



**ANDHRA PRADESH STATE COUNCIL OF HIGHER
EDUCATION**

**Model Syllabus for Artificial Intelligence (Minor) in consonance with
Curriculum framework w.e.f. AY 2025-26**

Prepared by Adikavai Nannaya University, Rajahmundry

COURSE STRUCTURE

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
II	III	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation Lab	2	1
	IV	2	Python Programming and Data structures	3	3
			Python programming and Data Structure lab	2	1
III	V	3	Artificial & Computational Intelligence	3	3
			Artificial & Computational Intelligence lab	2	1
		4	Data Science with R	3	3
			Data Science With R lab	2	1
	VI	5	Exploratory Data Analysis & Data Visualization	3	3
			Exploratory Data Analysis & Data Visualization Lab	2	1
		6	Machine Learning	3	3
			Machine Learning lab	2	1

SEMESTER-III

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Theory

Credits: 3

3 hrs/week

Course Objectives

1. **Understand foundational computing concepts**, including number systems, the evolution of computers, block diagrams, and generational progress.
2. **Develop knowledge of computer architecture**, focusing on system organization and networking fundamentals.
3. **Acquire practical skills in document creation**, formatting, and digital presentations using word processing tools.
4. **Gain proficiency in spreadsheet operations**, such as data entry, formulas, functions, and charting techniques.
5. **Introduce data visualization and basic modelling principles**, fostering analytical thinking in structuring and interpreting data sets.

Course Outcomes

1. At the End of the Course, The Students will be able to **explain different number systems**, the historical evolution of computers, and identify key components in a block diagram.
2. Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge**.
3. Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.
4. Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations**.
5. Learners will apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Unit 1. Number Systems, Evolution , Block Diagram and Generations:

Number Systems: Binary, Decimal, Octal, Hexadecimal; conversions between number systems.

Evolution of Computers: History from early mechanical devices to modern-day systems.

Block Diagram of a Computer: Components like Input Unit, Output Unit, Memory, CPU (ALU + CU).

Generations of Computers: First to Fifth Generation – technologies, characteristics, examples.

Unit 2. Basic organization and N/W fundamentals:

Computer Organization: Functional components – Input/Output devices, Storage types, Memory hierarchy.

Types of Computers: Micro, Mini, Mainframe, and Supercomputers.

Networking Fundamentals: Definition, need for networks, types (LAN, WAN, MAN), topology (Star, Ring, Bus).

Internet Basics: IP Address, Domain Name, Web Browser, Email, WWW.

Unit 3. Word Processing and presentations:

Word Processing Basics: Using MS Word/Google Docs – formatting, styles, tables, mail merge.

Presentation Tools: Using PowerPoint/Google Slides – slide design, animations, transitions.

Applications: Creating resumes, reports, brochures, and presentations.

Keyboard Shortcuts

Unit 4. Spreadsheet Basics:

Spreadsheet Concepts: Understanding rows, columns, cells in tools like MS Excel/Google Sheets, cell referencing.

Functions and Formulae: SUM, AVERAGE, IF, COUNT.

Charts and Graphs: Creating visual representations

Data Handling: Sorting, filtering, conditional formatting.

Text Functions: LEFT, RIGHT, MID, LEN, TRIM, CONCAT, TEXTJOIN

Advanced Functions: Logical: IF, AND, OR, IFERROR, **Lookup:** VLOOKUP, HLOOKUP, XLOOKUP, INDEX, MATCH

Unit 5. Data Analysis and Visualization:

Conditional Formatting: Custom rules, Color scales, Icon sets, Data bars

Data Analysis Tools: Pivot Tables and Pivot Charts, Data Validation (Drop-downs, Input Messages, Error Alerts), What-If Analysis: Goal Seek, Scenario Manager, Data Tables

Charts and Dashboards: Creating Interactive Dashboards, Using slicers with Pivot Tables, Combo Charts and Sparklines

Productivity Tips: Using Named Ranges, Freeze Panes, Split View

Textbooks:

1. **Fundamentals of Computers**, Reema Thareja, Oxford University Press, Second Edition
2. **Fundamentals of Computers**, V. Rajaraman – PHI Learning
3. **Introduction to Computers** by Peter Norton – McGraw Hill
4. **Microsoft Office 365 In Practice** by Randy Nordell – McGraw Hill Education

References:

1. **Excel 2021 Bible** by Michael Alexander, Richard Kusleika – Wiley
2. **Networking All-in-One For Dummies** by Doug Lowe – Wiley
3. **Microsoft Official Docs and Training:** <https://learn.microsoft.com>
4. **Google Workspace Learning Center:** <https://support.google.com/a/users/>

Activities:

Outcome: At the End of the Course, The Students will be able to **explain different number systems**, the historical evolution of computers, and identify key components in a block diagram.

Activity: Create a digital poster or infographic comparing number systems (binary, decimal, octal, hexadecimal) and illustrating the timeline of computer generations with key innovations.

Evaluation Method: Rubric-based assessment of the poster presentation on a 10-point scale focusing on:

- Accuracy of number system conversions
- Correct identification of block diagram components
- Visual organization and creativity

Outcome: Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge.**

Activity: Design a concept map showing the internal architecture of a computer and types of networks (LAN, WAN, MAN), including devices and topologies.

Evaluation Method: Checklist-based peer review and instructor validation:

- Completeness of the map
- Correctness of networking concepts

- Use of appropriate terminology
- Logical flow and structure of the map

Outcome: Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.

Activity: Prepare a formal report (e.g., project proposal) in a word processor and present it using a slide deck with transitions, embedded media, and design elements.

Evaluation Method: Performance-based evaluation using a 10-point scoring scale:

- Formatting and structure of the document
- Presentation aesthetics and clarity
- Communication skills during presentation

Outcome: Learners will manipulate data within spreadsheets, apply formulas, and generate accurate summaries and visualizations.

Activity: Analyze a dataset (e.g., student scores or sales data) using spreadsheet software. Apply formulas (SUM, AVERAGE, IF, VLOOKUP) and create relevant charts.

Evaluation Method: Practical test with a rubric:

- Correct use of formulas
- Accuracy of data summarie

Outcome: Learners will apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Activity: Prepare an interactive dashboard for a given data set using EXCEL.

Evaluation Method: Evaluation of the dashboard on a 10-point scoring scale:

- Presentation aesthetics and clarity
- Interactiveness
- Communication skills during presentation

SEMESTER-III

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Demonstration of Assembling and Desassembling of Computer Systems.
2. Identify and prepare notes on the type of Network topology of your institution.
3. Prepare your resume in Word.
4. Using Word, write a letter to your higher official seeking 10-days leave.
5. Prepare a presentation that contains text, audio and video.
6. Using a spreadsheet, prepare your class Time Table.
7. Using a Spreadsheet, calculate the Gross and Net salary of employees (Min 5) considering all the allowances.
8. Generate the class-wise and subject-wise results for a class of 20 students. Also generate the highest and lowest marks in each subject.
9. Using IF, AND, OR, and IFERROR to Automate Grade Evaluation.
 - a. Create a table of student scores in different subjects.
 - b. Use IF to assign grades (A/B/C/Fail).
 - c. Use IFERROR to handle missing scores or invalid data.
10. *Employee Database Search Using VLOOKUP, HLOOKUP, XLOOKUP, INDEX, and MATCH*
 - a. Create a database of employees (Name, ID, Department, Salary).
 - b. Implement VLOOKUP to search by employee ID.
 - c. Use HLOOKUP to extract department heads by role.
 - d. Apply XLOOKUP for more flexible searches.
 - e. Use INDEX + MATCH as an alternative to VLOOKUP.
11. Sales Report Analysis Using Pivot Tables and Charts
 - a. Use a dataset of product sales (Product, Region, Date, Quantity, Revenue).
 - b. Create Pivot Tables to summarize data by region/product.
 - c. Insert Pivot Charts for visual analysis (e.g., bar, line).
 - d. Add slicers to make the dashboard interactive.
12. Designing a Data Entry Form with Drop-downs and Input Rules
 - a. Create a student registration form.
 - b. Add drop-down lists for course selection using Data Validation.

- c. Add input messages to guide users.
- d. Add error alerts for wrong entries.

13. Monthly Budget Planning using Goal Seek and Scenario Manager

- a. Create a simple personal budget (income, expenses, savings).
- b. Use Goal Seek to determine income needed to save a desired amount.
- c. Use Scenario Manager to compare different budgeting scenarios (best/ worst/ realistic case).
- d. Create a one-variable Data Table to analyze how different expenses affect savings.

14. Dashboard Creation Using Combo Charts, Sparklines & Slicers

- a. Use existing sales or attendance data.
- b. Insert combo charts (e.g., column + line).
- c. Add sparklines to show trends.
- d. Use slicers with Pivot Tables to control dashboard elements.
- e. Finalize and format for interactivity.

SEMESTER-IV

COURSE 2: PYTHON PROGRAMMING AND DATA STRUCTURES

Theory

Credits: 3

3 hrs/week

Course Objectives

1. To introduce the fundamentals of Python programming, including environment setup, syntax, and core concepts.
2. To develop problem-solving skills using control flow, functions, and modules.
3. To provide knowledge of Python data structures, file handling, and exception handling for effective programming.
4. To impart object-oriented programming concepts and GUI development skills for building applications.

Course Outcomes (COs)

After successful completion of the course, students will be able to:

1. Explain the basic features, syntax, data types, and operators of Python programming.
2. Apply control flow constructs, functions, and modules to develop structured Python programs.
3. Demonstrate the use of sequences, sets, and dictionaries for effective data handling and manipulation.
4. Implement file handling techniques and apply exception handling mechanisms for robust applications.
5. Develop object-oriented and GUI-based applications using Python.

Unit 1: Basics of Python Programming:

Introduction to Python, Features of Python, Programming Modes - Interactive Mode & Script Mode, Identifiers, Naming Conventions, Keywords (Reserved Words), Built-in Data Types, Literals - Integer, Float, Complex, Boolean, String, Variables, Operators, Expressions, Assignment Statements, Input/Output Statements, Python Syntax (Lines, Comments, Indentation),

Operators & Operands, Classification of Operators - Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Assignment, Augmented Assignment, Identity Operators, Expressions & Precedence Rules

Unit 2: Control Flow, Functions & Modules:

Control Flow - if Statement, if-else, if-elif-else. Iterative Statements – while, for, Nested Loops, Loop Control Statements – break, continue, pass; else with loops

Need for Functions, Defining & Invoking User-defined Functions, Return Statement, Function Input/Output Cases, Scope of Variables - Local, Global, Nested Functions, Function Arguments - Required, Positional, Default, Variable-length, main() Function, Documentation Strings, Recursive Functions, Anonymous Functions (Lambda), Library Functions

Modules - Import, from..import, Creating & Using Modules, Namespaces

Unit 3: Core Data Types and Python Collections

Strings: Representation, Indexing, Slicing, Immutability, Operators, Methods, Formatting

Lists: Creation, Indexing, Slicing, Mutability, Common Methods, List Comprehension

Tuples: Immutability, Operations, Tuple Assignment

Sets and Frozensets: Methods, Mathematical Operations, Comprehension

Dictionaries: Key-Value Structure, Methods, Traversal, Nested Dictionaries

Unit 4: File Handling, Exception Management & Object-Oriented Programming

File Handling: Types, Opening, Reading, Writing, Closing, CSV Files, OS/Pathlib

Error Types, Exception Handling: try-except, raise, User-defined Exceptions, Assertions

OOP Concepts: Classes, Objects, Attributes, Methods, Constructor and Destructors

Encapsulation: Private and Public Members

Inheritance: Single, Multilevel, Multiple, Method Overriding

Unit 5: Abstract Data Structures and GUI Programming

Abstract Data Structures (ADTs): Concepts and Importance

Linked Lists: Definition, Types- Singly, Doubly, Circular; Node Structure, Insertion, Deletion, Traversal (Single Linked list implementation only)

Stacks: LIFO Principle, Implementation using List, Applications

Queues: FIFO Principle, Implementation using List, Priority Queues

GUI Programming with Tkinter: Widgets (Label, Button, Entry, Menu, Listbox, Canvas etc.), Event Handling, Building Simple GUI Apps

Textbooks:

1. Python Programming-An Object Oriented approach, Anita Goel, Universities Press
2. Python Programming using Problem Solving Approach Reema Thareja Oxford University Press 2020
3. Exploring Python, Budd T A, McGraw-Hill Education, 1st Edition, 2011.

Reference Book:

1. Python: The Complete Reference, Martin C. Brown, Mc Graw-Hill, 2018
2. Fundamentals of Python, Kenneth A. Lambert. (2019), First Programs,2nd Edition, CENGAGE Publication.

Activities:

Outcome: Explain the basic features, syntax, data types, and operators of Python programming.

Activity: Conduct a "Python Basics Lab" where students write small programs to demonstrate literals, variables, data types, and operators (e.g., swapping numbers, simple calculator).

Evaluation Method:

- Lab performance checklist (execution of 3 mini tasks)
- Short quiz with multiple-choice and fill-in-the-blanks on syntax, data types, and operators

Outcome: Apply control flow constructs, functions, and modules to develop structured Python programs.

Activity: Group activity - "Python Problem Solving Challenge": Students solve real-life problems (e.g., finding prime numbers, grade calculator, menu-driven calculator) using control structures, functions, and importing standard modules.

Evaluation Method:

- Code submission with proper use of functions/modules (20%)
- Viva-voce to explain logic and flow of control (40%)
- Unit test with scenario-based programming questions (40%)

Outcome: Demonstrate the use of sequences, sets, and dictionaries for effective data handling and manipulation.

Activity: Hands-on mini project – "Student Data Manager": Students create a program using lists, tuples, sets, and dictionaries to store and manipulate student records (e.g., marks, courses, hobbies).

Evaluation Method:

- Practical demo of program with at least 5 data operations (add, search, delete, update, traverse)
- Evaluation rubric for correctness, efficiency, and use of appropriate data structure

Outcome: Implement file handling techniques and apply exception handling mechanisms for robust applications.

Activity: Individual assignment - "File-Based Address Book": Students create a program to store, update, and retrieve data from files, with exception handling for invalid inputs or missing files.

Evaluation Method:

- Assessment of program correctness (file read/write, append, delete, exception handling)
- Short quiz with error-tracing and debugging questions (given code with errors, students identify and correct)

Outcome: Develop object-oriented and GUI-based applications using Python.

Activity: Mini Project – "Student Information System with GUI": Students design a simple Tkinter-based application with classes/objects for handling student data, including basic GUI widgets (Entry, Button, Listbox).

Evaluation Method:

- Project demo and presentation (50%)
- Rubric-based evaluation for OOP concepts (classes, inheritance, encapsulation) and GUI design (widgets, event handling) (30%)
- Peer review/feedback on usability (20%)

SEMESTER-IV

COURSE 2: PYTHON PROGRAMMING AND DATA STRUCTURES

Practical

Credits: 1

2 hrs/week

1. Basic Python Programs:
 - a. Write a program to display basic details (name, roll number, department) using print() and demonstrate different literal types (int, float, string, boolean, complex).
 - b. Write a program to perform arithmetic, relational, logical, bitwise, and assignment operations on given inputs.
2. Control Flow Practice
 - a. Write a program to find the largest of three numbers using if-elif-else.
 - b. Write a program to check whether a number is prime or not using loops.
 - c. Write a program to illustrate the use of loop control statements (break, continue, pass).
3. Functions and Recursion
 - a. Write a program to define a function to calculate factorial of a number (using recursion).
 - b. Write a program to demonstrate different types of function arguments (default, positional, keyword, variable-length).
4. Write a program to illustrate string slicing, concatenation, repetition, and built-in methods.
5. Write a program to create a list of numbers, perform insertion, deletion, searching, sorting, and list comprehension.
6. Write a program to demonstrate tuple packing, unpacking, and immutability.
7. Write a program to implement set operations (union, intersection, difference, subset, superset).
8. Write a program to create a dictionary of student roll numbers and marks, and perform add, update, delete, and traversal operations.
9. Write a program to read and display count of vowels, consonants, digits, and spaces of a text file.
10. Write a program to copy the contents of one file into another file.

11. Write a program to read and process student marks from a CSV file (calculate average, highest, lowest).
12. Write a program to demonstrate exception handling using try-except-finally.
13. Write a program to create a class Student with attributes and methods to display details.
14. Write a program to demonstrate single and multilevel inheritance.
15. Implement stack (LIFO) and queue (FIFO) using lists.
16. Implement singly linked lists: node creation, insertion, deletion, traversal.
17. Write a Tkinter program with Label, Entry, and Button widgets to take user input and display it.
- 18.** Write a Tkinter program to create a simple calculator application.

SEMESTER-V

COURSE 3: ARTIFICIAL & COMPUTATIONAL INTELLIGENCE

Theory

Credits: 3

3 hrs/week

Course Objectives

- Introduce foundational concepts and history of Artificial Intelligence (AI).
- Teach the PEAS framework and agent-based problem solving.
- Develop understanding of uninformed and informed search methods.
- Provide basic knowledge of machine learning concepts and computational intelligence.
- Instill awareness of ethical considerations in AI.
- Introduce logic programming fundamentals using Prolog as a practical tool.

Course Outcomes

Students completing this course will be able to:

- Explain basic AI terminology, agent models, and the PEAS framework.
- Formulate AI problems using search strategies and implement basic algorithms conceptually.
- Understand core machine learning categories and their applications.
- Describe computational intelligence approaches and their role in AI.
- Analyze ethical issues related to AI technologies.
- Write and execute simple Prolog programs demonstrating logic programming fundamentals.

Unit 1: Introduction to Artificial Intelligence and PEAS Framework

Introduction to AI: Definition, history, applications, and scope

The PEAS framework: Performance Measure, Environment, Actuators, Sensors, Examples of PEAS in real-world AI systems

Intelligent agents: Intelligent agents and their environments, **Types of intelligent agents:** Simple reflex, model-based, goal-based, utility-based, Agent architectures and rationality

Unit 2: Expert Systems

Definition and components of Expert Systems (Knowledge Base, Inference Engine, User Interface), Rule-based systems and knowledge representation, Examples of expert systems: medical diagnosis, decision support, Limitations and comparison with AI agents, Role and significance of expert systems in AI evolution

Unit 3: Search Strategies in AI

Problem-solving as search: problem formulation, states, actions, goal test, Traveller's problem

Uninformed (Blind) Search: Breadth-first search, Depth-first search, Uniform-cost search

Informed (Heuristic) Search: Greedy best-first search, A* algorithm

Applications of search in AI problems

Unit 4: Introduction to Machine Learning

What is machine learning, definitions, Types of learning: Supervised, Unsupervised, Reinforcement learning (basic ideas), classification, Regression, clustering and Association, Basic learning algorithms overview and applications

Unit 5: Computational Intelligence and Ethics in AI

Overview of computational intelligence (Basics of fuzzy logic, neural networks), Role of computational intelligence in AI, Ethics and societal challenges in AI, Responsible AI, fairness, transparency, and safety concerns

Unit 1: Introduction to AI, PEAS Framework & Intelligent Agents

Activities:

Interactive lecture with real-world AI examples

Group discussion on PEAS framework for different environments

Case study analysis of intelligent agents in everyday AI applications

Quiz on AI fundamentals and agent types

Outcomes:

Students will explain AI basics and describe the PEAS components.

Students will identify types of intelligent agents and their roles.

Evaluation Method:

Quiz and participation: 10%

Assignment on PEAS and agent modeling: 10%

Unit 2: Expert Systems

Activities:

Lecture with multimedia explaining components of expert systems

Hands-on group activity designing rule-based systems for simple decision problems

Case study review of medical diagnosis expert systems

Class debate on limitations and advantages of expert systems

Outcomes:

Understand expert system architecture and knowledge representation.

Develop simple rule-based systems for decision making.

Analyze real-world expert system examples critically.

Evaluation Method:

Group assignment designing rule base: 15%

Written test on expert system concepts: 10%

Unit 3: Search Strategies in AI

Activities:

Demonstration of uninformed search algorithms

Interactive exercises formulating search problems

Simulation of heuristic search and A* algorithm

Problem-solving sessions applying search to puzzles

Outcomes:

Formulate search problems and apply uninformed and heuristic algorithms conceptually.

Explain the working and use cases of different search strategies.

Evaluation Method:

Problem formulation assignment: 10%

Class test on search algorithms: 10%

Unit 4: Basics of Machine Learning

Activities:

Video lectures on different machine learning paradigms

Simple data classification exercises/classification demos

Group presentations on various ML types and applications

Quiz on ML types and terminology

Outcomes:

Explain supervised, unsupervised, and reinforcement learning basics.

Identify sample use cases for machine learning.

Evaluation Method:

Quiz and presentation: 10%

Written assignment on ML overview: 5%

Unit 5: Computational Intelligence & Ethics**Activities:**

Lecture introducing fuzzy logic, neural networks overview

Ethical dilemma discussions and case studies in AI technology

Role play on AI responsibility and fairness

Group project: Develop guidelines for responsible AI use

Outcomes:

Understand computational intelligence basics and ethical AI challenges.

Propose responsible AI practices for diverse scenarios.

Evaluation Method:

Participation in ethics discussions: 5%

Group project report and presentation: 15%

Recommended Textbooks and References

- Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition
- Elaine Rich, Kevin Knight, Artificial Intelligence, 3rd Edition
- Michael Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems
- Ivan Bratko, Prolog Programming for Artificial Intelligence, 4th Edition
- Online resources: AI course materials from Coursera, NPTEL, GeeksforGeeks AI tutorials.

SEMESTER-V

COURSE 3: ARTIFICIAL & COMPUTATIONAL INTELLIGENCE

Practical

Credits: 1

2 hrs/week

Note: Experiments have to be conducted using Prolog

1. Introduction to Prolog Environment and Syntax
 - Setting up Prolog, understanding facts, rules, and queries.
2. Defining Simple Facts and Queries
 - Write and test simple facts like family relationships, likes/dislikes.
3. Creating Rules in Prolog
 - Define logical rules with conditions and test queries.
4. List Handling in Prolog
 - Write programs to manipulate lists (head, tail, concatenation).
5. Recursion in Prolog
 - Implement recursive relations such as factorial and Fibonacci.
6. Search and Backtracking
 - Demonstrate Prolog's backtracking with sample queries and control cuts.
7. Family Relationship Programs
 - Model family trees and query relationships like siblings, ancestors.
8. Solve the Eight Queens Problem
 - Classic AI problem solved using backtracking.
9. Implement Simple Arithmetic Operations
 - Addition, subtraction, multiplication using Prolog predicates.
10. Monkey and Banana Problem
 - Logic problem modeling and solution.
11. Basic Expert System Prototype
 - Write rules for a simple medical diagnosis or recommendation system.
12. Implement Search Algorithms (Conceptual)
 - Demonstrate basic search algorithms like best-first search using Prolog rules.

SEMESTER-V
COURSE 4: DATA SCIENCE WITH R

Theory

Credits: 3

3 hrs/week

Course Objectives

1. Introduce the data science process, lifecycle, and applications in real-world domains.
2. Build proficiency in R programming for data manipulation, exploration, and visualization.
3. Train students in handling structured, unstructured, and time-based data effectively.
4. Familiarize with basic machine learning and statistical modeling using R.
5. Develop awareness of ethical, interpretability, and responsible use of data science.

Course Outcomes

At the end of the course, students will be able to:

1. Explain the Data Science process and perform EDA (Exploratory Data Analysis).
2. Write R programs using variables, functions, loops, and packages for basic analytics.
3. Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr, ggplot2).
4. Build and evaluate basic machine learning models such as regression and clustering.
5. Apply data science techniques to practical case studies.

Unit 1. Introduction to Data Science Process:

Introduction- Definition - Data Science in various fields - Examples - Impact of Data Science - Data Analytics Life Cycle - Data Science Toolkit - Data Scientist - Data Science Team, Exploratory Data Analysis (EDA), Feature Engineering & Data Transformation

Unit 2. Basics of R Programming:

Introduction to R and RStudio, Data Types, Variables, Operators, Control Structures (if, loops, apply), Functions and Packages, Data Input/Output (CSV, Excel, XML, JSON).

Unit 3. Data Handling & Visualization in R:

Data Frames, Lists, Matrices, Data Wrangling with dplyr and tidyr, Handling Missing Data, Working with Date/Time in R. Visualization with ggplot2: grammar of graphics, aesthetics, geometries, scales.

Faceting and layering techniques, Visualizing categorical and numerical data, Customizing and exporting plots

Unit 4. Applications & Case Studies in Data Science:

Simple Linear Regression, Multiple Regression

Model Evaluation Method: Accuracy, Confusion Matrix, ROC.

K-Means Clustering, Text Mining & Word Clouds, Recommender Systems Basics, Ethical Issues in Data Science

Unit 5. Advanced Topics in Data Science with R :

Introduction to Time Series Analysis in R (ARIMA basics)- Concept of time series (trend, seasonality, noise), Time series objects in R (ts, zoo, xts), Plotting and decomposing time series, Stationarity and differencing, Autocorrelation & Partial Autocorrelation (ACF/PACF), AR, MA, ARIMA model basics, Forecasting using forecast package

Creating interactive visualizations with plotly packages- Converting ggplot2 plots to interactive plots

Animations and sliders in plotly

R Shiny: Building interactive web applications- Introduction to Shiny framework, UI and server functions, Reactive expressions and reactivity in Shiny, Input and output widgets (sliders, dropdowns, text), Layouts and dashboard design

Textbooks

1. An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 2nd Edition, 2021
2. R for Data Science, Hadley Wickham and Garrett Golemund, O'Reilly Media, 2017.

Reference Books

1. The Art of R Programming, Norman Matloff, No Starch Press, 2011.
2. Modern Applied Statistics with S, W.N. Venables & B.D. Ripley, Springer, 2002.
3. Introduction to Data Science: Data Analysis and Prediction Algorithms with R, Rafael A. Irizarry, CRC Press, 2020.
4. Data Science from Scratch: First Principles with Python (for conceptual clarity only), Joel Grus,

Activities:

Outcome: Explain the Data Science process and perform EDA (Exploratory Data Analysis).

Activity: Use a real-world dataset (e.g., Titanic or COVID data) to:

- Outline the steps of the Data Science workflow
- Perform EDA using summary statistics and visualizations (histograms, boxplots, scatterplots)

Evaluation Method: Presentation and checklist (10-point scale):

- Clear explanation of workflow stages
- Quality of EDA insights
- Use of appropriate plots and summaries

Outcome: Write R programs using variables, functions, loops, and packages for basic analytics.

Activity: Write an R script that:

- Reads a CSV file
- Uses if, for, and while loops
- Defines and calls custom functions with arguments and return values

Evaluation Method: Code review and execution test to verify (10-point scale):

- Correctness of the syntax and logic
- Functionality of control structures
- Output accuracy and modularity

Outcome: Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr, ggplot2).

Activity: Clean a messy dataset using:

- dplyr for filtering, selecting, and mutating
- tidyr for reshaping and handling missing values
- Time-based operations (e.g., filling gaps, formatting dates)

Evaluation Method: Before-and-after comparison (10 point score):

- Completeness of cleaning steps
- Use of appropriate functions
- Handling of missing/time data

Outcome: Implement basic machine learning models and evaluate performance using appropriate metrics and visual tools.

Activity: Build a simple classification model (e.g., logistic regression or decision tree) using

R:

- Train/test split
- Predict outcomes
- Evaluate using confusion matrix, accuracy, precision, recall

Evaluation Method: Model report and demo (10 point scale):

- Correct implementation of model
- Use of evaluation metrics

SEMESTER-V
COURSE 4: DATA SCIENCE WITH R

Practical

Credits: 1

2 hrs/week

List of Practicals:

1. Compute Mean, Median, Mode, Variance, and Standard Deviation
2. Visualize Binomial, Normal, and Poisson Distributions
3. Perform t-test and Chi-Square Test in R
4. Calculate Correlation and Build a Simple Linear Regression Model
5. Conduct Exploratory Data Analysis (EDA) on a Real-World Dataset
6. Apply Feature Engineering: Scaling, Normalization, and Encoding
7. Practice R Programming: Variables, Control Structures, and Functions
8. Read and Write Data from CSV, Excel, JSON, and XML Files
9. Use dplyr and tidyr for Data Wrangling Tasks
10. Handle Missing Data and Detect Outliers
11. Work with Dates and Times in R
12. Visualize Data Using ggplot2 (Bar, Scatter, Histogram, Boxplot)
13. Perform K-Means Clustering and Visualize Clusters
14. Evaluate Models Using Confusion Matrix, Accuracy, and ROC Curve
15. Perform Text Mining and Create a Word Cloud
16. Time Series Forecasting with ARIMA on a real dataset (e.g., monthly airline passengers, stock prices, or temperature data).
17. Create interactive bar, line, and scatter plots using plotly. On a real dataset (e.g., COVID-19 cases, sales data, or student marks).
18. Develop a Shiny app that lets users upload a CSV file.

SEMESTER-VI

COURSE 5: EXPLORATORY DATA ANALYSIS & DATA VISUALIZATION

Theory

Credits: 3

3 hrs/week

Course Objectives

1. To introduce the concepts, importance, and workflow of exploratory data analysis.
2. To impart practical skills in data preprocessing, cleaning, and manipulation using Python.
3. To equip students with techniques in univariate, bivariate, and multivariate data analysis.
4. To develop proficiency in data visualization and interpretation using standard Python libraries.
5. To enable students to build reports and dashboards to communicate data insights effectively.

Course Outcomes

After completing this course, students will be able to:

1. Explain the importance and process of EDA in the data science pipeline.
2. Perform robust data preprocessing including missing value handling, outlier detection, transformation, and encoding.
3. Manipulate, filter, and reshape datasets using Python libraries like Pandas and NumPy.
4. Conduct data analysis and visualize results using Matplotlib, Seaborn, and Plotly.
5. Design interactive visualizations and dashboards to communicate findings in real-world scenarios.

Unit 1: Introduction to Exploratory Data Analysis (EDA)

Significance and objectives of EDA, Types and scales of data (nominal, ordinal, interval, ratio), Concepts of variability and central tendency, Role of EDA in the data science pipeline, Introduction to data quality and data issues, Differences between EDA, classical, and Bayesian analysis, Basic terminology and concepts

Unit 2: Data Preprocessing and Cleaning

Overview of data preprocessing, Handling missing types of missingness (MCAR, MAR, MNAR), Techniques for missing data imputation (mean, median, mode, forward/backward

fill), Detection and treatment of duplicates, Outlier detection methods (Z-score, IQR method), Data transformation: normalization and standardization, Encoding categorical variables: label encoding, one-hot encoding, Data integration and reduction techniques, Importance of preprocessing for machine learning models

Unit 3: Data Manipulation using Python with Pandas and NumPy

NumPy Array Creation: Use `np.array()`, `np.zeros()`, `np.ones()` to create arrays of various dimensions. **Array Manipulation:** Reshape, transpose, flatten arrays for flexible data formats.

Element-wise Arithmetic Operations, Statistical Functions on NumPy Arrays

Indexing & Slicing: Access array elements via slicing, masking, and fancy indexing for selective data analysis.

Pandas DataFrames: Series and Data frames, Create from dictionaries, Series, CSV, Excel, and JSON files using Dataframe or reader functions

Data Frame Manipulation: Indexing & Selection - Label-based and Position-based indexing; conditional filtering with boolean masks, Aggregation & Grouping- Group data by columns with `.groupby()`, summarize using `.sum()`, `.mean()`, `.count()`.

Data Reshaping & Pivoting and stack/unstack hierarchical index data.

Merging & Joining Data Frames

Datetime & Categorical Data: Handle and transform temporal data with `pd.to_datetime()`, categorical conversions with `.astype('category')`.

Unit 4: Basic Data Analysis and Visualization

What is visualization, what is its importance? Types: histograms, bar charts, Boxplots and violin plots for data distribution, Scatter plots and correlation analysis (Pearson, Spearman), Cross-tabulation and contingency tables, Analyzing relationships: correlation vs causation, Visualizing categorical vs numerical data

Unit 5: Advanced Visualization and Reporting

Principles of effective graphical representation, Introduction to visualization libraries: Matplotlib, Seaborn, Plotly, Advanced plots: heatmaps, pair plots, joint plots, Interactive visualizations and dashboards, Geographical data visualization basics, Storytelling with using visualization for communication, Introduction to reporting tools and best practices

Activities

Unit 1: Introduction to EDA

Activity:

Analyze the Titanic dataset to explore data types, central tendency, and variability.^[1]^[SEP]

Outcome:

Develop a foundational understanding of data structures and EDA significance.

^[1]^[SEP]Evaluation Method:

Lab notebook submission and oral quiz.

Unit 2: Data Preprocessing and Cleaning

Activity:

Clean a health survey dataset: identify/impute missing values, standardize columns, encode categories, remove duplicates/outliers.^[1]^[SEP]

Outcome:

Demonstrate proficiency in data cleaning and preprocessing techniques in Python.^[1]^[SEP]

Evaluation Method:

Script submission and report on preprocessing steps with before/after summary.

Unit 3: Data Manipulation using Python

Activity:

Manipulate a retail transaction dataset by indexing, filtering, grouping, merging, and handling date/time columns.^[1]^[SEP]

Outcome:

Apply multiple Pandas/NumPy techniques for dataset transformation.

Evaluation Method:

Graded Jupyter notebook, peer code review.

Unit 4: Basic Data Analysis and Visualization

Activity:

Explore demographic data with descriptive statistics, create histograms, boxplots, scatter plots, and cross-tabulations.^[1]^[SEP]

Outcome:

Visualize and interpret univariate and bivariate data insights.^[1]^[SEP]

Evaluation Method:

Visualization notebook and critical analysis write-up.

Unit 5: Advanced Visualization and Reporting

Activity:

- Build a dashboard on sales or air quality data using Matplotlib/Seaborn/Plotly.
- Incorporate interactive and geospatial visualizations.^[11]_{SEP}Outcome:
- Develop professional dashboards and communicate insights through visual storytelling.

Evaluation Method:

Final project dashboard evaluation and presentation.

Preferred Textbooks

1. Suresh Kumar Mukhiya, Usman Ahmed, “Hands-On Exploratory Data Analysis with Python”, Packt Publishing, 2020.
2. Jake Vander Plas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.
3. Catherine Marsh, Jane Elliott, “Exploring Data: An Introduction to Data Analysis for Social Scientists”, Wiley, 2008.

References

1. Eric Pimpler, “Data Visualization and Exploration with R”, GeoSpatial Training Service, 2017.
2. Claus O. Wilke, “Fundamentals of Data Visualization”, O’Reilly, 2019.
3. Matthew O. Ward, Georges Grinstein, Daniel Keim, “Interactive Data Visualization: Foundations, Techniques, and Applications”, CRC Press, 2015.

SEMESTER-VI

COURSE 5: EXPLORATORY DATA ANALYSIS & DATA VISUALIZATION

Practical

Credits: 1

2 hrs/week

Note: This lab has to be executed using interactive computing environments like Jupyter Notebook or Google Colab.

1. Importing Data and Basic Exploration

- Load datasets from CSV/Excel/JSON
- Use `.head()`, `.info()`, `.describe()`, `.shape` to examine structure and summary statistics

2. Examining Data Types and Missing Values

- Identify data types for each column
- Detect and quantify missing values with `.isnull()` and `.sum()`.

3. Handling Missing Data Techniques

- Impute missing values using mean, median, mode, forward, and backward fill
- Drop rows/columns with missing data and compare results

4. Dealing With Duplicates and Outliers

- Find and remove duplicate entries
- Detect outliers using Z-score or IQR methods
- Treat or remove outliers and compare distributions

5. Data Transformation and Scaling

- Normalize, standardize, log-transform columns
- Encode categorical variables (Label Encoding, One-Hot Encoding)

6. Indexing, Filtering, and Slicing DataFrames

- Select specific rows/columns using index and Boolean filters
- Conditional selections and advanced slicing

7. Aggregating, Grouping, and Pivoting Data

- Use `.groupby()` for aggregate statistics
- Create pivot tables for categorical summaries

8. Merging and Concatenating Multiple Datasets

- Join, merge, concatenate DataFrames
- Handle merging keys and missing arguments

9. Univariate Visualization Techniques

- Create histograms, boxplots, violin plots for distributions

- Visualize frequency and central tendency

10. Bivariate and Multivariate Visualization

- Display scatter plots, correlation matrices, pair plots
- Visualize relationships and patterns

11. Time Series Data Exploration

- Parse and index datetime columns
- Visualize trends and seasonality

12. Interactive Visualizations and Dashboards

- Build simple interactive graphs with Plotly
- Design a dashboard showing multiple aspects of a dataset

SEMESTER-VI
COURSE 6: MACHINE LEARNING

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. Understand fundamental concepts, types, and applications of machine learning.
2. Develop, evaluate, and optimize machine learning models through preprocessing, training, and feature engineering techniques.
3. Apply supervised and unsupervised learning algorithms to real-world problems using appropriate tools and methods.

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Describe various machine learning paradigms, data types, and the overall structure of a machine learning pipeline.
2. Perform data preprocessing, feature engineering, and evaluate models using appropriate metrics.
3. Implement and analyze supervised learning algorithms for regression and classification tasks.
4. Apply unsupervised learning techniques for clustering and identify suitable machine learning approaches for specific application domains

Unit 1. Introduction to Machine Learning:

Introduction to Machine Learning: Types of human learning, What is machine learning?, Types of machine learning: supervised, unsupervised, semi-supervised and reinforcement learning, machine learning activities, applications of machine learning. Types of data in machine learning, Structure of data

Unit 2. Model Preparation, Evaluation and feature engineering:

Data pre-processing, Model selection and training (for supervised learning), Model representation and interpretability, Evaluating machine learning algorithms and performance enhancement of models. What is feature engineering?, Feature transformation, Feature subset selection. Principal component analysis.

Unit 3. Supervised Learning-Regression:

Regression: Introduction of regression, Regression algorithms: Simple linear regression, Multiple linear regression, Polynomial regression model, Logistic regression, Maximum likelihood estimation.

Unit 4. Supervised Learning- Classification:

Introduction of supervised learning, Classification model and learning steps, Classification algorithms: Naïve Bayes classifier, k-Nearest Neighbour (kNN), Decision tree, Support vector machines, Random Forest.

Unit 5. Unsupervised Learning:

Introduction of unsupervised learning, Unsupervised vs supervised learning, Application of unsupervised learning, Clustering and its types, Partitioning method: k-Means and KMedoids, Hierarchical clustering, Density-based methods - DBSCAN.

Case-study of ML applications: Image recognition, speech recognition, Email spam filtering, Online fraud detection and other.

Textbooks:

1. Introduction to Machine Learning, Ethem Alpaydin, MIT Press, Fourth Edition, 2020.
2. Machine Learning: Theory and Practice, M N Murthy, V.S Ananthanarayana, Universities press
3. Machine Learning, S. Sridhar, M. Vijayalakshmi, Second Edition, Oxford University Press

Reference Books:

1. Machine Learning: An Algorithmic Perspective, Second Edition, Stephen Marsland, CRC Press, 2014
2. Machine Learning, Tom Mitchell, McGraw Hill, 3rd Edition.
3. Python Machine Learning, Sebastain Raschka, Vahid Mirjalili , Packt publishing 3rd Edition, 2019.

Activities:

Outcome: Describe various machine learning paradigms, data types, and the overall structure of a machine learning pipeline.

Activity: Prepare a detailed comparative report/chart explaining supervised, unsupervised, and reinforcement learning paradigms, data types, and step-by-step machine learning pipeline stages.

Evaluation Method: Rubric-based assessment evaluating completeness, clarity, correctness, and presentation quality - scored on a 10-point scale.

Outcome: Perform data preprocessing, feature engineering, and evaluate models using appropriate metrics.

Activity: Conduct a hands-on lab exercise using a real dataset to perform data cleaning, normalization, feature extraction/selection, and evaluate model performance using metrics like accuracy, precision, recall, and F1-score.

Evaluation Method: Practical assessment including code correctness, applied techniques, and interpretation of evaluation metrics; assessed with a rubric out of 10.

Outcome: Implement and analyze supervised learning algorithms for regression and classification tasks.

Activity: Implement at least two supervised learning algorithms (e.g., Linear Regression and Decision Trees) to solve prediction tasks, followed by comparative analysis of their performance on test datasets.

Evaluation Method: Code and report evaluation focusing on implementation accuracy, performance comparison, and analysis depth; scored on a 10-point rubric.

Outcome: Apply unsupervised learning techniques for clustering and identify suitable machine learning approaches for specific application domains.

Activity: Perform clustering (e.g., K-Means, Hierarchical) on a given dataset and prepare a case study selecting and justifying machine learning methods suited for different application scenarios.

Evaluation Method: Lab practical combined with a written case study; assessed for correct algorithm application, cluster interpretation, and justification of approach - evaluated on a 10-point rubric.

SEMESTER-VI
COURSE 6: MACHINE LEARNING

Practical

Credits: 1

2 hrs/week

Lab Experiments:

1. Write a python program to import and export data using Pandas library functions.
2. Demonstrate various data pre-processing techniques for a given dataset
3. Implement Dimensionality reduction using the Principal Component Analysis (PCA) method.
4. Write a Python program to demonstrate various Data Visualization Techniques.
5. Implement MLE on a Dataset
6. Implement Simple and Multiple Linear Regression Models.
7. Develop Logistic Regression Model for a given dataset.
8. Develop Decision Tree Classification model for a given dataset and use it to classify a new sample.
9. Implement Naïve Bayes Classification in Python.
10. Develop K-Means for a Given Dataset
11. Build KNN Classification model for a given dataset.
12. Develop DBSCAN on a given Dataset