B.Sc. (DATA SCIENCE)

Syllabus (CBCS)

(w.e.f. 2025-2026)

FACULTY OF SCIENCE
SATAVAHANA UNIVERSITY
KARIMNAGAR – 505002

2025

B.SC. (CBCS) DATA SCIENCE SYLLABUS

(With Mathematics & Statistics Combination)

With effect from the Academic Year: 2024-25

Year	Sem	Paper code	Theory/ Practical	Paper Title	Credits	Work Load (Hours / Week)	Total Marks
I	I	**BSDS-101-T**	**Paper-I**	**Problem solving and Python Programming**	4	4	100
		BSDS-101-P	Practical -I	Problem solving and Python Programming	1	2	25
	II	**BSDS-202-T**	**Paper-II**	**Data Structures and Algorithms**	4	4	100
		BSDS-202-P	Practical -II	Data Structures and Algorithms Lab	1	2	25
II	III	**BSDS-303-T**	**Paper-III**	**Data Engineering with Python**	4	4	100
		BSDS-303-P	Practical-III	Data Engineering with Python (Lab)	1	2	25
		BSDS-303-SE	SEC-2	Information Technology	2	2	50
	IV	**BSDS-404-T**	**Paper-IV**	**Programming in Java**	4	4	100
		BSDS-404-P	Practical-IV	Programming in Java Lab	1	2	25
		BSDS-404-SE	SEC-4	Web Technology	2	2	50
III	V	**BSDS-505-T**	**Paper-V**	**Machine Learning**	4	4	100
		BS-505-P	Practical-5	Machine Learning (Lab)	1	2	25
		BS-505-GE	**G. E.**	Basic Statistics	4	4	100
	VI	**BS-606-T**	**Paper-VI**	**Deep Learning**	4	4	100
		BS-606-P	Paper-VI	Deep Learning Lab	1	2	25
		BS-606-O	Optional	Data Science Project	4	4	100

Note:

1. No of credits = No of theory hours for teaching per week = Twice the no. of credits for practical teaching.
2. Skill Enhancement Courses (SEC) are offered for Data Science students in Sem-III & IV with continuation (offered for who are willing improve the practical skills).
3. Generic Elective (GE) course is offered for *other than this Statistics combination Course students.*
4. Optional paper course is offered if Data Science students are willing to opt.
5. The practical examination is of duration two hours and answering any two out of four.

DATA SCIENCE SYLLABUS
B.SC. (Data Science) I-YEAR I-SEMESTER
Paper-I (Theory): Problem Solving and Python Programming
[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Course Objectives:

The main objective is to teach Computational thinking using Python.

- To know the basics of Programming for solving a problem with writing an algorithm and flow charts.
- To convert an algorithm into a Python program
- To construct Python programs with control structures.
- To structure a Python Program as a set of functions
- To use Python data structures-lists, tuples, dictionaries.
- To do input/output with files in Python.
- To construct Python programs as a set of objects.

UNIT-I

Introduction to Computing and Problem Solving: Fundamentals of Computing – Computing Devices – Identification of Computational Problems – Pseudo Code and Flowcharts – Instructions – Algorithms – Building Blocks of Algorithms.

Introduction to Python Programming: Python Interpreter and Interactive Mode– Variables and Identifiers – Arithmetic Operators – Values and Types – Statements, Reading Input, Print Output, Type Conversions, The type() Function and Is Operator, Dynamic and Strongly Typed Language.

Control Flow Statements: The if, The if…else,The if…elif…else Decision Control Statements, Nested if Statement, The while Loop, The for Loop, The continue and break Statements.

UNIT-II

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments. **Strings**: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

UNIT-III

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, merge sort, histogram.

Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file.

UNIT-IV

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, The Constructor Method, Classes with Multiple Objects, Class Attributes versus Data Attributes, Encapsulation, Inheritance, Polymorphism.

Functional Programming: Lambda. Iterators, Generators, List Comprehensions

References:

1. Introduction to Python Programming. Gowrishankar S., Veena A. CRC Press, Taylor & Francis Group, 2019
2. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist'', 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (http://greenteapress.com/wp/think- python/)
3. Learning To Program With Python. Richard L. Halterman. Copyright © 2011
4. Python for Everybody, Exploring Data Using Python 3. Dr. Charles R. Severance. 2016.

Course Outcomes:

- The main objective of this is to put into practice in computer lab with computational thinking and able to write, compile, run and debug Python programs with perfect usage of variables, conditionals statements, control structures, functions (both recursive and iterative), basic data types as well as compound data structures such as strings, lists, sets, tuples, dictionaries. object-oriented programming.
- able to develop algorithmic solutions to simple computational problems.
- able to Develop and execute simple Python programs.
- Structure a Python program into functions.
- Represent compound data using Python lists, tuples, dictionaries.
- Read and write data from/to files in Python Programs.

B.SC. (Data Science) I-YEAR I-SEMESTER

Paper-I (Practical): Problem Solving and Python Programming (Lab)

[2 HPW :: 1 Credit :: 25 Marks]

List of Practical's:

I. Programs to demonstrate the usage of operators and conditional statements

1. Write a program that takes two integers as command line arguments and prints the sum of two integers.
2. Program to display the information: Your name, Full Address, Mobile Number, College Name, Course Subjects
3. Program to find the largest number among 'n' given numbers.
4. Program that reads the URL of a website as input and displays contents of a webpage.

II. Programs to demonstrate usage of control structures

1. Program to find the sum of all prime numbers between 1 and 1000.
2. Program that reads set of integers and displays first and second largest numbers.
3. Program to print the sum of first 'n' natural numbers.
4. Program to find the product of two matrices.
5. Program to find the roots of a quadratic equation.

III. Programs to demonstrate the usage of Functions and Recursion

6. Write both recursive and non-recursive functions for the following:
   a. To find GCD of two integers
   b. To find the factorial of positive integer
   c. To print Fibonacci Sequence up to given number 'n'
   d. To convert decimal number to Binary equivalent
7. Program with a function that accepts two arguments: a list and a number 'n'. It should display all the numbers in the list that are greater than the given number 'n'.
8. Program with a function to find how many numbers are divisible by 2, 3,4,5,6 and 7 between 1 to 1000.

IV. Programs to demonstrate the usage of String functions

9. Program that accept a string as an argument and return the number of vowels and consonants the string contains.

10. Program that accepts two strings S1, S2, and finds whether they are equal are not.
11. Program to count the number of occurrences of characters in a given string.
12. Program to find whether a given string is palindrome or not.

V. Programs to demonstrate the usage of lists, sets, dictionaries, tuples and files.

13. Program with a function that takes two lists L_1 and L_2 containing integer numbers as parameters. The return value is a single list containing the pair wise sums of the numbers in L_1 and L_2.
14. Program to read the lists of numbers as L_1, print the lists in reverse order without using reverse function.
15. Write a program that combine lists L1 and L2 into a dictionary.
16. Program to find mean, median, mode for the given set of numbers in a list.
17. Program to find all duplicates in the list.
18. Program to o find all the unique elements of a list.
19. Program to find max and min of a given tuple of integers.
20. Program to find union, intersection, difference, symmetric difference of given two sets.
21. Program to display a list of all unique words in a text file
22. Program to read the content of a text file and display it on the screen line wise with a line number followed by a colon
23. Program to analyse the two text files using set operations
24. Write a program to print each line of a file in reverse order.

VI. Programs to demonstrate the usage of Object-Oriented Programming

25. Program to implement the inheritance
26. Program to implement the polymorphism.

VII. Programs to search and sort the numbers

27. Programs to implement Linear search and Binary search
28. Programs to implement Selection sort, Insertion sort

DATA SCIENCE SYLLABUS
B.SC. I YEAR II SEMESTER (CBCS)
PAPER – II (Theory) : DATA STRUCTURES AND ALGORITHMS

UNIT – I

Fundamental Concepts: Introduction to Data Structures, Types of Data Structures, Introduction to Algorithm, Pseudo-code, Flow Chart, Analysis of Algorithms. Linear Data Structure Using Arrays: 1-D Arrays, 2-D Arrays, N-D Arrays, Memory Representation and Address Calculation of 1-D, 2-D, N-D Arrays, Concept of Ordered List, String Manipulation, Pros and Cons of Arrays. Stacks: Concept, Primitive Operations, Abstract Data Type, Representation Stacks Using Arrays, Prefix, Infix, Postfix Notations for Arithmetic Expression, Applications of Stacks– Converting Infix Expression to Postfix Expression, Evaluating the Postfix Expression, Checking Well-formed (Nested) Parenthesis, Processing of Function Calls, Reversing a String.

UNIT – II

Recursion: Introduction, Recurrence, Use of Stack in Recursion, Variants of Recursion, Execution of Recursive Calls, Recursive Functions, Iteration versus Recursion. Queues: Concept, Primitive Operations, Abstract Data Type, Representation Queues Using Arrays, Circular Queue, Double-Ended Queue, Applications of Queues. Linked Lists: Introduction, Concept, Terminology, Primitive Operations-creating, inserting, deleting, traversing, Representation of Linked Lists, Linked List Abstract Data Type, Linked List Variants - Singly Linked List, Doubly Linked List, Linear and Circular Linked List, Representation Stacks and Queues Using Linked Singly Lists, Application of Linked List–Garbage Collection.

UNIT – III

Trees: Introduction, Representation of a General Tree, Binary Tree Introduction, Binary Tree Abstract Data Type, Implementation of Binary Trees, Binary Tree Traversals – Preorder, Inorder, Postorder Traversals, Applications of Binary Trees Briefly. Graphs: Introduction, Graph Abstract Data Type, Representation of Graphs, Graph Traversal – Depth-First Search, Breadth-First Search, Spanning Tree – Prim's Algorithm, Kruskal's Algorithm. Hashing: Introduction, Hash Functions, Collision Resolution Strategies.

UNIT – IV

Searching and Sorting Algorithms: Sequential (Linear) Search, Binary Search, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort, and Comparison of Sorting Techniques. Heaps: Concept, Implementation, Abstract Data Type, Heap Sort.

References

1. "Computer Algorithms" - Ellis Horowitz, Sartaj Sahni and S. Rajasekaran
2. "Data Structure and Algorithmic Thinking with Python" - Narasimha Karumanchi
3. "Data Structures and Algorithms in Python"- Roberto Tamassia, M. H. Goldwasser, M.T. Goodrich.
4. "Problem Solving in Data Structures & Algorithms Using Python"- Hemant Jain

B.Sc. I-Year II-Semester (CBCS) Data Science Syllabus
Paper-II (Practical): Data Structures Using Python (Lab)

[2 HPW :: 1 Credit :: 25 Marks]

1. Write programs to implement the following using an array: a) Stack ADT b) Queue ADT.
2. Write a program to convert the given infix expression to postfix expression using stack.
3. Write a program to evaluate a postfix expression using stack.
4. Write a program to ensure the parentheses are nested correctly in an arithmetic expression.
5. Write a program to find following using Recursion a) Factorial of +ve Integer b) n^{th} term of the Fibonacci Sequence (c) GCD of two positive integers
6. Write a program to create a single linked list and write functions to implement the following operations. a) Insert an element at a specified position b) Delete a specified element in the list c) Search for an element and find its position in the list d) Sort the elements in the list ascending order
7. Write a program to create a double linked list and write functions to implement the following operations. a) Insert an element at a specified position b) Delete a specified element in the list c) Search for an element and find its position in the list d) Sort the elements in the list ascending order
8. Write a program to create singular circular linked lists and function to implement the following operations. a) Insert an element at a specified position b) Delete a specified element in the list c) Search for an element and find its position in the list
9. Write programs to implement the following using a single linked list: a) Stack ADT b) Queue ADT. 10 Write a program to implement Binary search technique using Iterative method and Recursive methods.
10. Write a program for sorting the given list numbers in ascending order using the following technique: Bubble sort and Selection sort
11. Write a program for sorting the given list numbers in ascending order using the following technique: Insertion sort and Quicksort
12. Write a program for sorting the given list numbers in ascending order using the following technique: Merge sort and Heapsort
13. Write a program to traverse a binary tree in following way. a) Pre-order b) In-order c)Post-order 15 Write a program to the implementation graph traversals – BFS and DFS.
14. Write a program to find the minimum spanning tree for a weighted graph using a) Prim's Algorithm b) Kruskal's Algorithm.

Note: Write the Pseudo Code, flowcharts and Python program code for the above problems/methods/ algorithms with different possibilities like with and without oops, functions, etc. is mandatory.

B.Sc. II-Year, III-Semester (CBCS) Data Science Syllabus

Paper-III (Theory): Data Engineering with Python
[4 HPW :: 4 Credits :: 100 Marks (External: 80, Internal: 20)]

Course Objectives and Outcomes:

1. Understanding the basic concepts on Data Engineering.
2. Database creation using MySQL and accessing data and performing ETL operations.
3. Able to handle different types of data files using Python.
4. Use of regular expression operations, relational databases via SQL, tabular numeric data, data structures: data series and frames. Usage of PyPlot, Numpy, Pandas on data sets.

UNIT – I

Data Science: Data sets, Data variables, understanding of Datasets, Data Analysis Sequence, Data Acquisition Pipeline, Report Structure. **Files and Working with Text Data:** Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules. **Working with Text Data:** JSON and XML in Python.

UNIT – II

Working with Text Data: Processing HTML Files, Processing Texts in Natural Languages. **Regular Expression Operations:** Using Special Characters, Regular Expression Methods, Named Groups in Python Regular Expressions, Regular Expression with *glob* Module. **Working with Databases:** Setting Up a MySQL Database, using a MySQL Database: Command Line, Using a MySQL Database, Taming Document Stores: MongoDB

UNIT – III

Working with Tabular Numeric Data (NumPy with Python): Introduction to NumPy, NumPy Arrays Creation Using *array()* Function, Array Attributes, NumPy Arrays Creation with Initial Placeholder Content, Integer Indexing, Array Indexing, Boolean Array Indexing, Slicing and Iterating in Arrays, Basic Arithmetic Operations on NumPy Arrays, Mathematical and Statistical Functions in NumPy, Changing the Shape of an Array, Stacking and Splitting of Arrays, Broadcasting in Arrays.

UNIT – IV

Pandas: Introduction to Pandas**,** Working with Pandas Data Structures: Data Series and Frames, Renaming and Reshaping Data Using Pandas, Handling Missing Data using Pandas, Merging the Data using Pandas, Ordering and Describing Data, Transforming Data, Taming Pandas File I/O (Explanation of data exchange between frames and series). **Plotting**: Introduction to Basic Plotting with PyPlot, Initializing the Plots object, Types of PlotsGetting to Know Other Plot Types, Mastering Embellishments, Plotting with Pandas.

References:

1. Gowrishankar S., Veena A. (2019): Introduction to Python Programming**.** CRC Press, T&F.
2. Charles R Severance (2016): Python for Everybody: Exploring Data Using Python.
3. Fabio Nelli, (2015): Python Data Analytics – Data Analysis and Science using Pandas, matplotlib and the Python Programming Language, Apress.
4. Chris Albon (2018): Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning. O'Reilly.
5. Dmitry Zinoriev (2016): Data Science Essentials in Python: Collect, Organize, Explore, Predict, Value. The Pragmatic Programmers LLC.

B.Sc. II-Year, III-Semester (CBCS) Data Science Syllabus

Paper-III (Practical): Data Engineering with Python (Lab)

[2 HPW :: 1 Credit :: 25 Marks]

Course Objectives & Outcomes:

1. Able to perform the operations Extract, Transform, Load (ETL) the input data with different file formats. text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc.
2. Able to cleaning and visualize the data.
3. Able to use Python libraries/modules: pandas, NumPy, Beautiful Soup, PyMySql, pymongo, nltk, matplotlib.
4. Able to perform all the above on various data sets with different file formats.

List Practical's:

1. Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.
2. Write programs for reading and writing binary files
3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions
4. Design a relational database for a small application and populate the database. Using SQL do the CRUD (create, read, update and delete) operations.
5. Create a Python MongoDB client using the Python module pymongo. Using a collection object practice functions for inserting, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes
6. Write programs to create Numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements
7. Write programs to use the Pandas data structures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:
   - Single-level and hierarchical indexing
   - Handling missing data
   - Arithmetic and Boolean operations on entire columns and tables
   - Database-type operations (such as merging and aggregation)
   - Plotting individual columns and whole tables
   - Reading data from files and writing data to files

Note: Student has to practice the above on various publicly available *appropriate datasets for implementation*. Some of the datasets are like: MNIST (http://yann.lecun.com/exdb/mnist/);UCI Machine Learning Repository(https://archive.ics.uci.edu/ml/datasets.html), Kaggle datasets (https://www.kaggle.com/datasets); Twitter Data

B.Sc. II-Year, IV-Semester (CBCS) Data Science Syllabus

SEC-2 Paper (Theory): Information Technologies

[2 HPW :: 2 Credits :: 50 Marks (External:40, Internal:10)]

Course Objectives & Outcomes:

1. Able to know the basic concepts on computer networks, basics of Email, computer security, Internet browsing.

UNIT – I

Computer Networks: Introduction, Connection Media, Data Transmission Mode, Data Multiplexing, Data Switching, Network Topologies, Types of Networks, Networking Devices, OSI Model.

Internet: Internet Services, Types of Internet Connections, Internet Security. Introduction to Emerging Computer Technologies: Distributed Networking, Peer-to-peer Computing, Grid Computing, Cloud Computing, Utility Computing, On-demand Computing, Wireless Network, Bluetooth, Artificial Intelligence.

UNIT – II

Basics of Email: Email, Functions of Email, Browser, Web Browser, Internet Service Providers. Introduction to Information Security – Need for Information Security, Threats to Information Systems, Information Assurance, Cyber Security.

Security: Introduction to Application Security and Counter Measures – Introduction to Application Security, Data Security Considerations, Security Technologies, Security Threats, Security Threats to E-Commerce, ECash and Electronic Payment System.

References:

1. Surya Prakash T, Ritendra G, Praveen Kumar S.: Introduction to Information Security and Cyber Laws, Dream tech Publications.

B.Sc. II-Year, IV-Semester (CBCS) Data Science Syllabus
Paper-IV (Theory): Programming in Java
[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Course Objectives & Outcomes:

1. Able to know the basic concepts Java Programming.
2. Able to write program using various Object-Oriented Programming concepts.
3. Able to write programs using I/O streams, Applet creation, Event handling and Swings

UNIT – I

Introduction: Java Essentials, JVM, Java Features, Creation and Execution of Programs, Data Types, Structure of Java Program, Type Casting, Conditional Statements, Loops, Classes, Objects, Class Declaration, Creating Objects.

UNIT - II

Methods: Declaration and Invocation, Method Overloading, Constructors – Parameterized Constructors, Constructor Overloading, Cleaning-up unused Objects. Class Variables &Method-static Keyword, this Keyword, One-Dimensional Arrays, Two-Dimensional Arrays, Command-Line Arguments, Inner Class. Inheritance: Introduction, Types of Inheritance, extends Keyword, Examples, Method Overriding, super, final Keyword, Abstract classes, Interfaces, Abstract Classes Verses Interfaces. **Packages:** Creating and Using Packages, Access Protection, Wrapper Classes, String Class, String Buffer Class.

UNIT – III

Exception: Introduction, Types, Exception Handling Techniques, User-Defined Exception. **Multithreading:** Introduction, Main Thread and Creation of New Threads –By Inheriting the Thread Class or Implementing the Runnable Interface, Thread Lifecycle, Thread Priority and Synchronization. **Input/Output:** Introduction, java.io Package, File Streams, File Input Stream Class, File Output Stream Class, Scanner Class, Buffered Input Stream Class, Buffered Output Stream Class, Random Access File Class.

UNIT - IV

Applets: Introduction, Example, Life Cycle, Applet Class, Common Methods Used in Displaying the Output (Graphics Class). **Event Handling:** Introduction, Types of Events, Example. AWT: Introduction, Components, Containers, Button, Label, Checkbox, Radio Buttons, Container Class, Layouts. **Swings:** Introduction, Differences between Swing and AWT, JFrame, JApplet, JPanel, Components in Swings, Layout Managers, JTable.

References:

1. Sachin Malhotra, Saurabh Choudhary: Programming in Java (2e)
2. Bruce Eckel: Thinking in Java (4e)
3. Herbert Schildt: Java: The Complete Reference (9e)
4. Y. Daniel Liang: Introduction to Java Programming (10e)
5. Paul Deitel, Harvey Deitel: Java: How To Program (10e)
6. Cay S. Horsttnann: Core Java Volume I –Fundamentals (10e)

B.Sc. II-Year, IV-Semester (CBCS) Data Science Syllabus

Paper-IV (Practical): Programming in Java (Lab)

[2 HPW :: 1 Credit :: 25 Marks]

Course objectives and Outcomes:

1. Able to write java programs by developing logic to the given simple problem.
2. Able write java program with using OOPs concepts
3. Able to write java programs with usage of I/O streams, applets, string operations, event handling, swings.

List of Programs:

1. Write a java program to find the largest of 'n' natural numbers.
2. Write a java program to find whether a given number is prime or not.
3. Write a menu driven program for following: Display a Fibonacci series and Compute Factorial of a number.
4. Write a java program to check whether a given number is odd or even.
5. Write a java program to check whether a given string is palindrome or not.
6. Write a java program to print the sum and product of digits of an Integer & reverse the Integer.
7. Write a program to create an array of 10 integers. Accept values from the user in that Array. Input another number from the user and find out how many numbers are equal to the number passed, how many are greater and how many are less than the number passed.
8. Write a java program that will prompt the user for a list of 5 prices. Compute the average of the prices and find out all the prices that are higher than the calculated average.
9. Write a java program in java to input N numbers in an array and print out the Armstrong numbers from the set.
10. Write java program for the following matrix operations: Addition of two matrices and Transpose of a matrix
11. Write a java program that computes the area of a circle, rectangle and a Cylinder using function overloading.
12. Write a java program for the implementation of multiple inheritance using interfaces to calculate the area of a rectangle and triangle.
13. Write a java program to create a frame window in an Applet. Display your name, address and qualification in the frame window.
14. Write a java program to draw a line between two coordinates in a window.
15. Write a java program to display the following graphics in an applet window. a. Rectangles b. Circles c. Ellipses d. Arcs e. Polygons
16. Write a program that reads two integer numbers for the variables a and b. If any other character except number (0-9) is entered then the error is caught by Number Format Exception object. After that ex.get Message () prints the information about the error occurring causes.
17. Write a program for the following string operations: a. Compare two strings b. concatenate two strings c. Compute length of a string.
18. Create a class called Fraction that can be used to represent the ratio of two integers. Include appropriate construction.

B.Sc. II-Year, IV-Semester (CBCS) Data Science Syllabus

SEC-4 Paper (Theory): Web Technology

[2 HPW :: 2 Credits :: 50 Marks (External:40, Internal:10)]

Course objectives and Outcomes:

1. Able to write web programs using HTML for developing web pages
2. Able to write web programs using Java script.

UNIT – I

Introduction to HTML: Introduction, first HTML, Headings, Linking, Images, special characters and horizontal rules, Lists, Tables, Frames, Forms, internal linking, meta-Elements. CASCADING STYLE SHEETS – Introduction, Inline Styles, Embedded Style Sheets, Conflicting Styles, linking external sheets, position Elements, box model and text flow, media types, building a CSS drop-down menu, user style sheets, CSS3.

UNIT – II

Introduction to Java Scripting: introduction, simple program, prompt dialog and alert boxes, memory concepts, operators. Conditional and loop Statements: decision making, control structures, if… else statement, while, counter-controlled repetitions, switch statement, do… while statement, break and continue statements. Functions: program modules in JavaScript, programmer–defined functions, functions definition, scope rules, global functions, Recursion.

Reference:

H. M. Deitel, P.J. Deitel: Internet & World Wide Web: How to Program, 4e -Pearson edition.

B.Sc. III-Year, V-Semester (CBCS) Data Science Syllabus
Paper-V(Theory): Machine Learning
[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Course objectives and Outcomes:

1. Able to know the basic principles and limitations of machine learning.
2. Able to know the various supervised, unsupervised, bio-inspired Probabilistic Modeling and Association Rule Mining.
3. Able to implement the various machine learning algorithms for data analysis using Python and Weka.

UNIT-I

Introduction: Introduction to Machine Learning (ML), Types of ML: Supervised, Unsupervised, Reinforcement**,** Canonical Learning Problems, The Decision Tree Model of Learning, Formalizing the Learning Problem. **Limits of Learning:** Data Generating Distributions, Inductive Bias, Underfitting and Overfitting, Separation of data to training and test, parameters and Hyperparameters, Real World Applications of Machine Learning. **Geometry and Nearest Neighbors:** From Data to Feature Vectors, K-Nearest Neighbors, Decision Boundaries, K-Means Clustering, High Dimensions.

UNIT-II

Practical Issues: Importance of Good Features, Irrelevant and Redundant Features, Feature Pruning and Normalization, Combinatorial Feature Explosion, Model Evaluation: Cross-Validation, Performance Metrics (Accuracy, Precision, Recall, F1 Score, ROC, AUC), Hypothesis Testing and Statistical Significance, Debugging Learning Algorithms, Bias Variance tradeoff. **Linear Models:** The Optimization Frame work for Linear Models, Convex Surrogate Loss Functions, Weight Regularization, Optimization and Gradient Descent, Support Vector Machines.

UNIT-III

Probabilistic Modeling: Classification by Density Estimation, Statistical Estimation, Naïve Bayes Models, Prediction. **Neural Networks:** Basics of Neural Networks, Activation Functions, The Perceptron Algorithm**,** Limitations of the Perceptron, Bio-inspired Multi-Layer Networks, The Back-propagation Algorithm, Initialization and Convergence of Neural Networks, Beyond two layers, Breadth vs Depth, Basis Functions.

UNIT-IV

Unsupervised Learning: Clustering Introduction, Similarity and Distance Measures, Agglomerative Algorithms, Divisive Clustering, Minimal Spanning Tree. **Association Rules:** Introduction, large Itemset, Apriori Algorithm.

References:

1. Hal Daume III (2017): A Course in Machine Learning (CIML). (http://ciml.info/)
2. Margaret H Dunham (2003): Data Mining: Introductory and Advanced Topics, Pearson Education.
3. Aurélien Géron (2019): Hands on Machine Learning with SciKit-Learn, Keras and Tensor Flow. O'Reily.
4. Chris Albo (2018): Machine Learning with Python Cookbook., O'Reily.
5. Andreas C Miller, Sarah Guido (2017): Introduction to Machine Learning with Python: A guide. O'Reily.

B.Sc. III-Year, V-Semester (CBCS) Data Science Syllabus
Paper-V(Practical): Machine Learning (Lab)
[2 HPW :: 1 Credit :: 25 Marks]

Course objectives and Outcomes:

1. Able to implement the machine learning algorithms: Decision tree, K-means, Perceptron learning, Naïve Bayes, Apriori and agglomerative clustering algorithms using Scikit learn and Weka on the datasets.

List of Practical's:

1. Write a Python program using Scikit-learn to split the Iris dataset consisting of 150 records into 70% train data and 30% test data. Also split data set into the training set will contain 120 records and the test set contains 30 of those records. Print both datasets. Examine the similar and dissimilar records.
2. Write Python program to use sklearn's Decision Tree Classifier to build a decision tree for the sklearn's datasets. Implement functions to find the importance of a split (entropy, information gain, gini measure)
3. Write a Python program to implement your own version of the K-means algorithm. Then apply it to different datasets and evaluate the performance.
4. Design a Perceptron classifier to classify handwritten numerical digits (0-9). Implement using scikit or Weka.
5. Write a Python program to classify text as spam or not spam using the Naïve Bayes Classifier
6. Use WEKA and experiment with the following classifiers: Association Rule Mining (Apriori), Agglomerative and Divisive Clustering.

Note: Students are expected to use machine learning toolkits: Scikit and weka.

1. Scikit-learn(https://scikit-learn.org/) an open-source machine learning Python library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.
2. The sklearn datasets package embeds small toy datasets. It includes utilities to load these datasets. It also includes methods to load and fetch popular reference datasets and features some artificial data generators. Students are expected to study and make use of these datasets.
3. Weka (http://www.cs.waikato.ac.nz/ml/weka/)is another widely used ML toolkit. Weka also to be implemented on various data sets.

References:

1. Ian Witten, Eibe Frank, and Mark Hall, Chris Pal.: Scikit-learn user guide. https://scikit-learn.org/stable//_downloads/scikit-learn-docs.pdf

2. Morgan Kaufmann: Data Mining: Practical Machine Learning Tools and Techniques, 4th Edition.

B.Sc. III-Year, VI-Semester (CBCS): Data Science Syllabus
Paper-VI (Theory): Deep Learning
[4 HPW :: 4 Credits :: 100 Marks (External: 80, Internal: 20)]

Course Objectives & Outcomes:

1. To know the basics of Deep learning concepts.
2. Evaluate deep neural networks for various real-world applications,
3. implementing and tuning deep learning models on real datasets.
4. Describe the fundamental principles of deep learning and differentiate between various neural network architectures and analyse the strengths and limitations.
5. Mathematical concepts underlying deep learning algorithms to solve real-world problems in image classification, object detection, natural language processing, and time series forecasting.

UNIT – I

Basics of Artificial Neural Networks (ANN), Human vs Computers, Organization of the Brain, Biological Activations of Neuron; Characteristics of ANN, Types of Neuron Activation Function, Signal functions and their properties, monotonicity. ANN Architecture, Classification Taxonomy of ANN. Un-supervised and Reinforcement learning. Learning tasks, Memory, Adaptation, Statistical nature of the learning process. Statistical learning theory**.** Gathering and partitioning of data for ANN and its pre and post processing

UNIT-II

Neural Networks, Data representations for neural networks, Scalars (0D tensors), Vectors (1D tensors), Matrices (2D tensors), 3D tensors and higher-dimensional tensors, Key attributes, Manipulating tensors in Numpy, The notion of data batches, Real-world examples of data tensors, Vector data, Timeseries data or sequence data, Image data, Video data.

UNIT-III

Perceptron Learning Algorithm, and its applications. Gradient-based optimization, Derivative of a tensor operation, Gradient Descent Learning, Stochastic gradient descent, Back-Propagation Learning Algorithms.

UNIT-IV

Introduction to Keras, Keras, TensorFlow, Theano, and CNTK. Understanding the LSTM and GRU layers. Applications of ANN in Classification, Clustering, Regression.

References:

1. Hay kin, S. (1994). Neural Networks: A Comprehensive Foundation. New York: Macmillan Publishing. A comprehensive book and contains a great deal of background theory
2. Yagna Narayana, B. (1999): “Artificial Neural Networks” PHI
3. Bart Kosko (1997): Neural Networks and Fuzzy systems, PHI
4. Jacek M. Zurada (1992): Artificial Neural Systems, West Publishing Company.
5. Carling, A. (1992). Introducing Neural Networks. Wilmslow, UK: Sigma Press.
6. Fausett, L. (1994). Fundamentals of Neural Networks. New York: Prentice Hall.

B.Sc. III-Year, VI-Semester (CBCS): Data Science Syllabus
Practical Paper-VI: Deep Learning (Lab)
[2 HPW :: 1 Credit :: 25 Marks]

Course Objectives and outcomes:

1. Implementation of basic deep learning (Keras) techniques to data sets.
2. Students are expected to learn Keras deep-learning framework (https://keras.io), which is open source and free to download. Keras workflow consisting of four steps (i) Define your training data: input tensors and target tensors (ii) Define a network of layers (or model) that maps your inputs to your targets (iii) Configure the learning process by choosing a loss function, an optimizer, and some metrics to monitor (iv) Iterate on your training data by calling the fit() method of your model

List of Practical's:

1. Consider the **IMDB dataset**, a set of 50,000 highly polarized reviews from the Internet Movie Database. Split into 25,000 reviews for training and 25,000 reviews for testing, each set consisting of 50% negative and 50% positive reviews. (practice with different %). The IMDB dataset comes packaged with Keras. *Build a network to classify* movie reviews as positive or negative, based on the text content of the reviews. Writing the inferences.
2. Consider the **Reuters dataset**, a set of short newswires and their topics, published by Reuters in 1986. It's a simple, widely used toy dataset for text classification. There are 46 different topics; some topics are more represented than others, but each topic has at least 10 examples in the training set. Reuters dataset comes packaged as part of Keras. Build a network to classify Reuters newswires into 46 mutually exclusive topics. *Each data point should be classified* into only one category (in this case, topic). The problem is more specifically an instance of single-label, *multiclass classification*.
3. Consider the **Boston Housing Price dataset** has an interesting difference from the two previous examples. It has relatively few data points: only 506, split between 404 training samples and 102 test samples. And each feature in the input data (for example, the crime rate) has a different scale. For instance, some values are proportions, which take values between 0 and 1; others take values between 1 and 12, others between 0 and 100, and so on. The two previous examples were classification problems, where the goal was to predict a single discrete label of an input data point. Another common type of machine-learning problem is *regression*, which consists of predicting a continuous value instead of a discrete label. You'll attempt to predict the median price of homes in a given Boston suburb in the mid-1970s, given data points about the suburb at the time, such as the crime rate, the local property tax rate, and so on.
4. **Datasets:** Publicly available appropriate datasets **like Fisher Iris, online foods, heart_failure_clinical_records, cancer, Mental Health Dataset, Indian Food Dataset, water-quality-1etc** can be used for the classification and clustering on similar lines.

B.Sc. III-Year, VI-Semester (CBCS): Data Science Syllabus
Paper-VI: Data Science Project
[4 HPW :: 4 Credit :: 100 Marks :: (External: 80, Internal: 20)]

Course Objectives and Outcomes:

1. To enhance practical and professional skills.
2. To familiarize tools and techniques of systematic Literature survey and documentation
3. To expose the students to industry practices and team work and Demonstrate effective written and oral communication skills.
4. To encourage students to work with innovative and entrepreneurial ideas.
5. Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to real-world problems.
6. Effectively plan a project and confidently perform all aspects of project management

Project is of duration maximum 4 months. Each student carries out a group project /individually. Use standard thesis format of O.U. for project Report (Times New Roman font, Title page, declaration by student, certificate from the supervisor, Contents page etc.). Should submit Hard bound copy covering the topics mentioned above before the last instruction date of the semester. A student can carry out the project at industry also if college permits. A student with the help of the guide formulates the problem. The aim of project work is to develop solutions to realistic problems (related to Data Science) applying the knowledge and skills obtained in different courses, new technologies and current industry practices. The department will appoint a project coordinator and Internal supervisor who will coordinate: formalise the project proposal with problem definition, scope, literature survey, Collection of dataset(s), applying the relevant data preprocessing techniques, Dats description, dataset objectives, Hypothesis framed, Descriptive statistics, Data visualization, machine and deep learning techniques implementation (minimum five), evaluation of metrics, drawing inferences and its interpretation should be placed in the Project Report along with code of implementation and outputs. Conducting seminar presentations (minimum three total carries 20 marks). Each student will be required to:

1. Submit a one-page synopsis before the seminar for display on notice board.
2. A 30 minutes presentation of project followed by 10 minutes discussion.
3. Submit a technical write-up on the talk.
