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