Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: linear algebra, calculus, and statistics/probability

Rationale : The course will introduce the fundamental concepts of linear algebra, probability and statistics required for a program in data science.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Basics of Data Science**Introduction; Typology of problems; Importance of linear algebra, statistics and optimization from a data science perspective; Structured thinking for solving data science problems. Linear Algebra: Understanding vectors, matrices, and their operations, Learning about eigenvalues, eigenvectors, Linear Algebra's applications in machine learning. Applying linear algebra to represent data and algorithms.	22	10
2	**Vector Spaces**General properties of Vector Space, Vector Subspaces - Algebra of subspaces, Linear Span- Linear sum of two subspaces, Linear Dependence and Linear Independence of vectors, Basis of a Vector Space – Finite Dimensional Vector Space	16	7
3	**Probability and Statistics**Probability, Statistics and Random Processes: Probability theory and axioms; Random variables; Understanding probability distributions, density functions (univariate and multivariate) and statistical inference. Expectations and moments; Covariance and correlation; Statistics and sampling distributions Learning about hypothesis testing and confidence intervals, Applying statistical methods for data analysis and prediction.	22	10
4	**Calculus**Understanding derivatives and integrals, learning about optimization techniques, Applying calculus to model complex systems and optimize functions.	18	8
5	**Optimization**Unconstrained optimization; Necessary and sufficiency, conditions for optima; Gradient descent methods; Constrained optimization, KKT conditions; Introduction to non-gradient techniques; Introduction to least squares optimization; Understanding different optimization algorithms and their applications in machine learning, Data Representation: Learning how to represent data using vectors, matrices, and other mathematical structures. Logical Thinking and Problem-Solving: Developing the ability to apply mathematical concepts to solve real-world data science problems. **Introduction to Data Science Methods:** - Linear regression as an exemplar function approximation problem, Linear classification problems.	22	10
	Total	100	45

Course Outcome

After Learning the Course the students shall be able to:
1. Demonstrate understanding of basic mathematical concepts in data science, relating to linear algebra, probability, and calculus.
2. Employ methods related to these concepts in a variety of data science applications.
3. Apply logical thinking to problem-solving in context.
4. Demonstrate skills in writing mathematics.

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Understanding of mathematics (including calculus, linear algebra, and probability), programming concepts (especially Python or R)

Rationale : The aim of this course is to equip students with a strong understanding of statistical concepts and tools essential for data analysis, modeling, and making informed decisions in data-driven contexts.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
4	-	-	-	4	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Statistics and the Data Analysis Process**The Data Analysis Process, Planning and Conducting a Statistical Study, Observational Studies, Experimental Studies, Types of Data and Some Simple Graphical Displays, Frequency Distributions and Bar Charts for Categorical Data, Dotplots for Numerical Data, Pie charts, Displaying Numerical data: Stem and Leaf Displays, Time series plots	23	10
2	**Probability**Properties Of Probability, Combinatorial Principles, Conditional Probability, Independence of Events, Baye's Theorem, Bayes Theorem Applications, Random Variables: - Probability Distributions for Discrete Random Variables, Probability Distributions for Continuous Random Variables, **Joint Probability:** Joint Probability Function, Marginal Distribution Function, Conditional Probability Density Function, Variance **Moments:** Moments About Arbitrary Point, Moments Bout Origin, Moments About Mean, Moment Generating Functions, Law of Large Numbers, Monte Carlo Approximations, Monte Carlo Methods,	26	12
3	**Sampling**Sampling, Bias in Sampling, **Sampling Techniques**: Simple Random Sample, Stratified Random Sampling, Cluster Sampling, Systematic Sampling, Purposive Sampling **Analysis Of Variance:** Linear Model, One Way ANOVA Assumption, One Way ANOVA the Process, **Statistical Inference:** Estimation, Point Estimation, Criteria of a Good Estimator, Methods of Estimation, Testing of Hypothesis: Neyman Pearson Theory, Likelihood Ratio Test, Bayesian Versus Frequentist Inference, Frequentist Statistics, Bayesian Statistics, Applications of Bayesian Inference in Data Science	26	12
4	**Stochastic Process**Markov Chain, Transition Probabilities, Classification of States. **Data Analysis Techniques:** Gaining practical skills in exploratory data analysis (EDA), data visualization, and identifying patterns and relationships in data. **Statistical Modeling:** Learning to build and evaluate statistical models for prediction, classification, and regression.	25	11
	Total	100	45

Course Outcome

After Learning the Course the students shall be able to:
1. To understand fundamental concepts of Probability and Statistics. 2. To gain a solid understanding of descriptive and inferential statistics, probability theory, and statistical distributions. 3. To apply techniques like hypothesis testing, confidence intervals, regression analysis, and ANOVA 4. To understand the concepts that will help to learn more advanced topics of Data Science.

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Solid understanding of mathematics, programming, and statistics.

Rationale : The aim of this course is to equip students with a strong understanding of statistical concepts and tools essential for data analysis, modeling, and making informed decisions in data-driven contexts.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction**Introduction to Data Science, Data Science Components, Evolution of Data Science, Data Science Roles, Stages in a Data Science Project, Applications of Data Science in various fields, Data Security Issues.	16	7
2	**Data Collection and Data Pre-Processing**Data Collection Strategies, Data Pre-Processing Overview, Data Preprocessing Techniques, Need for preprocessing, Data Cleaning- Missing Value, Noisy Data, Removing noisy a Using Smoothing technique, Regression, Clustering, Inconsistent Data, Data Cleaning as a process, Discrepancy detection tools, Data Integration- Issues in Data Integration, Data Reduction, Its strategies, parametric model-Regression, Log linear Models, Non parametric model- Histograms, Clustering, Sampling, Dimensionality Reduction- Dimension Reduction Types- Wavelet Transforms, Principal Component Analysis, Data Transformation and Discretization- Data Transformation & its Tool, **Data Transformation by Normalization:** Data Discretization, Concept Hierarchy Generation for Nominal Data	22	10
3	**Exploratory Data Analytics**Types of Data, Dimensionality of Data Sets, Numerical Summaries of Data **Descriptive Statistics** - Mean, Location Median, Which Location Measure Is Best? Scale: Variance, Standard Deviation, Why Squared Deviations? Scale: Quartiles and IQR, Skewness and Kurtosis **Univariate Data:** Histograms and Bar Plots, Box Plots, Bivariate Data, multivariate Data, Pivot Table, Heat Map, Correlation Statistics, ANOVA.	22	10
4	**Model Development**Simple and Multiple Regression, Model Evaluation using Visualization, Residual Plot, Distribution Plot, Polynomial Regression and Pipelines, Measures for In-sample Evaluation, Prediction and Decision Making.	18	8
5	**Model Evaluation**Generalization Error, Out-of-Sample Evaluation Metrics, Cross Validation, Overfitting and Under Fitting, Model Selection, Prediction by using Ridge Regression, Testing Multiple Parameters by using Grid Search	22	10
	Total	100	45

Course Outcome

After Learning the Course the students shall be able to:
1. Understand the basics of data science principles and methods. 2. Explore the different steps in data science, understanding how they are interconnected. 3. Learn ways to collect and analyze data, adapting methods to different situations.

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Understanding of Python Programming Fundamentals, Discrete Mathematics Concepts, Mathematical Induction and Problem-Solving and Logical Thinking

Rationale : The aim of this course is to equip students with a strong understanding of statistical concepts and tools essential for data analysis, modeling, and making informed decisions in data-driven contexts.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Foundational Concepts**Introduction to Programming and Algorithms: What are algorithms and data structures? Importance of efficient algorithms and data structures Basics of programming in Python, Variables, Data Types, Control Flow (Conditionals, Loops), and Functions. Oops Concepts- Class, Object, Constructors, Types of Variables, Types of Methods. Inheritance: Single, Multiple, Multi-level, Hierarchical, Hybrid, Polymorphism: with functions and objects, with class methods, with inheritance, Abstraction: Abstract classes. **Algorithmic Analysis:** Time and Space Complexity: Time and space complexity (Big O notation), Asymptotic analysis, Measuring algorithm efficiency Asymptotic Analysis: Learning about different types of complexities (e.g., linear, logarithmic, quadratic)	26	12
2	**Basic Data Structures- I**Arrays/Lists: Understanding how to store and manipulate data using Python lists, Basic Operations (insertion, deletion, searching), Array vs Linked List **Linked Lists**: Singly and doubly linked lists, Implementation in Python, Operations (insertion, deletion, searching) **Stacks and Queues**: Abstract data types (LIFO and FIFO), Implementation in Python using lists Applications (e.g., undo/redo functionality, task scheduling).	23	10
3	**Basic Data Structures- IIHash Tables (Dictionaries)**: Hashing functions and collision resolution Implementation in Python, Applications (e.g., symbol tables, caching) **Trees**: Binary trees, binary search trees, Tree traversals (inorder, preorder, postorder), Implementation in Python **Graphs**: Representations (adjacency matrix, adjacency list), Graph traversals (BFS, DFS), Applications (e.g., social networks, routing)	26	12
4	**Searching Algorithms:Searching Algorithms**: Linear search, binary search, Implementation in Python **Sorting Algorithms**: Bubble sort, insertion sort, selection sort, Merge sort, quicksort, Implementation in Python, Analysis of time and space complexity **Dynamic Programming**: Introduction to dynamic programming, Memorization, Examples (e.g., Fibonacci sequence, shortest path)	26	12
5	**Model Evaluation**Generalization Error, Out-of-Sample Evaluation Metrics, Cross Validation, Overfitting and Under Fitting, Model Selection, Prediction by using Ridge Regression, Testing Multiple Parameters by using Grid Search	25	11
	Total	126	57

Course Outcome

After Learning the Course the students shall be able to:
1. Develop Python program using basic syntactical constructs. 2. Perform operations on sequence structures in Python. 3. Implement Modules, Packages in Python for given problem. 4. Design classes for given problem. 5. Implement Linear Data Structure in Python. 6. Develop Python program to implement tree data structure.

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Solid foundation in programming fundamentals, Object-Oriented Programming (OOP), basic mathematical concepts and Problem-Solving Skills

Rationale : The course should enable the students to Understand basic data structures in python, Design and analyze simple linear data structures., Identify and apply the suitable data structure for the given real-world problem and Gain knowledge in practical applications of data structures

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Understand various data representation techniques in the real world. 2. Implement linear and non-linear data structures. 3. Design, implement, and analyze linear data structures, such as lists, queues, and stacks, according to the needs of different applications 4. Design, implement, and analyze efficient tree structures to meet requirements such as searching, indexing, and sorting 5. Analyze various algorithms based on their time and space complexity.

List of Practical

1.	**Write a Python program for class, Flower, that has three instance variables of type str, int, and float that respectively represent the name of the flower, its number of petals, and its price. Your class must include a constructor method that initializes each variable to an appropriate value, and your class should include methods for setting the value of each type, and retrieving the value of each type.**
2.	**Develop an inheritance hierarchy based upon a Polygon class that has abstract methods area( ) and perimeter( ). Implement classes Triangle, Quadrilateral, Pentagon, that extend this base class, with the obvious meanings for the area( ) and perimeter( ) methods. Write a simple program that allows users to create polygons of the various types and input their geometric dimensions, and the program then outputs their area and perimeter.**
3.	**Write a python program to implement Method Overloading and Method Overriding.**
4.	**Write a Python program to illustrate the following comprehensions: a) List Comprehensions b) Dictionary Comprehensions c) Set Comprehensions d) Generator Comprehensions**
5.	**Write a Python program to generate the combinations of n distinct objects taken from the elements of a given list. Example: Original list: [1, 2, 3, 4, 5, 6, 7, 8, 9] Combinations of 2 distinct objects: [1, 2] [1, 3] [1, 4] [1, 5] .... [7, 8] [7, 9] [8, 9].**
6.	**Write a program for Linear Search and Binary search.**
7.	**Write a program to implement Bubble Sort and Selection Sort.**
8.	**Write a program to implement Merge sort and Quick sort.**
9.	**Write a program to implement Stacks and Queues.**
10.	**Write a program to implement Singly Linked List.**
11.	**Write a program to implement Doubly Linked list.**
12.	**Write a program to implement Binary Search Tree.**

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Solid foundation in data analysis, data visualization, database management, and business acumen

Rationale : Upon completion of this course, students will be able to Explain the Business Intelligence, analytics and Decision Support system, list the technologies for Decision making, automated decision systems, explain sentiment analysis techniques, illustrate multi-criteria Decision-making systems, predictive modelling techniques

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Business Intelligence an Introduction**Introduction - Definition - History and Evolution -Business Intelligence Segments – Difference between Information and Intelligence - Defining Business Intelligence Value Chain - Factors of Business Intelligence System - Real time Business Intelligence Business Intelligence Applications: Marketing models: Relational marketing, Sales force management, Logistic and production models: Supply chain optimization, Optimization models for logistics planning, Revenue management systems. Data envelopment analysis: Efficiency measures, Efficient frontier, The CCR model, Identification of good operating practices	20	9
2	**Essentials of Business Intelligence**Introduction - Creating Business Intelligence Environment - Business Intelligence Landscape - Types of Business Intelligence -Business Intelligence Platform -Applications in Business Analytics -Dynamic roles in Business Intelligence - Challenges - Business Intelligence Tools - Modern Business Intelligence - Enterprise Business Intelligence - Information Workers.	15	7
3	**Decision Making**Introduction and Definitions, Phases of the Decision, Making Process, The Intelligence Phase, Design Phase, Choice Phase, Implementation Phase, Decision Support Systems Capabilities, Decision Support Systems Classification, Decision Support Systems Component	15	7
4	**Model Based Decision Making**Decision Support Systems modeling, Structure of mathematical models for decision support, Certainty, Uncertainty, and Risk, Decision modeling with spreadsheets, Mathematical programming optimization, Decision Analysis with Decision Tables and Decision Trees, Multi-Criteria Decision Making With Pairwise Comparison	16	7
5	**Business Intelligence User Model**Introduction - Business Intelligence Opportunity Analysis Overview - Content Management System - End User Segmentation - Basic Reporting and Querying - Online Analytical Processing - OLAP Techniques - OLAP Applications - Applying OLAP to Data Warehousing - Benefits of using OLAP – Dashboard -Key Performance Indicators -Advanced/Emerging BI Technologies - Future of Business Intelligence- Critical Challenges for Business Intelligence success.	18	8
6	**AI & EXPERT Systems**Concepts and Definitions of Artificial Intelligence, Artificial Intelligence Versus Natural Intelligence, Basic Concepts of Expert Systems, Applications of Expert Systems, Structure of Expert Systems, Knowledge Engineering, Development of Expert Systems	16	7
	Total	100	45

Reference Books	
1.	**Business Intelligence and Analytics: System for Decision Support** By Ramesh Sharda, Dursun Delen, Efraim Turban, J.E. Aronson, Ting-Peng Liang, David King | 10th Edition, Pearson Global Edition, Pub. Year 2013
2.	**Business Intelligence: A Managerial Approach** By Efraim Turban, Ramesh Sharda, Jay E. Aronson, David King | 9th Prentice Hall, Pub. Year 2013
3.	**Decision support and Business Intelligence Systems** By Efraim Turban, Ramesh Sharda, Dursun Delen,Pearson Ninth, Pub. Year 2011

Course Outcome
After Learning the Course the students shall be able to:
1: Understand the basic concepts of business intelligence and its application. 2: Elucidate the role of business intelligence and its value chain analysis. 3: Examine the various models of business intelligence in organizational success. 4: Understand how best to apply BI&A methods and techniques in addressing strategic business problems 5: Understand the role of BI&A in helping organizations make better decisions

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: A good grasp of data structures, discrete mathematics, and problem-solving skills.

Rationale : Upon completion of this course, students will be able to analyze the asymptotic performance of algorithms, write rigorous correctness proofs for algorithms, demonstrate a familiarity with major algorithms and data structures, apply important algorithmic design paradigms and methods of analysis and synthesize efficient algorithms in common engineering design situations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Fundamentals of Algorithms and mathematics**Problem, algorithm definitions, Mathematics for algorithmic sets, Functions and relations, Combinations, Vectors and matrices, Linear inequalities and linear equations	18	8
2	**Analysis of Algorithms**Orders of Magnitude (Asymptotic notations), Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential), Average and worst-case analysis, Analyzing control statements, Recurrence Relations- substitution, change of variables, master's method	22	10
3	**Sorting and searching algorithms:**Selection sort, bubble sort, insertion sort, Sorting in linear time, count sort, Linear search Divide and conquer algorithms: Introduction, Quick sort, worst and average case complexity, Merge sort, Matrix multiplication, Binary search, Binary search tree	18	8
4	**Greedy Algorithms:**General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm- Activity selection problem, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem Dynamic programming: Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- Making Change Problem, Assembly Line Scheduling, Knapsack problem, Matrix chain multiplication, Longest Common Subsequence, Dynamic Programming using Memorization.	24	11
5	**String matching:**Introduction, The naïve string-matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata Introduction to Complexity Theory: The class P and NP, Polynomial reduction, NP- Complete Problems, NP-Hard Problems	18	8
	Total	100	45

Reference Books

1.	**Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.**
2.	**Fundamental of Algorithms by Gills Brassard** By Brassard, Paul Bratley, PHI.
3.	**Design and Analysis of Computer Algorithms** By Aho, Hopcroft and Ullman, Pearson
4.	**The Algorithm Design Manual** By Steve s. Skiena

Course Outcome

After Learning the Course the students shall be able to:
1. Check the correctness of algorithms using inductive proofs and loop in variants. 2. Compare functions using asymptotic analysis and describe the relative merits of worst-, average-, and best-case analysis. 3. Solve recurrences using the master, the iteration, and the substitution method. 4. Become familiar with a variety of sorting algorithms and their performance characteristics (eg, running time, stability, space usage) and be able to choose the best one under a variety of requirements. 5. Understand and identify the performance characteristics of fundamental algorithms and data structures and be able to trace their operations for problems such as sorting

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Understanding of several key areas like basic programming, database concepts (especially SQL), Data manipulation techniques and Fundamentals of ETL

Rationale : Upon completion of this course, students will be able to: Know the critical skills that are necessary in the handling the challenges in a variety of modern-day data-driven businesses, Acquire the knowledge in data analysis and visualization that helps in deeper quantitative reasoning needed for data and predictive analytics work, Learn the skills to turn raw datasets into interactive and dynamic and learn concepts of Web scraping.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1: Understand the Data Wrangling and ETL Concept
2: Understand the Data Cleaning Techniques.
3: Understand the Data Exploration and Data Analysis.
4: Understand the knowledge of Web Scraping.

List of Practical

1.	**Unit-I:** Introduction to Data Wrangling & ETL: What is Data Wrangling? Defining data wrangling, its importance, and its role in the data analysis process. What is ETL? Understanding the ETL process, its steps (Extract, Transform, Load), and its relationship with data warehousing. Data Wrangling vs. ETL: Exploring the similarities and differences between data wrangling and ETL. Importance of Data Quality: Understanding the impact of data quality on analysis and the need for data cleaning. Data Profiling: Exploring data to understand its structure, content, and potential issues. Data Governance: Understanding the importance of data quality, security, and compliance in data management.
2.	**LAB 1: Data Wrangling Techniques & Tools:** Data Gathering: o Sources: Working with different data sources (e.g., CSV files, databases, APIs). o Web Scraping: Extracting data from websites. o Flat Files: Understanding and working with various flat file formats (e.g., CSV, TXT).
3.	**LAB 2: Data Cleaning:** o Handling Missing Data: Techniques for identifying and managing missing values. o Removing Duplicates: Identifying and removing duplicate records. o Data Validation: Applying rules and checks to ensure data accuracy and consistency. o Outlier Detection and Treatment: Identifying and handling outliers in the data.
4.	**LAB 3: Data Transformation:** o Data Type Conversion: Converting data from one type to another (e.g., string to integer). o Aggregation and Summarization: Creating summary statistics and aggregates. o Reshaping Data: Transforming data from wide to long format or vice-versa. o Feature Engineering: Creating new features from existing ones to improve model performance.
5.	**LAB 4: Data Enrichment:** o Combining Datasets: Merging data from multiple sources. o Adding External Data: Incorporating external data sources to enhance the dataset.
6.	**LAB 5: Demonstration of Following Tools** • Python Libraries: Pandas, NumPy, etc. • SQL: Working with databases using SQL for data manipulation. • Spreadsheet Software: Excel, Google Sheets.

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Some foundational knowledge in data analytics, Data Visualization Concepts and Mathematical Formulas and Functions

Rationale : Upon completion of this course, students will be able to:students will be able to:Import, transform and cleanse data using Power Query Editor, build a data model for self-service reporting and manipulate the model with DAX, publish and share visualizations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1: Do the Data Handling and Preparation. 2: Prepare the Report and Dashboard Creation. 3: Do the Data Analysis and Interpretation. 4: Do the Sharing and Collaboration of Reports and Dashboard

List of Practical

1.	**Using MSEXCEL, create a dashboard to display the implementation of the Pivot Table, Slicers and Charts to show the change in trends of Sales Data Analysis with the change over time.**
2.	**Using MSEXCEL, create a Sheet to display the implementation of the Goal Seek, Scenario, and VLOOKUP to show the data as per the condition and User's Request.**
3.	**Using POWER BI, create a dashboard to display the implementation of the followings: Cards and Multi Row Cards, Building and Formatting Charts, Line Charts, KPI Cards, Basic Filtering Options**
4.	**Using POWER BI create a dashboard to display the implementation of the followings: Use of Donus and Filters in Power BI, Table & Matrix Visuals, Conditional Formatting, Top N Filters, Map Visuals, Slicers, Gauge Charts, Use of QnA Feature, Use of Gauge Charts, Waterfall Chart, Scatter Chart.**
5.	**Using POWER BI, Implement the Functions in DAX and show the Cumulative Sales and Moving Average.**
6.	**Students will choose a dataset and create a comprehensive data visualization project. The project should demonstrate their understanding of data visualization techniques and data storytelling**
7.	**Using Power BI for Creating and Formatting Visualizations**
8.	**Using Slicers and Filters to interact with your reports.**
9.	**Using Power BI to implement Conditional Formatting to highlight specific data points.**

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Solid foundation in front-end technologies like HTML, CSS, and JavaScript, including frameworks like React or Angular

Rationale : Upon completion of this course, students will be able to:become knowledgeable about the most recent web development technologies, build Idea for creating two tier and three tier architectural web applications, Design and analyze real time web applications, Constructing suitable client and server-side applications and to learn core concept of both front end and back-end programming.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Reference Books

1.	**Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.**
2.	**Jon Duckett, "JavaScript & jQuery: Interactive Front-End Web Development",**
3.	**Mastering Full Stack React Web Development Paperback** By TomaszDyl , Kamil Przeorski , Maciej Czarnecki., Pub. Year 2017
4.	**Full-Stack JavaScript Development by Eric Bush**

Course Outcome

After Learning the Course the students shall be able to:
1: Develop a fully functioning website and deploy on a web server. 2: Gain Knowledge about the front end and back-end Tools 3: Find and use code packages based on their documentation to produce working results in a project. 4: Create web pages that function using external data. Implementation of web application employing efficient database access.

List of Practical

1.	**Write a script that Logs "Hello, World!" to the console. Create a script that calculates the sum of two numbers and displays the result in an alert box. b. Create an array of 5 cities and perform the following operations: Log the total number of cities. Add a new city at the end. Remove the first city. Find and log the index of a specific city.**
2.	**Read a string from the user, Find its length. Extract the word "JavaScript" using substring() or slice(). Replace one word with another word and log the new string. Write a function is Palindrome(str) that checks if a given string is a palindrome (reads the same backward).**
3.	**Create an object student with properties: name (string), grade (number), subjects (array), displayInfo() (method to log the student's details) Write a script to dynamically add a passed property to the student object, with a value of true or false based on their grade. Create a loop to log all keys and values of the student object.**
4.	**Create a button in your HTML with the text "Click Me". Add an event listener to log "Button clicked!" to the console when the button is clicked. Select an image and add a mouseover event listener to change its border color. Add an event listener to the document that logs the key pressed by the user.**
5.	**Build a React application to track issues. Display a list of issues (use static data). Each issue should have a title, description, and status (e.g., Open/Closed). Render the list using a functional component.**
6.	**Create a component Counter with A state variable count initialized to 0. Create Buttons to increment and decrement the count. Simulate fetching initial data for the Counter component using useEffect (functional component) or componentDidMount (class component). Extend the Counter component to Double the count value when a button is clicked. Reset the count to 0 using another button.**
7.	**Install Express (npm install express). Set up a basic server that responds with "Hello, Express!" at the root endpoint (GET /). Create a REST API. Implement endpoints for a Product resource: GET : Returns a list of products. POST : Adds a new product. GET /:id: Returns details of a specific product. PUT /:id: Updates an existing product. DELETE /:id: Deletes a product. Add middleware to log requests to the console. Use express.json() to parse incoming JSON payloads.**
8.	**Install the MongoDB driver for Node.js.** Create a Node.js script to connect to the shop database. Implement insert, find, update, and delete operations using the Node.js MongoDB driver. Define a product schema using Mongoose. Insert data into the products collection using Mongoose. Create an Express API with a /products endpoint to fetch all products.Use fetch in React to call the /products endpoint and display the list of products. Add a POST /products endpoint in Express to insert a new product. Update the Product List, After adding a product, update the list of products displayed in Reactcts displayed in React

Miscellaneous

Useful Links

1.https://www.guvi.in/blog/best-ways-to-learn-full-stack-development/
2.https://www.geeksforgeeks.org/blogs/what-is-full-stack-development/
3.https://github.com/vasansr/pro-mern-stack
4.https://nptel.ac.in/courses/106106156

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: Basic knowledge of mathematics, statistics fundamentals, and introductory programming concepts.

Rationale : This course builds statistical and analytical skills required for data-driven decision making by combining statistical theory with practical data analysis tools and techniques.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Foundational Statistics** Descriptive Statistics: Summarizing data using measures like mean, median, mode, and standard deviation. Inferential Statistics: Making inferences about populations based on sample data, including hypothesis testing and confidence intervals. Probability and Distributions: Understanding probability, different probability distributions (e.g., normal, binomial), and their applications. Sampling Methods: Learning about different sampling techniques and their implications for data analysis.	**23**	**11**
2	**Data Analysis Techniques** Regression Analysis: Exploring the relationships between variables using linear and non-linear regression models. Time Series Analysis: Analyzing data that changes over time, including forecasting techniques. Multivariate Analysis: Analyzing multiple variables simultaneously, including techniques like Principal Component Analysis (PCA) and Factor Analysis. Exploratory Data Analysis (EDA): Using various methods, including visualization, to understand and summarize data.	**26**	**13**
3	**Statistical Modelling** Model Building: Learning about different types of statistical models, including linear models, logistic regression, and more advanced models. Model Evaluation: Assessing the performance of models using various metrics, including R-squared, AIC, and BIC. Model Selection: Choosing the best model for a given dataset and problem.	**26**	**13**
4	**Software and Tools** R: Basic of R Programming language and its usage in statistical computing. Python: Basic of Python, Libraries like Pandas and Scikit-learn for data analysis and machine learning. SQL: Basic of SQL, Use of SQL to manage and manipulate databases. Data Visualization Tools: Introduction to Tableau, Power BI, and SPSS	**25**	**11**
	Total	**100**	**48**

Course Outcome

After Learning the Course the students shall be able to:

1. Apply descriptive, inferential statistics, probability distributions, and sampling techniques to analyze data.
2. Use regression, time series, multivariate analysis, and exploratory data analysis techniques to extract insights from data.
3. Build, evaluate, and select appropriate statistical models for real-world data problems.
4. Perform data analysis and visualization using tools such as R, Python, SQL, and industry-standard visualization software.

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: Basic knowledge of statistics, databases, linear algebra, probability, and introductory programming concepts.

Rationale : This course introduces data mining and machine learning techniques to extract meaningful patterns from large datasets and build intelligent models for real-world decision-making and predictive analytics.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to Data Mining** What is data mining? Related technologies - Machine Learning, DBMS, OLAP, Statistics, Data Mining Goals, Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods, Applications, Example: weather data Data mining relationship with data warehousing, why data mining is a discipline? Overview of data mining tasks: Clustering, Classifications, Rules learning etc. **Data mining knowledge representation:** Task relevant data, Background knowledge, Interestingness measures, Representing input data and output knowledge, Visualization techniques, Experiments with Weka – visualization	**20**	**10**
2	**Data mining algorithms** Association rules, Motivation and terminology, Example: mining weather data, **Basic idea:** item sets, Generating item sets and rules efficiently Correlation analysis, Experiments with Weka - mining association rules **Data mining algorithms:** Classification, Basic learning/mining tasks, **Inferring rudimentary rules:** 1R algorithm, Decision trees Covering rules, Experiments with Weka - decision trees, rules **Data mining algorithms:** Prediction, The prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance- based methods (nearest neighbor), Linear models, Experiments with Weka - Prediction	**20**	**10**
3	**Foundational Concepts** Introduction to machine learning, different types of learning (supervised, unsupervised, reinforcement), and the role of mathematics (linear algebra, probability, statistics), Role of machine learning in computer science and problem solving **Supervised Learning:** Different algorithms- Linear regression, logistic regression, decision trees, and ensemble methods (bagging, boosting). **Unsupervised Learning:** Different techniques- clustering (K-means, hierarchical), and dimensionality reduction (PCA).Notion of Generalization and concern of Overfitting	**20**	**9**
4	**Artificial Neural Network** Introduces neural networks, Single Layer Neural Networks, Multi-layer perceptron, biological neuron, Artificial neuron, Activation functions, Architectures of neural network, Perceptron, Learning process in ANN, Role of Loss Functions and Optimization, Gradient Descent and Perceptron/Delta Learning, Back propagation. **Advance topics in Machine Learning (Deep Learning):** Convolutional neural networks (CNNs), Recurrent neural networks (RNNs), Advances in Backpropagation and Optimization for Neural Networks Adversarial Learning, Recurrent Neural Networks, Transformer **Natural Language Processing (NLP):** Focuses on text analytics, sentiment analysis, and word embeddings.	**20**	**10**

	Reinforcement Learning: Explores concepts like Markov Decision Processes and Q-learning. **Data Handling and Preprocessing:** Covers data cleaning, feature engineering, and data transformation techniques. Notion of Training, Validation and Testing; Connect to generalization and overfitting.		
5	**Model Preparation, Evaluation and feature engineering**Types of data in machine learning, Exploring structure of data, Data pre-processing, Model selection and training (for supervised learning), Model representation and interpretability, **Model Evaluation and Optimization:** Includes methods for evaluating model performance, cross-validation, and hyperparameter tuning. Evaluating machine learning algorithms and performance enhancement of models	**20**	**9**
	Total	**100**	**48**

Course Outcome

After Learning the Course the students shall be able to:

1. Explain data mining concepts, processes, tasks, and knowledge representation techniques using tools such as Weka.
2. Apply data mining algorithms including association rules, classification, and prediction techniques to real datasets.
3. Analyze machine learning methods such as supervised, unsupervised, neural networks, deep learning, NLP, and reinforcement learning.
4. Prepare, evaluate, and optimize machine learning models using preprocessing, feature engineering, validation, and performance evaluation techniques.

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: To learn this subject, learners must have a solid foundation in computer science and programming. Here are some prerequisites to consider: Basic Computer Skills: Understanding the basic functions of a computer and its operating system is crucial. Databases Knowledge: Familiarity with the structure of databases and the use of database management systems is beneficial. SQL: Knowledge of SQL is a prerequisite for learning DBMS. SQL is a standard language used to communicate with relational data

Rationale : 1. Demonstrate the basic elements of a relational database management system, 2. Ability to identify the data models for relevant problems 3. Ability to design entity relationship and convert entity relationship diagrams into RDBMS and 4. formulate SQL queries on the respect data 5. Apply normalization for the development of application software's.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to Databases and DBMS**What is a database and why use it? Different types of DBMS (e.g., relational, NoSQL). DBMS architecture and components, Data abstraction and independence, Database users and administrators. Data Modeling: Entity-Relationship (ER) Modeling: Entities, attributes, relationships, and constraints, Converting ER diagrams into relational database schemas, Data modeling for different types of applications.	**26**	**12**
2	**Relational Database Design**Relational model concepts (e.g., relations, tables, keys), Relational constraints (e.g., primary key, foreign key), Normalization: 1NF, 2NF, 3NF, BCNF, Functional dependencies and decomposition. SQL (Structured Query Language): DDL (Data Definition Language) for creating and managing database objects, DML (Data Manipulation Language) for querying, inserting, updating, and deleting data, SQL queries (e.g., SELECT, INSERT, UPDATE, DELETE), SQL for join operations, subqueries, and aggregate functions, Aggregate function, Null Values, nested sub queries, Joined relations. Triggers. Views: Introduction to views, data independence, security, updates on views, comparison between tables and views	**25**	**11**
3	**Relational Algebra and Calculus**Relational algebra operators (e.g., selection, projection, join), Tuple and domain relational calculus, Expressiveness of algebra and calculus. Transactions and Concurrency Control: Transaction properties (ACID), Concurrency control techniques (e.g., locking), Serializability and recovery.	**24**	**11**
4	**Query Processing and Optimization:**Query parsing and translation, Query optimization techniques, and indexing and hashing for efficient data retrieval. Database Storage and File Structures: Storage organizations (e.g., heap files, B-trees), File structures for data storage. Advanced Topics: NoSQL databases and their characteristics, Database security and access control, and Database administration and maintenance.	**25**	**11**
	Total	**100**	**45**

Reference Books	
1.	**“Database System and Concepts”,** By A Silberschatz, H Korth, S Sudarshan, | fifth Edition McGraw-Hill,
2.	**“Database Systems”,** By Rob, Coronel, | Seventh Edition,
3.	**Data base Management Systems,** By Raghurama Krishnan,
4.	**Fundamentals of Database Systems,**
5.	**An Introduction to Database systems,** By C.J. Date, A.Kannan, S.Swami Nadhan, | Eight Edition for UNIT III.
6.	**Neural Networks, Fuzzy Systems and Evolutionary Algorithms: Synthesis and Applications,** By Rajasekaran,G. A. Vijayalakshmi Pai, PHI.

Course Outcome
After Learning the Course the students shall be able to:
1. Demonstrate the basic elements of a relational database management system 2. Ability to identify the data models for relevant problems 3. Ability to design entity relationships and convert entity relationship diagrams into RDBMS and 4. Formulate SQL queries on the respective data, and apply normalization for the development of application software.

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: Solid foundation in programming fundamentals, Object-Oriented Programming (OOP), basic mathematical concepts and Problem-Solving Skills

Rationale : The aim of this course is to 1. To explain basic database concepts, applications, data models, schemas and instances. 2. To demonstrate the use of constraints and relational algebra operations. 3. Describe the basics of SQL and construct queries using SQL. 3. To emphasize the importance of normalization in databases. 4. To facilitate students in Database design 5. To familiarize issues of concurrency control and transaction management.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Reference Books

1.	**SQL, PL/SQL – The Programming Language of Oracle Paperback**

Course Outcome

After Learning the Course the students shall be able to:

1. Use SQL Query to Implement the Data Definition Language.
2. Use SQL Query to Implement the Data Manipulation Language.
3. Use SQL Query to Implement the Data Control Language.
4. Use PLSQL to implement. Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system.

List of Practical	
1.	Design a Database and create required tables. For e.g. Bank, College Database
2.	Apply the constraints like Primary Key, Foreign key, NOT NULL to the tables.
3.	Write a SQL statement for implementing ALTER, UPDATE and DELETE
4.	Write the queries to implement the joins
5.	Write the query for implementing the following functions: MAX(), MIN(), AVG(), COUNT(), DISTINCT
6.	Write the query to implement the concept of Integrity constrains
7.	Write the query to create the views
8.	Perform the queries for triggers
9.	Perform the following operation for demonstrating the insertion, updation and deletion using the referential integrity constraints
10.	Write the query for creating the users and their role.
11.	Programs On Pl/Sql- Synatx & Structure to Write PLSQL Program
12.	PLSQL Program to Implement the Procedures And Functions
13.	PLSQL Program to Implement Triggers
14.	PLSQL to Implement the Cursors
15.	CASE STUDY: BOOK PUBLISHING COMPANY
16.	CASE STUDY GENERAL HOSPITAL
17.	CASE STUDY: Student Progress & Monitoring System

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: Basic understanding of programming concepts, mathematics, and introductory statistics.

Rationale : This course equips learners with practical skills in Python and R for data manipulation, visualization, statistical analysis, and machine learning, supporting data-driven decision making and analytics applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Python-I Fundamentals**Installation, syntax, variables, data types, operators, control structures (if, while, for), functions, modules, packages, file handling, error handling. Data Structures: Lists, tuples, dictionaries, sets. Data Manipulation: Libraries like Pandas for data manipulation and analysis, NumPy for numerical computing	26	13
2	**Python-II Data Visualization:**Libraries like Matplotlib, Seaborn for creating charts and graphs. Data Analysis and Machine Learning: Basic statistical concepts, linear regression, classification, clustering, and advanced techniques using libraries like scikit-learn. Data Import and Export: Reading and writing data from various formats (CSV, Excel, databases).	22	11
3	**R-I Fundamentals**Installation, syntax, data types, operators, control structures, functions, packages. Data Structures: Vectors, lists, matrices, arrays, data frames, factors. Data Manipulation: Tidy verse packages for data wrangling and cleaning, dplyr for data manipulation.	26	12
4	**R- II Data Visualization**ggplot2 for creating publication-quality graphics. Data Analysis and Statistics: Descriptive statistics, hypothesis testing, regression, ANOVA, and other statistical methods. Machine Learning: Using packages like caret for building machine learning models.	26	12
	Total	100	48

Reference Books

1.	**“The Book of R”** By Tilman M. Davies,
2.	**“The Art of R programming”** By Norman Matloff,
3.	**“Automate the Boring Stuff with Python”,** By William Pollock,
4.	**"Think Python: How to Think Like a Computer Scientist”,** By Allen B. Downey, | 2nd Edition,

Course Outcome

After Learning the Course the students shall be able to:

1. Develop programs using Python and R fundamentals including control structures, functions, data structures, and file handling.
2. Manipulate and analyze datasets using Python (NumPy, Pandas) and R (tidyverse, dplyr) libraries.
3. Create effective data visualizations using Matplotlib, Seaborn, and ggplot2 for exploratory and presentation purposes.
4. Apply statistical analysis and machine learning techniques using Python and R libraries to solve real-world data problems.

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: The primary prerequisites for learning Python and R programming are basic computer literacy, a sound logical mind, and strong determination to practice problem-solving. No prior programming experience is strictly necessary for either language, as they are both beginner-friendly.

Rationale : The aim of this course is to 1. Develop proficiency in writing and executing code through coding exercises and projects. 2. Incorporate statistical methods and techniques into Python and R programs. 3. Create visualizations using R's built-in functions and Python libraries like Matplotlib. 4. Explore and utilize relevant packages and libraries in both languages, such as Pandas and NumPy for Python and GGplot2 for R.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Reference Books

1.	**Think Python How to Think Like a Computer Scientist,** | 2nd Edition,
2.	**Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language,**
3.	**Core PYTHON Applications Programming,**
4.	**"The Book of R"** By Tilman M. Davies,
5.	**"The Art of R programming"** By Norman Matloff,

Course Outcome

After Learning the Course the students shall be able to:

1. Understand the basic concepts in Python Programming. Develop programming logic using R – Packages.
2. Understand and demonstrate the use of advanced data types such as tuples, dictionaries and lists, Tuples and Sets
3. Use and apply the different Python Libraries for GUI Interface, Data Analysis and Data Visualization
4. Setup R Programming Environment. Understand and use R – Data types. Understand and use R – Data Structures.

List of Practical	
1.	Installation of Python environment (e.g., Anaconda), basic syntax, variables, data types, and operators.
2.	Implementing conditional statements (if-else, elif) and loops (for, while) to solve problems like finding GCD, Fibonacci series, or prime numbers.
3.	Creating and manipulating lists, tuples, sets, and dictionaries (add, delete, slice, index elements).
4.	Defining and calling user-defined functions, using arguments and return values, and importing standard modules.
5.	Performing string manipulations (slicing, concatenation) and reading from/writing to text and CSV files.
6.	Implementing classes and objects for real-world scenarios (e.g., student information system, employee payroll).
7.	Data Analysis Libraries- Using NumPy for array manipulation and Pandas for data loading, cleaning, preparation, and manipulation.
8.	Data Visualization- Creating basic plots and graphs using libraries like Matplotlib.
9.	Setting up R and RStudio, understanding R sessions, basic syntax, variables, and data types
10.	Implementing control statements (if, else, switch) and user-defined functions (mean, sum, min, max, etc.).
11.	Working with vectors, lists, matrices, arrays, factors, and data frames (accessing elements, adding/deleting elements, merging).
12.	Loading, handling, and exploring data from various file formats (CSV, Excel).
13.	Using packages like dplyr for data transformation and tidyr for tidying data.
14.	Calculating descriptive statistics such as mean, median, mode, standard deviation, and correlation.
15.	Calculating descriptive statistics such as mean, median, mode, standard deviation, and correlation.
16.	Write a program to read a csv file and analyze the data in the file in R
17.	Create pie charts and bar charts using R

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: Basic statistics and Math, Python programming & R Programming

Rationale : The objective of this course is to make the students: 1. To Introduce the concept of Data Analytics Lifecycle. 2. To Develop Mathematical concepts required for advance regression. 3. To Understand data modeling in time series and its process. 4. To create awareness about Text analytics and its applications. 5. To provide overview of Data analytics and visualization with R. 6. To provide overview of Data analytics and visualization with Python.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	-	-	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Data Analytics Lifecycle overview**Key Roles for a Successful Analytics, Background and Overview of Data Analytics Lifecycle Project Phase 1: Discovery: Learning the Business Domain, Resources Framing the Problem, Identifying Key Stakeholders. Interviewing the Analytics Sponsor, Developing Initial Hypotheses Identifying Potential Data Sources Phase 2: Data Preparation: Preparing the Analytic Sandbox, Performing ETLT, Learning About the Data, Data Conditioning, Survey and visualize, Common Tools for the Data Preparation Phase Phase 3: Model Planning: Data Exploration and Variable Selection, Model Selection, Common Tools for the Model Planning Phase Phase 4: Model Building: Common Tools for the Model Building Phase Phase 5: Communicate Results Phase 6: Operationalize	**22**	**11**
2	**Data mining algorithms** Association rules, Motivation and terminology, Example: mining weather data, Regression Models **Introduction to simple Linear Regression:** The Regression Equation, Fitted value and Residuals, Least Square, Introduction to **Multiple Linear Regression:** Assessing the Model, Cross-Validation, Model Selection and Stepwise Regression, Prediction Using Regression **Logistic Regression:** Logistic Response function and logit, Logistic Regression and GLM, Generalized Linear model, Predicted values from Logistic Regression, Interpreting the coefficients and odds ratios, Linear and Logistic Regression: similarities and Differences, Assessing the models. **Time Series:** Overview of Time Series Analysis Box-Jenkins Methodology, ARIMA Model Autocorrelation Function (ACF) , Autoregressive Models, Moving Average Models, ARMA and ARIMA Models, Building and Evaluating an ARIMA Model, Reasons to Choose and Cautions. **Text Analytics:** History of text mining, Roots of text mining overview of seven practices of text analytic, Application and use cases for Text mining: extracting meaning from unstructured text, Summarizing Text. Text Analysis Steps, A Text Analysis Example, Collecting Raw Text, Representing Text, Term Frequency—Inverse Document Frequency (TFIDF), Categorizing Documents by Topics,	**32**	**15**

	Determining Sentiments , Gaining Insights.		
3	**Data analytics and visualization with R** Introduction to R: Data Import and Export, Attribute and Data type, Descriptive statistics. Exploratory Data Analysis: Visualization before analysis, Dirty Data, visualizing single variable, examining Multiple variables, Data Exploration versus presentation.	**23**	**11**
4	**Data analytics and Visualization with Python** Essential Data Libraries for data analytics: Pandas, NumPy, SciPy. Plotting and visualization with python: Introduction to Matplotlib, Basic Plotting with Matplotlib, Create Histogram, BarChart, Pie chart, Box Plot, violin plot using Matplotlib. Introduction to seaborn Library, Multiple Plots, Regression plot, regplot.	**23**	**11**
	Total	**100**	**48**

Reference Books

1.	**Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data.(TextBook) Wiley Publications**
2.	**Data Analytics using Python By Bharati Motwani, Wiley Publications.**
3.	**Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python** By By Peter Bruce and Andrew Bruce | O'Reilly Publications | 2nd Edition
4.	**Practical Text Mining and statistical Analysis for non-structured text data applications.** By Grey Miner, Thomas Hill. | 1st edition,
5.	**Data Analytics using R,** By Bharati Motwani, | Wiley Publications
6.	**McKinney, W.(2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython. 2nd edition. O'Reilly Media**

Course Outcome

After Learning the Course the students shall be able to:

1. Comprehend basics of data analytics and visualization. Apply various regression models on given data set and perform prediction.
2. Demonstrate advance understanding of Time series concepts and analysis of data using various time series models.
3. Analyze Text data and gain insights. Experiment with different analytics techniques and visualization using R.
4. Experiment with different analytics techniques and visualization using Python.

Miscellaneous

Useful Links

1. http://varianceexplained.org/RData/
2. https://www.kaggle.com/code/iamleonie/time-series-interpreting-acf-and-pacf
3. https://www.geeksforgeeks.org/data-visualization-using-matplotlib/

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: Student should have the basic Knowledge of Data Bases and its Management, basic Knowledge of Data warehousing. Also, they have the awareness of Associate Analytics & Big data concepts and able to understand the Tools & technology related to the maintenance of Data bases

Rationale : At the end of the course, the students will be able to: 1. Teach the students and make them Understand Basics of Predictive Analytics, Understand Data types and Variable types, Understand Basic Modeling and work on Missing Data 2. Teach the students and make them Understand Logistic Regression and its components, execute variable transformation and Understand Tableau visualization. 3. Teach the students objective segmentation CHAID and CART 4. Teach the students Time series methods/ forecasting feature Extraction and various forecasting methods like Univariate stationary processes (ARMA) and Non- stationary, integrated processes (ARIMA), Extract features from the generated model as height, Average, Energy etc. and analyze prediction business capabilities and ETL Approach. 5. Develop an understanding of basic skills necessary to work with Geographic Information Systems (GIS), using ESRI's ArcGIS software 6. Learn about GIS data types, Learn spatial data visualization technique

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	-	-	-	3	60	40	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Unit- I: Introduction to Predictive Analytics & Linear Regression:** What and Why Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of data and variables, Data Modeling Techniques, Missing imputations etc. Need for Business Modeling, Regression — Concepts, Blue property-assumptions-Least Square Estimation, Variable Rationalization, and Model Building etc. Regression: Model Theory, Model fit Statistics, Model Conclusion, Analytics applications to various Business Domains etc. Data preparation and exploration: Cleaning, wrangling, and visualizing data, dealing with missing values and outliers, and using summary statistics. Simple and multiple linear regression, assessing model fit, and using cross-validation. Classification: Logistic regression, support vector machines (SVM), decision trees, and Naive Bayes classifiers. Advanced and ensemble methods: Neural networks, boosting, bagging, and model stacking. Other models: Time series analysis, clustering, and recommendation systems. Model evaluation: Confusion matrices, performance metrics, and comparing models.	24	11
2	**Unit-II: Objective Segmentation:** Regression Vs Segmentation — Supervised and Unsupervised Learning, Tree Building — Regression, Classification, over fitting, Pruning and complexity, Multiple Decision Trees etc. Develop Knowledge, Skill and Competences, Introduction to Knowledge skills & competences, Training & Development, Learning & Development, Policies and Record keeping. etc.	20	8
3	**Unit-III: Foundational concepts:** : History and theory of spatial analysis, archaeological data, and spatial legacy data. Introduction to GIS and spatial data: Data types, data models (vector/raster), coordinate systems, and projections. **Geospatial data input and editing:** Data creation, digitization, attribute linking, and database management. **Cartography and visualization**: Basic cartographic principles and map design. **GIS software:** Familiarity with software such as ArcGIS, including its tools and functions. **Spatial data and visualization**: Spatial legacy data, data visualization, and time and space in analysis.	28	13

	Geographical modeling: Settlement geography, settlement and environment, and historical landscape analysis. **Modeling techniques:** Spatial agent-based modeling, spatial networks, and spatial interaction modeling. **Spatial statistics:** Analyzing spatial point patterns and processes, and using space syntax. **Data-driven spatial analysis:** Spatial statistics and data analysis in general.		
4	**Unit-IV: Core spatial analysis techniques--Queries and reasoning:** Spatial and attribute queries to answer simple questions. **Measurements:** Calculating distance, density, and neighborhood statistics. **Transformations:** Data transformation and manipulation, including vector overlays and raster calculations. **Spatial autocorrelation:** Understanding and measuring spatial relationships using statistics like Moran's I. **Point pattern analysis:** Analyzing the distribution of point data. Advanced modeling and analysis **Surface analysis:** Analyzing elevation data using tools like TIN, slope, aspect, and watershed modeling. **Network analysis:** Performing routing and allocation analyses in networks. **Suitability and multicriteria analysis:** Combining multiple criteria to evaluate sites for a specific purpose. **Simulation and modeling**: Building models to simulate spatial processes and project future changes. **Hydrological and environmental modeling:** Specific applications of modeling in environmental and ecological contexts.	**28**	**13**
	Total	**100**	**45**

Reference Books

1.	**GIS for the Urban Environment.** By Julie Maantay and John Ziegler,
2.	**The Design and Analysis of Spatial Data Structures,**
3.	**Building a Geodatabase, Redlands** By Mac Donald,
4.	**Principles of Geographical Information Systems,** By Burrough, P. A. and McDonnell, R. A. (2000) | Oxford University Press,
5.	**To nurture students in the field of Geoinformatics.**
6.	**To provide an in depth understanding of fundamentals of GIS, Remote sensing and their applications.**

Course Outcome

After Learning the Course the students shall be able to:

1. Understand how Predictive analytics can be used in the IT environment. Grasp the meaning, benefits of Predictive analytics
2. Understand analyze prediction business capabilities using Time series/Forecasting methods and Extract features.
3. Learn how to analyze spatial data, interpret geographic information, use software, understand different data models like vector and raster.
4. Apply these skills in fields like urban planning or environmental management.

Miscellaneous

Useful Links

1. https://courses.spatialthoughts.com/
2. https://geospatialworld.net/article/spatial-analysis-modelling/
3. https://ebooks.inflibnet.ac.in/esp06/chapter/spatial-analysis-network-analysis/
4. https://www.mygreatlearning.com/academy/learn-for-free/courses/predictive-analytics1
5. https://www.youtube.com/@PredictiveModeling?app=desktop

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: To learn cloud-native development, you need a foundation in programming (like Python or Java), Linux, networking, and basic cloud computing concepts.

Rationale : By the end of the course, you will be able to: 1. Understand the concepts, principles, and practices of cloud-native architecture, microservices, and serverless computing. 2. Learn how to design, develop, test, and deploy cloud-native applications using various tools and frameworks. 3. Gain hands-on experience with microservices, Kubernetes through exercises and projects. 4. Explore the advanced topics in cloud-native architecture such as observability, security, resilience, scalability, and DevOps.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	-	-	-	3	60	40	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Unit- I: Cloud Deployment Models:** -Public Cloud, Private Cloud, Community Cloud, Hybrid Cloud, Multi-Cloud, Cloud Platforms AWS, Google Cloud, Azure, OpenStack Cloud Service Models: - IaaS, PaaS, SaaS, Managed/Serverless Services, IT Ops, SysOps, DevOps, Low Ops, No Ops Activity: Planning the Case Study Architecture What Is Cloud-Native? Introduction to Cloud Native: Principles, goals, and key technologies. Traditional Development vs. Cloud-Native, Agile Practices, Feature-Driven Development Value-Driven Development, Agile Teams, Writing User Stories Activity: Analyzing the Case Study Requirements, **Automation and DevOps:** Infrastructure as Code (IaC), Continuous Integration and Continuous Delivery (CI/CD) **Application Architecture:** Microservices versus monoliths, API integration, and designing for resilience. **Containerization:** Using Docker to build, run, and distribute container images. **Container Orchestration:** Deploying and managing containers at scale using Kubernetes. **DevOps and CI/CD:** Implementing continuous integration, continuous delivery, and automation to streamline the development lifecycle.	**26**	**12**
2	**Unit-II: Building and Deploying Applications Programming:** Using languages like JavaScript, Node.js, and NoSQL databases for backend development. **Cloud Platforms:** Choosing and using services from public, private, and hybrid cloud environments. **Data and Storage:** Leveraging cloud-based data storage services. **Serverless Computing:** Simplifying deployments with PaaS and serverless environments. **Application Life Cycle Management:** Package Management, Managing Application Dependencies, Maven and Gradle Pip, NPM, NuGet, Source Control, Git- asic Git Commands, GitHub-Version Control with Git on the Cloud **Activity:** Using Git with the Case Study Version Control and Collaboration Branching and Merging Strategies **Activity:** Branching and Collaboration	**24**	**11**
3	**Unit-III:Docker:** Understanding Docker, Containers, Advantages of Containers, Images, Using Docker Basic Docker Commands, Building Docker Images, Dockerfile, Starting Containers, Stopping Containers, Deleting Containers and Images **Activity:** Containerizing the Case Study	**25**	**11**

	Deploying Docker Containers: Container Registries, Push and Pull **Activity:** Managing Docker Containers in a Container Registry Management and Operations **Infrastructure as Code (IaC):** Automating infrastructure provisioning. **Observability:** Implementing monitoring, logging, and tracing. **Security:** Addressing security basics, compliance, and access management. **Reliability and Disaster Recovery:** Planning for fault tolerance and business continuity.		
4	**Unit-IV: Some Advanced Topics:** **Resilience and Fault Tolerance:** Building applications that can withstand failures. **AI and Machine Learning:** Integrating AI to make applications "smarter". **Observability:** Ensuring applications are easy to monitor and troubleshoot.	**25**	**11**
	Total	**100**	**45**

Reference Books

1.	**Cloud-Native Computing: How to Design, Develop, and Secure Microservices and Event-Driven Applications.** By Raj, P., Vanga, S., & Chaudhary, A. | John Wiley & Sons.
2.	**Serverless as a Game Changer: How to Get the Most Out of the Cloud. Addison-Wesley Signature Series** By Emison, J.
3.	**Cloud-Native Application Architecture: Microservice Development Best Practice, Springer** By FreeWheel Biz-UI Team

Course Outcome

After Learning the Course the students shall be able to:

1. Explain cloud deployment and service models, cloud-native principles, and DevOps practices for designing scalable application architectures.
2. Develop and deploy cloud-native applications using modern programming tools, cloud platforms, version control, and CI/CD pipelines.
3. Containerize and manage applications using Docker and Kubernetes, including image creation, registries, orchestration, and infrastructure automation.
4. Apply operational best practices such as security, observability, resilience, disaster recovery, and integration of advanced technologies in cloud-native systems.

Miscellaneous

Useful Links

1. https://aws.amazon.com/
2. https://cloud.google.com/
3. https://www.datacamp.com/blog/learn-cloud-computing
4. https://www.pluralsight.com/professional-services/cloud/cloud-native-development-practices.

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Some foundational knowledge in data analytics, Data Visualization Concepts and Mathematical Formulas and Functions

Rationale : Upon completion of this course, students will be able to:students will be able to:Import, transform and cleanse data using Power Query Editor, build a data model for self-service reporting and manipulate the model with DAX, publish and share visualizations.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1: Do the Data Handling and Preparation. 2: Prepare the Report and Dashboard Creation. 3: Do the Data Analysis and Interpretation. 4: Do the Sharing and Collaboration of Reports and Dashboard

List of Practical

1.	**Using MSEXCEL, create a dashboard to display the implementation of the Pivot Table, Slicers and Charts to show the change in trends of Sales Data Analysis with the change over time.**
2.	**Using MSEXCEL, create a Sheet to display the implementation of the Goal Seek, Scenario, and VLOOKUP to show the data as per the condition and User's Request.**
3.	**Using POWER BI, create a dashboard to display the implementation of the followings: Cards and Multi Row Cards, Building and Formatting Charts, Line Charts, KPI Cards, Basic Filtering Options**
4.	**Using POWER BI create a dashboard to display the implementation of the followings: Use of Donus and Filters in Power BI, Table & Matrix Visuals, Conditional Formatting, Top N Filters, Map Visuals, Slicers, Gauge Charts, Use of QnA Feature, Use of Gauge Charts, Waterfall Chart, Scatter Chart.**
5.	**Using POWER BI, Implement the Functions in DAX and show the Cumulative Sales and Moving Average.**
6.	**Students will choose a dataset and create a comprehensive data visualization project. The project should demonstrate their understanding of data visualization techniques and data storytelling**
7.	**Using Power BI for Creating and Formatting Visualizations**
8.	**Using Slicers and Filters to interact with your reports.**
9.	**Using Power BI to implement Conditional Formatting to highlight specific data points.**

Course: M.Sc. (Data Science) **Semester:** 1

Prerequisite: Solid foundation in front-end technologies like HTML, CSS, and JavaScript, including frameworks like React or Angular

Rationale : Upon completion of this course, students will be able to:become knowledgeable about the most recent web development technologies, build Idea for creating two tier and three tier architectural web applications, Design and analyze real time web applications, Constructing suitable client and server-side applications and to learn core concept of both front end and back-end programming.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Reference Books

1.	**Full Stack JavaScript: Learn Backbone.js, Node.js and MongoDB.**
2.	**Jon Duckett, "JavaScript & jQuery: Interactive Front-End Web Development",**
3.	**Mastering Full Stack React Web Development Paperback** By TomaszDyl , Kamil Przeorski , Maciej Czarnecki., Pub. Year 2017
4.	**Full-Stack JavaScript Development by Eric Bush**

Course Outcome

After Learning the Course the students shall be able to:
1: Develop a fully functioning website and deploy on a web server. 2: Gain Knowledge about the front end and back-end Tools 3: Find and use code packages based on their documentation to produce working results in a project. 4: Create web pages that function using external data. Implementation of web application employing efficient database access.

List of Practical

1.	**Write a script that Logs "Hello, World!" to the console. Create a script that calculates the sum of two numbers and displays the result in an alert box. b. Create an array of 5 cities and perform the following operations: Log the total number of cities. Add a new city at the end. Remove the first city. Find and log the index of a specific city.**
2.	**Read a string from the user, Find its length. Extract the word "JavaScript" using substring() or slice(). Replace one word with another word and log the new string. Write a function is Palindrome(str) that checks if a given string is a palindrome (reads the same backward).**
3.	**Create an object student with properties: name (string), grade (number), subjects (array), displayInfo() (method to log the student's details) Write a script to dynamically add a passed property to the student object, with a value of true or false based on their grade. Create a loop to log all keys and values of the student object.**
4.	**Create a button in your HTML with the text "Click Me". Add an event listener to log "Button clicked!" to the console when the button is clicked. Select an image and add a mouseover event listener to change its border color. Add an event listener to the document that logs the key pressed by the user.**
5.	**Build a React application to track issues. Display a list of issues (use static data). Each issue should have a title, description, and status (e.g., Open/Closed). Render the list using a functional component.**
6.	**Create a component Counter with A state variable count initialized to 0. Create Buttons to increment and decrement the count. Simulate fetching initial data for the Counter component using useEffect (functional component) or componentDidMount (class component). Extend the Counter component to Double the count value when a button is clicked. Reset the count to 0 using another button.**
7.	**Install Express (npm install express). Set up a basic server that responds with "Hello, Express!" at the root endpoint (GET /). Create a REST API. Implement endpoints for a Product resource: GET : Returns a list of products. POST : Adds a new product. GET /:id: Returns details of a specific product. PUT /:id: Updates an existing product. DELETE /:id: Deletes a product. Add middleware to log requests to the console. Use express.json() to parse incoming JSON payloads.**
8.	**Install the MongoDB driver for Node.js.** Create a Node.js script to connect to the shop database. Implement insert, find, update, and delete operations using the Node.js MongoDB driver. Define a product schema using Mongoose. Insert data into the products collection using Mongoose. Create an Express API with a /products endpoint to fetch all products.Use fetch in React to call the /products endpoint and display the list of products. Add a POST /products endpoint in Express to insert a new product. Update the Product List, After adding a product, update the list of products displayed in Reactcts displayed in React

Miscellaneous

Useful Links

1.https://www.guvi.in/blog/best-ways-to-learn-full-stack-development/
2.https://www.geeksforgeeks.org/blogs/what-is-full-stack-development/
3.https://github.com/vasansr/pro-mern-stack
4.https://nptel.ac.in/courses/106106156

Course: M.Sc. (Data Science) **Semester:** 2

Prerequisite: To learn time series analysis, students need a foundation in statistics, mathematics (including calculus and linear algebra), and a programming language like Python or R. Also required familiarity with concepts like mean, standard deviation, and regression is essential, and it's also helpful to have some knowledge of econometrics and how to use data analysis tools.

Rationale : By the end of the course, you will be able to: 1. Get acquainted with the main concepts of Time Series theory and methods of analysis. 2. Know how to use them in examining financial processes 3. Understand methods, ideas, results and conclusions of articles on economics and finance. 4. Become master in traditional methods of Time Series analysis, intended mainly for working with time series data. 5. Understand the differences between cross-sections and time series, and those specific economic problems, which occur while working with data of these types.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
-	-	4	-	2	-	20	20	-	60	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction To Trend** Definition, components (trend, seasonality, cyclicity, and noise), and visualization, Introduction to times series data, application of time series from various fields, Components of a time series, Decomposition of time series. Trend: Estimation of trend by free hand curve method, method of semi averages, fitting a various mathematical curve, and growth curves. **Probability and statistics review:** Foundational concepts required for understanding statistical models. **Stationarity and ergodicity**: Key properties of time series data. **Autocorrelation functions:** Auto-correlation function (ACF) and partial auto-correlation function (PACF).	**20**	**12**
2	**Trend And Seasonal Component** Method of moving averages. Detrending. Effect of elimination of trend on other components of the time series. Seasonal Component: Estimation of seasonal component by Method of simple averages, Ratio to Trend, Ratio to moving average and Link relatives. Modelling and forecasting: **Random processes:** Linear processes, autoregressive (AR) models, moving average (MA) models, and ARMA models. **Non-stationary models:** Trend and seasonality modelling, ARIMA, and SARIMA models. **Forecasting techniques:** Exponential smoothing, short-term forecasting methods, and forecasting for trend and seasonal processes. **Advanced models:** Introduction to ARCH and GARCH models, particularly for finance applications.	**25**	**14**
3	**Fitting and plotting of modified exponential curve**	**5**	**2**
4	**Fitting and plotting of Gompertz curve**	**5**	**2**
5	**Fitting and plotting of logistic curve**	**5**	**2**
6	**Fitting of trend by Moving Average Method**	**5**	**2**
7	**Measurement of Seasonal indices Ratio-to-Trend method**	**5**	**2**
8	**Measurement of Seasonal indices Ratio-to-Moving Average method.**	**5**	**2**
9	**Measurement of seasonal indices Link Relative method**	**5**	**2**
10	**Calculation of variance of random component by variate difference method**	**5**	**2**

11	**Forecasting by exponential smoothing**	**5**	**2**
12	**Forecasting by short term forecasting methods**	**5**	**2**
	Total	**95**	**46**

Reference Books

1.	**Enders W. Applied Econometric Time Series.** By John Wiley & Sons,, Pub. Year 1995
2.	**The Econometric Modelling of Financial Time Series.**
3.	**Time Series Models.**
4.	**The Econometric Analysis of Time Series.** By Philip Allan, Andrew C. Harvey., Pub. Year 1990
5.	**Co-Integration, Error Correction, and Econometric Analysis of Non-Stationary Data.** By Banerjee, A., J.J. Dolado, and D.V. Hendry. | Oxford University Press, Pub. Year 1993
6.	**Introduction to Time Series and Forecasting.**

Course Outcome

After Learning the Course the students shall be able to:

1. Explain the components of time series data, stationarity, autocorrelation measures, and methods for trend estimation and decomposition.
2. Apply techniques for modeling and estimating trend and seasonal components using moving averages, growth curves, and seasonal index methods.
3. Build and analyze time series forecasting models including AR, MA, ARMA, ARIMA, SARIMA, and exponential smoothing techniques.
4. Fit, plot, and evaluate time series models and curves to analyze variance and forecasting accuracy for real-world datasets.

Miscellaneous

Useful Links

1. https://github.com/thuml/Time-Series-Library.
2. https://timeserieslab.com/
3. https://www.youtube.com/watch?v=zPHqVveF_ko