
IV	**Sequential Logic Design** • **Latches:** SR latch, D latch, gated latches • **Flip-flops:** SR, JK, D, T, Master-slave, edge-triggered • **Flip-flop Conversions:** One type to another • **Registers:** Shift registers (SISO, SIPO, PISO, PIPO), Universal shift register • **Counters:** o Asynchronous (Ripple) counters o Synchronous counters o Mod-N counters, up/down counters o Ring counter, Johnson counter **Finite State Machines (FSM):** Mealy and Moore models	1	25%

Textbooks:

1. Digital Design by M. Morris Mano and Michael D. Ciletti
2. Fundamentals of Digital Logic with Verilog Design by Stephen Brown and Zvonko Vranesic
3. Digital Logic and Computer Design by M. Morris Mano

Reference books:

1. Digital Electronics by R.P. Jain
2. Verilog HDL: A Guide to Digital Design and Synthesis by Samir Palnitkar
Introduction to Logic Design by Alan B. Marcovitz

MK UNIVERSITY

PATAN, GUJARAT

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

SEMESTER-II

SUBJECT CODE: BTAI201

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.

SUBJECT CODE: BTAI202

SUBJECT NAME: PROBABLITY AND STATICS FOR AI

Course Objectives:

- To establish a strong foundation in probability theory and statistical methods essential for AI and machine learning.
- To understand and apply probability distributions, sampling methods, and statistical inference in AI contexts.
- To develop skills in hypothesis testing, regression analysis, and Bayesian reasoning for data-driven decision-making.
- To prepare students to model uncertainty, analyse data, and validate AI models using statistical tools.

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

CO1	Apply descriptive statistics and probability concepts to summarize and model AI datasets.
CO2	Utilize common probability distributions and sampling methods for AI simulations and inference.
C03	Perform hypothesis testing and confidence interval estimation for AI model validation.
C04	Conduct regression analysis and apply Bayesian methods to AI problems.

Unit	Content	Credit	Weightage
I	**Descriptive Statistics & Probability Fundamentals** • **Introduction to Statistics in AI:** Role of statistics in data science and AI. • **Descriptive Statistics:** Measures of central tendency (mean, median, mode), dispersion (variance, SD, IQR), skewness, kurtosis. • **Data Visualization for AI:** Histograms, box plots, scatter plots, Q-Q plots. • **Probability Basics:** Sample space, events, axioms, conditional probability, Bayes' theorem. • **Random Variables:** Discrete vs continuous, PMF, PDF, CDF, expectation, variance.	1	25%
II	**Probability Distributions & Sampling Theory** • **Discrete Distributions:** Bernoulli, Binomial, Poisson, Geometric. • **Continuous Distributions:** Uniform, Normal, Exponential, Beta, Gamma.	1	25%

	• **Central Limit Theorem:** Concept, significance, and applications in AI. • **Sampling Methods:** Random sampling, stratified, cluster, bootstrapping. • **Estimation Theory:** Point estimation, bias, efficiency, consistency, maximum likelihood estimation (MLE).		
III	**Statistical Inference & Hypothesis Testing** • **Introduction to Inference:** Population vs sample, parameter vs statistic. • **Confidence Intervals:** For mean, proportion, variance; interpretation in AI contexts. • **Hypothesis Testing:** Null and alternative hypotheses, p-value, significance level (α), type I/II errors. • **Parametric Tests:** Z-test, t-test (one-sample, two-sample, paired), chi-square test. • **Non-Parametric Tests:** Wilcoxon signed-rank, Mann–Whitney U test. • **ANOVA:** One-way and two-way ANOVA, F-test, post-hoc analysis.	1	25%
IV	**Regression, Bayesian Methods & AI Applications** • **Linear Regression:** Simple and multiple linear regression, assumptions, R^2, MSE. • **Model Diagnostics:** Residual analysis, multicollinearity, heteroscedasticity. • **Logistic Regression:** For binary classification, odds ratio, sigmoid function. • **Bayesian Inference:** Prior, likelihood, posterior, conjugate priors, Bayesian updating. • **AI Applications:** A/B testing, anomaly detection, probabilistic graphical models (intro). • **Statistical Software in AI:** Using Python (SciPy, Stats Models) and R for statistical analysis.	1	25%

Textbooks:

- Probability and Statistics for Engineering and the Sciences *Jay L. Devore (9th Edition, Cengage)*
- Introduction to Probability *Joseph K. Blitzstein & Jessica Hwang* *(2nd Edition, Chapman & Hall/CRC) *
- All of Statistics: A Concise Course in Statistical Inference *Larry Wasserman (Springer)*

Reference books:

- Bayesian Statistics for Beginners: A Step-by-Step Approach *Therese M. Donovan & Ruth M. Mickey (Oxford University Press)*
- Practical Statistics for Data Scientists *Peter Bruce & Andrew Bruce (2nd Edition, O'Reilly)*
- The Elements of Statistical Learning *Trevor Hastie, Robert Tibshirani, Jerome Friedman (2nd Edition, Springer)*
- Probability Theory: The Logic of Science *E. T. Jaynes (Cambridge University Press)*

Online Platforms:

- Coursera: *Statistics with Python Specialization* (University of Michigan)
- edX: *Probability and Statistics in Data Science using Python* (UC San Diego)

- NPTEL: *Probability and Statistics* (IIT Kanpur)
- Khan Academy: *Statistics & Probability* modules

SUBJECT CODE: BTAI203

SUBJECT NAME: LINEAR ALGEBRA FOR AI

Course Objectives:

- To build a strong mathematical foundation in linear algebra essential for AI, machine learning, and deep learning.
- To understand vector spaces, matrices, linear transformations, and their geometric interpretations in AI.
- To apply matrix decompositions, eigenvalues, and eigenvectors to dimensionality reduction and optimization.
- To develop computational skills using Python libraries for solving linear algebra problems in AI applications.

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

CO1	Perform vector and matrix operations and interpret them geometrically.
CO2	Solve systems of linear equations and apply matrix decompositions (LU, QR, SVD).
C03	Apply eigenvalues, eigenvectors, and diagonalization in dimensionality reduction (PCA).
C04	Use linear algebra concepts in AI models such as regression, NLP embeddings, and neural networks.

Unit	Content	Credit	Weightage
I	**Vectors, Matrices & Systems of Linear Equations** • **Vectors:** Definition, operations (addition, scalar multiplication), dot product, cross product, norms, orthogonality. • **Matrices:** Types (square, diagonal, identity, symmetric, skew-symmetric), operations (addition, multiplication, transpose, trace, determinant). • **Systems of Linear Equations:** Gaussian elimination, row reduction, rank, consistency, matrix inversion. • **Vector Spaces:** Subspaces, basis, dimension, linear independence, span. • **Geometric Interpretation:** Lines, planes, hyperplanes in AI context.	1	25%
II	**Matrix Decompositions & Linear 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. • **Linear Transformations:** Matrix representation, kernel, image, rank–nullity theorem.	1	25%

	• **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 (gradient descent). • **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 *Gilbert Strang (6th Edition, Cengage)*
- Mathematics for Machine Learning *Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong (Cambridge University Press)*
- Introduction to Linear Algebra *Gilbert Strang* *(5th Edition, Wellesley-Cambridge Press) *

Reference books:

- Linear Algebra Done Right *Sheldon Axler (3rd Edition, Springer)*
- The Matrix Cookbook *Kaare Brandt Petersen & Michael Syskind Pedersen (Technical Reference)*
- Deep Learning *Ian Goodfellow, Yoshua Bengio, Aaron Courville (MIT Press) – Chapters on Linear Algebra*
- Numerical Linear Algebra *Lloyd N. Trefethen & David Bau III*

Online Platforms:

- MIT Open Course Ware: *Linear Algebra* (Gilbert Strang)
- Coursera: *Mathematics for Machine Learning: Linear Algebra* (Imperial College London)
- edX: *Linear Algebra – Foundations to Frontiers* (University of Texas at Austin)
- Khan Academy: *Linear Algebra* modules

MK UNIVERSITY

PATAN, GUJARAT

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

COURSE CODE: BTAI204
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) ○ Modern applications (e-commerce, social media, IoT) ○ Emerging trends (big data, cloud databases)	1	25%
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	1	25%

	• **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)		

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
	  ○ Cursors, exception handling
	  ○ Embedded SQL, dynamic SQL
		1	25%

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

MK UNIVERSITY

PATAN, GUJARAT

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

SEMESTER-III

SUBJECT CODE: BTAI301

SUBJECT NAME: MACHINE LEARNING

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: BTAI302

SUBJECT NAME: COMPUTER VISION FUNDAMENTALS

Course Objectives:

- To introduce the fundamental concepts, techniques, and algorithms of computer vision.
- To develop skills in image processing, feature extraction, and object detection.
- To apply machine learning and deep learning methods to visual recognition tasks.
- To build practical experience in implementing computer vision pipelines using Python and OpenCV.
- To prepare students for advanced topics in video analysis, 3D vision, and real-time vision systems.

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

CO1	Explain the image formation process, digital image representation, and basic image processing techniques.
CO2	Apply feature detection, description, and matching techniques for image alignment and recognition.
C03	Implement object detection, segmentation, and classification using traditional and deep learning methods.
C04	Design and evaluate computer vision systems for real-world applications such as surveillance, medical imaging, and autonomous systems.

Unit	Content	Credit	Weightage
I	**Introduction to Computer Vision & Image Processing** • **Introduction to CV:** History, applications, and challenges. • **Image Formation:** Pinhole camera model, lenses, color spaces (RGB, HSV, LAB). • **Digital Image Basics:** Pixels, resolution, bit-depth, image file formats. • **Image Preprocessing:** Grayscale conversion, histogram equalization, filtering (Gaussian, median, Sobel). • **Morphological Operations:** Erosion, dilation, opening, closing. • **Edge Detection:** Canny, Sobel, Laplacian operators.	1	25%
II	**Feature Extraction & Image Matching** • **Interest Point Detection:** Harris corner detector, Shi-Tomasi, FAST. • **Feature Descriptors:** SIFT, SURF, ORB, BRIEF. • **Feature Matching:** Brute-force matching, FLANN,	1	25%

	RANSAC for outlier removal. • **Image Alignment:** Homography, affine transformation, image stitching (panoramas). • **Optical Flow:** Lucas-Kanade method, dense optical flow.		
III	**Object Detection & Recognition** • **Classical Detection Methods:** Haar cascades, HOG + SVM, sliding window approach. • **Deep Learning for CV:** o Convolutional Neural Networks (CNN) architecture. o Transfer learning with pre-trained models (VGG, ResNet). o Object detection frameworks: R-CNN, Fast R-CNN, YOLO, SSD. • **Evaluation Metrics:** IoU, mAP, precision-recall curves.	1	25%
IV	**Image Segmentation & Advanced Topics** • **Segmentation Techniques:** Thresholding, region-based, watershed, GrabCut. • **Deep Learning Segmentation:** FCN, U-Net, Mask R-CNN. • **3D Computer Vision:** Stereo vision, depth maps, point clouds. • **Video Analysis:** Motion detection, tracking (Kalman filter, Mean-Shift). • **Ethics in CV:** Bias in facial recognition, privacy concerns, responsible AI.	1	25%

Textbooks:

- Computer Vision: Algorithms and Applications *Richard Szeliski (2nd Edition, Springer)*
- Learning OpenCV: Computer Vision with the OpenCV Library *Gary Bradski & Adrian Kaehler (O'Reilly)*
- Deep Learning for Computer Vision *Rajalingappaa Shanmugamani (Packt)*

Reference books:

- Multiple View Geometry in Computer Vision *Richard Hartley & Andrew Zisserman (2nd Edition, Cambridge University Press)*
- Computer Vision: Models, Learning, and Inference *Simon J. D. Prince (Cambridge University Press)*
- Programming Computer Vision with Python *Jan Erik Solem (O'Reilly)*
- The Computer Vision Workshop *Hafsa Asad, et al. (Packt)*

Online Platforms:

- Coursera: *Deep Learning Specialization* (Andrew Ng) – CNN modules
- edX: *Computer Vision Basics* (University at Buffalo)
- Udacity: *Introduction to Computer Vision* (Georgia Tech)
- NPTEL: *Computer Vision* (IIT Madras**)**

MK UNIVERSITY

PATAN, GUJARAT

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

SUBJECT CODE: BTAI303

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,	1	25%

	data lakes vs data warehouses.		
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: BTAI304

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,	1	25%

	storytelling, interactivity. • **BI Tools Overview:** Tableau, Power BI, Qlik Sense, Looker. • **Visual Analytics:** Creating charts, maps, filters, 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: BTAI305

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

COURSE CODE: BTAI401
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: BTAI402

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)**	1	25%

	• **CNN Fundamentals:** Convolution operation, filters, padding, stride, pooling. • **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:

- Fast.ai – Practical deep learning courses
- Coursera: "Deep Learning Specialization" by Andrew Ng (deeplearning.ai)
- 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: BTAI403
COURSE NAME: CLOUD COMPUTING FOR AI
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

MK UNIVERSITY

PATAN, GUJARAT

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

COURSE CODE: BTAI404
COURSE NAME: WEB TECHNOLOGIES

Course Objective

- Design and develop responsive web applications using HTML5, CSS3, and JavaScript.
- Implement server-side programming using Node.js and PHP with database integration.
- Develop dynamic web applications using React.js and Angular frameworks.
- Apply web security principles, RESTful APIs, and web services.
- Deploy web applications using cloud platforms and DevOps practices.
- Evaluate web application performance, accessibility, and SEO optimization.

Course Outcomes:

CO1	Design responsive web interfaces using HTML5, CSS3, and Bootstrap.
CO2	Develop interactive client-side applications using JavaScript and jQuery
CO3	Implement server-side applications using Node.js and PHP with MySQL/MongoDB
CO4	Build single-page applications using React.js and Angular frameworks

Unit	Content	Credit	Weightage
I	Frontend Technologies & Responsive Design • Web Fundamentals, HTML5, CSS3, Responsive Web Design, Bootstrap Framework, Version Control,	1	25%
II	Client-Side Programming & JavaScript • JavaScript Fundamentals, DOM Manipulation, Advanced JavaScript, jQuery Library, Frontend Build Tools, Browser Storage	1	25%
III	Server-Side Development & Databases • Server-Side Programming Concepts, Node.js and Express.js, PHP Programming, Database Integration Authentication & Security, Real-time Communication	1	25%
IV	Modern Frameworks & Deployment • React.js Framework, Angular Framework, Web Services & APIs, Testing Web Applications, Performance Optimization, Deployment & DevOps, Progressive Web Apps (PWAs), SEO Fundamentals, Emerging Trends	1	25%

Textbooks:

- "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" *Jennifer Niederst Robbins* (5th Edition) *Excellent foundation for web design basics*
- "Eloquent JavaScript: A Modern Introduction to Programming" *Marijn Haverbeke* (3rd Edition) *Free online, comprehensive JavaScript guide*

Reference Books:

- "HTML and CSS: Design and Build Websites" *Jon Duckett Visually appealing, beginner-friendly HTML/CSS book*
- "JavaScript: The Definitive Guide" *David Flanagan* (7th Edition) *Comprehensive JavaScript reference*
- "Node.js Design Patterns" *Mario Casciaro & Luciano Mammino* (3rd Edition) *Advanced Node.js patterns and best practices*

Online Resources:

- MDN Web Docs (Mozilla Developer Network)
- W3Schools (Tutorials and references)
- Free Code Camp (Full web development curriculum)
- Frontend Masters (Video courses)
- CSS-Tricks (CSS tutorials and articles)
- React.js Official Documentation
- Angular Official Documentation
- Node.js Official Documentation
- NPTEL: "Web Technologies" by IIT Kharagpur

SUBJECT CODE: BTAI405

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: BTAI501

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. • **Advanced Algorithms:** A2C, A3C, PPO, TRPO,	1	25%

	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 & 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: BTAI502

SUBJECT NAME: AI ETHICS AND RESPONSIBLE AI

Course Objectives:

- To understand the ethical, social, and legal challenges posed by AI systems.
- To identify and mitigate biases, ensure fairness, and promote transparency in AI models.
- To apply ethical frameworks, guidelines, and responsible practices in real-world AI projects.
- To develop skills in AI governance, accountability, and human-centered design for trustworthy AI 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 AI/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 AI** • **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), Fairlearn, What-If Tool, IBM Watson OpenScale. • **Bias Mitigation Strategies:** Pre-processing (reweighting, sampling), in-processing (constraints, adversarial debiasing), post-processing (calibration). • **Legal & Regulatory Aspects:** GDPR, CCPA, Algorithmic Accountability Act, EU AI Act.	1	25%
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-	1	25%

	term societal impacts.		

Textbooks:

- Weapons of Math Destruction *Cathy O'Neil (Broadway Books)*
- The Ethical Algorithm *Michael Kearns & Aaron Roth (Oxford University Press)*
- Atlas of AI *Kate Crawford (Yale University Press)*

Reference books:

- Fairness and Machine Learning *Solon Barocas, Moritz Hardt, & Arvind Narayanan*
  The Alignment Problem *Brian Christian (Norton & Company)*
- Ethics of Artificial Intelligence and Robotics *Vincent C. Müller (Stanford Encyclopedia of Philosophy)*
- Data Feminism *Catherine D'Ignazio & Lauren F. Klein (MIT Press)*
- Human Compatible: Artificial Intelligence and the Problem of Control *Stuart Russell (Penguin Books)*

Online Platforms:

- Coursera: *AI Ethics: Global Perspectives* (University of California, Davis)
- edX: *Data Science Ethics* (University of Michigan)
- Future Learn: *Ethics of AI* (University of Helsinki)
- NPTEL: *Ethics in Engineering Practice* (IIT Kharagpur)

PRACTICAL LIST:

Module I Practicals:

- Lab 1: Ethical analysis of a real-world AI case study (e.g., COMPAS recidivism algorithm).
- Lab 2: Stakeholder mapping exercise for an AI system (e.g., autonomous vehicles).
- Lab 3: Debate session – "AI Ethics: Regulation vs Innovation".
- Lab 4: Developing an AI ethics charter for a hypothetical company.

Module II Practicals:

- Lab 5: Bias detection in a dataset using AI Fairness 360 (AIF360).
- Lab 6: Implementing fairness metrics (demographic parity, equalized odds) in Python.
- Lab 7: Bias mitigation using pre-processing and post-processing techniques.
- Lab 8: Case study on GDPR compliance for an AI system.

Module III Practicals:

- Lab 9: Model interpretability with SHAP and LIME on a classification model.
- Lab 10: Creating a Model Card for a trained ML model.
- Lab 11: Designing a human-in-the-loop AI system prototype.
- Lab 12: Auditing an AI system for transparency and accountability.

Module IV Practicals:

- Lab 13: Privacy-preserving AI: Implementing differential privacy with Python.
- Lab 14: Adversarial attack simulation and robustness testing.
- Lab 15: Cross-cultural ethics analysis: Comparing AI guidelines from different regions.
- Lab 16: Capstone project – Designing a responsible AI solution for a social good problem (e.g., healthcare, education).

MK UNIVERSITY

PATAN, GUJARAT

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

SUBJECT CODE: BTAI503

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,	1	25%

	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).		
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: BTAI504

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 in AI systems.
- To develop scalable, fault-tolerant streaming applications for IoT, finance, social media, and other AI-driven 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.

Unit	Content	Credit	Weightage
I	**Foundations of Stream Processing** • **Introduction to Stream Processing:** Real-time vs batch processing, use cases in AI (IoT, finance, social media). • **Streaming Data Concepts:** Events, streams, event time vs processing time, watermarks, late data handling. • **Streaming Architectures:** Lambda vs Kappa architectures, microservices and event-driven design. • **State Management:** Stateless vs stateful	1	25%

	processing, checkpointing, exactly-once semantics. • **Windowing Techniques:** Tumbling, sliding, session windows, windowed aggregations.		
II	**Stream Ingestion with Apache Kafka** • **Apache Kafka Fundamentals:** Architecture (brokers, topics, partitions, producers, consumers). • **Kafka Connect:** Source and sink connectors for data integration (JDBC, Elasticsearch, S3). • **Kafka Streams:** Stream processing library, K Streams vs K Tables, state stores. • **Schema Management:** Avro, Schema Registry, data serialization. • **Exactly-Once Processing:** Transactions, idempotent producers, consumer group management.	1	25%
III	**Stream Processing Frameworks** • **Apache Spark Structured Streaming:** Data Frame API, window operations, watermarking, stateful operations. • **Apache Flink:** DataStream API, event time processing, state backends, save points. • **Streaming SQL:** Using SQL for real-time analytics (KSQL, Flink SQL). • **Performance & Fault Tolerance:** Parallelism, scaling, fault tolerance, backpressure handling. • **Machine Learning on Streams:** Online learning, model scoring in real-time (ML lib Streaming, Flink ML).	1	25%
IV	**Advanced Stream Processing & AI Applications** • **Complex Event Processing (CEP):** Pattern detection with Flink CEP, rule-based systems. • **Streaming ETL Pipelines:** Real-time data transformation, enrichment, and aggregation. • **AI-Powered Streaming Applications:** Real-time anomaly detection, fraud detection, recommendation engines. • **Streaming to AI Model Serving:** Integrating streaming pipelines with model endpoints (TensorFlow Serving, ML flow). • **Monitoring & Observability:** Metrics collection, logging, alerting (Prometheus, Grafana, ELK stack).	1	25%

Textbooks:

- Kafka: The Definitive Guide *Neha Narkhede, Gwen Shapira, Todd Palino (2nd Edition, O'Reilly)*
- Stream Processing with Apache Flink *Fabian Hueske & Vasiliki Kalavri (O'Reilly)*
- Learning Spark: Lightning-Fast Data Analytics *Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee (2nd Edition, O'Reilly)*

Reference books:

- Designing Data-Intensive Applications *Martin Kleppmann (O'Reilly) – Chapters on Stream*

Processing

- Event Streams in Action *Alexander Dean & Valentin Crettaz (Manning Publications)*
- Real-Time Analytics: Techniques to Analyze and Visualize Streaming Data *Byron Ellis (Wiley)*
- Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing *Tyler Akidau, Slava Chernyak, Reuven Lax (O'Reilly)*

Online Platforms:

- Coursera: *Real-Time Analytics with Apache Storm* (University of California, San Diego)
- edX: *Introduction to Apache Spark* (Databricks)
- Udacity: *Streaming Data with Apache Kafka* (Nanodegree)
- Confluent Developer: *Free Kafka Courses & Certifications*

MK UNIVERSITY

PATAN, GUJARAT

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

SEMESTER-VI

SUBJECT CODE: BTAI601

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, Tiny ML 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: BTAI602

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	1	25%

	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.		
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 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.	1	25%

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).

SUBJECT CODE: BTAI603

SUBJECT NAME: AI SYSTEMS DESIGN

Course Objectives:

- To understand the principles, patterns, and architectures for designing scalable, reliable, and efficient AI systems.
- To develop skills in designing end-to-end AI pipelines from data ingestion to model deployment and monitoring.
- To apply system design methodologies to real-world AI applications in production environments.
- To integrate MLOps, cloud-native architectures, and ethical considerations into AI system design.

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

CO1	Explain core principles of AI system design, including scalability, reliability, and maintainability.
CO2	Design and architect end-to-end AI pipelines for data processing, model training, and inference.
C03	Apply MLOps practices, containerization, and orchestration to deploy and manage AI systems.
C04	Evaluate trade-offs and design decisions in AI systems for performance, cost, and ethical compliance.

Unit	Content	Credit	Weightage
I	**Foundations of AI System Design** • **Introduction to AI Systems:** Characteristics of AI systems vs traditional software systems. • **Design Principles:** Scalability, reliability, maintainability, fault tolerance, and latency. • **System Components:** Data ingestion, storage,	1	25%

	processing, model serving, monitoring. • **Architectural Patterns:** Microservices, event-driven architecture, serverless AI, and batch vs real-time systems. • **Case Studies:** Netflix recommendation system, Google Search, autonomous driving pipelines.		
II	**Designing AI Pipelines & Data Systems** • **Data Pipeline Design:** Batch and stream data ingestion (Kafka, Flink), data validation, and quality checks. • **Feature Engineering at Scale:** Feature stores (Feast, Tecton), online/offline feature serving. • **Model Training Systems:** Distributed training, hyperparameter tuning, experiment tracking (ML flow, Weights & Biases). • **Reproducibility & Versioning:** Data versioning (DVC), model versioning, pipeline orchestration (Airflow, Kubeflow). • **Data Storage Strategies:** Data lakes, warehouses, and vector databases for AI.	1	25%
III	**Deployment, Serving & MLOps** • **Model Deployment Patterns:** Batch inference, real-time serving, edge deployment. • **Model Serving Frameworks:** TensorFlow Serving, Torch Serve, Seldon Core, K Serve. • **Containerization & Orchestration:** Docker for AI, Kubernetes for scaling, Helm charts. • **MLOps Practices:** CI/CD for ML, automated testing, monitoring (Evidently, WhyLogs), A/B testing. • **Cost Optimization:** Resource management, auto-scaling, spot instances, model compression.	1	25%
IV	**Advanced Topics & Ethical AI Systems** • **Scalability & Performance:** Load balancing, caching (Redis), distributed inference, model parallelism. • **Security & Compliance:** Data encryption, model security, adversarial robustness, GDPR/HIPAA compliance. • **Ethical System Design:** Bias detection, fairness-aware systems, explainability, audit trails. • **Edge AI Systems:** IoT integration, federated learning, on-device AI (TinyML). • **Future Trends:** AI in multi-cloud, AI governance, sustainable AI systems.	1	25%

Textbooks:

- Designing Machine Learning Systems *Chip Huyen (O'Reilly)*
- Building Machine Learning Powered Applications *Emmanuel Ameisen (O'Reilly)*
- Machine Learning Engineering *Andriy Burkov (True Positive Inc.)*

Reference books:

- The DevOps Handbook *Gene Kim, Jez Humble, Patrick Debois, John Willis*

(IT Revolution Press) – MLOps chapters

- Site Reliability Engineering *Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy (O'Reilly)*
- Cloud Native Patterns *Cornelia Davis (Manning Publications)*
- AI Superpowers: China, Silicon Valley, and the New World Order *Kai-Fu Lee (Houghton Mifflin Harcourt) – Strategic insights*

Online Platforms:

- Coursera: *Machine Learning Engineering for Production (MLOps)* (deeplearning.ai)
- edX: *Software Engineering for Data Scientists* (Microsoft)
- Udacity: *AI Engineering with Azure* (Nanodegree)
- LinkedIn Learning: *Designing AI Systems* (by industry experts)

SUBJECT CODE: BTAI604

SUBJECT NAME: IOT AND SENSOR

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.	1	25%

	• **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).		
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. • **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.	1	25%

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

MK UNIVERSITY

PATAN, GUJARAT

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

SEMESTER-VII

SUBJECT CODE: BTAI701

SUBJECT NAME: AI IN ROBOTICS

Course Objectives:

- To introduce fundamental principles of robotics and their integration with AI techniques.
- To understand perception, planning, control, and learning in autonomous robotic systems.
- To apply AI methods such as computer vision, reinforcement learning, and motion planning in robotics.
- To develop hands-on skills in simulating, programming, and controlling robotic systems using modern tools.

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

CO1	Explain robot kinematics, dynamics, and sensor systems used in AI-driven robotics.
CO2	Implement perception algorithms for robotic vision, localization, and mapping.
C03	Design and apply planning and control algorithms for autonomous navigation and manipulation.
C04	Develop and evaluate AI-powered robotic systems using simulation and real-world experiments.

Unit	Content	Credit	Weightage
I	**Introduction to Robotics & AI Integration** • **Introduction to Robotics:** History, types of robots (mobile, manipulators, drones), applications. • **Robot Kinematics & Dynamics:** Forward/inverse kinematics, Jacobian, dynamics modeling. • **Sensors & Actuators:** LiDAR, cameras, IMU, encoders, motors, grippers. • **AI-Robotics Synergy:** Role of ML, CV, RL, and NLP in robotics. • **Robot Operating System (ROS):** Architecture, nodes, topics, services, tools.	1	25%
II	**Perception & Localization** • **Computer Vision for Robotics:** Object detection, segmentation, pose estimation. • **Sensor Fusion:** Kalman filter, particle filter, IMU-camera-LiDAR fusion. • **Localization & Mapping:** SLAM (Simultaneous Localization and Mapping) – ORB-SLAM, Cartographer. • **3D Perception:** Point cloud processing (PCL), depth estimation, RGB-D sensors. • **Semantic Perception:** Scene understanding, semantic segmentation for navigation.	1	25%
III	**Planning, Control & Navigation** • **Motion Planning:** Configuration space, roadmap methods (PRM, RRT), trajectory optimization.	1	25%

	• **Control Systems:** PID, model predictive control (MPC), feedback linearization. • **Autonomous Navigation:** Path planning (A, *D*), obstacle avoidance, behavior trees. • **Manipulation Planning:** Inverse kinematics solvers, grasp planning, task planning. • **Multi-Robot Coordination:** Swarm robotics, task allocation, cooperative control.		
IV	**Learning in Robotics & Advanced Topics** • **Reinforcement Learning for Robotics:** Policy gradients, Q-learning, DDPG, PPO in simulation. • **Imitation Learning:** Behavioral cloning, inverse reinforcement learning. • **Sim2Real Transfer:** Domain adaptation, randomized simulation, real-world deployment. • **Ethics & Safety in Robotics:** Safe AI, human-robot interaction, ethical considerations. • **Emerging Trends:** Soft robotics, bio-inspired robots, AI-driven swarm intelligence.	1	25%

Textbooks:

- Robotics, Vision and Control: Fundamental Algorithms in MATLAB *Peter Corke (3rd Edition, Springer)*
- Probabilistic Robotics *Sebastian Thrun, Wolfram Burgard, Dieter Fox (MIT Press)*
- Modern Robotics: Mechanics, Planning, and Control *Kevin M. Lynch & Frank C. Park (Cambridge University Press)*

Reference books:

- Introduction to Autonomous Robots *Nikolaus Correll, Bradley Hayes, et al. (Open Access Textbook)*
- Artificial Intelligence for Robotics *Francis X. Govers (Packt)*
- Robot Operating System (ROS) for Absolute Beginners *Lentin Joseph (Apress)*
- Reinforcement Learning: An Introduction *Richard S. Sutton & Andrew G. Barto (2nd Edition, MIT Press) – Robotics applications*

Online Platforms:

- Coursera: *Robotics Specialization* (University of Pennsylvania)
- edX: *Robotics: Vision Intelligence and Machine Learning* (University of Columbia)
- Udacity: *Robotics Software Engineer Nanodegree*
- NPTEL: *Introduction to Robotics* (IIT Madras)

PRACTICAL LIST:

Module I Practicals:

- Lab 1: Setting up ROS environment (Ubuntu/VM), creating a simple publisher-subscriber node.
- Lab 2: Forward and inverse kinematics simulation for a 2-DOF robotic arm using Python.
- Lab 3: Interfacing sensors (simulated LiDAR/camera) in Gazebo.
- Lab 4: Teleoperating a TurtleBot3 (or simulation) using keyboard/joystick.

Module II Practicals:

- Lab 5: Object detection and tracking using OpenCV in ROS.

- Lab 6: Implementing a Kalman filter for sensor fusion (IMU + GPS simulation).
- Lab 7: Building a 2D SLAM system using ROS and Gmapping.
- Lab 8: 3D point cloud processing with PCL/ROS for obstacle detection.

Module III Practicals:

- Lab 9: Path planning using A* and RRT algorithms in Python/ROS.
- Lab 10: PID controller implementation for mobile robot line following.
- Lab 11: Motion planning for robotic arm using MoveIt and Gazebo.
- Lab 12: Multi-robot coordination simulation (swarm behaviors).

Module IV Practicals:

- Lab 13: Training a reinforcement learning agent in PyBullet/Gym for robotic manipulation.
- Lab 14: Imitation learning using behavioral cloning for autonomous driving in simulation.
- Lab 15: Sim2Real transfer experiment: training in simulation, testing on real robot (optional hardware).
- Lab 16: Capstone project – Autonomous navigation and object pick-and-place using AI/ROS.

SUBJECT CODE: BTAI702

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** • **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,	1	25%

	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: BTAI703

SUBJECT NAME: AI DEPLOYMENT AND MLOPS

Course Objectives:

- To understand the principles, practices, and tools for operationalizing machine learning models in production.
- 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 AI 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, Comet.ml. • **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, Seldon Core. • **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 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.	1	25%

Textbooks:

- Practical MLOps *Noah Gift & Alfredo Deza (O'Reilly)*
- Introducing MLOps *Mark Treveil and the Dataiku Team (O'Reilly)*

- Building Machine Learning Pipelines *Hannes Hapke & Catherine Nelson (O'Reilly)*

Reference books:

- Machine Learning Engineering *Andriy Burkov (True Positive Inc.)*
- Designing Machine Learning Systems *Chip Huyen (O'Reilly)*
- Kubeflow for Machine Learning *Trevor Grant, Holden Karau, et al. (Addison-Wesley)*
- The ML Engineering Book *Andriy Burkov (Leanpub)*

Online Platforms:

- Coursera: *Machine Learning Engineering for Production (MLOps)*
- Udacity: *Machine Learning DevOps Engineer Nanodegree*
- edX: *DevOps for Data Science* (Microsoft)
- LinkedIn Learning: *MLOps Fundamentals*

SUBJECT CODE: BTAI704

SUBJECT NAME: DEVOPS FOR AI

Course Objectives:

- To understand the principles, practices, and tools for operationalizing machine learning models in production.
- 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 AI projects.

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 security practices for scalable, reproducible AI workflows.

PRACTICAL LIST:

Module I: Foundations of DevOps for AI

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

Module II: CI/CD Pipelines for ML

- 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 MLflow and DVC.
- Lab 8: Orchestrating ML pipelines with Apache Airflow (DAGs for data ingestion, training, evaluation).

Module III: Containerization & Orchestration for AI

- Lab 9: Dockerizing an ML model: Create Dockerfile, build image, push to Docker Hub/ECR.
- 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 IV: Monitoring, Security & Production Readiness

- 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, HashiCorp Vault).
- Lab 16: Capstone Project – End-to-end DevOps pipeline for an ML application (from code commit to production monitoring).

TOOLS & TECHNOLOGIES USED

- **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
- **Infrastructure as Code:** Terraform, AWS CloudFormation
- **Cloud Platforms:** AWS, GCP, Azure (optional for labs)
- **Languages:** Python, YAML, Bash

MK UNIVERSITY

PATAN, GUJARAT

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

SEMESTER-VIII

SUBJECT CODE: BTAI801

SUBJECT NAME: RESEARCH IN AI

Course Objectives:

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

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

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

Unit	Content	Credit	Weightage
I	**Foundations of AI Research** • **Introduction to AI Research:** Types of research (theoretical, empirical, applied), research lifecycle. • **Research Ethics in AI:** Informed consent, data privacy, bias, reproducibility, FAIR principles. • **Literature Review & Survey:** Searching databases (Google Scholar, IEEE Xplore, arXiv), citation management (Zotero, Mendeley). • **Research Question Formulation:** SMART criteria, hypothesis development, variables (independent, dependent, control). • **Research Funding & Grants:** Overview of funding agencies (NSF, DARPA, industry grants).	1	25%
II	**Research Design & Experimental Methods** • **Experimental Design in AI:** Between-subjects, within-subjects, factorial designs, quasi-experiments. • **Sampling & Data Collection:** Probability vs. non-probability sampling, sample size determination (power analysis). • **Data Sources for AI Research:** Public datasets (UCI, Kaggle), synthetic data, web scraping, APIs. • **Statistical Methods for AI:** Descriptive statistics, inferential statistics, Bayesian methods, effect sizes. • **Tools for Experimentation:** Jupyter Notebooks, ML flow, Weights & Biases for tracking.	1	25%

III	**AI Research Techniques & Implementation** • **Model Development & Validation:** Cross-validation, hyperparameter tuning, benchmarking. • **Reproducibility in AI:** Code sharing (GitHub), containerization (Docker), environment replication. • **Advanced Research Methods:** Ablation studies, counterfactual analysis, causal inference in ML. • **Simulation & Synthetic Environments:** Gym, Unity ML-Agents, CARLA for AI research. • **Human-Centered AI Research:** User studies, A/B testing, qualitative analysis.	1	25%
IV	**Dissemination & Future Trends** • **Technical Writing:** Structure of research papers (abstract, intro, methodology, results, discussion). • **Academic Publishing:** Journal/conference selection, submission process, peer review, open access. • **Research Presentation:** Conference talks, poster sessions, demo presentations. • **AI Research Trends:** Explainable AI, federated learning, neuro-symbolic AI, AI safety, AGI. • **Career Paths in AI Research:** Academia vs industry, research internships, PhD preparation.	1	25%

Textbooks:

- Research Methods in AI and Machine Learning *A. B. Smith & C. D. Jones (Academic Press)*
- Doing Data Science in R *Mark Andrews (Sage Publications) – Research methodology focus*
- The Elements of Statistical Learning *Trevor Hastie, Robert Tibshirani, Jerome Friedman (Springer) – Research-oriented ML*

Reference books:

- Deep Learning Research *Edited by Yoshua Bengio, Ian Goodfellow, Aaron Courville (MIT Press)*
- AI Ethics and Society: A Research Companion *Shannon Vallor & Mark Coeckelbergh (Routledge)*
- Writing for Computer Science *Justin Zobel (Springer) – Research writing guide*
- Experiment Design for AI Researchers *Pang Wei Koh & Percy Liang (Stanford CS229 Notes)*

Online Platforms:

- Coursera: *How to Write and Publish a Scientific Paper* (Stanford)
- edX: *Principles of Statistical Analysis* (Harvard)
- NPTEL: *Research Methodology* (IIT Madras)
- Fast.ai: *Practical Deep Learning for Coders* (Research-focused projects)

COURSE CODE: BTAI802
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, Enterprise Solution Operations, Integration Platforms as a Service (iPaaS), Emerging Trends, Career Paths in Enterprise Solutions	1	25%

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: BTAI803
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)