B.Sc. (DATA SCIENCE)

Semester Pattern Syllabus (CBCS) : w. e. f. : Academic Year : 2020-21

(With Mathematics Combination)

Year	Semester	Theory / Practical	Paper Title	Work Load (Hrs/Week)	# Credits	Marks
I	FIRST	**Paper - I (DSC - A)**	**Fundamentals of Information Technology**	**4**	**4**	**100**
		Practical – 1	Fundamentals of Information Technology (Lab)	*3*	*1*	*50*
	SECOND	**Paper - II (DSC - B)**	**Problem solving and Python Programming**	**4**	**4**	**100**
		Practical – 2	Problem solving and Python Programming (Lab)	*3*	*1*	*50*
II	THIRD	**SEC – 1**	**University Specified**	**2**	**2**	**50**
		SEC – 2	**Mini Project**	**2**	**2**	**50**
		Paper - III (DSC - C)	**Data Engineering with Python**	**4**	**4**	**100**
		Practical – 3	Data Engineering with Python (Lab)	*3*	*1*	*50*
	FOURTH	**SEC – 3**	**University Specified**	**2**	**2**	**50**
		SEC – 4	**Mini Project**	**2**	**2**	**50**
		Paper - IV (DSC - D)	**Machine Learning**	**4**	**4**	**100**
		Practical – 4	Machine Learning (Lab)	*3*	*1*	*50*
III	FIFTH	**Paper – V (A) (DSE - A)**	**Natural Language Processing**	***4***	***4***	***100***
		Paper – V (B) (DSE - A)	**No SQL Data Bases**	**4**	**4**	**100**
		Practical – 5(A)	Natural Language Processing (Lab)	*3*	*1*	*50*
		Practical – 5(B)	No SQL Data Bases (Lab)	*3*	*1*	*50*
		Paper VI – GE	*Data Structures and Algorithms*	***4***	***4***	***100***
	SIXTH	**Paper – VII (A) (DSE - B)**	**Big Data**	**4**	**4**	**100**
		Paper- VII (B) (DSE - B)	**Deep Learning**	**4**	**4**	**100**
		Practical – 7(A)	Big Data (Lab)	*3*	*1*	*50*
		Practical – 7(B)	Deep Learning (Lab)	*3*	*1*	*50*
		Paper VIII (Project)	***Major Project***	***4***	***4***	***100***

B.Sc. (DATA SCIENCE)
Theory Question Paper Pattern
w.e.f: Academic Year: 2020-21
(With Mathematics Combination)

Time: 3 hours] [Max. Marks: 80

Section - A

Answer any EIGHT questions. All questions carry equal marks. (8Qx4m=32)

1. From Unit I
2. From Unit I
3. From Unit I
4. From Unit II
5. From Unit II
6. From Unit II
7. From Unit III
8. From Unit III
9. From Unit III
10. From Unit IV
11. From Unit IV
12. From Unit IV

Section - B

Answer ALL questions. All questions carry equal marks. (4Qx12m=48)

13. a) From Unit I
    (or)
    b) From Unit I

14. a) From Unit II
    (or)
    b) From Unit II

15. a) From Unit III
    (or)
    b) From Unit III

16. a) From Unit IV
    (or)
    b) From Unit IV

***

B.Sc. I Year I Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - I)

Paper – I : Fundamentals of Information Technology

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objectives:

1. **To deal with the basic concepts of computers.**
2. **To discuss about the computer hardware, its components and basic computer architecture.**
3. **To understand the basic computer software including the operating system and its concepts.**
4. **To introduce the software development process**
5. **To introduce the basic concept of programming**

Outcomes:

Students should be able to

1. Identify the components of a computer and their functions.
2. Understand the concept of networking, LAN, Internet, and working of www.
3. Understand the notion of problem solving using computer by programming
4. Understand the notion of Software Project and the Process of software development

Unit-I

Data and Information: Introduction, Types of Data, Simple Model of a Computer, Data Processing Using a Computer, Desktop Computer [Reference 1]

Acquisition of Numbers and Textual Data: Introduction, Input Units, Internal Representation of Numeric Data, Representation of Characters in Computers, Error-Detecting Codes [Reference 1]

Unit-II

Data Storage: Introduction, Storage Cell, Physical Devices Used as Storage Cells, Random Access Memory, Read Only Memory, Secondary Storage, Compact Disk Read Only Memory (CDROM), Archival Store [Reference 1]

Central Processing Unit: Introduction, Structure of a Central Processing Unit, Specifications of a CPU, Interconnection of CPU with Memory and I/O Units, Embedded Processors [Reference 1]

Unit-III

Computer Networks: Introduction, Local Area Network (LAN), Applications of LAN, Wide Area Network (WAN), Internet, Naming Computers Connected to Internet, Future of Internet Technology [Reference 1]

Input Output Devices: Introduction, Keyboard, Video Display Devices, Touch Screen Display, E-Ink Display, Printers, Audio Output [Reference 1]

Computer Software: Introduction, Operating System, Programming Languages, Classification of Programming Languages, Classification of Programming Languages Based on Applications [Reference 1]

Unit-IV

The Software Problem: Cost, Schedule, and Quality, Scale and Change [Reference 2]
Software Processes: Process and Project, Component Software Processes, Software Development Process Models [Reference 2]
Programming Principles and Guidelines: Structured Programming, Information Hiding, Some Programming Practices, Coding Standards [Reference 2]

References

1. V Rajaraman. Introduction to Information Technology, 3rd Edition, PHI Learning Private Limited, 2018
2. Pankaj Jalote. Concise Introduction toSoftware Engineering, Springer, 2011

B.Sc. I Year I Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - I)

Practical - 1 : Fundamentals of Information Technology (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objective

The main objective of this laboratory is to familiarize the students with the basic hardware and software in computers

Exercises

1. Assembly and disassembly of a system box and identifying various parts inside the system box to recognize various parts of a typical computer system
2. Assembly and disassembly of peripheral devices- keyboard and mouse and study of their interface cables, connectors and ports.
3. Installation of Operating Systems-Windows and Linux
4. Disk defragmentation using system tool.
5. Procedure of disk partition and its operation (Shrinking, Extending, Delete, Format).
6. Installing and uninstalling of device drivers using control panel.
7. Working practice on windows operating system and Linux operating system: creating file, folder. Copying, moving, deleting file, folder
8. User Account creation and its feature on Windows Operating System and Changing resolution, color, appearances, and Changing System Date and Time.
9. Installation and using various wireless input devices (Keyboard/Mouse/Scanners etc.,)under Windows/Linux.
10. Study of various types of memory chips and various types of hard disk drives, partition and formatting of hard disk.
11. Installation of scanner, modem and network cards in Windows/Linux.
12. Assembly and disassembly of printer, installing a printer, taking test page, and using printer under Windows/Linux.
13. Installation of application software's – Office Automation, Anti-Virus.
14. Demonstrate the usage of Word and Power point in Windows and Linux
15. Configure Internet connection, Email Account creation, reading, writing and sending emails with attachment.

B.Sc. I Year II Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - II)

Paper – II : Problem Solving and Python Programming

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objectives

The main objective is to teach Computational thinking using Python.

- To know the basics of Programming
- 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.

Outcomes:

On completion of the course, students will be able to:

1. Develop algorithmic solutions to simple computational problems.
2. Develop and execute simple Python programs.
3. Develop simple Python programs for solving problems.
4. Structure a Python program into functions.
5. Represent compound data using Python lists, tuples, dictionaries.
6. Read and write data from/to files in Python Programs

Unit-I

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

Introduction to Python Programming: Python Interpreter and InteractiveMode– 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 andmethods; advanced list processing - list comprehension; Illustrative programs: selection sort,
insertion sort, mergesort, 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 The 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/)

Suggested Reading:

1. Learning To Program With Python. Richard L. Halterman. Copyright © 2011
2. Python for Everybody, Exploring Data Using Python 3. Dr. Charles R. Severance. 2016

B.Sc. I Year II Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - II)

Practical - 2 : Problem Solving and Python Programming (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objective

The main objective of this laboratory is to put into practice computational thinking. The students will be expected to write, compile, run and debug Python programs to demonstrate the usage of

- variables, conditionals and 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

Installing Python and Setting up the Environment

Python interpreter can be downloaded for Windows/Linux platform using the link below:
https://www.python.org/downloads/windows/

Exercises

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

5. Program to find the sum of all prime numbers between 1 and 1000.
6. Program that reads set of integers and displays first and second largest numbers.
7. Program to print the sum of first ‘n’ natural numbers.
8. Program to find the product of two matrices.
9. Program to find the roots of a quadratic equation

III. Programs to demonstrate the usage of Functions and Recursion

10. 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

11. 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’.
12. 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

13. Program that accept a string as an argument and return the number of vowels and consonants the string contains.
14. Program that accepts two strings S1, S2, and finds whether they are equal are not.
15. Program to count the number of occurrences of characters in a given string.
16. Program to find whether a given string is palindrome or not

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

17. Program with a function that takes two lists L1 and L2 containing integer numbers as parameters. The return value is a single list containing the pair wise sums of the numbers in L1 and L2.
18. Program to read the lists of numbers as L1, print the lists in reverse order without using reverse function.
22. Write a program that combine lists L1 and L2 into a dictionary.
19. Program to find mean, median, mode for the given set of numbers in a list.
20. Program to find all duplicates in the list.
21. Program to o find all the unique elements of a list.
22. Program to find max and min of a given tuple of integers.
23. Program to find union, intersection, difference, symmetric difference of given two sets.
24. Program to display a list of all unique words in a text file
25. 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
26. Program to analyze the two text files using set operations
27. Write a program to print each line of a file in reverse order.

VI. Programs to demonstrate the usage of Object Oriented Programming

28. Program to implement the inheritance
29. Program to implement the polymorphism

VII. Programs to search and sort the numbers

30. Programs to implement Linear search and Binary search
31. Programs to implement Selection sort, Insertion sort

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

(With Mathematics Combination)

(Examination at the end of Semester - III)

Paper – III : Data Engineering with Python

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to teach how to extract raw data, clean the data, perform transformations on data, load data and visualize the data

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

- Handle different types of files and work with text data
- Use regular expression operations
- Use relational databases via SQL
- Use tabular numeric data
- Use the data structures: data series and frames
- Use PyPlot for visualization

Unit – I
Data Science: Data Analysis Sequence, Data Acquisition Pipeline, Report Structure [Reference 1(Chapter 1-Unit1 to Unit 3)]]
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.pathModules. [Reference 2, Chapter 9)]
Working with Text Data: JSON and XML in Python[Reference 2, Section12.2]

Unit – II
Working with Text Data: Processing HTML Files, Processing Texts in Natural Languages [Reference 1(Chapter3 –Unit 13, and Unit16)
Regular Expression Operations: Using Special Characters, Regular Expression Methods, Named Groups in Python Regular Expressions, Regular Expression with *glob* Module [Reference 2-Chapter 10]

Unit – III
Working with Databases: Setting Up a MySQL Database, Using a MySQL Database: Command Line, Using a MySQL Database, Taming Document Stores: MongoDB [Reference 1 (Chapter4-Unit17toUnit20)]
Working with Tabular Numeric Data(Numpy with Python): NumPy Arrays Creation Using *array()* Function, Array Attributes, NumPy Arrays Creation with Initial Placeholder Content, Integer Indexing, Array Indexing, Boolean ArrayIndexing, Slicing and Iterating in Arrays, Basic Arithmetic Operations on NumPy Arrays, Mathematical Functions in NumPy, Changing the Shape of an Array, Stacking and Splitting of Arrays, Broadcasting in Arrays. [Reference 2: Section 12.3)]

Unit – IV

Working with Data Series and Frames: Pandas Data Structures, Reshaping Data, Handling Missing Data, Combining Data, Ordering and Describing Data, Transforming Data, Taming Pandas File I/O [Reference 1 (Chapter 6-Unit 31 to Unit 37)]

Plotting: Basic Plotting with PyPlot, Getting to Know Other Plot Types, Mastering Embellishments, Plotting with Pandas [Reference 1 (Chapter8-Unit 41 to Unit 44)]

References:

1. Data Science Essentials in Python: Collect, Organize, Explore, Predict, Value. Dmitry Zinoriev, The Pragmatic Programmers LLC, 2016
2. Introduction to Python Programming**.** Gowrishankar S., Veena A. CRC Press, Taylor & Francis Group, 2019

Suggested Reading

3. Python for Everybody: Exploring Data Using Python 3. Charles R Severance, 2016
4. Python Data Analytics – Data Analysis and Science using Pandas, matplotlib and the Python Programming Language. Fabio Nelli, Apress, 2015
5. Website Scraping with Python. Using BeautifulSoup and Scrapy. GáborLászlóHajba, Apress, 2018
6. Machine Learning with Python Cookbook:.Practical Solutions from Preprocessing to Deep Learning. Chris Albon, O'Reilly 2018

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

(With Mathematics Combination)

(Examination at the end of Semester - III)

Practical - 3 : Data Engineering with Python (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objective:
The main objective of this laboratory is to put into practice the ETL (extract, transform, load) pipeline which will extract raw data, clean the data, perform transformations on data, load data and visualize the data.

This requires mentoring by TCS.

Libraries
In this course students are expected to extract, transform and load input data that can be text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc.,. For doing this, they should learn the following Python libraries/modules:
pandas, numpy, BeautifulSoup, pymysql, pymongo, nltk, matplotlib

Datasets
For this laboratory, appropriate publicly available datasets, can be studied and used.
Example:
MNIST (http://yann.lecun.com/exdb/mnist/),
UCI Machine Learning Repository(https://archive.ics.uci.edu/ml/datasets.html),
Kaggle(https://www.kaggle.com/datasets)
Twitter Data

Exercises

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 datastructures: 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

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

(With Mathematics Combination)

(Examination at the end of Semester - IV)

Paper – IV : Machine Learning

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objectives: The main objective of this course is to teach the principles and foundations of machine learning algorithms

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

- Basics of Machine Learning and its limitations
- Machine Learning Algorithms: supervised, unsupervised, bio-inspired
- Probabilistic Modeling and Association Rule Mining

Unit-I
Introduction: What does it mean to learn, Some canonical Learning Problems, The Decision Tree Model of Learning, Formalizing the Learning Problem [Reference 1], ID3 Algorithm [[Reference 2]
Limits of Learning: Data Generating Distributions, Inductive Bias, Not Everything is learnable, Underfitting and Overfitting, Separation of training and test Data, Models, parameters and Hyperparameters, Real World Applications of Machine Learning [Reference 1]
Geometry and Nearest Neighbors: From Data to Feature Vectors, k-Nearest Neighbors, Decision Boundaries, k-means Clustering, High Dimensions [Reference 1]

Unit-II
The Perceptron: Bio-inspired Learning, The Perceptron Algorithm, Geometric Interpretation, Interpreting Perceptron Weights, Perceptron Convergence and Linear Separability, Improved Generalization, Limitations of the Perceptron [Reference 1]
Practical Issues: Importance of Good Features, Irrelevant and Redundant Features, Feature Pruning and Normalization, Combinatorial Feature Explosion, Evaluating Model Performance, Cross Validation, Hypothesis Testing and Statistical Significance, Debugging Learning Algorithms, Bias Variance tradeoff [Reference 1]
Linear Models: The Optimization Framework for Linear Models, Convex Surrogate Loss Functions, Weight Regularization, Optimization and Gradient Descent, Support Vector Machines [Reference 1]

Unit-III
Probabilistic Modeling: Classification by Density Estimation, Statistical Estimation, Naïve Bayes Models, Prediction [Reference 1]
Neural Networks: Bio-inspired Multi-Layer Networks, The Back-propagation Algorithm, Initialization and Convergence of Neural Networks, Beyond two layers, Breadth vs Depth, Basis Functions [Reference 1]

Unit IV

Unsupervised Learning: Clustering Introduction, Similarity and Distance Measures, Agglomerative Algorithms, Divisive Clustering, Minmum Spanning Tree [Reference 2]
Association Rules: Introduction, large Itemsets, Apriori Algorithm [Reference 2]

References:

1. A Course in Machine Learning (CIML). Hal Daume III, 2017 (freely available online) http://ciml.info/
2. Data Mining: Introductory and Advanced Topics. Margaret H Dunham, Pearson Education, 2003

Suggested Reading:

3. Hands on Machine Learning with SciKit-Learn, Keras and Tensor Flow. AurélienGéron. O'Reily, 2019
4. Machine Learning with Python Cookbook. Chris Albo, O'Reily, 2018
5. Introduction to Machine Learning with Python: A guide. Andreas C Miller, Sarah Guido. O'Reily, 2017

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

(With Mathematics Combination)

(Examination at the end of Semester - IV)

Practical - 4 : Machine Learning (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objective:
The main objective of this laboratory is to put into practice the various machine learning algorithms for data analysis using Python and Weka.

ML Toolkits
Students are expected to learn

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. Weka (http://www.cs.waikato.ac.nz/ml/weka/)is another widely used ML toolkit.

Datasets

1. 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
2. Weka also has provides various data sets.

References:

1. scikit-learn user guide.https://scikit-learn.org/stable//_downloads/scikit-learn-docs.pdf
2. Ian Witten, Eibe Frank, and Mark Hall, Chris Pal. DATA MINING: Practical Machine Learning Tools and Techniques, 4th Edition. Morgan Kaufmann.

Exercises

8. Write a Python program using Scikit-learn to split the iris dataset into 70% train data and 30% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Print both datasets
9. Write Python program to use sklearn'sDecisionTreeClassifier to build a decision tree for the sklearn's datasets. Implement functions to find the importance of a split (entropy, information gain, gini measure)
10. Write a Python program to implement your own version of the K-means algorithm. Then apply it to different datasets and evaluate the performance.
11. Design a perceptron classifier to classify handwritten numerical digits (0-9). Implement using scikit or Weka.
12. Write a Python program to classify text as spam or not spam using the Naïve Bayes Classifier
13. Use WEKA and experiment with the following classifiers: Association Rule Mining (Apriori), Agglomerative and Divisive Clustering.

B.Sc. III Year V Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - V)

Paper – V(A) : Natural Language Processing

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to give a practical introduction to NLP. It deals with morphological processing, syntactic parsing, information extraction, probabilistic NLP and classification of text using Python's NLTK Library.

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

- Write Python programs to manipulate and analyze language data
- Understand key concepts from NLP and linguistics to describe and analyze language
- Understand the data structures and algorithms that are used in NLP
- Classify texts using machine learning and deep learning

Unit-I
Language Processing and Python: Computing with Language: Texts and Words, A Closer Look at Python: Texts as Lists of Words, Computing with Language: Simple Statistics, Back to Python: Making Decisions and Taking Control, Automatic Natural Language Understanding [Reference 1]
Accessing Text Corpora and Lexical Resources: Accessing Text Corpora, Conditional Frequency Distributions, Lexical Resources, WordNet [Reference 1]

Unit-II
Processing Raw Text: Accessing Text from the Web and from Disk, Strings: Text Processing at the Lowest Level, Text Processing with Unicode, Regular Expressions for Detecting Word Patterns, Useful Applications of Regular Expressions, Normalizing Text, Regular Expressions for Tokenizing Text, Segmentation, Formatting: From Lists to Strings. [Reference 1]
Categorizing and Tagging Words: Using a Tagger, Tagged Corpora, Mapping Words to Properties Using Python Dictionaries, Automatic Tagging, N-Gram Tagging, Transformation-Based Tagging, How to Determine the Category of a Word [Reference 1]

Unit-III
Learning to Classify Text: Supervised Classification, Evaluation, Naive Bayes Classifiers [Reference 1]
Deep Learning for NLP: Introduction to Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks, Classifying Text with Deep Learning [Reference 2]

Unit-IV
Extracting Information from Text
Information Extraction, Chunking, Developing and Evaluating Chunkers, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction. [Reference 1]

Analyzing Sentence Structure
Some Grammatical Dilemmas, What's the Use of Syntax. Context-Free Grammar, Parsing with Context-Free Grammar, [Reference 1]

References:

1. Natural Language Processing with Python. Steven Bird, Ewan Klein, and Edward Lope, O'Reily, 2009
2. Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. Akshay Kulkarni, AdarshaShivananda, Apress, 2019

Suggested Reading:

3. Allen James, Natural Language Understanding, Benjamin/Cumming,1995.
4. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.

B.Sc. III Year V Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - V)

Practical – 5(A) : Natural Language Processing (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objective: The main objective of this laboratory is to write programs that manipulate and analyze language data using Python

This lab requires mentoring sessions from TCS.

Python Packages

Students are expected to know/ learn the following Python NLP packages

- NLTK (www.nltk.org/ (http://www.nltk.org/))
- Spacy (https://spacy.io/)
- TextBlob (http://textblob.readthedocs.io/en/dev/
- Gensim (https://pypi.python.org/pypi/gensim)
- Pattern (https://pypi.python.org/pypi/Pattern)

Datasets:

1. NLTK includes a small selection of texts from the Project Gutenberg electronic text archive, which contains some 25,000 free electronic books, hosted at *http://www.gutenberg.org/*.
2. The Brown Corpus contains text from 500 sources, and the sources have been categorized by genre, such as *news*, *editorial*, and so on (*http://icame.uib.no/brown/bcm-los.html*).
3. Wikipedia Articles

Or any other dataset of your choice

Reference:

Jacob Perkins. Python 3 Text Processing with NLTK 3 Cookbook. Packt Publishing. 2014

Exercises:

1. Text segmentation: Segment a text into linguistically meaningful units, such as paragraphs, sentences, or words. Write programs to segment text (in different formats) into tokens (words and word-like units) using regular expressions. Compare an automatic tokenization with a gold standard
2. Part-of-speech tagging: Label words (tokens) with parts of speech such as noun, adjective, and verb using a variety of tagging methods , e.g., default tagger, regular expression tagger, unigram tagger, and n-gram taggers.
3. Text classification: Categorize text documents into predefined classes using Naïve Bayes Classifier and the Perceptron model
4. Chunk extraction, or partial parsing: Extract short phrases from a part-of-speech tagged sentence. This is different from full parsing in that we're interested in standalone chunks, or phrases, instead of full parse trees

5. Parsing: parsing specific kinds of data, focusing primarily on dates, times, and HTML. Make use of the following preprocessing libraries:
   - dateutil which provides datetime parsing and timezone conversion
   - lxml and BeautifulSoup which can parse, clean, and convert HTML
   - charade and UnicodeDammit which can detect and convert text character encoding
6. Sentiment Analysis: Using Libraries TextBlob and nltk, give the sentiment of a document

B.Sc. III Year V Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - V)

Paper – V(B) : NoSQL Data Bases

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to cover core concepts of NoSQL databases, along with an example database for each of the key-value, document, column family, and graph databases

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

- Understand the need for NoSQL databases and their characteristics
- Understand the concepts of NoSQL databases
- Implement the concepts of NoSQL databases using four example databases: Redis for key-value databases, MongoDB for document databases, Cassandra for column-family databases, and Neo4J for graph databases.

Unit-I

Why NoSQL: The Value of Relational Databases, Impedance Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL

Aggregate Data Models: Aggregates, Column-Family Stores, Summarizing Aggregate-Oriented Databases

More Details on Data Models: Relationships, Graph Databases, Schemaless Databases, Materialized Views, Modeling for Data Access

Unit-II

Distribution Models: Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication

Consistency: Update Consistency, Read Consistency, Relaxing Consistency, Relaxing Durability, Quorums

Version Stamps: Business and System Transactions, Version Stamps on Multiple Nodes

Map-Reduce: Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations

Unit-III

Key-Value Databases: What Is a Key-Value Store, Key-Value Store Features, Suitable Use Cases, When Not to Use

Document Databases: What Is a Document Database, Features, Suitable Use Cases, When Not to Use

Unit-IV

Column-Family Stores: What Is a Column-Family Data Store, Features, Suitable Use Cases, When Not to Use

Graph Databases: What Is a Graph Database, Features, Suitable Use Cases, When Not to Use

Reference:

1. Pramod J. Sadalage, Martin Fowler. NoSQL Distilled, Addison Wesley 2013

Suggested Reading

2. Luc Perkins, Eric Redmond, Jim R. Wilson. Seven Databases in Seven Weeks. The Pragmatic Bookshelf, 2018
3. Guy Harrison. Next Generation Databases: NoSQL, NewSQL, and Big Data. Apress, 2015

B.Sc. III Year V Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - V)

Practical – 5(B) : NoSQL Data Bases (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objective: The main objective of this lab is to become familiar with the four NoSQL databases: Redis for key-value databases, MongoDB for document databases, Cassandra for column-family databases, and Neo4J for graph databases

NoSQL Databases:
Redis (http://redis.io)
MongoDB (http://www.mongodb.org)
Cassandra (http://cassandra.apache.org)
Neo4j (http://neo4j.com)

Exercises:

1. Installation of NoSQL Databases: Redis,MongoDB, Cassandra, Neo4j on Windows & Linux
2. Practice CRUD (*Create*, *Read*, *Update*, and *Delete)* operations on the four databases: Redis,MongoDB, Cassandra, Neo4j
3. Usage of Where Clause equivalent in MongoDB
4. Usage of operations in MongoDB – AND in MongoDB, OR in MongoDB, Limit Records and Sort Records. Usage of operations in MongoDB – Indexing, Advanced Indexing, Aggregation and Map Reduce.
5. Practice with ' macdonalds ' collection data for document oriented database. Import restaurants collection and apply some queries to get specified output.
6. Write a program to count the number of occurrences of a word using MapReduce

B.Sc. III Year V Semester (CBCS) : Statistics Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - V)

Paper – VI - GE : Data Structures and Algorithms

[4 HPW :: 4 Credits :: 100 Marks]

Objectives:

- To introduce the time and space complexities ofalgorithms.
- To discuss the linear and non-linear data structures and theirapplications.
- To introduce the creation, insertion and deletion operations on binary search trees and balanced binary searchtrees.
- To introduce various internal sorting techniques and their timecomplexities

Outcomes:

Students will be

- Able to analyze the time and space complexities ofalgorithms.
- Able to implement linear, non-linear data structures and balanced binarytrees
- Able to analyse and implement various kinds of searching and sorting techniques.
- Able to find a suitable data structure and algorithm to solve a real world problem.

UNIT–I

Performance and Complexity Analysis: Space Complexity, Time Complexity, Asymptotic Notation (Big-Oh), Complexity AnalysisExamples.

Linear List-Array Representation: Vector Representation, Multiple Lists Single Array. **Linear List-Linked Representation:** Singly Linked Lists, Circular Lists, Doubly Linked Lists, Applications (Polynomial Arithmetic).

Arrays and Matrices: Row and Column Major Representations, Sparse Matrices.

Stacks: Array Representation, Linked Representation, Applications (Recursive Calls, Infix to Postfix, Postfix Evaluation).

Queues: Array Representation, Linked Representation.

Skip Lists and Hashing: Skip Lists Representation, Hash Table Representation, Application- Text Compression.

UNIT- II

Trees: Definitions and Properties, Representation of Binary Trees, Operations, Binary TreeTraversal.

Binary Search Trees: Definitions, Operations on Binary Search Trees.

Balanced Search Trees: AVL Trees, and B-Trees.

UNIT –III

Graphs: Definitions and Properties, Representation, Graph Search Methods (Depth First Search and Breadth First Search)

Application of Graphs: Shortest Path Algorithm (Dijkstra), Minimum Spanning Tree (Prim's and Kruskal'sAlgorithms).

UNIT –IV

Searching : Linear Search and Binary Search Techniques and their complexity analysis.

Sorting and Complexity Analysis: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, and Heap Sort.

Algorithm Design Techniques: Greedy algorithm, divide-and-conquer, dynamic programming.

Suggested Reading:

1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, *Data Structures and Algorithms Python* John Wiley & Sons,2013.
2. Mark Allen Weiss, *Data Structures and Problem Solving using C++*, Pearson Education International,2003.
3. SartajSahni, *Data Structures--Algorithms and Applications in C++*, 2nd Edition, Universities Press (India) Pvt. Ltd.,2005.

B.Sc. III Year VI Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - VI)

Paper – VII(A) : Big Data

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

UNIT – I

Getting an overview of Big Data: Introduction to Big Data, Structuring Big Data, Types of Data, Elements of Big Data, Big Data Analytics, Advantages of Big Data Analytics.

Introducing Technologies for Handling Big Data: Distributed and Parallel Computing for Big Data, Cloud Computing and Big Data, Features of Cloud Computing, Cloud Deployment Models, Cloud Services for Big Data, Cloud Providers in Big Data Market.

UNIT – II

Understanding Hadoop Ecosystem: Introducing Hadoop, HDFS and MapReduce, Hadoop functions, Hadoop Ecosystem. **Hadoop Distributed File System**- HDFS Architecture, Concept of Blocks in HDFS Architecture, Namenodes and Datanodes, Features of HDFS. MapReduce.

Introducing HBase- HBase Architecture, Regions, Storing Big Data with HBase, Combining HBase and HDFS, Features of HBase, Hive, Pig and Pig Latin, Sqoop, ZooKeeper, Flume, Oozie.

UNIT- III

Understanding MapReduce Fundamentals and HBase: The MapReduceFramework ,Exploring the features of MapReduce, Working of MapReduce, Techniques to optimize MapReduce Jobs, Hardware/Network Topology, Synchronization, File system, Uses of MapReduce, Role of HBase in Big Data Processing- Characteristics of HBase.

Understanding Big Data Technology Foundations: Exploring the Big Data Stack, Data Sources Layer, Ingestion Layer, Storage Layer, Physical Infrastructure Layer, Platform Management Layer, Security Layer, Monitoring Layer, Visualization Layer.

UNIT – IV

Storing Data in Databases and Data Warehouses: RDBMS and Big Data, Issues with Relational Model, Non – Relational Database, Issues with Non Relational Database, Polyglot Persistence, Integrating Big Data with Traditional Data Warehouse, Big Data Analysis and Data Warehouse.

NoSQL Data Management: Introduction to NoSQL, Characteristics of NoSQL, History of NoSQL, Types of NoSQL Data Models- Key Value Data Model, Column Oriented Data Model, Document Data Model, Graph Databases, Schema-Less Databases, Materialized Views, CAP Theorem.

Reference

1. BIG DATA, Black Book TM, DreamTech Press, 2016 Edition.

Suggested Reading:

2. Seema Acharya, SubhasniChellappan , "BIG DATA and ANALYTICS", Wiley publications, 2016

3. Nathan Marz and James Warren, "BIG DATA- Principles and Best Practices of Scalable Real-Time Systems", 2010

B.Sc. III Year VI Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - VI)

Practical – 7(A) : Big Data (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objectives:

- Installation and understanding of working of HADOOP
- Understanding of MapReduce program paradigm.
- Writing programs in Python using MapReduce
- Understanding working of Pig, Hive
- Understanding of working of Apache Spark Cluster

1. Setting up and Installing Hadoop in its two operating modes:
   - Pseudo distributed,
   - Fully distributed.
2. Implementation of the following file management tasks in Hadoop:
   - Adding files and directories
   - Retrieving files
   - Deleting files
3. Implementation of Word Count Map Reduce program
   - Find the number of occurrence of each word appearing in the input file(s)
   - Performing a MapReduce Job for word search count (look for specific keywords in a file)
4. Map Reduce Program for Stop word elimination:
   - Map Reduce program to eliminate stop words from a large text file.
5. Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented. Data available at: https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all.
   - Find average, max and min temperature for each year in NCDC data set?
   - Filter the readings of a set based on value of the measurement, Output the line of input files associated with a temperature value greater than 30.0 and store it in a separate file.
6. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.
7. Write a Pig Latin script for finding TF-IDF value for book dataset (A corpus of eBooks available at: Project Gutenberg)
8. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes.
9. Install, Deploy & configure Apache Spark Cluster. Run apache spark applications using Scala.
10. Perform Data analytics using Apache Spark on Amazon food dataset, find all the pairs of items frequently reviewed together.

B.Sc. III Year VI Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - VI)

11.

Paper – VII(B) : Deep Learning

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to give a practical introduction to Deep Learning using Keras. It covers the concepts of deep learning and their implementation.

Outcomes:

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

1. Understand the basics of deep learning
2. Understand the usage of tensors in deep learning
3. Use Python deep-learning framework Keras, with Tensor-Flow as a backend engine.

Unit-I

Introduction: History, Hardware, Data, Algorithms

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-II

Tensor operations: Element-wise operations, Broadcasting, Tensor dot,. Tensor reshaping, Geometric interpretation of tensor operations, A geometric interpretation of deep learning,

Unit-III

Gradient-based optimization, Derivative of a tensor operation, Stochastic gradient descent,. Chaining derivatives: the Backpropagation algorithm

Neural networks: Anatomy, Layers, Models, Loss functions and optimizers

Unit-IV

Introduction to Keras, Keras, TensorFlow, Theano, and CNTK

Recurrent neural networks: A recurrent layer in Keras, Understanding the LSTM and GRU layers

Reference:

1. François Chollet. Deep Learning with Python. Manning Publications, 2018

Suggested Reading:

2. AurélienGéron. Hands on Machine Learning with SciKit-Learn, Keras and Tensor Flow. O'Reily, 2019

Andrew W. Trask. Grokking Deep Learning. Manning Publications, 2019

B.Sc. III Year VI Semester (CBCS) : Data Science Syllabus

(With Mathematics Combination)

(Examination at the end of Semester - VI)

Practical – 7(B) : Deep Learning (Lab)

[3 HPW :: 1 Credit :: 50 Marks]

Objectives: The main objective of this lab is to develop deep learning models using Keras

Deep Learning Tools

Students are expected to learn Keras deep-learning framework (https://keras.io), which is open source and free to download. They should have access to a UNIX machine; though it's possible to use Windows, too. It is also recommended that they work on a recent NVIDIA GPU

Exercises:

Note: The exercises should following **Keras workflow** consisting of four steps

1. Define your training data: input tensors and target tensors
2. Define a network of layers (or *model*) that maps your inputs to your targets
3. Configure the learning process by choosing a loss function, an optimizer, and some metrics to monitor
4. Iterate on your training data by calling the fit() method of your model

Exercise 1:

Dataset:

IMDB dataset, a set of 50,000 highly polarized reviews from the Internet Movie Database. They're split into 25,000 reviews for training and 25,000 reviews for testing, each set consisting of 50% negative and 50% positive reviews. the IMDB dataset comes packaged with Keras

Binary Classification Task:

Build a network to classify movie reviews as positive or negative, based on the text content of the reviews.

Exercise 2:

Dataset:

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.

Single-label Multi class Classification Task:

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*.

Exercise 3:

Dataset:

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.

Regression Task:

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.

3. More exercises can be defined on similar lines.