# 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 Engineering

| DIPLOMA (ARTIFICIAL INTELLIGENCE ENGINEERING) SEM-I | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SR NO. | COURSE TYPE | COURSE CODE | COURSE NAME | LECTURE (HRS.)/WEEK | PRACTICAL (HRS.)/WEEK | CREDITS | EXAMINATION | | TOTAL MARKS |
| | | | | | | | INTERNAL | EXTERNAL | |
| 1 | MAJOR | DAI101 | ENGINEERING MATHEMATICS-I | 4 | 0 | 4 | 40 | 60 | 100 |
| 2 | MAJOR | DAI102 | ENGINEERING PHYSICS | 4 | 2 | 6 | 90 | 60 | 150 |
| 3 | MAJOR | DAI103 | ENGINEERING CHEMISTRY | 4 | 2 | 6 | 90 | 60 | 150 |
| 4 | MAJOR | DAI104 | FUNDAMENTALS OF PROGRAMMING | 4 | 2 | 6 | 90 | 60 | 150 |
| 5 | MINOR | DAI105 | WORKSHOP PRACTICE | 0 | 2 | 2 | 50 | 00 | 50 |
| TOTAL | | | | 16 | 8 | 24 | 360 | 240 | 600 |

| DIPLOMA (ARTIFICIAL INTELLIGENCE ENGINEERING) SEM-II | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SR NO. | COURSE TYPE | COURSE CODE | COURSE NAME | LECTURE (HRS.)/WEEK | PRACTICAL (HRS.)/WEEK | CREDITS | EXAMINATION | | TOTAL MARKS |
| | | | | | | | INTERNAL | EXTERNAL | |
| 1 | MAJOR | DAI201 | ENGINEERING MATHEMATICS-II | 4 | 0 | 4 | 40 | 60 | 100 |
| 2 | MAJOR | DAI202 | DATA STRUCTURES & ALGORITHMS | 4 | 2 | 6 | 90 | 60 | 150 |
| 3 | MAJOR | DAI203 | DATA BASE MANAGEMENT SYSTEMS | 4 | 2 | 6 | 90 | 60 | 150 |
| 4 | MINOR | DAI204 | WEB TECHNOLOGIES | 4 | 2 | 6 | 90 | 60 | 150 |
| 5 | SEC | DAI205 | COMMUNICATION SKILL | 2 | 0 | 2 | 00 | 50 | 50 |
| TOTAL | | | | 18 | 6 | 24 | 310 | 290 | 600 |

# MK UNIVERSITY

## PATAN, GUJARAT

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

| DIPLOMA (ARTIFICIAL INTELLIGENCE ENGINEERING) SEM-III | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SR NO. | COURSE TYPE | COURSE CODE | COURSE NAME | LECTURE (HRS.)/ WEEK | PRACTICAL (HRS.)/ WEEK | CREDITS | EXAMINATION | | TOTAL MARKS |
| | | | | | | | INTERNAL | EXTERNAL | |
| 1 | MAJOR | DAI301 | STATISTICS FOR AI | 4 | 2 | 6 | 90 | 60 | 150 |
| 2 | MAJOR | DAI302 | INTRODUCTION TO MACHINE LEARNING | 4 | 2 | 6 | 90 | 60 | 150 |
| 3 | MAJOR | DAI303 | COMPUTER NETWORKS | 4 | 2 | 6 | 90 | 60 | 150 |
| 4 | MINOR | DAI304 | INDUSTRIAL VISIT REPORT | 0 | 2 | 2 | 50 | 00 | 50 |
| 5 | IKS | DAI305 | IKS-ANNCIENT INDIAN ENGINEERING PRACTICE | 0 | 2 | 2 | 50 | 00 | 50 |
| TOTAL | | | | 12 | 10 | 22 | 370 | 180 | 550 |

| DIPLOMA (ARTIFICIAL INTELLIGENCE ENGINEERING) SEM-IV | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SR NO. | COURSE TYPE | COURSE CODE | COURSE NAME | LECTURE (HRS.)/ WEEK | PRACTICAL (HRS.)/ WEEK | CREDITS | EXAMINATION | | TOTAL MARKS |
| | | | | | | | INTERNAL | EXTERNAL | |
| 1 | MAJOR | DAI401 | DEEP LEARNING FUNDAMENTALS | 4 | 2 | 6 | 90 | 60 | 150 |
| 2 | MAJOR | DAI402 | NATURAL LANGUAGE PROCESSING | 4 | 2 | 6 | 90 | 60 | 150 |
| 3 | MAJOR | DAI403 | COMPUTER VISION | 4 | 0 | 4 | 40 | 60 | 100 |
| 4 | MINOR | DAI404 | CLOUD COMPUTING BASICS | 4 | 0 | 4 | 40 | 60 | 100 |
| 5 | VAC | DAI405 | ENVIRONMENTAL SCIENCE | 2 | 0 | 2 | 00 | 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**

| DIPLOMA (ARTIFICIAL INTELLIGENCE ENGINEERING) SEM-V | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SR NO. | COURSE TYPE | COURSE CODE | COURSE NAME | LECTURE (HRS.)/ WEEK | PRACTICAL (HRS.)/WEEK | CREDITS | EXAMINATION | | TOTAL MARKS |
| | | | | | | | INTERNAL | EXTERNAL | |
| 1 | MAJOR | DAI501 | AI ETHICS & GOVERNANCE | 4 | 2 | 6 | 90 | 60 | 150 |
| 2 | MAJOR | DAI502 | REINFORCEMENT LEARNING | 4 | 0 | 4 | 40 | 60 | 100 |
| 3 | MAJOR | DAI503 | BIG DATA ANALYTICS | 4 | 2 | 6 | 90 | 60 | 150 |
| 4 | MINOR | DAI504 | ROBOTICS & AUTOMATION | 4 | 0 | 4 | 40 | 60 | 100 |
| 5 | SEC | DAI505 | MIN PROJECT | 0 | 2 | 2 | 50 | 00 | 50 |
| TOTAL | | | | 16 | 6 | 22 | 310 | 240 | 550 |

| DIPLOMA (ARTIFICIAL INTELLIGENCE ENGINEERING) SEM-VI | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| SR NO. | COURSE TYPE | COURSE CODE | COURSE NAME | LECTURE (HRS.)/ WEEK | PRACTICAL (HRS.)/WEEK | CREDITS | EXAMINATION | | TOTAL MARKS |
| | | | | | | | INTERNAL | EXTERNAL | |
| 1 | MAJOR | DAI601 | AI IN INDUSTRY | 4 | 2 | 6 | 90 | 60 | 150 |
| 2 | MAJOR | DAI602 | AI DEPLOYMENT & MLOPs | 4 | 2 | 6 | 90 | 60 | 150 |
| 3 | MAJOR | DAI603 | IOT & INDUSTRY AUTOMATION | 4 | 2 | 6 | 90 | 60 | 150 |
| 4 | MINOR | DAI604 | DIPLOMA PROJECT | 0 | 6 | 6 | 150 | 00 | 150 |
| TOTAL | | | | 12 | 12 | 24 | 420 | 180 | 600 |

# MK UNIVERSITY

PATAN, GUJARAT

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


## SEMESTER-I

**COURSE CODE: DAI101**

**COURSE NAME: ENGINEERING MATHEMATICS-I**

**Course Objectives:**

- To develop foundational mathematical skills essential for mechanical engineering applications
- To apply algebraic, trigonometric, and calculus concepts to solve engineering problems
- To interpret and analyze data using statistical methods
- To build problem-solving abilities through applied mathematics
- To prepare students for advanced engineering mathematics in subsequent semesters

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

| | |
|---|---|
| CO1 | **Solve** algebraic equations and apply them to engineering problems |
| CO2 | **Apply** trigonometric functions to analyze mechanical systems |
| C03 | **Perform** basic differentiation and integration relevant to engineering applications |
| C04 | **Analyze** data using measures of central tendency and dispersion |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Algebra and Trigonometry**<br>**Topics:**<br>• **Algebra:** Quadratic equations, simultaneous linear equations (2 and 3 variables)<br>• Arithmetic and geometric progressions<br>• **Trigonometry:** Trigonometric ratios, identities, compound angles<br>• Heights and distances (engineering applications)<br>• Complex numbers: basics and operations<br>• **Applications:** Simple harmonic motion, projectile motion, force resolution | 1 | 25% |
| II | **Differential Calculus**<br>**Topics:**<br>• Functions, limits, and continuity<br>• Derivatives: Standard formulas<br>• Rules of differentiation: Product, quotient, chain rule<br>• Applications of derivatives:<br>  ○ Rate of change (velocity, acceleration)<br>  ○ Maxima and minima (optimization problems)<br>  ○ Tangents and normals<br>• Partial differentiation (introduction)<br>• **Applications:** Optimization in design, motion analysis, slope of curves | 1 | 25% |
| III | **Integral Calculus**<br>**Topics:**<br>• Indefinite integrals: Standard formulas | 1 | 25% |

| | | | |
|---|---|---|---|
| | • Methods of integration: Substitution, integration by parts<br>• Definite integrals and properties<br>• Applications of integration:<br>  ○ Area under curves<br>  ○ Volume of solids of revolution<br>  ○ Center of gravity/centroid (basic concepts)<br>• **Applications:** Area calculation in engineering drawings, volume of tanks/containers | | |
| IV | **Statistics and Probability**<br>**Topics:**<br>• **Statistics:** Data classification, frequency distribution<br>• Measures of central tendency: Mean, median, mode<br>• Measures of dispersion: Range, standard deviation, variance<br>• Graphical representation: Histogram, frequency polygon, ogive<br>• **Probability:** Basic concepts, addition and multiplication theorems<br>• **Applications:** Quality control, measurement analysis, manufacturing data interpretation | 1 | 25% |

**Textbooks:**

- Primary: *Engineering Mathematics* — NP Bali & Dr. Manish Goyal
- Primary: *A Textbook of Engineering Mathematics* — B.S. Grewal

**Reference books:**

- *Advanced Engineering Mathematics* — H.K. Das
- *Engineering Mathematics* — D. G. Gupta
- *Basic Technical Mathematics with Calculus* — Allyn J. Washington
- *Mathematics for Mechanical Engineering* — B.V. Ramana

**Online Platforms:**

1. NPTEL Videos: "Basic Course in Mathematics" for engineering
2. Coursera: "Pre-Calculus" by University of California, Irvine

**COURSE CODE: DAI102**

**COURSE NAME: ENGINEERING PHYSICS**

**Course Objectives:**

- To understand fundamental physics principles relevant to mechanical engineering
- To apply physics concepts to solve practical engineering problems
- To develop skills in measurement, experimentation, and data analysis
- To correlate theoretical physics with mechanical systems and applications
- To build foundation for advanced engineering courses

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

| | |
|---|---|
| CO1 | **Apply** mechanics principles to analyze forces, motion, and energy in mechanical systems |
| CO2 | **Explain** thermal physics concepts relevant to heat engines and refrigeration |
| C03 | **Demonstrate** understanding of optics and acoustics in |

| | |
|---|---|
| | engineering contexts |
| C04 | **Perform** measurements using physical instruments and analyze experimental data |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Mechanics & Properties of Matter**<br>**Topics:**<br>• Scalars and vectors, force resolution, moment of force<br>• Laws of motion, friction, work, energy, power<br>• Circular motion, centripetal force<br>• Elasticity: Stress, strain, Hooke's law, Young's modulus<br>• Surface tension and viscosity (basic concepts)<br>• **Applications:** Machine design, material strength, fluid mechanics basics | 1 | 25% |
| II | **Thermal Physics & Thermodynamics**<br>**Topics:**<br>• Heat and temperature, thermal expansion<br>• Calorimetry, specific heat capacity<br>• Laws of thermodynamics (zeroth, first, second)<br>• Heat transfer: conduction, convection, radiation<br>• Kinetic theory of gases (basic)<br>• **Applications:** Heat engines, refrigeration, insulation materials | 1 | 25% |
| III | **Waves, Optics & Acoustics**<br>**Topics:**<br>• Simple harmonic motion, wave motion<br>• Sound: characteristics, intensity, Doppler effect<br>• Ultrasonics and applications<br>• Reflection, refraction, lenses, optical instruments<br>• Fiber optics (basic principles)<br>• **Applications:** Machine vibration, NDT, optical measurements, noise control | 1 | 25% |
| IV | **Modern Physics & Material Science**<br>**Topics:**<br>• Quantum physics basics: photons, matter waves<br>• Lasers: principles, types, applications<br>• Semiconductors: basics<br>• Superconductivity (elementary concepts)<br>• Nanotechnology introduction<br>• **Applications:** Laser machining, sensors, advanced materials | 1 | 25% |

**Textbooks:**

- Primary: *Engineering Physics* — D. R. Khanna & H. N. Srivastava
- Primary: *Engineering Physics* — R. K. Gaur & S. L. Gupta

**Reference books:**

- *Fundamentals of Physics* — Halliday, Resnick & Walker
- *Concepts of Physics* — H. C. Verma

- *Engineering Physics* — M. N. Avadhanulu & P. G. Kshirsagar
- *Practical Physics* — C. L. Arora

**Online Platforms:**

- SWAYAM/NPTEL: "Engineering Physics" courses by IITs/NITs

**PRACTICAL LIST:**

**Section A: Mechanics**

1. Vernier Calipers & Screw Gauge: Measurement of dimensions of given objects
2. Simple Pendulum: Determination of 'g' and study of laws of pendulum
3. Young's Modulus: By Searle's method or cantilever
4. Coefficient of Friction: Using inclined plane
5. Force Table: Verification of law of parallelogram of forces

**Section B: Thermal Physics**

6. Specific Heat Capacity: Of solid/liquid using calorimeter
7. Thermal Conductivity: Of good conductor (Searle's apparatus)
8. Mechanical Equivalent of Heat: Using Joule's calorimeter
9. Coefficient of Linear Expansion: Using optical lever

**Section C: Waves & Optics**

10. Sonometer: Verification of laws of vibrating strings
11. Melde's Experiment: Transverse and longitudinal modes
12. Compound Pendulum: Determination of 'g' and radius of gyration
13. Optical Bench: Focal length of convex lens
14. Prism: Refractive index using spectrometer

**Section D: Modern Physics**

15. LASER: Determination of wavelength using diffraction grating
16. Photoelectric Effect: Verification of Einstein's equation
17. PN Junction Diode: Characteristics
18. Thermistor: Temperature-resistance characteristics

**COURSE CODE: DAI103**

**COURSE NAME: ENGINEERING CHEMISTRY**

**Course Objectives:**

- To provide fundamental knowledge of chemistry relevant to mechanical engineering applications
- To understand material properties, corrosion, fuels, and lubricants from chemical perspective
- To develop skills in chemical analysis, quality control, and material testing
- To correlate chemical principles with mechanical systems and manufacturing processes
- To build foundation for materials science, metallurgy, and environmental engineering

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

| | |
|---|---|
| CO1 | **Explain** water treatment processes for industrial applications |
| CO2 | **Analyze** properties of fuels and lubricants used in mechanical systems |
| C03 | **Identify** corrosion mechanisms and prevention methods |
| C04 | **Apply** principles of electrochemistry to batteries and corrosion control |

| Unit | Content | Credit | Weightage |
|---|---|---|---|

| | | | |
|---|---|---|---|
| I | **Water Chemistry & Treatment**<br>**Topics:**<br>• Water impurities: hardness, alkalinity, pH<br>• Water softening methods: lime-soda, ion exchange<br>• Boiler feed water treatment: scale and sludge formation, prevention<br>• Cooling water treatment<br>• Drinking water standards<br>• **Applications:** Boiler operations, cooling systems, industrial water supply | 1 | 25% |
| II | **Fuels & Combustion**<br>**Topics:**<br>• Classification of fuels: solid, liquid, gaseous<br>• Calorific value determination: bomb calorimeter<br>• Solid fuels: coal analysis (proximate & ultimate)<br>• Liquid fuels: petroleum refining, petrol, diesel, octane/cetane number<br>• Gaseous fuels: LPG, CNG, biogas<br>• Combustion calculations<br>• **Applications:** IC engines, furnaces, power generation | 1 | 25% |
| III | **Lubricants & Corrosion**<br>**Topics:**<br>• Lubrication: mechanisms, types of lubricants<br>• Properties: viscosity index, flash point, pour point<br>• Additives in lubricants<br>• Corrosion: types, mechanisms (electrochemical)<br>• Factors affecting corrosion<br>• Corrosion prevention methods<br>• **Applications:** Machine maintenance, automotive, industrial equipment | 1 | 25% |
| IV | **Engineering Materials & Polymers**<br>**Topics:**<br>• Cement: composition, setting and hardening<br>• Refractories: properties, classification<br>• Polymers: addition, condensation, engineering plastics<br>• Composite materials: introduction<br>• Batteries: primary, secondary, fuel cells<br>• **Applications:** Construction materials, polymers in engineering, energy storage | 1 | 25% |

**Textbooks:**

- Primary: *Engineering Chemistry* — Jain & Jain
- Primary: *Engineering Chemistry* — Dr. O. P. Verma

**Reference books:**

- *A Textbook of Engineering Chemistry* — S. S. Dara & S. S. Umare
- *Engineering Chemistry* — R. P. Mani & K. N. Mishra
- *Chemistry for Engineering Students* — B. S. Jai Prakash & R. Venugopal
- *Applied Chemistry* — H. D. Gesser

**Online Platforms:**

- SWAYAM/NPTEL: "Engineering Chemistry" courses by IITs
- Khan Academy: Chemistry fundamentals
- MERLOT: Chemistry learning materials

**PRACTICAL LIST:**

**Section A: Water Analysis**

1. Determination of Hardness: By EDTA titration method
2. Alkalinity Determination: Using acid-base titration
3. pH Measurement: Using pH meter/universal indicator
4. Chloride Content: By argentometric method

**Section B: Fuels & Lubricants**

5. Viscosity Measurement: Using Ostwald viscometer/Redwood viscometer
6. Flash Point Determination: Using Abel/Pensky Martens apparatus
7. Calorific Value: Bomb calorimeter demonstration
8. Saponification Value: Of oil sample

**Section C: Corrosion & Electrochemistry**

9. Corrosion Rate Measurement: Weight loss method
10. Galvanic Series Determination
11. Electroplating: Copper plating on iron
12. EMF Measurement: Of simple galvanic cell

**Section D: Materials & Polymers**

13. Cement Setting Time: Initial and final setting time
14. Polymer Identification Tests
15. Preparation of Polymer: Phenol-formaldehyde/Bakelite
16. Refractory Properties: Porosity, thermal shock resistance

**COURSE CODE: DAI104**

**COURSE NAME: FUNDAMENTALS OF PROGRAMMING**

**Course Objectives:**

- To introduce the fundamental concepts of programming and problem-solving.
- To develop logical thinking and algorithmic reasoning using structured programming.
- To enable students to write, debug, and execute programs in C language.
- To prepare a strong foundation for advanced programming and AI/ML implementations.
- To familiarize students with good programming practices and documentation.

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

| | |
|---|---|
| CO1 | **Explain** basic programming concepts, algorithms, and flowchart design. |
| CO2 | **Develop** programs using control structures, loops, and functions. |
| C03 | **Implement** arrays, strings, and structures to handle data. |
| C04 | **Apply** pointers and file handling concepts in real-world applications. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Programming & C Basics** | 1 | 25% |

| | | | |
|---|---|---|---|
| | Problem-solving approaches: Algorithms, Flowcharts, Pseudocode<br>- Introduction to C: History, features, structure of C program<br>- C tokens: Keywords, identifiers, constants, operators<br>- Data types: int, float, char, double<br>- Input/output functions: printf(), scanf(), getchar(), putchar()<br>- Operators: Arithmetic, relational, logical, assignment, bitwise<br>- **Applications:** Simple calculator, temperature conversion | | |
| II | **Control Structures & Loops**<br>Decision making: if, if-else, nested if, switch-case<br>- Loops: while, do-while, for loops<br>- Loop control statements: break, continue, goto<br>- Nested loops and pattern printing<br>- **Applications:** Menu-driven programs, number series, star patterns | 1 | 25% |
| III | **Arrays, Strings & Functions**<br>Arrays: 1D and 2D arrays, initialization, traversal<br>- Strings: declaration, standard library functions (strlen, strcpy, strcmp)<br>- Functions: definition, declaration, call by value, call by reference<br>- Recursion: concept, examples (factorial, Fibonacci)<br>- Storage classes: auto, static, register, extern<br>- **Applications:** Matrix operations, sorting, searching, string manipulation | 1 | 25% |
| IV | **Pointers, Structures & File Handling**<br>Pointers: concept, arithmetic, pointer to array, pointer to function<br>- Structures: definition, accessing members, array of structures<br>- File handling: opening, closing, reading, writing files<br>- Dynamic memory allocation: malloc(), calloc(), free()<br>- **Applications:** Student record system, file-based data storage, memory management | 1 | 25% |

**Textbooks:**

- *Let Us C* — Yashwant Kanetkar
- *Programming in ANSI C* — E. Balagurusamy

**Reference books:**

- *The C Programming Language* — Brian W. Kernighan & Dennis M. Ritchie
- *C: The Complete Reference* — Herbert Schildt
- *Problem Solving and Program Design in C* — Jeri R. Hanly & Elliot B. Koffman
- *Programming with C* — Byron Gottfried

**Online Platforms:**

- NPTEL:
  - *Programming in C* by Prof. Satyadev Nandakumar
  - *Problem Solving through Programming in C* by IIT Kharagpur
- Coursera:

- *C for Everyone: Programming Fundamentals* by University of California, Santa Cruz

**PRACTICAL LIST:**

- Section A: Basics & Control Structures

- Write a program to find the sum and average of three numbers.
- Write a program to swap two numbers using a third variable and without using a third variable.
- Write a program to check whether a number is even or odd.
- Write a program to find the largest among three numbers using nested if-else.
- Write a program to display the multiplication table of a given number using a for loop.
- Write a program to print the Fibonacci series up to n terms.
- Write a program to check whether a number is prime or not.

- Section B: Arrays & Strings

- Write a program to find the largest and smallest element in an array.
- Write a program to add two matrices.
- Write a program to multiply two matrices.
- Write a program to sort an array using bubble sort.
- Write a program to search an element in an array using linear search.
- Write a program to reverse a string without using library functions.
- Write a program to check whether a string is palindrome or not.

- Section C: Functions & Recursion

- Write a function to calculate factorial of a number using recursion.
- Write a function to check whether a number is Armstrong or not.
- Write a program to find GCD of two numbers using recursion.
- Write a program to swap two numbers using call by reference.
- Write a program to demonstrate the use of static variable.

- Section D: Pointers, Structures & Files

- Write a program to demonstrate pointer arithmetic.
- Write a program to store student information (roll, name, marks) using structures and display it.
- Write a program to read from a file and count vowels, consonants, and digits.
- Write a program to copy contents from one file to another.
- Write a program to dynamically allocate memory for an array and find its sum.

# MK UNIVERSITY

PATAN, GUJARAT

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

## SEMESTER-II

**COURSE CODE: DAI201**

**COURSE NAME: ENGINEERING MATHEMATICS-II**

**Course Objectives:**

- To build advanced mathematical skills for engineering problem-solving
- To apply differential equations to model mechanical systems
- To understand vector algebra and its applications in mechanics
- To develop skills in numerical methods for engineering computations
- To prepare mathematical foundation for subjects like Mechanics, Thermodynamics, and Machine Design

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

| | |
|---|---|
| CO1 | **Solve** ordinary differential equations relevant to engineering systems |
| CO2 | **Apply** vector algebra to analyze forces and motions in 3D space |
| C03 | **Perform** numerical computations using interpolation, differentiation, and integration methods |
| C04 | **Analyze** data using probability distributions and statistical methods |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Differential Equations**<br>**Topics:**<br>• First order differential equations: variable separable, homogeneous, exact<br>• Linear differential equations of first order<br>• Applications: Newton's law of cooling, growth and decay, simple circuits<br>• Second order linear differential equations with constant coefficients<br>• Complementary function and particular integral methods<br>• **Applications:** Spring-mass systems, electrical circuits, vibration analysis | 1 | 25% |
| II | **Vector Algebra & 3D Geometry**<br>**Topics:**<br>• Vectors: dot product, cross product, scalar triple product<br>• Vector differentiation<br>• Gradient, divergence, curl (basic concepts)<br>• Lines and planes in 3D space<br>• Direction cosines and ratios<br>• **Applications:** Force analysis, moment of force, work done by force | 1 | 25% |
| III | **Numerical Methods**<br>**Topics:**<br>• Solution of algebraic equations: Bisection method, | 1 | 25% |

| | | | |
|---|---|---|---|
| | Newton-Raphson method<br>• Interpolation: Newton's forward and backward difference formulas<br>• Numerical differentiation and integration<br>• Trapezoidal rule and Simpson's rules<br>• **Applications:** Root finding in design equations, area calculations, data analysis | | |
| IV | **Probability & Laplace Transforms**<br>**Topics:**<br>• **Probability:** Basic concepts, addition and multiplication theorems<br>• Random variables, probability distributions (Binomial, Poisson, Normal)<br>• Mean, variance, standard deviation<br>• **Laplace Transforms:** Definition, basic transforms<br>• Properties: linearity, shifting, differentiation<br>• Application to differential equations<br>• **Applications:** Quality control, reliability analysis, system dynamics | 1 | 25% |

**Textbooks:**

- Primary: *Higher Engineering Mathematics* — B.S. Grewal
- Primary: *Engineering Mathematics* — NP Bali & Dr. Manish Goyal

**Reference books:**

- *Advanced Engineering Mathematics* — H.K. Das
- *Numerical Methods* — S.S. Sastry
- *Probability and Statistics for Engineers* — Dr. J. Ravichandran
- *Mathematical Methods* — B.V. Ramana

**Online Platforms:**

- NPTEL Videos: "Differential Equations for Engineers"
- Khan Academy: Complete probability and statistics
- MIT OCW: "Single Variable Calculus" continuation
  Coursera: "Introduction to Numerical Methods"

**COURSE CODE: DAI202**

**COURSE NAME: DATA STRUCTURES AND ALGORITHMS**

**Course Objectives:**

- To introduce fundamental data structures and their implementation in C/C++.
- To develop problem-solving skills using appropriate data structures.
- To analyze algorithm efficiency using time and space complexity.
- To prepare students for advanced topics in AI, machine learning, and software development.
- To enable students to implement and apply data structures in real-world scenarios.

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

| | |
|---|---|
| CO1 | **Explain** the concepts of arrays, linked lists, stacks, and queues. |
| CO2 | **Implement** trees and graphs for hierarchical and network data representation. |
| C03 | **Apply** sorting and searching algorithms to solve computational |

| | |
|---|---|
| | problems. |
| C04 | **Analyze** algorithm efficiency and select appropriate data structures for given problems. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Data Structures & Linear Lists**<br>Introduction: Data types, ADTs, time and space complexity<br>- Arrays: 1D, 2D, operations, applications<br>- Linked Lists: Singly, doubly, circular linked lists<br>- Stacks: Array and linked list implementation, applications<br>- Queues: Simple, circular, priority queues<br>- **Applications:** Expression evaluation, job scheduling, memory management | 1 | 25% |
| II | **Trees & Hierarchical Data Structures**<br>Trees: Terminology, binary trees, traversal (inorder, preorder, postorder)<br>- Binary Search Trees (BST): Insertion, deletion, searching<br>- AVL Trees: Rotations, balancing<br>- Heaps: Min-heap, max-heap, heap operations<br>- **Applications:** File systems, database indexing, priority queues | 1 | 25% |
| III | **Graphs & Hashing**<br>Graphs: Terminology, representation (adjacency matrix, list)<br>- Graph traversals: BFS, DFS<br>- Hashing: Hash functions, collision resolution (chaining, open addressing)<br>- **Applications:** Social networks, GPS navigation, spell checking | 1 | 25% |
| IV | **Sorting, Searching & Algorithm Design**<br>Sorting algorithms: Bubble, selection, insertion, merge, quick, heap sort<br>- Searching algorithms: Linear, binary, interpolation search<br>- Algorithm design techniques: Greedy, divide and conquer, dynamic programming (intro)<br>- **Applications:** Data analysis, AI search algorithms, database query optimization | 1 | 25% |

**Textbooks:**

- *Data Structures and Algorithms in C* — Reema Thareja
- *Data Structures Using C* — Aaron M. Tenenbaum

**Reference books:**

- *Introduction to Algorithms* — Thomas H. Cormen et al.
- *Data Structures and Algorithm Analysis in C* — Mark Allen Weiss
- *The Algorithm Design Manual* — Steven S. Skiena
- *Data Structures Through C* — Yashavant Kanetkar

**Online Platforms:**

- NPTEL:
  1. *Data Structures and Algorithms* by Prof. Naveen Garg (IIT Delhi)

2. *Programming, Data Structures and Algorithms* by IIT Madras

- Coursera:
  1. *Data Structures and Algorithms* by University of California, San Diego
  2. *Algorithms Specialization* by Stanford University

**PRACTICAL LIST:**

- Section A: Linear Data Structures
- Implement stack using array and linked list with push, pop, display.
- Implement queue using array and linked list with enqueue, dequeue, display.
- Implement circular queue with basic operations.
- Implement singly linked list with insertion, deletion, traversal.
- Implement doubly linked list with insertion, deletion, display.
- Implement polynomial addition using linked list.
- Section B: Trees & Heaps
- Implement binary tree traversal (inorder, preorder, postorder) recursively.
- Implement binary search tree with insertion, deletion, searching.
- Implement heap (max-heap) with insertion and deletion.
- Implement AVL tree with rotations (LL, RR, LR, RL).
- Implement expression tree from postfix expression and evaluate it.
- Section C: Graphs & Hashing
- Implement graph using adjacency matrix and perform BFS and DFS.
- Implement graph using adjacency list and perform BFS and DFS.
- Implement hashing with separate chaining for collision resolution.
- Implement hashing with linear probing for collision resolution.
- Find shortest path in unweighted graph using BFS.
- Section D: Sorting & Searching Algorithms
- Implement bubble sort, selection sort, and insertion sort.
- Implement merge sort and quick sort.
- Implement heap sort.
- Implement binary search recursively and iteratively.
- Implement linear search and count occurrences in an array.
- Implement a menu-driven program to compare sorting algorithm performance.

# MK UNIVERSITY

**PATAN, GUJARAT**

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

**COURSE CODE: DAI203**

**COURSE NAME: DATABASE MANAGEMENT SYSTEMS**

**Course Objectives:**

- To introduce fundamental concepts of databases, data models, and DBMS architecture.
- To develop skills in designing databases using ER modeling and normalization.
- To enable students to write SQL queries for data definition, manipulation, and control.
- To understand transaction management, concurrency control, and database security.
- To prepare students for real-world database applications in AI and data science.

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

| | |
|---|---|
| CO1 | **Explain** DBMS architecture, data models, and ER diagrams. |
| CO2 | **Design** normalized relational databases using ER modeling and normalization. |
| C03 | **Write** SQL queries for data definition, manipulation, and retrieval. |
| C04 | **Analyze** transaction management, concurrency control, and database security mechanisms. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to DBMS & Data Modeling**<br>Database concepts: Data, database, DBMS, advantages<br>- DBMS architecture: 3-tier architecture, data independence<br>- Data models: Hierarchical, network, relational, ER model<br>- ER diagrams: Entities, attributes, relationships, keys<br>- **Applications:** Banking, inventory, student management systems | 1 | 25% |
| II | **Relational Database Design & SQL**<br>Relational model: Relations, tuples, attributes, keys<br>- Relational algebra: Operations, selection, projection, join<br>- SQL: DDL, DML, DCL commands<br>- SQL queries: SELECT, WHERE, GROUP BY, HAVING, ORDER BY<br>- Joins: Inner, outer, self, cross joins<br>- **Applications:** Querying real databases, report generation | 1 | 25% |
| III | **Normalization & Advanced SQL**<br>Normalization: 1NF, 2NF, 3NF, BCNF, functional dependencies<br>- Advanced SQL: Subqueries, views, indexes, sequences<br>- PL/SQL basics: Cursors, procedures, functions, triggers<br>- **Applications:** Data integrity, performance tuning, automation | 1 | 25% |
| IV | **Transaction Management & Security**<br>Transaction concepts: ACID properties, states<br>- Concurrency control: Locks, two-phase locking, deadlock<br>- Database recovery: Log-based recovery, checkpoints<br>- Database security: Authentication, authorization, encryption | 1 | 25% |

| | | | |
|---|---|---|---|
| | - **Applications:** Banking transactions, multi-user systems, secure databases | | |

**Textbooks:**

- *Database System Concepts* — Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- *Fundamentals of Database Systems* — Ramez Elmasri, Shamkant B. Navathe

**Reference books:**

- *SQL: The Complete Reference* — James R. Groff, Paul N. Weinberg
- *Database Management Systems* — Raghu Ramakrishnan, Johannes Gehrke
- *An Introduction to Database Systems* — C.J. Date
- *Oracle PL/SQL Programming* — Steven Feuerstein

**Online Platforms:**

- NPTEL:
  1. *Database Management System* by Prof. P.K. Biswas (IIT Kharagpur)
  2. *Introduction to Database Systems* by IIT Madras
- Coursera:
  1. *Databases and SQL for Data Science* by IBM
  2. *SQL for Data Science* by University of California, Davis

**PRACTICAL LIST:**

**Section A: Database Design & ER Modeling**

1. Draw ER diagrams for:
   - Library Management System
   - Hospital Management System
   - Online Shopping System
2. Convert ER diagrams into relational schemas.
3. Create tables using DDL commands with primary keys, foreign keys, and constraints.

**Section B: SQL Queries – Basic to Intermediate**

4. Create a database for Student Management System with tables: Student, Course, Enrollment.
5. Insert sample data into all tables.
6. Write SQL queries to:
   - Retrieve all students enrolled in a specific course.
   - Find the average marks of students.
   - List students with marks above average.
   - Count number of students per course.
7. Implement different types of joins (inner, left, right, full) on the database.

**Section C: Advanced SQL & PL/SQL**

8. Create views for:
   - Students with marks > 75%
   - Course-wise student count
9. Write SQL queries using:
   - Subqueries (nested, correlated)
   - GROUP BY, HAVING, ORDER BY
10. Create indexes on frequently queried columns.
11. Write a PL/SQL procedure to calculate total marks of a student.
12. Create a trigger to log changes in the Student table.

**Section D: Normalization & Transaction Control**

13. Take an unnormalized table and normalize it to 3NF.
14. Demonstrate transaction control commands:

- COMMIT, ROLLBACK, SAVEPOINT

15. Simulate concurrent transactions and demonstrate locking.
16. Implement user roles and grant/revoke permissions.
17. Perform backup and recovery operations on a sample database.

**COURSE CODE: DAI204**

**COURSE NAME: WEB TECHNOLOGIES**

**Course Objectives:**

- To introduce the fundamentals of web technologies, internet protocols, and web architecture.
- To develop skills in designing responsive web pages using HTML, CSS, and JavaScript.
- To enable students to create dynamic web applications using front-end and back-end technologies.
- To understand web APIs, AJAX, and integration with databases.
- To prepare students for building web-based AI applications and interactive dashboards.

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

| | |
|---|---|
| CO1 | **Explain** web architecture, internet protocols, and client-server models. |
| CO2 | **Design** responsive and interactive web pages using HTML, CSS, and JavaScript. |
| C03 | **Develop** dynamic web applications using front-end frameworks and back-end scripting. |
| C04 | **Integrate** databases and web APIs to create data-driven web applications. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Web Fundamentals & HTML5**<br>Internet basics: WWW, HTTP/HTTPS, DNS, web browsers<br>- Web architecture: Client-server model, static vs dynamic websites<br>- HTML5: Structure, tags, forms, multimedia, semantic elements<br>- HTML5 APIs: Geolocation, local storage, canvas basics<br>- **Applications:** Portfolio websites, forms, simple web pages | 1 | 25% |
| II | **CSS3 & Responsive Web Design**<br>CSS3: Selectors, box model, positioning, flexbox, grid<br>- Responsive design: Media queries, mobile-first approach<br>- CSS frameworks: Bootstrap basics<br>- Animations and transitions<br>- **Applications:** Responsive layouts, navigation bars, styled forms | 1 | 25% |
| III | **JavaScript & Front-End Development**<br>JavaScript basics: Syntax, DOM manipulation, events<br>- ES6 features: let/const, arrow functions, promises<br>- AJAX and Fetch API for async requests<br>- Front-end frameworks: Introduction to React.js/Vue.js<br>- **Applications:** Interactive UI, form validation, dynamic content loading | 1 | 25% |

| | | | |
|---|---|---|---|
| IV | **Back-End Basics & Web APIs**<br>Back-end introduction: Node.js, Express.js basics<br>- RESTful APIs: CRUD operations, JSON handling<br>- Database integration: MongoDB/MySQL with Node.js<br>- Authentication: JWT, session-based auth basics<br>- **Applications:** Simple web apps, API integration, user authentication | 1 | 25% |

**Textbooks:**

- *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML and AJAX* — Uttam K. Roy
- *HTML & CSS: Design and Build Websites* — Jon Duckett

**Reference books:**

- *Eloquent JavaScript* — Marijn Haverbeke
- *Learning PHP, MySQL & JavaScript* — Robin Nixon
- *Node.js Design Patterns* — Mario Casciaro
- *React: Up & Running* — Stoyan Stefanov

**Online Platforms:**

- NPTEL:
  1. *Web Technologies* by Prof. D. Janakiram (IIT Madras)
  2. *Introduction to Modern Application Development* by IIT Madras
- Coursera:
  1. *HTML, CSS, and JavaScript for Web Developers* by Johns Hopkins University
  2. *Web Design for Everybody* by University of Michigan

**PRACTICAL LIST:**

**Section A: HTML5 & CSS3**

1. Create a personal portfolio website using HTML5 and CSS3.
2. Design a responsive registration form with validation using HTML5 form elements.
3. Build a webpage layout using CSS Grid and Flexbox.
4. Implement a navigation bar with dropdown menus using CSS.
5. Create a photo gallery with hover effects and transitions.

**Section B: JavaScript & DOM Manipulation**

6. Create a calculator using JavaScript.
7. Build a to-do list application with add, edit, delete, and mark-as-done features.
8. Develop a digital clock with date and time display.
9. Implement form validation (email, password, phone) using JavaScript.
10. Create a slideshow/carousel using JavaScript and CSS.

**Section C: Responsive Design & Bootstrap**

11. Convert a given webpage into a mobile-responsive layout using media queries.
12. Design a landing page using Bootstrap 5 (navbar, cards, modal, forms).
13. Create a responsive dashboard layout with Bootstrap grid system.
14. Build a blog template with Bootstrap components.

**Section D: Back-End Integration & APIs**

15. Create a simple REST API using Node.js and Express that returns JSON data.
16. Build a weather app using a public API (e.g., OpenWeatherMap) with Fetch/AJAX.
17. Develop a note-taking app with local storage (CRUD operations).

## SEMESTER-III

# MK UNIVERSITY

**PATAN, GUJARAT**

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

**COURSE CODE: DAI301**

**COURSE NAME: STATISTICS FOR AI**

**Course Objectives:**

- To introduce fundamental statistical concepts essential for data analysis and AI.
- To develop skills in descriptive statistics, probability, and inferential statistics.
- To enable students to apply statistical methods for data summarization, visualization, and hypothesis testing.
- To prepare students for advanced AI/ML topics requiring statistical foundations.
- To foster the ability to interpret and communicate statistical results effectively.

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

| | |
|---|---|
| CO1 | **Calculate** and interpret descriptive statistics for data summarization. |
| CO2 | **Apply** probability concepts and distributions to model real-world data. |
| C03 | **Perform** hypothesis testing and confidence interval estimation. |
| C04 | **Analyze** relationships between variables using correlation and regression. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Descriptive Statistics & Data Visualization**<br>Introduction to statistics: Role in AI, types of data<br>- Measures of central tendency: Mean, median, mode<br>- Measures of dispersion: Range, variance, standard deviation, quartiles<br>- Data visualization: Histograms, box plots, scatter plots, bar charts<br>- **Applications:** Exploratory Data Analysis (EDA), outlier detection | 1 | 25% |
| II | **Probability & Probability Distributions**<br>Probability basics: Sample space, events, axioms<br>- Conditional probability, Bayes' theorem<br>- Random variables: Discrete and continuous<br>- Probability distributions: Binomial, Poisson, Normal, Uniform<br>- **Applications:** Risk assessment, classification models, simulation | 1 | 25% |
| III | **Inferential Statistics**<br>Sampling methods and sampling distributions<br>- Central Limit Theorem<br>- Estimation: Point and interval estimation<br>- Hypothesis testing: Z-test, t-test (one-sample, two-sample)<br>- **Applications:** A/B testing, quality control, model validation | 1 | 25% |
| IV | **Correlation, Regression & ANOVA**<br>Correlation: Pearson, Spearman rank correlation<br>- Simple linear regression: Model fitting, interpretation, $R^2$<br>- Multiple linear regression (introduction) | 1 | 25% |

| | | | |
|---|---|---|---|
| | - ANOVA: One-way ANOVA, F-test<br>- **Applications:** Predictive modeling, feature selection, experiment analysis | | |

**Textbooks:**

- *Statistics for Management and Economics* — Gerald Keller
- *Introductory Statistics* — Sheldon M. Ross

**Reference books:**

- *The Elements of Statistical Learning* — Trevor Hastie, Robert Tibshirani, Jerome Friedman
- *Practical Statistics for Data Scientists* — Peter Bruce & Andrew Bruce
- *Naked Statistics* — Charles Wheelan
- *Statistics in Plain English* — Timothy C. Urdan

**Online Platforms:**

- NPTEL:
  1. *Probability and Statistics* by Prof. S. K. Gupta (IIT Roorkee)
  2. *Statistics for Data Science* by IIT Madras
- Coursera:
  1. *Statistics with Python* by University of Michigan
  2. *Introduction to Statistics* by Stanford University

**PRACTICAL LIST:**

Section A: Descriptive Statistics & Visualization

1. Calculate mean, median, mode, variance, and standard deviation for a given dataset using Python/R.
2. Create a histogram and box plot to visualize data distribution and identify outliers.
3. Generate scatter plots and interpret relationships between two variables.
4. Build a frequency distribution table and cumulative frequency curve.

Section B: Probability & Distributions

5. Simulate coin tosses/dice rolls and compute empirical probabilities.
6. Calculate probabilities using Binomial and Poisson distributions.
7. Use Normal distribution to find probabilities and percentiles (Z-scores).
8. Apply Bayes' theorem to solve a classification problem (e.g., spam detection).

Section C: Inferential Statistics

9. Perform a one-sample t-test to compare sample mean with population mean.
10. Conduct a two-sample t-test to compare means of two independent groups.
11. Calculate confidence intervals for population mean and proportion.
12. Simulate the Central Limit Theorem using random sampling.

Section D: Correlation, Regression & ANOVA

13. Calculate Pearson and Spearman correlation coefficients and interpret results.
14. Fit a simple linear regression model and interpret slope, intercept, and $R^2$.
15. Perform one-way ANOVA to test differences among group means.
16. Use multiple linear regression to predict an outcome using two or more predictors.

# MK UNIVERSITY

PATAN, GUJARAT

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

**COURSE CODE: DAI302**

**COURSE NAME: INTRODUCTION TO MACHINE LEARNING**

**Course Objectives:**

- To introduce fundamental concepts, types, and applications of machine learning.
- To develop skills in data preprocessing, feature engineering, and model evaluation.
- To enable students to implement and evaluate supervised and unsupervised learning algorithms.
- To prepare students for advanced ML topics and real-world AI applications.
- To foster ethical awareness and responsible use of ML systems.

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

| | |
|---|---|
| CO1 | **Explain** types of ML, data preprocessing steps, and model evaluation metrics. |
| CO2 | **Implement** supervised learning algorithms for classification and regression. |
| C03 | **Apply** unsupervised learning techniques for clustering and dimensionality reduction. |
| C04 | **Evaluate** model performance and interpret results using real datasets. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to ML & Data Preprocessing**<br>What is ML? Types: Supervised, Unsupervised, Reinforcement Learning<br>- ML workflow: Problem definition, data collection, preprocessing, modeling, evaluation<br>- Data preprocessing: Handling missing values, encoding, normalization, train-test split<br>- Feature engineering and selection<br>- **Applications:** Spam detection, recommendation systems, customer segmentation | 1 | 25% |
| II | **Supervised Learning – Regression & Classification**<br>Linear Regression: Simple and multiple, assumptions, evaluation metrics (MSE, $R^2$)<br>- Logistic Regression: Binary classification, sigmoid function, confusion matrix<br>- Decision Trees: Splitting criteria, pruning, interpretation<br>- Model evaluation: Accuracy, precision, recall, F1-score, ROC-AUC<br>- **Applications:** House price prediction, customer churn, medical diagnosis | 1 | 25% |
| III | **Advanced Supervised Learning & Ensemble Methods**<br>k-Nearest Neighbors (k-NN): Distance metrics, k-value selection<br>- Support Vector Machines (SVM): Linear and kernel SVM, hyperparameters<br>- Ensemble methods: Bagging (Random Forest), Boosting (AdaBoost, Gradient Boosting)<br>- Hyperparameter tuning: Grid Search, Random Search | 1 | 25% |

| | | | |
|---|---|---|---|
| | - **Applications:** Image classification, fraud detection, credit scoring | | |
| IV | **Unsupervised Learning & Model Deployment Basics**<br>Clustering: k-Means, Hierarchical clustering, evaluation (silhouette score)<br>- Dimensionality reduction: PCA, t-SNE basics<br>- Introduction to model deployment: Saving models (pickle, joblib), Flask basics<br>- Ethical considerations: Bias, fairness, interpretability in ML<br>- **Applications:** Market segmentation, anomaly detection, visualization | 1 | 25% |

**Textbooks:**

- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — Aurélien Géron
- *Introduction to Machine Learning with Python* — Andreas C. Müller & Sarah Guido

**Reference books:**

- *Pattern Recognition and Machine Learning* — Christopher M. Bishop
- *Machine Learning Yearning* — Andrew Ng
- *The Hundred-Page Machine Learning Book* — Andriy Burkov
- *Python Machine Learning* — Sebastian Raschka & Vahid Mirjalili

**Online Platforms:**

- NPTEL:
  - *Introduction to Machine Learning* by Prof. Balaraman Ravindran (IIT Madras)
  - *Machine Learning* by IIT Kharagpur
- Coursera:
  - *Machine Learning* by Andrew Ng (Stanford University)
  - *Applied Data Science with Python* by University of Michigan

**PRACTICAL LIST:**

Section A: Data Preprocessing & EDA

1. Load a dataset (e.g., Titanic, Iris) and perform exploratory data analysis (EDA).
2. Handle missing values and encode categorical variables.
3. Normalize/standardize features and split data into train/test sets.
4. Visualize feature distributions and correlations using seaborn/matplotlib.

Section B: Supervised Learning – Regression & Classification

5. Implement Simple Linear Regression to predict housing prices (Boston Housing dataset).
6. Implement Logistic Regression for binary classification (Titanic survival prediction).
7. Build and visualize a Decision Tree classifier and evaluate using confusion matrix.
8. Compare model performance (accuracy, precision, recall) for different classifiers.

Section C: Advanced Supervised Learning

9. Implement k-NN classifier for Iris dataset and find optimal k using cross-validation.
10. Train an SVM classifier with different kernels (linear, RBF) and compare results.
11. Implement Random Forest classifier and analyze feature importance.
12. Perform hyperparameter tuning using GridSearchCV for any classifier.

Section D: Unsupervised Learning & Mini-Project

13. Apply k-Means clustering for customer segmentation (Mall Customer dataset).
14. Perform PCA for dimensionality reduction and visualize clusters in 2D.
15. Build an end-to-end ML pipeline: from data loading to model deployment using Flask.

16. Mini-Project: Develop a complete ML model (e.g., spam classifier, sales predictor) with a detailed report.

**COURSE CODE: DAI303**

**COURSE NAME: COMPUTER NETWORKS**

**Course Objectives:**

- To introduce fundamental concepts, architectures, and models of computer networks.
- To develop understanding of network protocols, data communication, and network devices.
- To enable students to design, configure, and troubleshoot basic network setups.
- To prepare students for understanding network-related aspects in AI, IoT, and cloud systems.
- To foster awareness of network security, performance, and ethical considerations.

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

| | |
|---|---|
| CO1 | **Explain** network models, topologies, and transmission media. |
| CO2 | **Describe** the functions and protocols of data link, network, and transport layers. |
| C03 | **Configure** basic network services and understand routing principles. |
| C04 | **Analyze** network security threats and apply basic security measures. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Computer Networks & Physical Layer**<br>Basics: Components, network types (LAN, MAN, WAN), topologies<br>- Network models: OSI model, TCP/IP model, layers and functions<br>- Transmission media: Guided (twisted pair, coaxial, fiber) and unguided (radio, microwave)<br>- Switching techniques: Circuit, packet, message switching<br>- **Applications:** Home networks, office networks, internet connectivity | 1 | 25% |
| II | **Data Link Layer & Media Access Control**<br>Functions of data link layer: Framing, error detection (CRC), flow control<br>- MAC protocols: CSMA/CD, CSMA/CA, Ethernet (IEEE 802.3)<br>- Network devices: Hubs, switches, bridges<br>- VLANs (Virtual LANs) – basic concepts<br>- **Applications:** Ethernet LANs, Wi-Fi networks, network segmentation | 1 | 25% |
| III | **Network Layer & Routing**<br>IPv4 addressing: Classes, subnetting, CIDR<br>- IPv6 basics<br>- Routing algorithms: Static vs dynamic routing, distance vector, link state<br>- Routers and routing tables<br>- ARP, ICMP, DHCP basics | 1 | 25% |

| | | | |
|---|---|---|---|
| | - **Applications:** IP addressing, routing in networks, internet connectivity | | |
| IV | **Transport Layer, Application Layer & Network Security**<br>Transport layer: TCP and UDP, ports, sockets, congestion control<br>- Application layer protocols: HTTP, HTTPS, DNS, SMTP, FTP<br>- Network security: Threats (malware, DoS, phishing), firewalls, encryption basics<br>- Introduction to IoT networks and cloud networking<br>- **Applications:** Web browsing, email, secure communication, IoT devices | 1 | 25% |

**Textbooks:**

- *Data Communications and Networking* — Behrouz A. Forouzan
- *Computer Networks* — Andrew S. Tanenbaum

**Reference books:**

- *TCP/IP Illustrated* — W. Richard Stevens
- *Computer Networking: A Top-Down Approach* — James F. Kurose & Keith W. Ross
- *Network Security Essentials* — William Stallings
- *CCNA Routing and Switching Complete Study Guide* — Todd Lammle

**Online Platforms:**

- NPTEL:
  1. *Computer Networks* by Prof. Sujoy Ghosh (IIT Kharagpur)
  2. *Introduction to Computer Networks* by IIT Bombay
- Coursera:
  1. *The Bits and Bytes of Computer Networking* by Google
  2. *Computer Networks* by University of Washington

**PRACTICAL LIST:**

Section A: Network Basics & Configuration

1. Identify network components and create a simple LAN topology diagram.
2. Configure IP addresses on multiple devices and verify connectivity using ping.
3. Use basic network commands: ipconfig, ifconfig, ping, tracert/traceroute.
4. Create a crossover cable and test connectivity between two PCs.

Section B: Data Link Layer & Switching

5. Configure a VLAN on a switch and assign ports to VLANs.
6. Analyze Ethernet frames using Wireshark.
7. Simulate CSMA/CD in a network simulator (e.g., Cisco Packet Tracer).
8. Configure port security on a switch.

Section C: Network Layer & Routing

9. Perform subnetting exercises: Calculate subnets, network addresses, broadcast addresses.
10. Configure static routing between two or more routers.
11. Configure DHCP server on a router and assign IPs dynamically.
12. Analyze IP packets using Wireshark.

Section D: Transport/Application Layer & Security

13. Configure a simple web server and access it via HTTP/HTTPS.
14. Use nslookup and dig commands for DNS query analysis.
15. Set up a basic firewall rule (using Windows Firewall or iptables basics).

16. Capture and analyze TCP/UDP packets using Wireshark.
17. Mini-project: Design and simulate a small office network with multiple VLANs, routing, and basic security.

# MK UNIVERSITY

PATAN, GUJARAT

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


## SEMESTER-IV

**COURSE CODE: DAI401**

**COURSE NAME: DEEP LEARNING FUNDAMENTALS**

**Course Objectives:**

- To introduce the fundamental concepts, architectures, and mathematical foundations of deep learning.
- To develop proficiency in implementing, training, and evaluating neural networks using modern frameworks.
- To enable students to design and apply convolutional and recurrent neural networks to real-world problems.
- To prepare students for advanced topics in computer vision, natural language processing, and generative AI.
- To foster awareness of ethical considerations, model interpretability, and best practices in deep learning.

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

| | |
|---|---|
| CO1 | **Explain** the architecture, components, and training dynamics of artificial neural networks. |
| CO2 | **Implement** and optimize convolutional neural networks (CNNs) for image-based tasks. |
| C03 | **Design** and apply recurrent neural networks (RNNs) and LSTMs for sequential data. |
| C04 | **Evaluate** deep learning models, apply regularization techniques, and understand ethical implications. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Neural Networks**<br>What is Deep Learning? Evolution, key milestones, and applications<br>- Artificial neurons, perceptrons, activation functions (Sigmoid, Tanh, ReLU, Softmax)<br>- Feedforward neural networks: Architecture, forward propagation, loss functions<br>- Backpropagation, gradient descent, optimizers (SGD, Adam, RMSprop)<br>- **Applications:** Digit recognition, binary classification, simple regression | 1 | 25% |
| II | **Convolutional Neural Networks (CNNs)**<br>CNN fundamentals: Convolutional layers, pooling, padding, stride<br>- CNN architectures: LeNet, AlexNet, VGG, ResNet (overview)<br>- Transfer learning and fine-tuning using pre-trained models<br>- Data augmentation and regularization techniques | 1 | 25% |

| | | | |
|---|---|---|---|
| | (Dropout, BatchNorm)<br>- **Applications:** Image classification, object detection, medical image analysis | | |
| III | **Recurrent Neural Networks (RNNs) & Sequence Models**<br>Introduction to sequential data and RNN architecture<br>- Challenges: Vanishing/exploding gradients<br>- Advanced RNNs: LSTM, GRU, Bidirectional RNNs<br>- Sequence-to-sequence models and attention mechanism (basics)<br>- **Applications:** Time-series forecasting, sentiment analysis, text generation | 1 | 25% |
| IV | **Model Optimization, Ethics & Deployment**<br>Hyperparameter tuning: Learning rate, batch size, epochs<br>- Regularization: L1/L2, dropout, early stopping, data augmentation<br>- Model evaluation: Overfitting vs underfitting, cross-validation<br>- Ethical AI: Bias in datasets, fairness, transparency, and interpretability<br>- **Applications:** Deploying models with TensorFlow Serving/Flask, bias auditing | 1 | 25% |

**Textbooks:**

- *Deep Learning* — Ian Goodfellow, Yoshua Bengio, Aaron Courville
- *Deep Learning with Python* — François Chollet

**Reference books:**

- *Neural Networks and Deep Learning: A Textbook* — Charu C. Aggarwal
- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — Aurélien Géron
- *Practical Deep Learning for Cloud, Mobile, and Edge* — Anirudh Koul et al.
- *Deep Learning for Computer Vision* — Rajalingappaa Shanmugamani

**Online Platforms:**

- NPTEL:
  1. *Deep Learning* by Prof. Mitesh Khapra (IIT Madras)
  2. *Deep Learning for Computer Vision* by IIT Ropar
- Coursera:
  1. *Deep Learning Specialization* by Andrew Ng
  2. *Introduction to Deep Learning* by National Research University Higher School of Economics

**PRACTICAL LIST:**

Neural Networks & Optimization

1. Implement a multi-layer perceptron (MLP) from scratch using NumPy.
2. Build and train a neural network with Keras/TensorFlow for the MNIST dataset.
3. Experiment with different activation functions and optimizers; compare performance.
4. Visualize loss curves, accuracy, and weight distributions during training.

Section B: Convolutional Neural Networks (CNNs)

5. Design and train a CNN for CIFAR-10 image classification.
6. Implement data augmentation techniques (rotation, flip, zoom) and evaluate impact.

7. Use transfer learning (VGG16/ResNet50) on a custom image dataset.
8. Visualize convolutional filters and feature maps to interpret model decisions.
   Section C: Recurrent Neural Networks (RNNs)
9. Build an LSTM model for time-series prediction (e.g., stock prices).
10. Implement a sentiment analysis model using RNNs on the IMDB dataset.
11. Create a character-level text generation model using LSTMs.
12. Compare Simple RNN, LSTM, and GRU on a sequence classification task.
    Section D: Model Tuning & Mini-Project
13. Apply dropout, batch normalization, and L2 regularization to a CNN; measure effect on overfitting.
14. Perform hyperparameter tuning using Keras Tuner or Optuna.
15. Deploy a trained deep learning model using Flask or TensorFlow Serving.
16. Mini-Project: Develop an end-to-end deep learning application (e.g., facial emotion recognition, handwritten equation solver, or spam detector with LSTMs).

**COURSE CODE: DAI402**

**COURSE NAME: NATURAL LANGUAGE PROCESSING**

**Course Objectives:**

- To introduce fundamental concepts, tasks, and applications of natural language processing.
- To develop skills in text preprocessing, feature extraction, and language modeling.
- To enable students to implement and evaluate NLP models for classification, generation, and understanding tasks.
- To prepare students for advanced topics in conversational AI, sentiment analysis, and language translation.
- To foster awareness of ethical considerations and biases in NLP systems.

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

| | |
|---|---|
| CO1 | **Explain** NLP pipelines, text preprocessing techniques, and linguistic fundamentals. |
| CO2 | **Implement** traditional NLP methods: tokenization, stemming, TF-IDF, and word embeddings. |
| C03 | **Build** and evaluate NLP models for tasks like sentiment analysis, text classification, and named entity recognition. |
| C04 | **Apply** transformer-based models and sequence-to-sequence architectures for advanced NLP tasks. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Neural Networks**<br>What is NLP? History, applications, and challenges<br>- Linguistic basics: Morphology, syntax, semantics, pragmatics<br>- Text preprocessing: Tokenization, stopword removal, stemming, lemmatization<br>- Regular expressions for text pattern matching<br>- Applications: Chatbots, search engines, document | 1 | 25% |

| | | | |
|---|---|---|---|
| | summarization | | |
| II | **Convolutional Neural Networks (CNNs)**<br>CNN architecture: Convolutional layers, pooling, flattening, fully connected layers<br>- Popular CNN architectures: LeNet, AlexNet, VGG (overview)<br>- Transfer learning and fine-tuning<br>- Data augmentation for image data<br>- **Applications:** Image classification, object detection, medical imaging | 1 | 25% |
| III | **Recurrent Neural Networks (RNNs) & Sequence Models**<br>RNN basics: Sequential data handling, hidden states, vanishing gradient problem<br>- Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU)<br>- Sequence-to-sequence models (Seq2Seq) basics<br>- Time-series forecasting and text generation intro<br>- **Applications:** Sentiment analysis, language modeling, stock prediction | 1 | 25% |
| IV | **Model Optimization, Regularization & Ethics**<br>Regularization: Dropout, batch normalization, L1/L2 regularization<br>- Hyperparameter tuning: Learning rate, batch size, epochs<br>- Model evaluation: Overfitting vs underfitting, confusion matrix for DL<br>- Ethical considerations: Bias in AI, fairness, interpretability<br>- **Applications:** Model deployment basics, bias detection in models | 1 | 25% |

**Textbooks:**

- *Deep Learning* — Ian Goodfellow, Yoshua Bengio, Aaron Courville
- *Neural Networks and Deep Learning* — Michael Nielsen (Online free book)

**Reference books:**

- *Deep Learning with Python* — François Chollet
- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* — Aurélien Géron
- *Practical Deep Learning for Cloud, Mobile, and Edge* — Anirudh Koul et al.
- *Deep Learning for Computer Vision* — Rajalingappaa Shanmugamani

**Online Platforms:**

- NPTEL:
  1. *Deep Learning* by Prof. Mitesh Khapra (IIT Madras)
  2. *Deep Learning for Computer Vision* by IIT Ropar
- Coursera:
  1. *Deep Learning Specialization* by Andrew Ng
  2. *Introduction to Deep Learning* by MIT

**PRACTICAL LIST:**

**Section A: Neural Networks Basics**

1. Implement a perceptron from scratch using NumPy.

2. Build a feedforward neural network using TensorFlow/Keras for MNIST digit classification.
3. Experiment with different activation functions (Sigmoid, ReLU, Tanh) and compare performance.
4. Visualize weight updates and loss curves during training.

**Section B: Convolutional Neural Networks (CNNs)**

5. Build a CNN using Keras for CIFAR-10 image classification.
6. Implement data augmentation (rotation, zoom, flip) and observe impact on accuracy.
7. Use transfer learning with a pre-trained model (VGG16/ResNet) for a custom image dataset.
8. Visualize CNN feature maps and filters to understand layer activations.

**Section C: Recurrent Neural Networks (RNNs)**

9. Implement an RNN/LSTM for time-series prediction (e.g., stock prices, temperature).
10. Build a sentiment analysis model using LSTM on IMDB movie reviews.
11. Create a text generation model using RNNs on a small text corpus.
12. Compare performance of Simple RNN vs LSTM vs GRU on a sequence task.

**Section D: Optimization & Mini-Project**

13. Apply dropout and batch normalization to a CNN and evaluate effect on overfitting.
14. Perform hyperparameter tuning using Keras Tuner or Grid Search.
15. Mini-Project: Develop an end-to-end deep learning application (e.g., facial expression recognition, spam detection using LSTMs).
16. Analyze model bias using a fairness toolkit (e.g., IBM AI Fairness 360 demo).

**COURSE CODE: DAI403**

**COURSE NAME: COMPUTER VISION**

**Course Objectives:**

- To introduce fundamental concepts, techniques, and applications of computer vision.
- To develop skills in image processing, feature extraction, and object detection using traditional and deep learning methods.
- To enable students to implement and evaluate computer vision models for real-world problems.
- To prepare students for advanced topics in video analysis, 3D vision, and AI-driven imaging systems.
- To foster awareness of ethical issues, biases, and responsible deployment of vision systems.

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

| | |
|---|---|
| CO1 | **Explain** image fundamentals, processing techniques, and feature extraction methods. |
| CO2 | **Implement** image filtering, edge detection, and morphological operations using OpenCV. |
| C03 | **Design** and train CNN-based models for image classification, object detection, and segmentation. |
| C04 | **Evaluate** vision models, apply transfer learning, and understand ethical implications in CV systems. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Computer Vision & Image Processing**<br>What is Computer Vision? History, applications, and challenges<br>- Digital image fundamentals: Pixels, channels, color spaces | 1 | 25% |

| | | | |
|---|---|---|---|
| | (RGB, HSV, Grayscale)<br>- Image preprocessing: Resizing, cropping, normalization, histogram equalization<br>- Image filtering: Smoothing, sharpening, Gaussian blur, median filter<br>- **Applications:** Medical imaging, surveillance, augmented reality | | |
| II | **Feature Detection & Image Segmentation**<br>Edge detection: Sobel, Canny, Laplacian<br>- Corner detection: Harris, Shi-Tomasi<br>- Feature descriptors: SIFT, SURF, ORB (overview)<br>- Image segmentation: Thresholding, region-based, watershed algorithm<br>- **Applications:** Object recognition, facial landmark detection, medical segmentation | 1 | 25% |
| III | **Deep Learning for Computer Vision**<br>CNN architectures for vision: LeNet, AlexNet, VGG, ResNet<br>- Transfer learning and fine-tuning for image classification<br>- Object detection: R-CNN, YOLO, SSD (concepts and implementation)<br>- Image segmentation: U-Net, Mask R-CNN (introduction)<br>- **Applications:** Autonomous driving, facial recognition, industrial inspection | 1 | 25% |
| IV | **Advanced Topics & Ethical Considerations**<br>Video analysis: Optical flow, object tracking (Mean-Shift, Kalman filter)<br>- 3D vision basics: Stereo imaging, depth estimation<br>- Generative models for vision: GANs for image synthesis (introduction)<br>- Ethics in CV: Bias in datasets, privacy concerns, adversarial attacks<br>- **Applications:** Action recognition, depth sensing, synthetic data generation | 1 | 25% |

**Textbooks:**

- *Computer Vision: Algorithms and Applications* — Richard Szeliski
- *Learning OpenCV 4* — Adrian Kaehler & Gary Bradski

**Reference books:**

- *Deep Learning for Computer Vision* — Rajalingappaa Shanmugamani
- *Programming Computer Vision with Python* — Jan Erik Solem
- *Computer Vision: Models, Learning, and Inference* — Simon J. D. Prince
- *Practical Deep Learning for Cloud, Mobile, and Edge* — Anirudh Koul et al.

**Online Platforms:**

- NPTEL
  1. *Computer Vision* by Prof. Jayanta Mukhopadhyay (IIT Kharagpur)
  2. *Deep Learning for Computer Vision* by IIT Ropar
- Coursera:
  1. *Deep Learning Specialization* (Course 4: Convolutional Neural Networks)
  2. *Introduction to Computer Vision* by University at Buffalo

# MK UNIVERSITY

PATAN, GUJARAT

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

**COURSE CODE: DAI404**

**COURSE NAME: CLOUD COMPUTING BASICS**

**Course Objectives:**

- To introduce the fundamental concepts, models, and architectures of cloud computing.
- To develop skills in deploying, managing, and scaling applications using major cloud platforms.
- To enable students to leverage cloud services for AI/ML model training, storage, and deployment.
- To prepare students for cloud-native development and understanding of security, compliance, and cost management.
- To foster awareness of emerging trends such as serverless computing, containers, and hybrid cloud.

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

| | |
|---|---|
| CO1 | **Explain** cloud computing models, service models, and deployment models. |
| CO2 | **Deploy** and manage virtual machines, storage, and networking in a cloud environment. |
| C03 | **Utilize** cloud-based AI/ML services for model training and deployment. |
| C04 | **Apply** cloud security best practices and manage cloud costs effectively. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Cloud Computing**<br>What is Cloud Computing? Evolution, characteristics, benefits, and challenges<br>- Cloud service models: IaaS, PaaS, SaaS, FaaS<br>- Cloud deployment models: Public, private, hybrid, community cloud<br>- Key cloud providers: AWS, Azure, GCP overview<br>- **Applications:** Web hosting, data storage, scalable computing | 1 | 25% |
| II | **Cloud Infrastructure & Services**<br>Virtualization: Hypervisors, virtual machines, containers (Docker basics)<br>- Compute services: EC2 (AWS), VM (Azure), Compute Engine (GCP)<br>- Storage services: S3, Blob Storage, Cloud Storage<br>- Networking: VPC, load balancers, CDN basics<br>- **Applications:** Deploying web apps, media streaming, backup solutions | 1 | 25% |
| III | **Cloud for AI & DevOps**<br>AI/ML cloud services: AWS SageMaker, Azure ML, GCP AI Platform<br>- Serverless computing: AWS Lambda, Azure Functions<br>- Containers and orchestration: Docker, Kubernetes basics<br>- DevOps in the cloud: CI/CD pipelines (GitHub Actions, Jenkins basics)<br>- **Applications:** Training ML models, auto-scaling apps, | 1 | 25% |

| | | | |
|---|---|---|---|
| | microservices | | |
| IV | **Cloud Security, Management & Trends**<br>Cloud security: Identity and Access Management (IAM), encryption, compliance<br>- Cost management: Pricing models, budgeting, monitoring tools<br>- Migration strategies: Lift-and-shift, re-platforming, cloud-native<br>- Emerging trends: Edge computing, hybrid cloud, sustainable cloud<br>- **Applications:** Secure cloud deployments, cost-optimized architectures, green IT | 1 | 25% |

**Textbooks:**

- *Cloud Computing: Concepts, Technology & Architecture* — Thomas Erl et al.
- *Cloud Computing for Dummies* — Judith Hurwitz et al.

**Reference books:**

- *Architecting the Cloud* — Michael J. Kavis
- *Cloud Native Transformation* — Pini Reznik et al.
- *AWS Certified Solutions Architect Study Guide* — Ben Piper & David Clinton
- *Cloud Computing Bible* — Barrie Sosinsky

**Online Platforms:**

- NPTEL:
  1. *Cloud Computing* by Prof. Soumya Kanti Ghosh (IIT Kharagpur)
  2. *Introduction to Cloud Computing* by IIT Madras
- Coursera:
  1. *Cloud Computing Specialization* by University of Illinois
  2. *AWS Fundamentals* by Amazon Web Services

# MK UNIVERSITY

PATAN, GUJARAT

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

## SEMESTER-V

**COURSE CODE: DAI501**

**COURSE NAME: AI ETHICS AND GOVERNANCE**

**Course Objectives:**

- To introduce fundamental ethical principles, challenges, and societal impacts of artificial intelligence.
- To develop skills in identifying, assessing, and mitigating ethical risks in AI systems.
- To enable students to understand and apply governance frameworks, regulations, and standards for responsible AI.
- To prepare students for roles requiring ethical oversight, compliance, and responsible innovation in AI.
- To foster critical thinking around fairness, transparency, accountability, and inclusivity in AI design and deployment.

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

| | |
|---|---|
| CO1 | **Explain** core ethical principles, biases, and societal impacts of AI systems. |
| CO2 | **Identify** ethical risks and apply fairness, accountability, and transparency (FAT) frameworks. |
| C03 | **Evaluate** AI systems for compliance with legal, regulatory, and governance standards. |
| C04 | **Design** ethical AI solutions and draft governance policies for responsible deployment. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to AI Ethics & Societal Impact**<br>What is AI Ethics? History, need, and core principles<br>- Ethical theories: Utilitarianism, deontology, virtue ethics<br>- Societal impacts: Job displacement, privacy, surveillance, digital divide<br>- Case studies: Biased hiring algorithms, facial recognition misuse<br>- **Applications:** Ethical AI design, public policy, tech accountability | 1 | 25% |
| II | **Bias, Fairness & Transparency in AI**<br>Types of bias: Data bias, algorithmic bias, human bias<br>- Measuring fairness: Demographic parity, equal opportunity, disparate impact<br>- Explainable AI (XAI): Techniques (LIME, SHAP), model interpretability<br>- Tools for bias detection: IBM AI Fairness 360, Google What-If Tool<br>- **Applications:** Fair credit scoring, unbiased recruitment, transparent healthcare AI | 1 | 25% |
| III | **AI Governance, Regulations & Standards**<br>Global AI regulations: GDPR (EU), AI Act (EU), | 1 | 25% |

| | | | |
|---|---|---|---|
| | Algorithmic Accountability Act (US)<br>- National frameworks: India's AI Strategy, NITI Aayog guidelines<br>- Standards: IEEE Ethically Aligned Design, ISO/IEC 42001<br>- Corporate governance: Ethics boards, compliance officers, audit trails<br>- **Applications:** Regulatory compliance, corporate AI policy, certification | | |
| IV | **Responsible AI Deployment & Future Challenges**<br>Responsible innovation: Human-centered design, participatory AI<br>- Ethical deployment: Monitoring, feedback loops, incident response<br>- Future challenges: Autonomous weapons, deepfakes, AGI ethics<br>- Sustainability: Green AI, carbon footprint of AI systems<br>- **Applications:** Ethical AI toolkits, audit reports, sustainability assessments | 1 | 25% |

**Textbooks:**

- *Ethics of Artificial Intelligence and Robotics* — Vincent C. Müller
- *Weapons of Math Destruction* — Cathy O'Neil

**Reference books:**

- *The Alignment Problem* — Brian Christian
- *Atlas of AI* — Kate Crawford
- *AI Ethics* — Mark Coeckelbergh
- *Practical Fairness* — Aileen Nielsen

**Online Platforms:**

- NPTEL:
  1. *Ethics in AI* by Prof. Ponnurangam Kumaraguru (IIIT Delhi)
  2. *Responsible AI* by IIT Hyderabad
- Coursera:
  1. *AI Ethics* by University of Helsinki
  2. *Responsible AI* by Google Cloud

**PRACTICAL LIST:**

Section A: Ethical Analysis & Case Studies

1. Analyze a real-world AI ethics case study (e.g., COMPAS recidivism algorithm) and write a critique.
2. Conduct a stakeholder impact analysis for an AI system (e.g., automated hiring tool).
3. Use ethical frameworks (e.g., EU Ethics Guidelines) to evaluate a given AI application.
4. Debate ethical dilemmas in AI (e.g., trolley problem for autonomous vehicles).

Section B: Bias Detection & Fairness Assessment

5. Detect bias in a dataset using Python (e.g., using aif360 or fairlearn).
6. Apply fairness metrics (demographic parity, equal opportunity) to a classification model.
7. Mitigate bias using pre-processing, in-processing, or post-processing techniques.
8. Use SHAP or LIME to interpret model predictions and identify unfair influences.

Section C: Governance & Compliance Simulation

9. Draft an AI ethics policy for a fictional company deploying facial recognition.

10. Conduct a GDPR compliance checklist for an AI system handling personal data.
11. Role-play an AI ethics board meeting to approve/reject a new AI product.
12. Create an AI incident response plan for a bias-related failure.
    Section D: Responsible AI Project & Audit
13. Design a "Responsible AI Checklist" for deployment covering fairness, transparency, privacy, and accountability.
14. Perform an ethical audit of an open-source AI model or dataset.
15. Develop an awareness campaign or educational material on AI ethics for non-technical stakeholders.
16. Final Project: Create a comprehensive "AI Ethics & Governance Report" for a real or hypothetical AI system, covering risk assessment, mitigation strategies, and compliance recommendations.

**COURSE CODE: DAI502**

**COURSE NAME: REINFORCEMENT LEARNING**

**Course Objectives:**

- To introduce fundamental concepts, frameworks, and mathematical foundations of reinforcement learning (RL).
- To develop skills in modeling decision-making problems as Markov Decision Processes (MDPs).
- To enable students to implement and evaluate classic and modern RL algorithms.
- To prepare students for applying RL to real-world problems in robotics, gaming, finance, and autonomous systems.
- To foster understanding of ethical considerations, safety, and scalability in RL systems.

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

| | |
|---|---|
| CO1 | **Explain** RL concepts, MDPs, value functions, and exploration-exploitation trade-offs. |
| CO2 | **Implement** tabular RL methods: Dynamic Programming, Monte Carlo, and Temporal Difference. |
| C03 | **Apply** value-based and policy-based deep RL algorithms (DQN, Policy Gradients). |
| C04 | **Evaluate** RL systems, handle partial observability, and understand real-world deployment challenges. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Reinforcement Learning**<br>What is RL? Key concepts: Agent, environment, state, action, reward<br>- Markov Decision Processes: States, transitions, rewards, discount factor<br>- Value functions: State-value, action-value (Q-function)<br>- Exploration vs exploitation: ε-greedy, softmax<br>- **Applications:** Game playing, robotic control, recommendation systems | 1 | 25% |
| II | **Tabular RL Methods**<br>Dynamic Programming: Policy evaluation, policy iteration, value iteration<br>- Monte Carlo methods: Prediction and control | 1 | 25% |

| | | | |
|---|---|---|---|
| | - Temporal Difference learning: TD(0), SARSA, Q-learning<br>- N-step TD and eligibility traces (introduction)<br>- **Applications:** Grid-world problems, simple games, inventory management | | |
| III | **Deep Reinforcement Learning**<br>Need for function approximation in RL<br>- Deep Q-Networks (DQN): Experience replay, target networks<br>- Policy gradient methods: REINFORCE, Actor-Critic (A2C, A3C)<br>- Advanced algorithms: DDPG, PPO (overview)<br>- **Applications:** Atari games, autonomous driving, robotics simulation | 1 | 25% |
| IV | **Advanced Topics & Real-World RL**<br>Partially Observable MDPs (POMDPs) and belief states<br>- Multi-agent RL: Competitive and cooperative settings<br>- Safe RL: Constrained optimization, risk-aware policies<br>- Ethical RL: Reward hacking, alignment, fairness<br>- **Applications:** Healthcare treatment policies, financial trading, smart grids | 1 | 25% |

**Textbooks:**

- *Reinforcement Learning: An Introduction* — Richard S. Sutton & Andrew G. Barto
- *Deep Reinforcement Learning Hands-On* — Maxim Lapan

**Reference books:**

- *Algorithms for Reinforcement Learning* — Csaba Szepesvári
- *Reinforcement Learning with Python* — Sudharsan Ravichandiran
- *Grokking Deep Reinforcement Learning* — Miguel Morales
- *Reinforcement Learning: State-of-the-Art* — Marco Wiering & Martijn van Otterlo

**Online Platforms:**

- NPTEL:
  1. *Reinforcement Learning* by Prof. Balaraman Ravindran (IIT Madras)
  2. *Deep Reinforcement Learning* by IIT Hyderabad
- Coursera:
  1. *Reinforcement Learning Specialization* by University of Alberta & Alberta Machine Intelligence Institute
  2. *Deep Reinforcement Learning* by Hugging Face

**PRACTICAL LIST:**

Section A: Foundational RL & MDPs

1. Implement a simple grid-world environment and define states, actions, rewards.
2. Solve a given MDP using Value Iteration and Policy Iteration.
3. Implement ε-greedy and softmax exploration strategies in a bandit problem.
4. Visualize value function and policy updates in a grid-world.

Section B: Tabular RL Algorithms

5. Implement Monte Carlo prediction and control for Blackjack.
6. Code SARSA and Q-learning for the FrozenLake environment.
7. Compare TD(0) vs Monte Carlo methods on the Cliff Walking environment.

8. Implement N-step TD learning and analyze its performance.
   Section C: Deep Reinforcement Learning
9. Train a DQN agent to play CartPole using OpenAI Gym.
10. Implement experience replay and target networks in DQN.
11. Code the REINFORCE algorithm for a simple RL task.
12. Build a basic Actor-Critic agent for LunarLander.
    Section D: Advanced Applications & Mini-Project
13. Implement a multi-armed bandit with non-stationary rewards.
14. Adapt an RL algorithm for a partially observable environment (e.g., using LSTMs).
15. Simulate a simple multi-agent scenario (cooperative/competitive) using RLlib or PettingZoo.
16. Mini-Project: Develop an end-to-end RL solution (e.g., train an agent to play a custom game, optimize a trading strategy, or control a simulated robot arm).

**COURSE CODE: DAI502**

**COURSE NAME: BIG DATA ANALYTICS**

**Course Objectives:**

- To introduce fundamental concepts, challenges, and technologies in big data.
- To develop skills in processing, storing, and analyzing large-scale datasets using distributed computing frameworks.
- To enable students to apply big data analytics techniques for data-driven decision-making and AI model training.
- To prepare students for real-world big data pipelines, data lakes, and cloud-based analytics.
- To foster awareness of data governance, privacy, and ethical handling of big data.

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

| | |
|---|---|
| CO1 | **Explain** big data characteristics, architecture, and storage technologies. |
| CO2 | **Process** and analyze large datasets using Hadoop, Spark, and NoSQL databases. |
| C03 | **Apply** data mining and machine learning techniques on big data platforms. |
| C04 | **Design** and implement scalable data pipelines for analytics and AI applications. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Big Data**<br>What is Big Data? 5 V's: Volume, Velocity, Variety, Veracity, Value<br>- Big data ecosystem: Hadoop, Spark, NoSQL, data lakes<br>- Data storage: HDFS, cloud storage (AWS S3, Google Cloud Storage)<br>- Big data challenges: Scalability, fault tolerance, real-time processing<br>- **Applications:** Social media analytics, IoT data, log processing | 1 | 25% |
| II | **Big Data Processing with Hadoop & Spark** | 1 | 25% |

| | | | |
|---|---|---|---|
| | Hadoop architecture: HDFS, MapReduce, YARN<br>- Apache Spark: RDDs, DataFrames, Spark SQL<br>- Spark MLlib for machine learning on big data<br>- Stream processing: Spark Streaming, Kafka basics<br>- **Applications:** Batch processing, real-time analytics, ETL pipelines | | |
| III | **NoSQL Databases & Data Warehousing**<br>NoSQL databases: Types (Document, Key-Value, Columnar, Graph)<br>- MongoDB, Cassandra, HBase basics<br>- Data warehousing: OLAP vs OLTP, star schema, snowflake schema<br>- ETL tools: Apache NiFi, Talend (overview)<br>- **Applications:** Customer 360, time-series data, recommendation systems | 1 | 25% |
| IV | **Big Data Analytics & AI Integration**<br>Data mining techniques: Clustering, association rule mining<br>- Machine learning at scale: Distributed training, model serving<br>- Big data visualization: Tableau, Power BI, Apache Superset<br>- Ethics and governance: Data privacy, GDPR, data anonymization<br>- **Applications:** Predictive maintenance, fraud detection, business intelligence | 1 | 25% |

**Textbooks:**

- *Big Data: Principles and Best Practices* — Thomas Erl et al.
- *Hadoop: The Definitive Guide* — Tom White

**Reference books:**

- *Spark: The Definitive Guide* — Bill Chambers & Matei Zaharia
- *NoSQL Distilled* — Pramod J. Sadalage & Martin Fowler
- *Designing Data-Intensive Applications* — Martin Kleppmann
- *Big Data Analytics with Python* — Venkat Ankam

**Online Platforms:**

- NPTEL:
  1. *Big Data Computing* by Prof. Rajiv Misra (IIT Patna)
  2. *Introduction to Big Data* by IIT Kharagpur
- Coursera:
  1. *Big Data Specialization* by University of California San Diego
  2. *Apache Spark with Python* by IBM

**PRACTICAL LIST:**

Section A: Big Data Fundamentals & HDFS

1. Set up a single-node Hadoop cluster (using Docker or VM).
2. Perform basic HDFS operations: Upload, download, list, delete files.
3. Write a MapReduce program (WordCount) in Java or Python (MRJob).
4. Analyze large log files using MapReduce to find top IP addresses.

Section B: Apache Spark & Data Processing

5. Install PySpark and perform basic RDD operations (map, filter, reduceByKey).
6. Use Spark DataFrames to clean and analyze a large dataset (e.g., NYC taxi data).
7. Perform SQL queries on big data using Spark SQL.
8. Implement a streaming word count using Spark Streaming (or Structured Streaming).
   Section C: NoSQL Databases & ETL
9. Perform CRUD operations in MongoDB using PyMongo.
10. Design a schema and insert data into Apache Cassandra.
11. Build a simple ETL pipeline using Apache NiFi to move data from CSV to HDFS.
12. Query data from a data warehouse (e.g., using Amazon Redshift or Google BigQuery).
    Section D: Big Data Analytics & Mini-Project
13. Perform clustering on big data using Spark MLlib (K-means).
14. Build a recommendation system using collaborative filtering on a large dataset.
15. Create a dashboard using Tableau Public or Apache Superset to visualize big data insights.
16. Mini-Project: Design and implement an end-to-end big data pipeline for a real-world problem (e.g., social media sentiment analysis, IoT sensor data processing, or sales forecasting).

**COURSE CODE: DAI503**

**COURSE NAME: ROBOTICS AND AUTOMATION**

**Course Objectives:**

- To introduce fundamental concepts of robotics, automation systems, and their integration with AI.
- To develop skills in robot kinematics, sensors, actuators, and control systems.
- To enable students to program and simulate robots using modern tools and frameworks.
- To prepare students for roles in industrial automation, smart manufacturing, and autonomous systems.
- To foster awareness of safety standards, ethical considerations, and future trends in robotics.

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

| | |
|---|---|
| CO1 | **Explain** robotics fundamentals, components, and types of automation systems. |
| CO2 | **Apply** kinematics, dynamics, and control theory to robotic motion planning. |
| C03 | **Program** robots using ROS (Robot Operating System) and simulate robotic tasks. |
| C04 | **Design** automation solutions integrating AI, computer vision, and IoT for smart systems. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Robotics & Automation**<br>What is Robotics? History, types (industrial, service, mobile)<br>- Automation systems: Fixed, programmable, flexible automation<br>- Robot components: Sensors, actuators, controllers, end-effectors<br>- Robot configurations: Cartesian, cylindrical, spherical, articulated | 1 | 25% |

| | | | |
|---|---|---|---|
| | - **Applications:** Manufacturing, healthcare, logistics, domestic robots | | |
| II | **Robot Kinematics, Dynamics & Control**<br>Kinematics: Forward and inverse kinematics for robotic arms<br>- Dynamics: Newton-Euler and Lagrangian formulations (intro)<br>- Robot control: PID control, trajectory planning, feedback systems<br>- Sensors: Proximity, vision, force/torque, LiDAR<br>- **Applications:** Pick-and-place, welding, painting, assembly lines | 1 | 25% |
| III | **Robot Programming & ROS**<br>Introduction to ROS: Nodes, topics, services, messages<br>- Programming robots with Python and C++ in ROS<br>- Simulation tools: Gazebo, RViz, CoppeliaSim<br>- SLAM (Simultaneous Localization and Mapping) basics<br>- **Applications:** Autonomous navigation, mobile robots, drone simulation | 1 | 25% |
| IV | **AI in Robotics & Smart Automation**<br>AI integration: Machine learning for robot perception and decision-making<br>- Computer vision for robotics: Object detection, pose estimation<br>- Industrial IoT and cyber-physical systems<br>- Ethics, safety standards (ISO 10218, R15.06), and human-robot collaboration<br>- **Applications:** Collaborative robots (cobots), AGVs, smart factories, robotic surgery | 1 | 25% |

**Textbooks:**

- *Introduction to Robotics: Mechanics and Control* — John J. Craig
- *Robotics, Vision and Control* — Peter Corke

**Reference books:**

- *Probabilistic Robotics* — Sebastian Thrun, Wolfram Burgard, Dieter Fox
- *ROS Robotics by Example* — Carol Fairchild & Thomas L. Harman
- *Robotics: Everything You Need to Know* — Peter McKinnon
- *Automation, Production Systems, and Computer-Integrated Manufacturing* — Mikell P. Groover

**Online Platforms:**

- NPTEL:
  1. *Robotics* by Prof. C. Amarnath (IIT Bombay)
  2. *Introduction to Robotics* by IIT Madras
- Coursera:
  1. *Robotics Specialization* by University of Pennsylvania
  2. *Modern Robotics* by Northwestern University

# MK UNIVERSITY

PATAN, GUJARAT

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

## SEMESTER-VI

**COURSE CODE: DAI601**

**COURSE NAME: AI IN INDUSTRY**

**Course Objectives:**

- To introduce real-world applications, challenges, and implementation strategies of AI across major industries.
- To develop skills in designing, deploying, and managing AI solutions in industrial settings.
- To enable students to integrate AI with existing business processes, IoT, and automation systems.
- To prepare students for roles as AI solution architects, industry consultants, and technical leads.
- To foster understanding of ROI, scalability, ethics, and change management in industrial AI projects.

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

| | |
|---|---|
| CO1 | **Analyze** industry-specific AI use cases, challenges, and business impacts. |
| CO2 | **Design** AI-driven solutions for manufacturing, healthcare, finance, and retail. |
| C03 | **Implement** AI integration pipelines with IoT, cloud, and legacy systems. |
| C04 | **Evaluate** AI projects for ROI, scalability, ethics, and operational feasibility. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to Industrial AI & Digital Transformation**<br>What is Industrial AI? Industry 4.0, smart factories, digital twins<br>- AI adoption framework: Strategy, data readiness, technology stack<br>- Business impact: Cost reduction, efficiency, innovation, new revenue streams<br>- Case studies: Siemens, GE, Toyota AI implementations<br>- **Applications:** Predictive maintenance, quality inspection, supply chain optimization | 1 | 25% |
| II | **AI in Manufacturing & Supply Chain**<br>Smart manufacturing: Robotics, computer vision for defect detection<br>- Predictive maintenance: Sensor data, failure prediction models<br>- Supply chain AI: Demand forecasting, inventory optimization, logistics<br>- Digital twin simulation for process optimization<br>- **Applications:** Automotive assembly lines, pharmaceutical packaging, warehouse automation | 1 | 25% |
| III | **AI in Healthcare, Finance & Retail**<br>Healthcare AI: Medical imaging, drug discovery, patient monitoring | 1 | 25% |

| | | | |
|---|---|---|---|
| | - FinTech AI: Fraud detection, algorithmic trading, robo-advisors<br>- Retail AI: Recommendation systems, customer sentiment analysis, smart pricing<br>- Ethical and regulatory considerations (HIPAA, GDPR, SEBI)<br>- **Applications:** MRI analysis, credit scoring, e-commerce personalization | | |
| IV | **AI Deployment, Scaling & Governance**<br>MLOps for industry: Model deployment, monitoring, retraining<br>- Integration with IoT/ERP/SCADA systems<br>- Measuring ROI: KPIs, performance metrics, cost-benefit analysis<br>- Change management: Upskilling workforce, stakeholder buy-in<br>- **Applications:** Edge AI for factories, cloud AI platforms, compliance reporting | 1 | 25% |

**Textbooks:**

- *AI in Practice* — Bernard Marr
- *The AI-Powered Enterprise* — Seth Earley

**Reference books:**

- *Applied Artificial Intelligence: A Handbook for Business Leaders* — Mariya Yao, Adelyn Zhou, Marlene Jia
- *AI Superpowers* — Kai-Fu Lee
- *The Future of Work* — Darrell M. West
- *Industrial AI* — Fei Tao & Qinglin Qi

**Online Platforms:**

- NPTEL:
  1. *Industry 4.0* by IIT Kharagpur
  2. *AI for Business* by IIM Bangalore

**PRACTICAL LIST:**

Section A: Industrial AI Use Case Analysis

1. Research and present a case study on AI adoption in a chosen industry (e.g., automotive, pharma).
2. Perform a SWOT analysis for implementing an AI solution in a manufacturing unit.
3. Identify and map data sources for a predictive maintenance project in a factory.
4. Design a business case presentation for an AI project including ROI estimation.

Section B: AI Solution Design for Industry

5. Design a computer vision system for defect detection in product assembly lines.
6. Build a demand forecasting model using time-series data (e.g., sales data).
7. Create a chatbot for customer service in the retail sector using Dialogflow or Rasa.
8. Simulate a digital twin for a conveyor belt system using Python and simulation tools.

Section C: Integration & Deployment

9. Deploy a trained machine learning model as a REST API using Flask/FastAPI.
10. Integrate an AI model with an IoT sensor stream (simulated using MQTT).
11. Set up a basic MLOps pipeline using GitHub Actions for model retraining.
12. Create a dashboard using Power BI or Grafana to monitor AI model performance in production

Section D: Capstone Industry Project

13. Capstone Project: Develop an end-to-end AI solution for a real-world industry problem (e.g., predictive maintenance for HVAC systems, inventory optimization for a retail store, or patient readmission prediction for hospitals).
14. Prepare a detailed project report including:
    - Problem statement
    - Data collection & preprocessing
    - Model development
    - Deployment strategy
    - Business impact & ROI analysis
    - Ethical considerations

**COURSE CODE: DAI602**

**COURSE NAME: AI DEPLOYMENT & MLOPs**

**Course Objectives:**

- To introduce the end-to-end lifecycle of machine learning models from development to deployment and monitoring.
- To develop skills in building scalable, reproducible, and maintainable ML pipelines using MLOps principles.
- To enable students to deploy models in cloud, on-premise, and edge environments with containerization and orchestration.
- To prepare students for roles in ML engineering, DevOps for AI, and production ML system management.
- To foster understanding of model versioning, monitoring, governance, and ethical deployment practices.

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

| | |
|---|---|
| CO1 | **Explain** the MLOps lifecycle, deployment strategies, and CI/CD for ML. |
| CO2 | **Build** automated ML pipelines with version control, testing, and containerization. |
| C03 | **Deploy** models using cloud platforms, Kubernetes, and serverless architectures. |
| C04 | **Monitor** and manage models in production with logging, alerting, and performance tracking. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to MLOps & Model Lifecycle**<br>What is MLOps? Differences between DevOps and MLOps<br>- ML project lifecycle: Data collection, training, validation, deployment, monitoring<br>- Model versioning: Git, DVC (Data Version Control), MLflow<br>- CI/CD for ML: Automated testing, pipeline orchestration<br>- **Applications:** Reproducible research, automated retraining, A/B testing | 1 | 25% |
| II | **Containerization & Orchestration for ML** | 1 | 25% |

| | | | |
|---|---|---|---|
| | Docker basics: Images, containers, Dockerfile for ML models<br>- Kubernetes fundamentals: Pods, services, deployments for scaling ML<br>- Model serving frameworks: TensorFlow Serving, TorchServe, KServe<br>- Edge deployment: TensorFlow Lite, ONNX Runtime<br>- **Applications:** Scalable model APIs, multi-model serving, edge AI | | |
| III | **Cloud Deployment & Serverless ML**<br>Cloud ML platforms: AWS SageMaker, Google AI Platform, Azure ML<br>- Serverless deployment: AWS Lambda, Google Cloud Functions, Azure Functions<br>- Infrastructure as Code (IaC): Terraform/CloudFormation for ML stacks<br>- Cost optimization and auto-scaling strategies<br>- **Applications:** Real-time inference, batch processing, event-driven ML | 1 | 25% |
| IV | **Model Monitoring, Governance & Ethics**<br>Monitoring: Performance drift, data drift, concept drift detection<br>- Logging and alerting: Prometheus, Grafana, ELK stack basics<br>- Model governance: Compliance, audit trails, reproducibility<br>- Ethical deployment: Bias monitoring, fairness, explainability in production<br>- **Applications:** Financial fraud detection, healthcare diagnostics, recommendation systems | 1 | 25% |

**Textbooks:**

- *Introducing MLOps* — Mark Treveil et al.
- *Practical MLOps* — Noah Gift & Alfredo Deza

**Reference books:**

- *Machine Learning Engineering* — Andriy Burkov
- *Building Machine Learning Pipelines* — Hannes Hapke & Catherine Nelson
- *MLOps: From Model to Production* — Rishabh Bhargava
- *The DevOps Handbook* — Gene Kim et al. (MLOps adaptation)

**Online Platforms:**

- NPTEL:
  1. *MLOps: Machine Learning Operations* by IIT Hyderabad
  2. *Cloud Computing for AI* by IIT Kharagpur

**PRACTICAL LIST:**

Section A: MLOps Foundations & Pipeline Automation

1. Set up a Git repository for an ML project with DVC for data versioning.
2. Build an automated ML pipeline using GitHub Actions for training and validation.
3. Use MLflow to track experiments, log parameters, and register models.

4. Create unit tests for data validation, model training, and inference code.

Section B: Containerization & Model Serving

5. Containerize a trained ML model using Docker and create a REST API with Flask/FastAPI.
6. Deploy a TensorFlow/PyTorch model using TensorFlow Serving or TorchServe.
7. Orchestrate multiple model containers using Docker Compose.
8. Convert a model to TensorFlow Lite for edge deployment and test on mobile/embedded simulation.

Section C: Cloud Deployment & Scaling

9. Deploy a model on AWS SageMaker/Google AI Platform and set up auto-scaling.
10. Create a serverless ML inference pipeline using AWS Lambda or Google Cloud Functions.
11. Use Terraform to provision cloud infrastructure for an ML deployment.
12. Implement A/B testing for models using cloud-based traffic splitting.

Section D: Monitoring, Governance & Capstone

13. Set up monitoring for model drift using Evidently AI or Amazon SageMaker Model Monitor.
14. Build a dashboard with Grafana to track model latency, throughput, and error rates.
15. Implement a feedback loop for continuous retraining using new inference data.
16. Capstone Project: Design and deploy a full MLOps pipeline for a real-world problem

**COURSE CODE: DAI603**

**COURSE NAME: IOT AND INDUSTRY AUTOMATION**

**Course Objectives:**

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

| | |
|---|---|
| CO1 | **Explain** IoT architecture, protocols, and industrial automation frameworks. |
| CO2 | **Design** and implement IoT systems using sensors, microcontrollers, and communication modules. |
| C03 | **Integrate** IoT with AI/ML for predictive maintenance, quality control, and process optimization. |
| C04 | **Develop** secure, scalable IIoT solutions for smart factories and automated systems. |

| Unit | Content | Credit | Weightage |
|---|---|---|---|
| I | **Introduction to IoT & Industrial Automation**<br>What is IoT? Evolution, components, and architecture (sensors, gateways, cloud)<br>- Industry 4.0: Smart factories, digital twins, cyber-physical systems<br>- Industrial automation: PLCs, SCADA, DCS, and robotics<br>- IoT communication protocols: MQTT, CoAP, HTTP, Modbus, OPC UA<br>- **Applications:** Smart manufacturing, energy management, predictive maintenance | 1 | 25% |
| II | **IoT Hardware & Embedded Systems**<br>Microcontrollers: Arduino, ESP32, Raspberry Pi for IoT<br>- Sensors and actuators: Temperature, humidity, motion, relays, motors | 1 | 25% |

| | | | |
|---|---|---|---|
| | - Connectivity modules: Wi-Fi, Bluetooth, LoRa, Zigbee, NB-IoT<br>- Edge computing: Local processing and decision-making<br>- **Applications:** Environmental monitoring, asset tracking, smart agriculture | | |
| III | **IoT Data, Cloud & AI Integration**<br>IoT data pipelines: Collection, storage, processing<br>- Cloud platforms: AWS IoT, Azure IoT Hub, Google Cloud IoT Core<br>- AI at the edge: Real-time analytics, anomaly detection, computer vision<br>- Digital twin modeling and simulation<br>- **Applications:** Quality inspection, energy optimization, remote monitoring | 1 | 25% |
| IV | **IIoT Security, Standards & Deployment**<br>IIoT security: Threats, encryption, authentication, secure boot<br>- Industrial standards: ISA-95, IEC 62443, Industry 4.0 compliance<br>- Deployment strategies: Pilot testing, scalability, maintenance<br>- Ethical and sustainable IoT: Energy efficiency, data privacy, e-waste<br>- **Applications:** Secure smart grids, automated supply chain, ethical automation | 1 | 25% |

**Textbooks:**

- *Internet of Things: A Hands-On Approach* — Arshdeep Bahga & Vijay Madisetti
- *Industrial Internet of Things (IIoT)* — Sabina Jeschke et al.

**Reference books:**

- *Building the Internet of Things* — Maciej Kranz
- *IoT and Edge Computing for Architects* — Perry Lea
- *Industry 4.0: The Industrial Internet of Things* — Alasdair Gilchrist
- *Practical Industrial Internet of Things Security* — Sravani Bhattacharjee

**Online Platforms:**

- NPTEL:
  1. *Introduction to Internet of Things* by Prof. Sudip Misra (IIT Kharagpur)
  2. *Industry 4.0* by IIT Roorkee
- Coursera:
  1. *IoT Specialization* by University of California Irvine
  2. *Industrial IoT on Google Cloud* by Google Cloud

**PRACTICAL LIST:**

Section A: IoT Fundamentals & Sensor Integration

1. Set up an Arduino/ESP32 board and interface with temperature and humidity sensors.
2. Send sensor data to a local server using MQTT protocol.
3. Build a simple PLC-based automation system using ladder logic (simulation with LogixPro).
4. Create a Node-RED dashboard to visualize real-time sensor data.

Section B: IoT Communication & Edge AI

5. Implement LoRa communication for long-range IoT data transmission.
6. Use Raspberry Pi with a camera module for image capture and edge-based object detection.
7. Develop an anomaly detection system for machine vibration using accelerometer data.
8. Build a digital twin simulation of a conveyor belt system using MATLAB/Simulink or Python.

Section C: Cloud Integration & Industrial Automation

9. Connect an IoT device to AWS IoT Core and store data in DynamoDB.
10. Implement a predictive maintenance model using cloud-based ML (Azure ML/AWS SageMaker).
11. Automate a smart irrigation system using IoT sensors and cloud-triggered actuators.
12. Design a SCADA-like monitoring system using Grafana and InfluxDB.

Section D: Security, Deployment & Capstone

13. Implement secure firmware updates and encrypted communication for an IoT device.
14. Perform a security audit on an IoT network using Wireshark and vulnerability scanners.
15. Deploy a full IIoT pilot project for a mini-factory setup (e.g., smart packaging line).