Course Code: PCAA201C	Machine learning and Data analytics using Python	Credits : 04
Hours/Week (L:T:P) : 2:2:0		CIE Marks : 50
Total Hours: 52		SEE Marks : 50

Course Type: Theory

Course Learning Objectives:

1. Understand foundational concepts in machine learning and data analytics.
2. Gain proficiency in Python for data analysis and machine learning tasks.
3. Learn and apply various machine learning algorithms and techniques.
4. Develop skills in data preprocessing, visualization, and model evaluation.
5. Prepare students for industry roles involving data-driven decision making and predictive modeling.

Module-1	12 Hrs.

Introduction to Machine Learning and Python:

Introduction to Machine Learning: Definition and importance of machine learning, Types of machine learning: Supervised, unsupervised, and reinforcement learning, Applications of machine learning in various domains. **Python for Data Analysis:** Introduction to Python programming, Python libraries for data analysis: NumPy, Pandas, Matplotlib, Data manipulation and visualization using Pandas and Matplotlib.

Data Preprocessing: Data cleaning and transformation, Handling missing values and outliers

Module-2	10 Hrs.

Supervised Learning: Regression: Linear regression, Polynomial regression, Model evaluation metrics: MAE, MSE, RMSE. Classification: Logistic regression, K-Nearest Neighbors (KNN), Decision Trees and

Random Forests, Model evaluation metrics: Accuracy, precision, recall, F1-score, ROC-AUC.

Model Training and Evaluation: Train-test split: Hold out methods and cross-validation, Bootstrap

Model evaluation and tuning: Overfitting , underfitting. And Traid-off bias and variance.

Module-3	10 Hrs.

Unsupervised Learning: Clustering: Applications of clustering, types of clustering, K-Means clustering, Hierarchical clustering, Evaluation of clustering results. **Dimensionality Reduction:** Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) t-Distributed Stochastic Neighbor Embedding (t-SNE).

Module-4	10 Hrs.

Finding patterns using Association Rule Learning: Apriori algorithm, Market Basket Analysis, Evaluation metrics for association rules. **Basics of Neural Networks :** Introduction, Understanding the boilogical neuron, Exploring the artificial neurons, Architecture of Neural network: Single layer feed forward network, Multilayer feed forward ANN's. Convolutional Neural Networks (CNN). Learning process in ANN.

Module-5	10 Hrs.

Data Analytics and Real-World Applications: Exploratory Data Analysis (EDA): Data visualization techniques, Statistical analysis and hypothesis testing, Identifying patterns and insights from data.

Time Series Analysis: Introduction to time series data, Time series forecasting using ARIMA and Prophet, Evaluating time series models.

Suggested Learning resources

Text Books:

1. Practical Machine Learning for Data Analysis Using Python Abdulhamit Subasi.
2. Advance Machine Learning with Python by John Hearty.

Reference books:

1. "Python Machine Learning: Machine Learning and Deep Learning with Python, scikitlearn, and Tensor Flow " by Sebastian Raschka and Vahid Mirjalili.
2. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron.
3. "Introduction to Machine Learning with Python".

Sl.NO	Experiments
1	Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2	For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate- Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples
3	Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample
4	Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
5	Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.
6	Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.
7	Write a program to demonstrate Regression analysis with residual plots on a given data set.
8	Write a program to compute summary statistics such as mean, median, mode, standard deviation and variance of the given different types of data.
9	Write a program to implement k-Means clustering algorithm to cluster the set of data stored in .CSV file.

Course Outcomes:

CO1: Demonstrate an understanding of machine learning concepts and techniques.

CO2: Perform data preprocessing and exploratory data analysis using Python.

CO3: Develop and evaluate machine learning models using Python libraries.

CO4: Apply machine learning algorithms to real-world data problems.

CO5: Integrate data analytics and machine learning models into practical applications.

Course Outcomes	Programme Outcomes							
	1	2	3	4	5	6	7	8
CO1	3	3						2
CO2	3	2						
CO3			2	2				
CO4		2		2				2
CO5			2					2

PCAA202C	Object Oriented Programming using Java	Credits:04
Hrs/Week:L:T:P:S **4:0:0:0**		CIEMarks:50
TotalHours/Week: 50Hrs		SEEMarks:50

MODULE-I	08Hrs.

Introduction: Object Oriented Programming: objects, classes, Abstraction, Encapsulation, Inheritance, Polymorphism, OOP in Java, Characteristics of Java, The Java Environment, Java Virtual Machine, Java Source File Structure, and Compilation.

Fundamental Programming Structures in Java: Defining classes in Java, constructors, methods, The this Keyword, Overloading Constructors. Access specifies, static members, Comments, Data Types, The Scope and Lifetime of Variables,, Operators, Control Flow, Arrays.

MODULE-II	08Hrs.

Inheritance : Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Superclass constructors, Using super to Access Superclass Members, Method Overriding, Overridden Methods support polymorphism, Why Overridden Methods, Using Abstract Classes, Using final.

String Handling: String Constructor, String length, Special string Operations, Character Extraction, String comparison, Modifying a string, String Buffer

MODULE-III	08Hrs.

Interface: Creating an Interface, Implementing an Interface, Using Interface References, Implementing Multiple Interfaces .

Packages : Package Fundamentals, Packages and Member Access, Importing Packages Exception Handling : The Exception Hierarchy, Exception Handling Fundamentals, using Multiple catch clauses, **User defined exception**: Throwing an Exception, A Closer look at Throwable, using finally, System defined Exception using throws.

MODULE-IV	08Hrs.

Multithreaded Programming: Multithreading fundamentals, The Thread Class and Runnable Interface, Creating Thread, Thread Priorities, Synchronization, using Synchronization Methods, The Synchronized Statement, Thread Communication using notify(), wait() and notify All(), suspending, Resuming and stopping Threads.

MODULE-V	08Hrs.

JDBC: Talking to Database, Types of JDBC, Essential JDBC program, using prepared Statement JDBC in Action Result sets, Batch updates, Mapping, Basic JDBC data types, Advanced JDBC data types, immediate solutions.

Reference Books:

1) Introduction to JAVA Programming 9th Edition by Y. Daniel Liang , Pearson education, 2012

2) Programming in JAVA 5.0 1st Edition by James P Cohoon, Jack W Davidson, TATA McGraw hill,2006

Course Outcomes:

At the end of the course the student will be able to :

1. Understand the basic principles of the object-oriented programming
2. Demonstrate an introductory understanding of graphical user interfaces.
3. Apply the knowledge of Java concepts to find the solution for a given problem.
4. Analyse the given Java application for correctness/functionalities.
5. Develop Java programs / applications for a given requirement.

Course Outcomes	Programme Outcomes							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CO1	3		3					
CO2			3					2
CO3	3			3				2
CO4			2					
CO5				3				2

Course Code: PCAA203C	Data Structures and Algorithms	Credits : 04
Hours/Week (L:T:P) : 4:0:0		CIE Marks : 50
Total Hours: 52		SEE Marks : 50

Course Type: Theory

Course Objectives:

1. To understand and implement fundamental data structures.
2. To develop efficient algorithms for solving problems.
3. To analyze the time and space complexity of algorithms.
4. To gain practical experience in applying data structures and algorithms to real-world problems.
5. To prepare students for industry roles requiring strong foundations in data structures and algorithmic thinking.

Module-1	11 Hrs.

Introduction to Data Structures and Algorithms:
Basic Concepts: Definition and importance of data structures, Abstract Data Types (ADTs), Algorithm analysis: Time and space complexity, Big O notation. **Arrays:** Definition and operations: Insertion, deletion, traversal, Multidimensional arrays, Applications of arrays. **Linked Lists**: Singly linked list: Creation, insertion, deletion, traversal, Doubly linked list and circular linked list, Applications of linked lists.

Module-2	10 Hrs.

Stacks and Queues,: Stacks: Definition and operations: Push, pop, peek, Applications: Expression evaluation, backtracking, function calls. **Queues:** Definition and operations: Enqueue, dequeue, front, rear, Types: Circular queue, priority queue, double-ended queue (deque), Applications of queues.

Module-3	10 Hrs.

Recursion: Definition and principles of recursion, Recursive algorithms: Factorial, Fibonacci series, Tower of Hanoi, Analysis of recursive algorithms. **Trees: Definition and terminology**: Root, leaf, internal node, height, depth, **Binary trees:** Traversal (preorder, inorder, postorder), creation, insertion, deletion, Binary search trees (BST), AVL trees, B-trees.

Module-4	10 Hrs.

Graphs: Definition and terminology: Vertices, edges, adjacency list, adjacency matrix, Graph traversal algorithms: Depth-first search (DFS), breadth-first search(BFS), **Shortest path Algorithms:** Dijkstra's algorithm Floyd -Warshall algorithm.

Module-5	11 Hrs.

Sorting and Searching Algorithms: Sorting Algorithms: Basic concepts and classification, Comparison-based sorting: Bubble sort, selection sort, insertion sort, quicksort, mergesort, heapsort, Non-comparison-based sorting: Radix sort, counting sort. Searching Algorithms: Linear search and binary search, Search in linked lists, trees, and graphs, Hashing: Hash functions, collision resolution techniques (chaining, open addressing). Advanced Data Structures and Applications: Advanced Data Structures: Heaps: Definition, operations, heap sort, applications, Trie: Definition, operations, applications in dictionary and spell-checking.

Suggested Learning resources

Text Books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2023.
2. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
3. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

Reference books:

1. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
2. A M Tenenbaum, Data Structures using C, PHI, 1989
3. Introduction to the Design and Analysis of Algorithms, Anany Levitin: 2nd Edition, 2009. Pearson.
4. Algorithms, Kenneth A Berman and Jerome L Paul, Cengage Learning India Pvt Ltd, 2002 edition.

Course Outcomes:

CO1: Demonstrate an understanding of fundamental data structures and algorithms
CO2: Implement and manipulate data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
CO3: Develop algorithms for searching, sorting, and optimization problems.
CO4: Analyze the efficiency and correctness of algorithms.
CO5: Apply data structures and algorithms to solve complex problems in various domains.

Course Outcomes	Programme Outcomes							
	1	2	3	4	5	6	7	8
CO1	3	3	2	1				
CO2	3	3	2	2				
CO3	3	2	3	2				
CO4	3	3	2					
CO5	3	3						2

PCAA204C	Software Engineering	Credits:03
Hrs/Week:L:T:P:S **3:0:0:0**		CIEMarks:50
TotalHours/Week: 40Hrs		SEEMarks:50

MODULE-I	08Hrs.
Software Process & Software Development Methods: Professional Software Development, software engineering ethics, Software process models, Process activities, Coping with change. **Agile Software Development:** Agile methods, Plan-driven and agile development, Extreme programming, **Requirements Engineering**: Functional and non-functional requirements, The software requirements document, Requirements specification	
MODULE-II	**08Hrs.**
Requirements engineering processes: Requirements elicitation and analysis, Requirements validation. Requirements change. System Models: Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering, **Design and Implementation:** Introduction to RUP, Design Principles.	
MODULE-III	**08Hrs.**
Object-oriented design using the UML Design patterns, Implementation issues, Open source development. Software testing: Development testing, Test-driven development, Release testing, User testing. **Project management:** Risk management, Managing people, Team work, Project planning, Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques.	
MODULE-IV	**08Hrs.**
Quality Management: Software quality, Software standards, Reviews and inspections, Software measurement and metrics. Scrum Framework Foundation of Scrum, pillars of empiricism, Scrum Values, Identifying a Scrum Team; **Scrum Events: Spring** planning, Implementation and review, Scrum Artifacts: Creating, Managing and refining product backlog. **Scrum in Action**: Planning and Estimating with Scrum: Estimation Scale, Bucket method	
MODULE-V	**08Hrs.**
Envisioning the product journey with a product roadmap; **Sprint Journey**: Refining the Product Backlog, Tracking progress with a Scrum Board, Defects in Sprint; **Facets of Scrum**: software development practices for Scrum, Source control model for continuous integration, Continuous delivery and continuous deployment, Leveraging testing methods for Scrum, Applying Scrum to remote teams.	

ReferenceBooks:

1. Ian Sommerville, Software Engineering, 10th Edition, Pearson India Education Services Pvt. Ltd., 2020.
2. RogerS.Pressman,SoftwareEngineering-APractitionersapproach,7thEdition,Tata McGrawHill.
3. PankajJalote, An Integrated Approach to Software Engineering,WileyIndia.
4. Fred heath, The Professional Scrum Master Guide, Packt Publishing,2021.
5. Stacia Viscardi, The Professional Scrum Master's Handbook,PacktPublishing,2013
6. Andrew T. Pham and David K. Pham, Business-Driven IT-Wide Agile (Scrum) and Kanban(Lean) Implementation, CRC Press.

Course Outcomes:

1. Comprehend software process and process models and decide which process model has to be adopted for the given scenarios.
2. For given case study List functional and nonfunctional requirements.
3. Ability to develop system model and architectural design.
4. Analyze the importance of various of software testing methods and the role of project planning and quality management in software development.
5. Understand Scrum frameworks and its utility in software development.

Course Outcomes	Programme Outcomes							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CO1			2	2				
CO2			2					
CO3				2		2		1
CO4				3				
CO5		2						2

PCAA205C	Web Application Development	Credits:03
Hrs/Week:L:T:P:S **3:0:0:0**		CIEMarks:50
TotalHours/Week: 40Hrs		SEEMarks:50

MODULE-I	08Hrs.
Introduction to Web Development and HTML5: Web Development Basics: Introduction to web technologies and protocols, Client-server architecture, Overview of front-end and back-end development **HTML5 Fundamentals:** HTML5 elements and attributes, Semantic HTML5 tags, Forms and input types, Multimedia elements (audio, video) Advanced HTML5: Canvas and SVG for graphics, HTML5 APIs(Geolocation, Web Storage, Web Workers), Offline web applications using AppCache.	
MODULE-II	**08Hrs.**
CSS3 and Responsive Web Design: CSS3 Basics: Introduction to CSS3, Selectors, properties, and values, Box model, layout, and positioning, Flexbox and Grid layouts. **Responsive Web Design**: Media queries, Responsive design principles, Fluid grids and flexible images, Mobile-first design approach. **CSS Frameworks:** Introduction to Bootstrap, Bootstrap components and utilities, Customizing Bootstrap with Sass.	
MODULE-III	**08Hrs.**
JavaScript and DOM Manipulation: JavaScript Basics: Introduction to JavaScript, Variables, data types, and operators, Control structures (if-else, loops), **Functions and scope Document Object Model (DOM):** DOM structure and manipulation, Event handling and event listeners, Creating and modifying DOM elements, **Form validation using JavaScript Advanced JavaScript:** Asynchronous JavaScript (callbacks, promises, async/await), AJAX and Fetch API, Introduction to JavaScript libraries (e.g., jQuery).	
MODULE-IV	**08Hrs.**
Front-End Frameworks and AngularJS: Introduction to Front-End Frameworks: Importance of front-end frameworks, **Overview of popular frameworks (React, Angular, Vue) AngularJS Basics:** Introduction to AngularJS, Modules, controllers, and scope, Directives, expressions, and filters Advanced AngularJS: Services and dependency injection, Routing and single-page applications (SPAs), Data binding and form handling, Custom directives and components.	
MODULE-V	**08Hrs.**
Back-End Integration and Deployment: Back-End Development: Introduction to server-side programming, Overview of server-side languages (Node.js, PHP, Python), RESTful web services and APIs, Database integration (SQL, NoSQL) Full-Stack Development: Integrating front-end and back-end technologies, Developing fullstack web applications, Case studies on full-stack applications Deployment and Security: Web application deployment (cloud platforms, hosting services), Security best practices for web applications, Authentication and authorization, Performance optimization.	

Reference Books:

1. Web Programming By Chris Bates , Wiley Publications
2. HTML5 Black Book by Dreamtech
3. Angular JS By Krishna Rungta
4. Bootstrap essentials by Snig by Packt-open source .

Course Outcomes:

At the end of the course the student will be able to :

1. Demonstrate an understanding of web technologies and protocols.
2. Develop and deploy web applications using HTML5, CSS3, JavaScript, and modern frameworks.
3. Apply responsive design principles using frameworks like Bootstrap.
4. Implement dynamic web applications using AngularJS.
5. Integrate front-end and back-end technologies to create full stack web applications.

Course Outcomes	Programme Outcomes							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CO1	3							
CO2			3	3				2
CO3	1			2				2
CO4					2			
CO5			2			2		

Course Code: PCAA206L	Object Oriented Programming using JAVA Laboratory	Credits : 02
Hours/Week (L:T:P) : 0:2:0		CIE Marks : 50
Total Hours of Pedagogy : 30		SEE Marks : 50

Course Type: Practical

Course Objectives:

1. Using java programming to develop programs for solving real-world problems.
2. Reinforce the understanding of basic object-oriented programming concepts..

Sl. No.	Experiments
01	Write a Java program to print the following triangle of numbers 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5
02	Write a Java program To find the area and circumference of the circle by accepting the radius from the user. To accept a number and find whether the number is Prime or not
03	Write a JAVA program to read 3 subjects marks, calculate the total and average marks. Display the grade based on the following criteria: Note: Percentage>=90% : Grade A Percentage>=80% : Grade B Percentage>=70% : Grade C Percentage>=60% : Grade D Percentage>=40% : Grade E Percentage.
04	Write a JAVA program to create a class called Person with p_name and age. Which includes constructor to initialize these fields and a method to display the person information. In the main method, an instance of the Person class will be created and Person information is displayed.
05	Define a class named Animal with two methods: eat() and sleep(). Create a class named Dog that extends the Animal which includes a method called bark() and also create a class called Cat that extends Animal which includes Meow() method. Create a class called MainClass to write main() method and demonstrate the following: Inside the main method: a. Create objects of the Dog and Cat classes (myDog and myCat). b. Demonstrate calling methods from the parent class (eat() and sleep()). c. Illustrate calling methods from the Dog class (bark()). d. Showcase calling methods from the Cat class (meow()).
06	Write a JAVA program calculate factorial of a given number n.
07	Write a JAVA program that prompts the user to enter a number between 1 and 7. The program then uses a switch case statement to determine the corresponding day of the week based on the user's input and prints the result.
08	Create a Shape class serves as the base class for all shapes. It contains a method draw() that prints a generic message indicating the act of drawing a shape. The Circle class extends the Shape class and overrides the draw() method to provide a specialized implementation for drawing a circle. Similarly, the Square class extends the Shape class and provides its own implementation of the draw() method to handle drawing a square. Create MainClass contains the main method, serving as the entry point of the program. Two objects, shape1 and shape2,

	are declared of type Shape but instantiated as Circle and Square objects, respectively. This demonstrates the polymorphic behavior, allowing objects of derived classes to be treated as objects of the base class.
	Demonstration Experiments (For CIE) if any
09	Write java program to implement the concept of abstract class and interface.
10	Write a Java program to implement a Queue using user defined Exception Handling (also make use of throw, throws). a. Complete the following: b. Create a package named shape. c. Create some classes in the package representing some common shapes like Square, Triangle, and Circle. d. Import and compile these classes in other program.
	Course Outcome
	At the end of the course the student will be able to : 1. Demonstrate the fundamental data types and constructs of Java Programming by writing executable/interpretable programs. 2. Illustrate the object oriented principles with the help of java programs. 3. Develop reusable and efficient applications using inheritance concepts of java. 4. Learn the object oriented concepts and its implementation in Java..

Course Outcomes	**Programme Outcomes**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CO1	3	2						
CO2	2		2	2				2
CO3			2					
CO4	3				2			2

Course Code: PCAA207L	Data Structures with Algorithms Laboratory	Credits : 02
Hours/Week (L:T:P) : 0:2:0		CIE Marks : 50
Total Hours of Pedagogy : 30		SEE Marks : 50

Course Type: Practical

Course Objectives:

1. Evaluate the Expressions like postfix, prefix conversions.
2. Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs.

Sl. No.	Experiments
01	Implement a Program in C for converting an Infix Expression to Postfix Expression.
02	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide)
03	Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display
04	Write a C program to simulate the working of a singly linked list providing the following operations: a. Display & Insert b. Delete from the beginning/end c. Delete a given element
05	Write a C program to Implement the following searching techniques a. Linear Search b. Binary Search.
06	Write a C program to implement the following sorting algorithms using user defined functions: a. Bubble sort (Ascending order) b. Selection sort (Descending order).
07	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm (C programming)
08	From a given vertex in a weighted connected graph, find shortest paths to other vertices Using Dijkstra's algorithm (C programming)
	Demonstration Experiments (For CIE) if any
09	Using circular representation for a polynomial, design, develop, and execute a program in C to accept two polynomials, add them, and then print the resulting polynomial.
10	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).
	Course Outcome
	Course Outcomes: At the end of the course the student will be able to : 1. Implement data structures (namely Stacks, Queues, Circular Queues, Linked Lists, Trees and Graphs), its operations and algorithms. 2. Implement sorting / searching techniques, and validate input/output for the given problem. 3. Implement the techniques for evaluating the given expression

		Course Outcomes	Programme Outcomes								
			1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
		CO1	3		3	3					
		CO2	2		2	3					
		CO3	2		2	2					