AC – 28/03/2025
Item No. – 6.3 (N) (3ab) Sem. III & IV

As Per NEP 2020

University of Mumbai

Syllabus for Minor Vertical 2		
Faculty of Science and Technology		
Board of Studies in Data Science		
Second Year Programme in Minor (Data Science)		
Semester	**III & IV**	
Title of Paper	**Sem.**	**Total Credits 4**
I) Python for Data Science	**III**	**2**
II) Scala for DS		**2**
Title of Paper		**Credits**
I) Advance Python for Data Science	**IV**	**2**
II) Data Analysis with SAS / SPSS /R		**2**
From the Academic Year		**2025-26**

Sem. - III

Syllabus
Minor in Data Science
(Sem.- III)

Title of Paper: Python for Data Science

Sr. No.	Heading	Particulars
1	**Description of the course :** **Including but Not limited to :**	Advanced python programming practical modules make able to acquire knowledge for implementing python code for various applications such as handling data, analysing and visualizing data. Database Management System's practical approach is useful to gain the knowledge for software backend development. It benefits to user by providing data definition, data access, reduced data redundancy, data integrity, data sharing, data organizing, data consistency, data accuracy, and security.
2	**Vertical :**	Minor
3	**Type :**	Practical
4	**Credit**:	2 credits (1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted**:	50 Marks
7	**Course Objectives**: 1. Implement Python for Data Processing – Utilize tuples, regular expressions, date-time functions, and libraries like NumPy and Pandas for data manipulation. 2. Understand Relational Databases & SQL – Identify entities, relationships, and relational structures while implementing constraints using SQL. 3. Perform Data Retrieval & Manipulation in SQL – Execute DML operations, apply built-in functions, retrieve and aggregate data, and work with joins and nested queries. 4. Manage Database Security & Access Control – Implement user access controls, security measures, and database backup strategies.	
8	**Course Outcomes**: 1. Apply Python for Data Handling – Utilize lists, tuples, regular expressions, date-time functions, and libraries like NumPy and Pandas for data processing. 2. Execute SQL Queries for Data Operations – Perform CRUD (Create, Read, Update, Delete) operations, table modifications, and database backup/restoration using SQL. 3. Retrieve & Analyze Data Using SQL – Use aggregate functions, joins, and nested queries to extract meaningful insights from relational databases. 4. Manage Database Security & Optimization – Implement access control, create virtual tables, and optimize database structures for secure and efficient data management.	

9	**Modules**:-	
	Module 1:	
	1a. Write a python code to print your profile. 1b. write a python code to print addition of two numbers. 1c. Write a python code to print square root of number. 1c. Write a python code to calculate area of Triangle. 1d. Write a python code to swap two variables. 2a. Write a python code to create nested tuples. 2b. Write a python code to sort the nested tuple using sorted() function. 2c. Write a python code to copy or clone list. 2d.Write a python code to check immutability property of python tuples. 3a. Write a python code for creating a variable and storing the text that we want to search 3b. Write a python code to retrieve data from HTML file. 3c. Write a python code to print current date in different format. 3d. Write a python code to convert time stamp to date stamp. 3e. Write a python code to develop calendar module. 3f. Write a python code to compare two dates.	
	Module 2:	
	4a. Write a python code to create Numpy Array. 4b. Write a python code to demonstrate basic operations on single array. 4c. Write a python code to create array with 10 elements and slice element from 1st to 5th element. 4d. Write a python code to sort an array alphabetically. 4e. Write a python code to create a filter array that will return maximum values from an array. 5a. Write a python code to demonstrate importing pandas libraries and create data frame object. 5b. Write a python code to show statistical information on given data set. 5c.Write a python code to create pandas series from dictionaries. 5d. Write a python code to demonstrate filter pandas series with Boolean arrays.	
10	**Text Books:** 1. Programming through Python M. T. Savaliya, R.K Maurya, G.M Magar, Staredu Solutions, 1st edition (2018) 2. Python DataScience Handbook, Jake VanderPlas, O'Reilly Media, 1st edition (2016)	
11	**Reference Books:** **1.** Let Us Python, Yashwant Kanetkar, BPB publication , 1st edition (2019) **2.** Programming in Python3, Mark Summerfield, Pearson Education, 2nd edition (2018) **3.** Learning Python, LutzM, O'Reilly- Shroff, 5th edition, 2013. **4.** Beginning Python, Magnus LieHetland, Apress, 2nd edition, 2009. Star Python, Star Certification, Star Certification,1st , 2018.	
12	**Internal Continuous Assessment: 40%**	**External, Semester End Examination 60% Individual Passing in Internal and External Examination**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation.	

Title of Paper Scala for DS

Sr. No.	Heading	Particulars
1	**Description of the course :** **Including but Not limited to :**	This course provides hands-on experience with Scala and its ecosystem for data analysis and machine learning. Students will learn statistical methods, machine learning algorithms, and data processing techniques using Breeze and Apache Spark. The course also covers time-series analysis, feature engineering, and building scalable data pipelines. Through practical exercises, students will gain proficiency in implementing regression models and clustering while handling real-world datasets effectively.
2	**Vertical :**	Minor
3	**Type :**	Practical
4	**Credit**:	2 credits (1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted**:	50 Marks
7	**Course Objectives**: CO1: To set up and configure Scala, SBT, and Apache Spark for programming, data analysis, and large-scale data processing. CO2: To perform statistical calculations, including correlation, frequency distribution, and moving averages using Scala and Breeze, and visualize data insights with Breeze-viz. CO3: To implement machine learning models such as linear regression, logistic regression, and k-means clustering, along with feature engineering for predictive modeling. CO4: To utilize the Breeze library for numerical computations, matrix operations, and time-series data analysis to extract meaningful insights.	
8	**Course Outcomes**: OC1: Students will set up a functional Scala development environment with SBT and execute basic programs for data analysis. OC2: Students will utilize Breeze for numerical operations, matrix manipulations, and statistical computations such as correlation and frequency distribution. OC3: Students will create data visualizations using Breeze-viz and implement machine learning models, including regression and clustering, using Breeze. OC4: Students will work with Apache Spark for large-scale data processing, machine learning pipelines, and time-series analysis to extract meaningful insights.	

9	**Modules**:-
	Module 1:
	0. Set up Scala and SBT on your system. 1. Write a simple Scala program that prints a welcome message for data scientists. 2. Calculate mean, median, and mode of a list of numbers. Implement basic statistical calculations using Scala collections. 3. Generate a random dataset of 10 numbers and calculate its variance and standard deviation. 4. Create a dense vector using Breeze and calculate its sum, mean, and dot product with another vector. 5. Generate a random matrix using Breeze and compute its transpose and determinant. 6. Slice a Breeze matrix to extract a sub-matrix and calculate its row and column sums. 7. Write a program to perform element-wise addition, subtraction, multiplication, and division of two Breeze matrices. 8. Read a CSV file and calculate basic statistics for each numeric column. Use the scala-csv library or similar tools. 9. Handle missing values in a dataset. Replace missing values with the column mean. 10. Filter rows in a dataset where a specific column value exceeds a threshold. 11. Write a program to tokenize and count the frequency of words in a text file. 12. Implement one-hot encoding for a categorical column in a dataset. 13. Create a scatter plot of random data using Breeze-viz. Label the axes and customize the color of points. 14. Generate a histogram of a dataset using Breeze-viz. Experiment with different bin sizes. 15. Plot a line graph for a dataset showing a trend over time. 16. Combine two plots (e.g., scatter and line plot) in a single visualization using Breeze-viz.
	Module 2:
	1. Find the correlation between two lists of numbers. Implement the formula for Pearson correlation coefficient. 2. Calculate the moving average of a time series data using Scala collections. 3. Write a program to compute frequency distribution and cumulative frequency of a dataset. 4. Sort a dataset by a specific column and extract the top 5 rows. 5. Implement linear regression using Breeze. Fit a model to a small dataset and predict a value. 6. Perform logistic regression using Breeze. Classify a dataset with binary labels. 7. Compute the Euclidean distance between two Breeze vectors. Use it for nearest neighbor classification. 8. Cluster a dataset into two groups using k-means clustering in Breeze. 9. Set up Apache Spark locally and count the frequency of words in a text file. 10. Filter rows in a CSV file using Spark DataFrames where a numeric column exceeds a threshold. 11. Perform a group-by operation in Spark DataFrames to compute the average of each group. 12. Join two CSV files in Spark DataFrames based on a common column and write the output to a file.

	13. Create a simple Spark MLlib pipeline to classify data. Use logistic regression or decision trees. 14. Perform basic time series analysis in Scala. Generate synthetic time series data (e.g., daily sales over a month). 15. Create polynomial features from a dataset. Given a list of numbers (e.g., [1, 2, 3]), generate polynomial features up to degree 3 (e.g., [1, 1^2, 1^3, 2, 2^2, 2^3, 3, 3^2, 3^3]).	
10	**Text Books:** 1. Scala for Data Science, by Pascal Bugnion, Packt Publishing, 1st edition (28 January 2016) 2. Mastering Scala by Dennis Alexander, Packt Publishing, 1st edition (2023) 3. Scala 3 Mastery by John Hunt, Apress, 1st edition (2023) 4. Mastering Scala 3 by John Hunt, Apress, 1st edition (2023)	
11	**Reference Books:** 1. Programming Scala by Dean Wampler and Alex Payne, O'Reilly Media, 3rd edition (2021) 2. Scala Cookbook by Alvin Alexander, O'Reilly Media, 2nd edition (2021) 3. Functional Programming in Scala by Paul Chiusano and Rúnar Bjarnason, Manning Publications, 2nd edition (2023)	
12	**Internal Continuous Assessment: 40%**	**External, Semester End Examination 60% Individual Passing in Internal and External Examination**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation.	

Sem. - IV

Syllabus
(Sem.- IV)

Title of Paper : Data Analysis with SAS / SPSS /R

Sr. No.	Heading	Particulars
1	**Description of the course :** **Including but Not limited to :**	Data Analysis with SAS / SPSS /R course provides hands-on training in data analysis techniques using industry-standard tools — SAS, SPSS, or R. It covers essential skills such as data importing, cleaning, transformation, and visualization, along with performing statistical analysis like t-tests, ANOVA, chi-square tests, and regression. Learners will gain practical experience in managing datasets, applying statistical methods, and generating professional reports, preparing them for real-world data analysis tasks across various domains.
2	**Vertical :**	Minor
3	**Type :**	Practical
4	**Credit**:	2 credits (1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted**:	50 Marks
7	**Course Objectives**: CO 1. To understand the SAS/SPSS/R as a tool to data analysis. CO 2. Understand different techniques of data manipulation. CO 3. Use different functions for descriptive statistics. CO 4. To generate the reports after data manipulation.	
8	**Course Outcomes**: OC 1. To Understand the basis of data analytics using a software tool. OC 2. To use data from multiple sources relevant for Data Analytics. OC 3. To Categorize and utilize data for inferential data analytics. CO 4. To integrate the statistical tool with an analytical problem to bring the proficiency.	

9	**Modules**:- Per credit One module can be created
	Module 1: Data Handling, Preparation and Transformation (using SAS or SPSS or R)
	1 Introduction to the SAS, SPSS and R environments – installation, interface overview, and loading data files. 2 Creating datasets from raw data (text files, CSV files, Excel sheets) and importing data into SAS/SPSS/R. 3 Exploring data: Displaying datasets using PROC PRINT (SAS), Data Viewer (SPSS), and View() or print() (R). 4 Applying conditional filters using IF, WHERE, and IF-THEN in SAS; Select Cases in SPSS; and subset() or filter() in R. 5 Sorting data using PROC SORT in SAS, Sort Cases in SPSS, and arrange() in R. 6 Combining and appending datasets using MERGE in SAS, Merge Files in SPSS, and merge() or bind_rows() in R. 7 Selecting and dropping variables using KEEP, DROP in SAS, Variable View in SPSS, and select() in R. 8 Applying basic data cleaning functions: handling missing values using MISSING (SAS), Define Missing Values (SPSS), and na.omit()/replace_na() in R. 9 Performing text manipulation using substr, scan (SAS), String functions (SPSS), and str_sub(), str_split() (R). 10 Creating new variables using transformations and calculations (all three software). 11 Reshaping data using PROC TRANSPOSE (SAS), Restructure Data Wizard (SPSS), and pivot_longer()/pivot_wider() (R). 12 Combining datasets vertically (concatenation) using SET statement (SAS), Merge Files - Add Cases (SPSS), and rbind() (R). 13 Identifying and handling duplicates using PROC SORT NODUPKEY (SAS), Identify Duplicate Cases (SPSS), and distinct() (R). 14 Extracting date components using DATE functions (SAS), Date & Time Wizard (SPSS), and lubridate:: functions (R). 15 Generating basic summaries using PROC CONTENTS (SAS), Variable View (SPSS), and str() or summary() (R).
	Module 2: Statistical Analysis and Reporting (using SAS or SPSS or R)
	1 Generating descriptive statistics using PROC MEANS (SAS), Descriptive Statistics (SPSS), and summary() or describe() (R). 2 Generating frequency tables using PROC FREQ (SAS), Frequencies (SPSS), and table() or count() (R). 3 Creating cross-tabulations and two-way tables using PROC FREQ (SAS), Crosstabs (SPSS), and table() (R). 4 Performing one-sample t-tests using PROC TTEST (SAS), T-Test (SPSS), and t.test() (R). 5 Performing independent two-sample t-tests using PROC TTEST (SAS), T-Test (SPSS), and t.test() with grouping (R). 6 Performing paired t-tests using PROC TTEST (SAS), Paired Samples T-Test (SPSS), and t.test(paired=TRUE) (R). 7 Performing one-way ANOVA using PROC ANOVA (SAS), One-Way ANOVA (SPSS), and aov() (R). 8 Performing two-way ANOVA using PROC GLM (SAS), Univariate Analysis (SPSS), and aov() (R). 9 Conducting Chi-square tests using PROC FREQ (SAS), Crosstabs with Chi-Square (SPSS), and chisq.test() (R).

	10 Creating graphical reports using PROC REPORT (SAS), Chart Builder (SPSS), and ggplot2 (R). 11 Generating histograms and box plots using PROC SGPLOT (SAS), Graphs (SPSS), and ggplot2 (R). 12 Generating correlation matrices using PROC CORR (SAS), Correlation (SPSS), and cor() (R). 13 Performing linear regression analysis using PROC REG (SAS), Regression (SPSS), and lm() (R). 14 Performing logistic regression using PROC LOGISTIC (SAS), Binary Logistic Regression (SPSS), and glm() (R). 15 Exporting results into external files (Excel, CSV, PDF) using ODS (SAS), Export Wizard (SPSS), and write.csv()/writexl (R).	
10	**Text Books:** 1. The Little SAS Book: A Primer – Lora D. Delwiche & Susan J. Slaughter 2. Learning SAS by Example: A Programmer's Guide – Ron Cody 3. Discovering Statistics Using IBM SPSS Statistics – Andy Field 4. IBM SPSS for Introductory Statistics: Use and Interpretation – George A. Morgan, Nancy L. Leech, 5. R for Data Science – Hadley Wickham & Garrett Grolemund 6. The Book of R: A First Course in Programming and Statistics – Tilman M. Davies	
11	**Reference Books:** 1. Applied Statistics and the SAS Programming Language – Ron P. Cody & Jeffrey K. Smith 2. Data Analysis with IBM SPSS Statistics – Kenneth Stehlik-Barry & Anthony J. Babinec 3. Hands-On Programming with R – Garrett Grolemund	
12	**Internal Continuous Assessment: 40%**	**External, Semester End Examination 60% Individual Passing in Internal and External Examination**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation.	

Title of Paper : Advance Python for Data Science

Sr. No.	Heading	Particulars
1	**Description of the course :** **Including but Not limited to :**	Advanced Python for Data Science is designed to equip learners with essential data manipulation, analysis, and visualization skills using Python's powerful libraries – Pandas, NumPy, and Seaborn. The course focuses on efficiently handling large datasets, performing complex data transformations, statistical analysis, and creating insightful visualizations. Learners will gain practical experience through hands-on labs, enabling them to apply these techniques to real-world data science problems.
2	**Vertical :**	Minor
3	**Type :**	Practical
4	**Credit**:	2 credits (1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted**:	50 Marks
7	**Course Objectives**: CO1: To introduce students to data manipulation and analysis using Pandas and NumPy, including data loading, transformation, and filtering techniques. CO2: To equip students with the ability to perform statistical computations, matrix operations, and time-series analysis using Pandas and NumPy. CO3: To develop proficiency in data visualization using Seaborn, enabling students to create meaningful visual representations of datasets. CO4: To train students in using Seaborn for advanced plotting techniques, including heatmaps, pairplots, and time-series visualizations for data-driven insights.	
8	**Course Outcomes**: OC1: Students will be able to manipulate and preprocess datasets using Pandas and NumPy, including handling missing values, filtering data, and performing statistical analysis. OC2: Students will perform matrix operations, generate random datasets, execute Boolean indexing, and apply time-series analysis using NumPy and Pandas. OC3: Students will create and customize various Seaborn plots, including histograms, boxplots, scatter plots, and violin plots for effective data visualization. OC4: Students will apply advanced Seaborn visualization techniques such as heatmaps, facet grids, and time-series plots to extract and present meaningful insights from data.	

9	**Modules**:-
	Module 1: Data Manipulation and Analysis using Pandas and NumPy
	1. Load a CSV dataset into a Pandas DataFrame. Perform basic data inspection such as displaying the first few rows, checking data types, handling missing values, and removing duplicate rows. 2. Using Pandas, load a dataset and perform column-wise operations such as adding a new calculated column, renaming columns, changing data types, and dropping unnecessary columns. 3. Using NumPy, generate a random dataset of 1000 values. Calculate basic statistical measures such as mean, median, variance, standard deviation, minimum, and maximum using NumPy functions. 4. Write a Pandas program to group a dataset by one or more categorical columns and calculate summary statistics such as count, mean, and standard deviation for each group. 5. Load a dataset into Pandas and filter rows based on complex conditions using .loc and .query(). For example, filter rows where sales are above a threshold and region equals "North". 6. Using NumPy, create two matrices (3x3) filled with random integers. Perform matrix addition, subtraction, multiplication, element-wise division, and calculate the determinant and inverse. 7. Write a Pandas program to merge two DataFrames using different types of joins (inner, outer, left, right). Use sample data representing customer details and order details. 8. Load a dataset and perform time series analysis using Pandas DateTime features. Extract year, month, day, and weekday from a date column, and group data by month to calculate monthly sales. 9. Using NumPy, generate a 1D array of 100 random integers between 1 and 1000. Use Boolean indexing to filter all values greater than 500 and less than 800, and calculate the mean of filtered values. 10. Write a Pandas program to pivot a DataFrame to create a pivot table summarizing the data. Use sample sales data to show product-wise sales for each region.
	Module 2: Data Visualization using Seaborn
	1. Load a dataset into Pandas and use Seaborn to plot a histogram showing the distribution of a numerical column. Customize the bin size, color, and add a title. 2. Using Seaborn, plot a boxplot for a numerical column grouped by a categorical column (e.g., salary distribution across different departments) from a given dataset. 3. Load a dataset into Pandas and create a pairplot using Seaborn to visualize pairwise relationships between all numerical columns. Add hue to distinguish different categories. 4. Create a Seaborn heatmap using a correlation matrix generated from a DataFrame. Customize the color palette, annotations, and title. 5. Using Seaborn, create a barplot comparing the average values of a numerical column for different categories of a categorical column. Customize axes labels and titles. 6. Load a dataset and create a Seaborn scatter plot between two numerical columns. Add hue to differentiate categories and customize markers and plot size.

	7. Using Seaborn, create a line plot to visualize trends in a time series dataset. Customize the plot with appropriate labels, grid lines, and title. 8. Load a dataset with multiple numerical columns and create a Seaborn violin plot to show the distribution of values for each column grouped by a categorical column. 9. Create a Seaborn count plot to visualize the frequency distribution of values in a categorical column from a dataset. Customize colors, orientation, and add value labels on top of bars. 10. Load a dataset and create a Seaborn facet grid of scatter plots, showing relationships between two numerical columns for different values of a third categorical column. .	
10	**Text Books:** 1. Python for Data Analysis by Wes McKinney 2nd Edition Publisher: O'Reilly Media	
11	**Reference Books:** 1. Python Data Science Handbook by Jake VanderPlas 1st Edition Publisher: O'Reilly Media 2. Hands-On Data Analysis with Pandas by Stefanie Molin 2nd Edition Publisher: Packt Publishing	
12	**Internal Continuous Assessment: 40%**	**External, Semester End Examination 60% Individual Passing in Internal and External Examination**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation.	

QUESTION PAPER PATTERN
(External and Internal)

Design Practical Examination Slips

Q1. Based on Module 1 -- 15 marks
Q2. Based on Module 2 -- 15 marks

Sd/-	**Sd/-**	**Sd/-**
Sign of the BOS Chairman **Dr. Srivaramangai R** **BOS in Data Science**	**Sign of the Offg. Associate Dean** **Dr. Madhav R. Rajwade** **Faculty of Science & Technology**	**Sign of the Offg. Dean** **Prof. Shivram S. Garje** **Faculty of Science & Technology**