

B.TECH - DS

SEM-I. BCO572A	Decoding Data: Learn AI, ML, DS & Analytics	2-0-2 [4]
---------------------------	--	------------------

OJECTIVE:

- To introduce the fundamentals of Data Science, Big Data, and their ecosystem, tools, and applications.
- To understand data types, data processing steps, and the complete Data Science process.
- To provide foundational knowledge of Machine Learning concepts, algorithms, and applications.
- To explore Artificial Intelligence, its history, approaches, and problem-solving techniques.
- To develop practical data analytics skills using Excel, Power BI, and visualization tools.

UNIT 1	Introduction to Data Science: Defining Data Science and Big Data, Benefits and Uses of Data Science and Big Data, Facets of Data, Structured Data, Unstructured Data, Natural Language, Machine generated Data, Graph based or Network Data, Audio, Image, Video, Streaming data, Data Science Process, Big data ecosystem and data science, distributed file systems, Distributed programming framework, data integration framework, machine learning framework, No SQL Databases, scheduling tools, benchmarking tools, system deployments
UNIT 2	Data Science Processes: Six steps of data science processes, define research goals, data retrieval, cleansing data, correct errors as early as possible, integrating – combine data from different sources, transforming data, exploratory data analysis, Data modelling, model and variable selection, model execution, model diagnostic and model comparison, presentation and automation.
UNIT 3	Introduction to Machine Learning: What is Machine Learning, Learning from Data, History of Machine Learning, Big Data for Machine Learning, Leveraging Machine Learning, Descriptive vs Predictive Analytics, Machine Learning and Statistics, Artificial Intelligence and Machine Learning, Types of Machine Learning – Supervised, Unsupervised, Semi-supervised, Reinforcement Learning, Types of Machine Learning Algorithms, Classification vs Regression Problem, Bayesian, Clustering, Decision Tree, Dimensionality Reduction, Neural Network and Deep Learning, Training machine learning systems

UNIT 4	Introduction to AI: What is AI, Turing test, cognitive modelling approach, law of thoughts, the relational agent approach, the underlying assumptions about intelligence, techniques required to solve AI problems, level of details required to model human intelligence, successfully building an intelligent problem, history of AI
UNIT 5	Introduction to Data Analytics: Working with Formula and Functions, Introduction to Power BI & Charts, Logical functions using Excel, Analysing Data with Excel.

Course Outcome (CO):

At the ends of this course students will have:

CO1: Understand the core concepts of Data Science, Big Data, and their ecosystem frameworks.

CO2: Apply data science processes for data collection, cleaning, integration, transformation, and analysis.

CO3: Demonstrate knowledge of Machine Learning types, algorithms, and their practical applications.

CO4: Analyze Artificial Intelligence principles, approaches, and techniques to solve problems.

CO5: Perform data analysis and visualization using Excel functions, charts, and Power BI.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	M	M	M								H		
CO2	M	M	H	M	H				L				H	M	
CO3	M	M	H	M	L									M	
CO4	H	L	M	L	H				M			M	L		M
CO5	M	M	H	H	M							H	M	H	M

H = Highly Related; M = Medium L = Low

Text Books:

1. Foster Provost & Tom Fawcett, *Data Science for Business*, O'Reilly Media, 1st Edition.
2. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2nd Edition.
3. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, 4th Edition.

Reference Books:

1. Trevor Hastie, Robert Tibshirani & Jerome Friedman, *The Elements of Statistical Learning*, Springer, 2nd Edition.
2. Ian Goodfellow, Yoshua Bengio & Aaron Courville, *Deep Learning*, MIT Press, 1st Edition.

BCO573A	Decoding Data: Learn AI, ML, DS & Analytics LAB	0-0-2 [2]
---------	---	-----------

List of Experiments

1. Data Import & Profiling

Aim: Load heterogeneous data and generate a first-pass profile.

Dataset: “Titanic” or “Iris” (CSV).

Tasks: a) Import CSV/Excel; b) Inspect dtypes, nulls, unique counts; c) Basic stats; d) Save a profiling report.

2. Data Cleaning & Missing-Value Treatment

Aim: Detect and fix data quality issues.

Dataset: Retail sales with intentional missing/outliers.

Tasks: a) Identify missing by column; b) Impute (mean/median/mode/ML); c) Handle outliers (IQR/winsorize); d) Validate with before/after stats.

3. Deduplication & Data Integrity Checks

Aim: Enforce uniqueness and referential integrity.

Dataset: Customers + Transactions (two tables).

Tasks: a) Find duplicates by composite keys; b) Resolve conflicts; c) Check FK integrity; d) Document data quality rules.

4. Data Integration (Joins) & Schema Alignment

Aim: Combine multi-source data into a single, tidy table.

Dataset: Orders, Order_Items, Products.

Tasks: a) Normalize column names; b) Inner/left joins; c) Resolve key mismatches; d) Verify row counts.

5. Feature Engineering & Encoding

Aim: Create informative features for ML.

Dataset: Bank marketing or Telco churn.

Tasks: a) Create ratios/buckets/dates→age/tenure; b) One-hot vs label encoding; c) Scale/standardize; d) Train a quick baseline to compare.

6. Exploratory Data Analysis (EDA) & Correlations

Aim: Identify structure, relationships, and anomalies.

Dataset: Housing prices.

Tasks: a) Univariate (hist/box); b) Bivariate (scatter/corr heatmap); c) Detect multicollinearity (VIF); d) Insights list.

7. Visualization Best Practices

Aim: Communicate findings clearly.

Dataset: Any from above.

Tasks: a) Build hist/box/scatter/line; b) Choose correct chart per question; c) Add labels/axes/notes; d) Avoid clutter; e) Export PNGs.

8. Train/Test Split & Cross-Validation

Aim: Estimate generalization reliably.

Dataset: Iris or Telco churn.

Tasks: a) Stratified split; b) k-fold CV; c) Compare metrics variance; d) Fix data leakage; e) Persist splits.

9. Linear Regression & Diagnostics

Aim: Model a continuous target and validate assumptions.

Dataset: Housing prices (with numeric predictors).

Tasks: a) Fit baseline LR; b) Residual plots; c) Check normality/heteroscedasticity; d) RMSE/MAE/R²; e) Add regularization (Ridge/Lasso) and compare.

10. Logistic Regression for Classification

Aim: Binary classification with interpretability.

Dataset: Telco churn or bank marketing.

Tasks: a) Fit logistic model; b) Confusion matrix; c) ROC-AUC/PR-AUC; d) Threshold tuning; e) Explain coefficients/odds ratios.

11. Clustering with K-Means

Aim: Segment data without labels.

Dataset: Customer RFM features.

Tasks: a) Scale features; b) Elbow & silhouette; c) Fit K-means; d) Profile clusters (size, means); e) Visualize in 2D (PCA).

12. Dimensionality Reduction with PCA

Aim: Reduce dimensionality and visualize structure.

Dataset: Any 20+ feature dataset.

Tasks: a) Standardize; b) Fit PCA; c) Variance explained plot; d) Biplot; e) Refit a model on reduced features and compare.

13. Decision Trees & Feature Importance

Aim: Build interpretable non-linear models.

Dataset: Credit risk or breast cancer.

Tasks: a) Train tree; b) Control depth/min_samples; c) Evaluate accuracy/F1; d) Plot tree; e) Feature importances and stability across seeds.

14. Text Classification with Naive Bayes (TF-IDF)

Aim: Classify short texts (spam/ham or reviews).

Dataset: SMS Spam Collection or IMDb subset.

Tasks: a) Clean text; b) Tokenize + TF-IDF; c) Multinomial NB; d) Evaluate F1/ROC-AUC; e) Top features per class.

15. Excel & Power BI Analytics Mini-Suite

Aim: Hands-on business analytics from Excel to BI.

Dataset: Sales orders (date, region, product, revenue, cost).

Tasks: a) Excel: clean + build PivotTables; b) KPIs (Revenue, Profit, ROI) via formulas; c) Create charts; d) Power BI: import, relationships, basic DAX (SUMX, CALCULATE); e) Interactive dashboard (slicers).

Course Outcome (CO):

At the ends of this course students will have:

CO1: Understand the core concepts of Data Science, Big Data, and their ecosystem frameworks.

CO2: Apply data science processes for data collection, cleaning, integration, transformation, and analysis.

CO3: Demonstrate knowledge of Machine Learning types, algorithms, and their practical applications.

CO4: Analyze Artificial Intelligence principles, approaches, and techniques to solve problems.

CO5: Perform data analysis and visualization using Excel functions, charts, and Power BI.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	M	M	M								H		
CO2	M	M	H	M	H				L				H	M	
CO3	M	M	H	M	L									M	
CO4	H	L	M	L	H				M			M	L		M
CO5	M	M	H	H	M							H	M	H	M

H = Highly Related; M = Medium L = Low

SEM-II BCO169C	Data Analysis Using Python	2-0-2 [4]
---------------------------	-----------------------------------	------------------

OJECTIVE:

- Understand the Python programming environment, syntax, and basic control flow.
- Learn to work with Python data structures, functions, and file operations.
- Apply NumPy for efficient array and matrix computations.
- Perform data manipulation and analysis using pandas.
- Create informative and customizable visualizations using matplotlib and seaborn.

UNIT 1	Python programming Basic: Python interpreter, IPython Basics, Tab completion, Introspection, %run command, magic commands, matplotlib integration, python programming, language semantics, scalar types. Control flow.
UNIT 2	Data Structure, functions, files: tuple, list, built-in sequence function, dict, set, functions, namespace, scope, local function, returning multiple values, functions are objects, lambda functions, error and exception handling, file and operation systems
UNIT 3	NumPy: Array and vectorized computation: Multidimensional array object. Creating ndarrays, arithmetic with numpy array, basic indexing and slicing, Boolean indexing, transposing array and swapping axes, universal functions, array-oriented programming with arrays, conditional logic as arrays operations, file input and output with array
UNIT 4	Pandas: Pandas data structure, series, DataFrame, Index Object, Reindexing, dropping entities from an axis, indexing, selection and filtering, integer indexes, arithmetic and data alignment, function application and mapping, soring and ranking, correlation and covariance, unique values, values controls and membership, reading and writing data in text format
UNIT 5	Visualization with Matplotlib: Figures and subplots, colors, markers, line style, ticks, labels, legends, annotation and drawing on subplots, matplotlib configuration. Plotting with pandas and seaborn: line plots, bar plots, histogram, density plots, scatter and point plots, facet grids and categorical data

Course Outcome (CO):

At the ends of this course students will have:

CO1: Describe the Python interpreter, control flow, and basic syntax.

CO2: Use built-in data structures, functions, and manage file I/O and exceptions.

CO3: Implement data processing using NumPy arrays and functions.

CO4: Analyze and manipulate structured data using pandas.

CO5: Visualize data using matplotlib and seaborn plotting libraries.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	M	H								H	M	
CO2	H	M	H	M	M				L				M	H	
CO3	M	H	M	M	L									M	M
CO4	M	L	M	L	H				M			M	L		M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. "Python for Data Analysis" – Wes McKinney, 3rd Edition, O'Reilly Media, 2022.
2. "Learning Python" – Mark Lutz, 5th Edition, O'Reilly Media, 2013.
3. "NumPy Beginner's Guide" – Ivan Idris, 3rd Edition, Packt Publishing, 2015.

Reference Books:

1. "Fluent Python" – Luciano Ramalho, 2nd Edition, O'Reilly Media, 2022.
2. "Think Python" – Allen B. Downey, 2nd Edition, Green Tea Press, 2015.

List of Experiments

1. Verify Anaconda installation and list available packages
 - a) Learn to launch and use Anaconda Navigator and Anaconda Prompt.
 - b) Use the conda list command to verify package installations.
 - c) Confirm Anaconda's version and Python environment details using CLI.
2. Write a program to install, search, and uninstall packages using Conda
 - a) Use conda search, conda install, and conda uninstall commands.
 - b) Explore Anaconda repositories to locate scientific Python packages.
 - c) Understand the difference between base and environment-level installations.
3. Write a program to modify a list inside a tuple and attempt to modify the tuple itself
 - a) Understand how tuple immutability works and how nested mutable objects behave.
 - b) Access and update nested structures like a list within a tuple.
 - c) Recognize the limitations and exceptions when trying to change a tuple.
4. Write a program to swap variables and iterate over tuples using unpacking
 - a) Practice tuple unpacking to simplify variable assignments.
 - b) Use unpacking with loops and the asterisk * operator to handle multiple values.
 - c) Implement variable swapping without a temporary variable.
5. Write a Python script to demonstrate the role of indentation in code blocks
 - a) Understand how Python uses whitespace instead of braces for blocks.
 - b) Practice consistent indentation with recommended spacing (4 spaces).
 - c) Avoid syntax errors by applying indentation rules in control structures.
6. Write a Python script to assign and update variables using dynamic binding
 - a) Explore variable assignment and reference behavior.
 - b) Observe how variables share references to the same object.
 - c) Differentiate between re-binding and object mutation.
7. Use identity and comparison operators to test object equality and type
 - a) Use is, is not, and == to test object identity and value.
 - b) Compare behavior of copied vs referenced objects.
 - c) Understand the use of is None as a best practice.
8. Write a program to accept user input and personalize output messages
 - a) Use input() to capture data from the user.
 - b) Understand how input() returns a string and requires casting for other types.
 - c) Concatenate and format output using variables and text.
9. Write a program to square a number using repeated addition and a while loop
 - a) Simulate multiplication through loop-based addition.
 - b) Practice while loops with decrementing counters.
 - c) Understand loop entry, termination, and flow control.
10. Write a program to print coordinate pairs with nested for loops and break logic
 - a) Nest for loops to iterate over grid-like structures.

- b) Use conditional breaks to control inner loop execution.
- c) Practice tuple-based printing for structured output.

11. NumPy Array Operations

- a) Create 1D, 2D, and 3D NumPy arrays using `np.array` and `np.arange`.
- b) Perform arithmetic operations (addition, subtraction, multiplication, division) on NumPy arrays.
- c) Apply universal functions like `np.sqrt`, `np.exp`, and `np.maximum` on arrays.

12. Indexing and Array Manipulation in NumPy

- a) Demonstrate basic indexing, slicing, and Boolean indexing.
- b) Transpose a matrix and swap axes of a 3D array.
- c) Implement conditional logic using NumPy (e.g., element-wise if-else using `np.where`).

13. Working with pandas Series and DataFrames

- a) Create a pandas Series and DataFrame, and explore index, columns, and values.
- b) Perform indexing, filtering, and selection using label-based and position-based accessors.
- c) Apply functions and mapping on Series/DataFrame, and demonstrate arithmetic and alignment.

14. Reading, Writing, and Analyzing Data with pandas

- a) Read data from a CSV file using `pd.read_csv()` and write to a new file.
- b) Drop rows/columns, sort data, and use `rank()`, `unique()`, `value_counts()`.
- c) Compute correlation and covariance between DataFrame columns.

15. Data Visualization using Matplotlib, pandas, and seaborn

- a) Plot line, bar, and scatter plots using `matplotlib.pyplot` and `pandas.plot()`.
- b) Create subplots with titles, labels, ticks, and legends.
- c) Use seaborn to create histograms, density plots, facet grids, and visualize categorical data.

Course Outcome (CO):

At the ends of this course students will have:

CO1: Describe the Python interpreter, control flow, and basic syntax.

CO2: Use built-in data structures, functions, and manage file I/O and exceptions.

CO3: Implement data processing using NumPy arrays and functions.

CO4: Analyze and manipulate structured data using pandas.

CO5: Visualize data using matplotlib and seaborn plotting libraries.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	M	L	M	H								H	M	
CO2	H	M	H	M	M				L				M	H	
CO3	M	H	M	M	L									M	M
CO4	M	L	M	L	H				M			M	L		M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-III BCO591A	Statistical Modelling and Data Reasoning with Python	2-0-2 [4]
--------------------	---	-----------

OJECTIVE:

- Understand fundamental statistical concepts and techniques for data summarization and analysis.
- Apply probability theory and distributions in modeling and interpreting real-world data.
- Explore and analyze random variables, expectations, and joint distributions.
- Learn methods of point and interval estimation for parameters in statistical models.
- Conduct hypothesis testing and perform Bayesian and frequentist inference using Python.

UNIT 1	<p>Introduction to Statistics: Introduction to Statistics. Role of statistics in scientific methods, current applications of statistics.</p> <p>Scientific data gathering: Sampling techniques, scientific studies, observational studies, data management.</p> <p>Data description: Displaying data on a single variable (graphical methods, measure of central tendency, measure of spread), displaying relationship between two or more variables, measure of association between two or more variables.</p>
UNIT 2	<p>Probability Theory: Sample space and events, probability, axioms of probability, independent events, conditional probability, Bayes' theorem.</p>
UNIT 3	<p>Random Variables: Discrete and continuous random variables. Probability distribution of discrete random variables, binomial distribution, poisson distribution. Probability distribution of continuous random variables, The uniform distribution, normal (gaussian) distribution, exponential distribution, gamma distribution, beta distribution, t-distribution, χ^2 distribution. Expectations, variance and covariance. Probability Inequalities. Bivariate distributions</p>
UNIT 4	<p>Point Estimations: Methods of finding estimators, method of moments, maximum likelihood estimators, bayes estimators. Methods of evaluating estimators, mean squared error, best unbiased estimator, sufficiency and unbiasedness</p> <p>Interval Estimations: Confidence interval of means and proportions, Distribution free confidence interval of percentiles</p>
UNIT 5	<p>Test of Statistical Hypothesis and p-values: Tests about one mean, tests of equality of two means, test about proportions, p-values, likelihood ratio test, Bayesian tests</p>

	<p>Bayesian Statistics: Bayesian inference of discrete random variable, Bayesian inference of binomial proportion, comparing Bayesian and frequentist inferences of proportion, comparing Bayesian and frequentist inferences of mean</p> <p>Univariate Statistics using Python: Mean, Mode. Median, Variance, Standard Deviation, Normal Distribution, t-distribution, interval estimation, Hypothesis Testing, Pearson correlation test, ANOVA F-test</p>
--	---

Course Outcome (CO):

At the ends of this course students will have:

CO1: Explain descriptive statistics, data collection techniques, and measures of association.

CO2: Apply probability rules, conditional probability, and Bayes' theorem to solve problems.

CO3: Use probability distributions and evaluate expectations, variances, and joint distributions.

CO4: Estimate parameters using various techniques and construct confidence intervals.

CO5: Perform hypothesis testing and compare Bayesian and frequentist methods using Python.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. **"Introduction to the Practice of Statistics"** – David S. Moore, George P. McCabe, Bruce A. Craig, 9th Edition, W.H. Freeman, 2017.
2. **"Probability and Statistics for Engineers and Scientists"** – Ronald E. Walpole et al., 9th Edition, Pearson, 2012.

3. **"Think Stats: Probability and Statistics for Programmers"** – *Allen B. Downey*, 2nd Edition, O'Reilly Media, 2014.

Reference Books:

1. **"All of Statistics"** – *Larry Wasserman*, 1st Edition, Springer, 2004.
2. **"Bayesian Statistics: An Introduction"** – *Peter M. Lee*, 4th Edition, Wiley, 2012.
3. **"Practical Statistics for Data Scientists"** – *Peter Bruce, Andrew Bruce, Peter Gedeck*, 2nd Edition, O'Reilly, 2020.

BCO593A	Statistical Modelling and Data Reasoning with Python LAB	0-0-2 [2]
----------------	---	------------------

List of Experiments

1. a. Create different types of graphical representations (bar chart, histogram, boxplot) for univariate data using Python
b. Calculate measures of central tendency (mean, median, mode) using Python for a given dataset
2. a. Demonstrate sampling techniques: simple random, stratified, and systematic sampling
b. Compare sample statistics to population statistics for bias and variability
3. a. Perform data cleaning and management using pandas
b. Handle missing values and outliers in a dataset
4. a. Calculate and interpret variance and standard deviation for a dataset
b. Compare spread of data using boxplots
5. a. Simulate and plot the binomial distribution using Python
b. Compute probabilities using the binomial formula
6. a. Generate Poisson distribution data and plot using matplotlib
b. Apply Poisson model to real-world count data
7. a. Create and analyze normal distribution data
b. Use z-scores to compute probabilities and percentiles
8. a. Simulate uniform and exponential distributions using NumPy
b. Visualize and compare their shapes
9. a. Derive and visualize t-distribution and χ^2 -distribution
b. Use them in hypothesis testing scenarios
10. a. Estimate parameters using the method of moments
b. Estimate parameters using maximum likelihood estimation
11. a. Construct confidence intervals for population mean and proportion
b. Visualize margin of error and interpret the results
12. a. Perform hypothesis testing for one sample mean
b. Interpret results using p-values

13. a. Test for equality of two means (independent and paired samples)
 - b. Perform hypothesis test for population proportions
14. a. Conduct Bayesian inference on a binomial proportion
 - b. Compare Bayesian and frequentist interval estimates
15. a. Calculate and interpret univariate statistics (mean, variance, standard deviation) using Python
 - b. Overlay normal curve on histogram and evaluate fit

Course Outcome (CO):

At the ends of this course students will have:

CO1: Explain descriptive statistics, data collection techniques, and measures of association.

CO2: Apply probability rules, conditional probability, and Bayes’ theorem to solve problems.

CO3: Use probability distributions and evaluate expectations, variances, and joint distributions.

CO4: Estimate parameters using various techniques and construct confidence intervals.

CO5: Perform hypothesis testing and compare Bayesian and frequentist methods using Python.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-III BCO280C	R Programming for Data Science and Data Analysis	2-0-2 [4]
----------------------------	---	------------------

OJECTIVE:

- Understand the R programming environment and workspace management using RStudio.
- Learn to create and manipulate data structures such as vectors, matrices, arrays, lists, and data frames in R.
- Apply control structures, conditional statements, loops, and custom functions to perform dynamic data analysis.
- Perform data analysis, handle missing values, and use statistical and mathematical functions for data computation.
- Visualize and analyze data using various graphical methods and apply basic statistical modeling techniques in R.

UNIT 1	Getting Started with R and R Workspace: Introducing R, R as a programming Language, the need of R, Installing RStudio, RStudio's user interface, console, editor, environment pane, history pane, file pane, plots pane, package pane, help and viewer pane
UNIT 2	R Workspace, R's working directory, R Project in R Studio, absolute and relative path, Inspecting an Environment, Inspect existing Symbols, View the structure of object, Removing symbols, Modifying Global Options, Modifying warning level, Library of Packages, Getting to know a package, Installing a Package from CRAN, Updating Package from CRAN, Installing package from online repository, Package Function, Masking and name conflicts
UNIT 3	Basic Objects and Basic Expressions: Vectors, Numeric Vectors, Logical Vectors, Character Vectors, subset vectors, Named Vectors, extracting element, converting vector, Arithmetic operators, create Matrix, Naming row and columns, subsetting matrix, matrix operators, creating and subsetting an Array, Creating a List, extracting element from list, subsetting a list, setting value, creating a value of data frame, subsetting a data frame, setting values, factors, useful functions of a data frame, loading and writing data on disk, creating a function, calling a function, dynamic typing, generalizing a function. Assignment Operators, Conditional Expression, using if as expression and statement, using if with vectors, vectorized if: ifelse, using switch, using for loop, nested for loop, while loop
UNIT 4	Working with Basic Objects and Strings: Working with object function, getting data dimensions, reshaping data structures, iterating over one dimension, logical operators, logical functions, dealing with missing values, logical coercion, math function, number rounding functions, trigonometric functions, hyperbolic functions, extreme functions, finding roots, derivatives and integration, Statistical function, sampling from a vector, Working with random distributions, computing

	summary statistics, covariance and correlation matrix, printing string, concatenating string, transforming text, Formatting text, formatting date and time, formatting date and time to string, finding string pattern, using group to extract data, reading data
UNIT 5	Working with Data – Visualize and Analyze Data: Reading and Writing Data, importing data using built-in-function, READR package, export a data frame to file, reading and writing Excel worksheets, reading and writing native data files, loading built-in data sets, create scatter plot, bar chart, pie chart, histogram and density plots, box plot, fitting linear model and regression tree.

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate understanding of the R programming environment, workspace, and package management.

CO2: Apply core data structures such as vectors, matrices, lists, and data frames for basic data operations.

CO3: Implement control structures, conditional logic, and custom functions for data manipulation and computation.

CO4: Use built-in mathematical, statistical, and string manipulation functions to perform exploratory data analysis.

CO5: Visualize and analyze datasets using R graphics and statistical modeling tools like linear regression and regression trees.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. *The Art of R Programming* by Norman Matloff, No Starch Press, 1st Edition
2. *Hands-On Programming with R* by Garrett Golemund, O'Reilly Media, 1st Edition

3. *R for Data Science* by Hadley Wickham and Garrett Golemund, O'Reilly Media, 1st Edition

Reference Books:

1. *Advanced R* by Hadley Wickham, CRC Press
2. *Practical Data Science with R* by Nina Zumel and John Mount, Manning Publications

BCO592A	R Programming for Data Science and Data Analysis LAB	0-0-2 [2]
----------------	---	------------------

List of Experiments

1.
 - a. Install R and RStudio
 - b. Explore RStudio interface panes
 - c. Set and inspect working directories
2.
 - a. Create and manipulate numeric, logical, and character vectors
 - b. Subset and name vectors
 - c. Perform vector arithmetic
3.
 - a. Construct matrices and arrays
 - b. Subset matrices by rows/columns
 - c. Apply matrix operations
4.
 - a. Create and extract elements from lists
 - b. Modify and set values in lists
 - c. Use named lists in functions
5.
 - a. Construct data frames from vectors
 - b. Subset rows/columns in data frames
 - c. Apply basic functions to data frames
6.
 - a. Create and work with factors
 - b. Summarize factor levels
 - c. Convert factors to numeric or character
7.
 - a. Load data from CSV and Excel using base and readr package
 - b. Export data to file
 - c. Load built-in datasets
8.
 - a. Write conditional expressions using if/else and switch
 - b. Use vectorized ifelse() for fast decisions
 - c. Combine logical operators for filtering
9.
 - a. Write custom functions in R
 - b. Generalize functions with arguments
 - c. Call functions dynamically
10.
 - a. Use for, while, and repeat loops
 - b. Nest loops for multidimensional data
 - c. Avoid loops using vectorization
11.
 - a. Handle missing values using is.na() and na.omit()
 - b. Perform logical coercion and filtering
 - c. Apply extreme value and root-finding functions
12.
 - a. Generate summary statistics (mean, median, sd, IQR)
 - b. Compute correlation and covariance matrices

- c. Sample data from vectors and distributions
- 13. a. Manipulate and transform strings using paste(), substr(), and stringr
 - b. Use regex to extract data
 - c. Format date and time
- 14. a. Create scatter plots, bar charts, pie charts, histograms
 - b. Customize plots using parameters
 - c. Save plots to file
- 15. a. Fit and interpret a linear model using lm()
 - b. Visualize model results
 - c. Apply regression tree modeling using rpart

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate understanding of the R programming environment, workspace, and package management.

CO2: Apply core data structures such as vectors, matrices, lists, and data frames for basic data operations.

CO3: Implement control structures, conditional logic, and custom functions for data manipulation and computation.

CO4: Use built-in mathematical, statistical, and string manipulation functions to perform exploratory data analysis.

CO5: Visualize and analyze datasets using R graphics and statistical modeling tools like linear regression and regression trees.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-IV BCO 0239C	Machine Learning and Pattern Recognition	2-0-2 [4]
-----------------------------	---	------------------

OJECTIVE:

- To introduce the fundamentals of machine learning, including learning systems, real-world applications, and key terminologies.
- To understand different types of machine learning approaches such as supervised, unsupervised, and reinforcement learning.
- To explore important concepts in model evaluation and selection including bias-variance trade-off, overfitting, and the curse of dimensionality.
- To develop the ability to implement and evaluate linear regression and multiple linear regression models.
- To gain knowledge in classification techniques such as logistic regression, linear and quadratic discriminant analysis, and apply Bayes' theorem in classification tasks.

UNIT 1	Introduction: Learning systems, real world applications of machine learning, why machine learning, variable types and terminology, function approximation
UNIT 2	Types of machine learning: Supervised learning, unsupervised learning, reinforcement learning
UNIT 3	Important concepts of machine learning: Parametric vs non-parametric models, the trade- off between prediction accuracy and model interpretability, the curse of dimensionality, measuring the quality of fit, bias-variance trade off, overfitting, model selection, no free lunch theorem
UNIT 4	Linear Regression: Linear regression, estimating the coefficients, accessing the accuracy of coefficient estimates, accessing the accuracy of the model, multiple linear regression, qualitative predictors
UNIT 5	Classification: Logistic regression, estimating regression coefficients, making predictions, multiple logistic regressions, linear discriminant analysis, bayes' theorem of classification, LDA for $p=1$, LDA for $p>1$, quadratic discriminant analysis

Course Outcome (CO):

At the ends of this course students will have:

CO1: Understand the fundamentals of machine learning, including learning systems, applications, and basic terminology.

CO2: Explore and differentiate between supervised, unsupervised, and reinforcement learning techniques.

CO3: Analyze key machine learning concepts such as model selection, overfitting, bias-variance trade-off, and the curse of dimensionality.

CO4: Apply and evaluate linear and multiple linear regression models for predictive analysis.

CO5: Implement classification algorithms including logistic regression, linear discriminant analysis, and quadratic discriminant analysis using Bayes' theorem.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. *"Introduction to Statistical Learning"* by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, Springer, 2nd Edition
2. *"Pattern Recognition and Machine Learning"* by Christopher M. Bishop, Springer, 1st Edition
3. *"Machine Learning: A Probabilistic Perspective"* by Kevin P. Murphy, MIT Press, 1st Edition

Reference Books:

1. *"The Elements of Statistical Learning"* by Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer, 2nd Edition
2. *"Machine Learning"* by Tom M. Mitchell, McGraw-Hill, 1st Edition

List of Experiments

1.
 - a. Explore different types of variables (categorical, numerical)
 - b. Load and preprocess datasets using Python (NumPy, pandas)
 - c. Handle missing data and perform data cleaning
2.
 - a. Visualize data using histograms, scatter plots, and box plots
 - b. Identify relationships between variables
 - c. Use pair plots to explore multidimensional data
3.
 - a. Implement simple linear regression using Python
 - b. Estimate regression coefficients manually and using sklearn
 - c. Plot regression line and interpret results
4.
 - a. Evaluate model performance using metrics like RMSE and R^2
 - b. Perform residual analysis
 - c. Detect and interpret overfitting visually
5.
 - a. Implement multiple linear regression using real-world dataset
 - b. Analyze influence of multiple predictors
 - c. Use statsmodels to view coefficient significance
6.
 - a. Handle qualitative (categorical) predictors using one-hot encoding
 - b. Integrate encoded variables into regression models
 - c. Interpret model output with categorical variables
7.
 - a. Implement logistic regression for binary classification
 - b. Evaluate predictions using confusion matrix and accuracy
 - c. Plot decision boundary
8.
 - a. Apply logistic regression on real-world classification data (e.g., Titanic dataset)
 - b. Use precision, recall, F1-score as evaluation metrics
 - c. Plot ROC curve and calculate AUC
9.
 - a. Explore the bias-variance trade-off with polynomial regression
 - b. Visualize underfitting and overfitting scenarios
 - c. Use cross-validation for model selection
10.
 - a. Demonstrate curse of dimensionality using synthetic data
 - b. Apply PCA for dimensionality reduction
 - c. Visualize high-dimensional data in 2D

11. a. Implement Linear Discriminant Analysis (LDA) for classification
b. Compare performance with logistic regression
c. Visualize LDA projection and class separation
12. a. Apply Quadratic Discriminant Analysis (QDA)
b. Compare QDA and LDA decision boundaries
c. Evaluate using accuracy and confusion matrix
13. a. Explore Bayes' theorem in the context of classification
b. Implement Naive Bayes classifier
c. Apply it on a text classification problem
14. a. Implement k-Fold cross-validation
b. Use it to compare model performance (e.g., LDA vs. QDA)
c. Interpret variance across folds
15. a. Perform model selection using grid search
b. Optimize logistic regression with regularization
c. Analyze results of hyperparameter tuning

Course Outcome (CO):

At the ends of this course students will have:

CO1: Understand the fundamentals of machine learning, including learning systems, applications, and basic terminology.

CO2: Explore and differentiate between supervised, unsupervised, and reinforcement learning techniques.

CO3: Analyze key machine learning concepts such as model selection, overfitting, bias-variance trade-off, and the curse of dimensionality.

CO4: Apply and evaluate linear and multiple linear regression models for predictive analysis.

CO5: Implement classification algorithms including logistic regression, linear discriminant analysis, and quadratic discriminant analysis using Bayes' theorem.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-IV BCO644A	Machine Learning with R, Python, Scikit-learn, Matplotlib, TensorFlow	2-0-2 [4]
---------------------------	--	------------------

OJECTIVE:

- Understand and apply resampling methods for model assessment and selection, including cross-validation and the bootstrap.
- Explore and implement regularization techniques such as ridge and lasso regression for improving model performance.
- Analyze tree-based learning models including classification and regression trees, bagging, random forests, and boosting.
- Gain proficiency in support vector machines and their use in linear and non-linear classification problems.
- Learn key unsupervised learning techniques including PCA, clustering algorithms, ICA, LSI, and Hidden Markov Models.

UNIT 1	Resampling Methods, Model Selection and Regularization: Cross-validation, leave-one-out cross- validation, k-fold cross-validation, the bootstrap, subset selection, shrinkage methods, ridge and lasso regression, dimension reduction methods, principal components regression, partial least square
UNIT 2	Tree Based Methods: Advantages and disadvantages of trees, regression Trees, classification trees, bagging, random forest, boosting
UNIT 3	Support Vector Machine: Maximum margin classifier, classification using a separating hyperplane, the maximal margin classifier, support vector classifier, support vector machines
UNIT 4	classification with non-linear decision boundaries, support vector machine, one-versus-one classification, one-versus- many classification
UNIT 5	Unsupervised Learning: Principle component analysis, what are principal components, clustering methods, k-means clustering, hierarchical clustering, Independent component analysis, latent semantic indexing, Markov Models, Hidden Markov Models

Course Outcome (CO):

At the ends of this course students will have:

CO1: Understand and apply various resampling methods such as cross-validation and the bootstrap for model evaluation.

CO2: Demonstrate knowledge of model selection and regularization techniques including subset selection, ridge regression, and lasso.

CO3: Analyze and implement tree-based methods such as regression trees, classification trees, bagging, random forests, and boosting.

CO4: Apply support vector machine techniques for classification using both linear and non-linear decision boundaries.

CO5: Perform unsupervised learning using principal component analysis, clustering methods, and probabilistic models like Hidden Markov Models.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. "Designing Machine Learning Systems" by Chip Huyen, O'Reilly Media, 1st Edition
2. "Machine Learning Engineering" by Andriy Burkov, True Positive Inc., 1st Edition
3. "Introducing MLOps" by Mark Treveil and Alok Shukla, O'Reilly Media, 1st Edition

Reference Books:

1. "Building Machine Learning Powered Applications" by Emmanuel Ameisen, O'Reilly Media, 1st Edition
2. "Machine Learning Systems: Designs that scale" by Jeff Smith, Manning Publications, 1st Edition

BCO645A	Machine Learning Lab with R, Python, Scikit-learn, Matplotlib, TensorFlow LAB	0-0-2 [2]
----------------	--	------------------

List of Experiments

1.
 - a. Implement k-fold cross-validation
 - b. Evaluate model performance using leave-one-out cross-validation
 - c. Apply the bootstrap method for estimating prediction error
2.
 - a. Perform best subset selection
 - b. Compare forward and backward stepwise selection
 - c. Visualize model complexity vs. error
3.
 - a. Apply ridge regression on a multivariate dataset
 - b. Tune the regularization parameter using cross-validation
 - c. Analyze coefficient shrinkage
4.
 - a. Perform lasso regression
 - b. Identify important features selected by lasso
 - c. Compare ridge and lasso results
5.
 - a. Conduct principal component analysis (PCA)
 - b. Visualize data in reduced dimensions
 - c. Interpret variance explained by each principal component
6.
 - a. Implement principal component regression (PCR)
 - b. Evaluate performance vs. number of components
 - c. Compare PCR with standard regression
7.
 - a. Apply partial least squares (PLS) regression
 - b. Compare PLS with PCR
 - c. Analyze variable importance in projection
8.
 - a. Build a regression tree
 - b. Prune the tree and compare results
 - c. Visualize tree structure
9.
 - a. Construct a classification tree
 - b. Evaluate accuracy with confusion matrix
 - c. Apply tree pruning and analyze improvement
10.
 - a. Implement bagging with decision trees
 - b. Compare with a single tree model
 - c. Plot OOB error
11.
 - a. Apply random forest classifier
 - b. Tune number of trees and max features
 - c. Analyze feature importance
12.
 - a. Use boosting for regression/classification
 - b. Compare performance with bagging and random forest
 - c. Plot training vs. test error

13. a. Implement linear SVM for binary classification
 - b. Tune cost parameter
 - c. Visualize decision boundary
14. a. Use SVM with RBF kernel for non-linear classification
 - b. Perform one-vs-one and one-vs-all classification
 - c. Evaluate performance metrics
15. a. Apply k-means clustering
 - b. Perform hierarchical clustering with dendrogram
 - c. Use PCA before clustering to improve results

Course Outcome (CO):

At the ends of this course students will have:

CO1: Understand and apply various resampling methods such as cross-validation and the bootstrap for model evaluation.

CO2: Demonstrate knowledge of model selection and regularization techniques including subset selection, ridge regression, and lasso.

CO3: Analyze and implement tree-based methods such as regression trees, classification trees, bagging, random forests, and boosting.

CO4: Apply support vector machine techniques for classification using both linear and non-linear decision boundaries.

CO5: Perform unsupervised learning using principal component analysis, clustering methods, and probabilistic models like Hidden Markov Models.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

. SEM-V BCO578A	Neural Networks and Deep Learning	2-0-2 [4]
----------------------------	--	------------------

OJECTIVE:

- To understand TensorFlow fundamentals, architecture, and how tensors operate within it.
- To perform tensor operations, broadcasting, reshaping, and automatic differentiation using TensorFlow.
- To gain conceptual and practical understanding of neural network architecture, activation functions, and training mechanisms.
- To build, train, and optimize neural networks using Keras, applying regularization and hyperparameter tuning techniques.
- To explore and implement deep learning models including CNNs, RNNs, LSTMs, and GANs for real-world applications.

UNIT 1	Tensorflow and Tensors: About TensorFlow, TensorFlow Architecture, TensorFlow 1.x vs 2.x, Setting Up TensorFlow, Introduction to Tensor: Tensor, Real-Word Analogy, Tensor vs. Numpy Array, Tensor Data Types, Ranks, Shape, Attributes
UNIT 2	Tensor Operations & Broadcasting: Element-wise Operation, Basic Arithmetic, Matrix Manipulation, Broadcasting, Tensor Slicing, Indexing, and Reshaping, <code>expand_dims()</code> , <code>squeeze()</code> , TensorFlow Variables, Automatic Differentiation, <code>tf.GradientTape()</code> , Functions and Graphs: <code>@tf.function</code> , eager execution, graph execution
UNIT 3	Neural Networks : What is Neural Network, Neural Networks vs. Traditional Programming, Biological vs Artificial Neurons, Architecture of Neural Networks, Structure of Neural Network, Input Layer, Hidden Layer, Output Layer, Activation Functions, Sigmoid Function, Tanh Function, ReLU, Softmax Function, Forward Propagation, How Forward Propagation Works? Weights and Bias, Nodes and Layers, Loss Functions, MSE, Cross Entropy, Common Loss Functions, Gradient Descent & Learning Rate, Gradients, Local vs Global Minimum, Local Maximum, Saddle Point, Optimizers, SGD, Momentum, Adam Optimizer, RMS Prop, Backpropagation, Chain Rule, Gradients in Backpropagation, Epochs, Batches, Iterations
UNIT 4	Implementing and Training Neural Network : Data Pre-processing, Missing Data, Feature Scaling, Encoding Categorical Data, Shuffle Data, Train, Validation & Test Split, Feature Engineering, Feature Selection, Feature Transformation, Feature Creation, Keras for Neural Networks: Build a Neural Network with Keras, Regularization Techniques, Overfitting, L1 Regularization, L2 Regularization, Dropout, Batch Normalization, Early Stopping, Model Checkpointing, Hyperparameter Tuning, Layers, Neurons per Layer, Learning

	Rate, Batch Size, Grid Search, Random Search, Model Evaluation, Confusion Matrix, Accuracy, Precision, Recall, F1 Score
UNIT 5	Deep Learning : About Deep Learning, CNN Overview, How CNN See Image, CNN Architecture, Convolution, Filter, Activation Function, Pooling, Stride, Padding, Flattening, Softmax Layer, Applications of CNN, RNN, RNN Architecture, Vanishing Gradient Problem, LSTM, How LSTM Works, Foget, Inout, Output Gates, Transfer Learning, GANs (Generative Adversarial Networks), How GANs works, Generator and Discriminator, Real World Applications of Deep Learn

Course Outcome (CO):

At the ends of this course students will have:

CO1: To understand TensorFlow fundamentals, architecture, and how tensors operate within it.

CO2: To perform tensor operations, broadcasting, reshaping, and automatic differentiation using TensorFlow.

CO3: To gain conceptual and practical understanding of neural network architecture, activation functions, and training mechanisms.

CO4: To build, train, and optimize neural networks using Keras, applying regularization and hyperparameter tuning techniques.

CO5: To explore and implement deep learning models including CNNs, RNNs, LSTMs, and GANs for real-world applications.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. *Deep Learning with Python* by François Chollet, Manning Publications, 2nd Edition.
2. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* by Aurélien Géron, O'Reilly Media, 3rd Edition.
3. *Neural Networks and Deep Learning* by Michael Nielsen, Determination Press, 1st Edition.

Reference Books:

1. *Pattern Recognition and Machine Learning* by Christopher M. Bishop, Springer, 1st Edition.
2. *Deep Learning* by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 1st Edition.

BCO581A	Neural Networks and Deep Learning LAB	0-0-2 [2]
----------------	--	------------------

List of Experiments

1. Tensor creation, tensor operations, broadcasting
2. Tensor slicing, indexing, reshaping
3. TensorFlow variable creation, tf.GradientTape usage, auto-differentiation
4. Build a simple neural network, compile model, train with dummy data
5. Apply activation functions: ReLU, Sigmoid, Tanh
6. Loss functions implementation: MSE, Cross Entropy, visualize loss
7. Optimizer comparison: SGD, Adam, RMSprop
8. Train/test split, data normalization, encoding categorical variables
9. Train neural network on MNIST dataset, evaluate performance
10. Apply regularization: L1, L2, Dropout, Early Stopping
11. Confusion matrix, calculate precision, recall, F1 score
12. Design and train a CNN for image classification
13. Build an RNN and LSTM for sequence modeling
14. Use Transfer Learning with pre-trained models (e.g., MobileNet, VGG16)
15. Implement and train a basic GAN (Generator, Discriminator, loss tracking)

Course Outcome (CO):

At the ends of this course students will have:

CO1: To understand TensorFlow fundamentals, architecture, and how tensors operate within it.

CO2: To perform tensor operations, broadcasting, reshaping, and automatic differentiation using TensorFlow.

CO3: To gain conceptual and practical understanding of neural network architecture, activation functions, and training mechanisms.

CO4: To build, train, and optimize neural networks using Keras, applying regularization and hyperparameter tuning techniques.

CO5: To explore and implement deep learning models including CNNs, RNNs, LSTMs, and GANs for real-world applications.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-V BCO334C	Scala for Data Science	2-0-2 [4]
------------------	------------------------	-----------

OJECTIVE:

- To introduce the fundamentals of Scala programming and its relationship with Java.
- To understand variables, data types, conditional statements, and pattern matching in Scala.
- To develop functions, code blocks, and functional programming concepts.
- To explore collections, classes, objects, packages, and exception handling.
- To provide hands-on experience with Scala REPL, SBT, and compiling Scala applications.

UNIT 1	Scala Language: Getting to know Scala programming language, Scala and Java, Statically typed language, Apache Spark and Scala, Scala Performance Benefits, Installing Scala, Using Scala REPL/Shell, getting help from Scala shell, Hello World, Paste mode, retrieving history, auto-complete feature, exiting from Scala REPL
UNIT 2	Variables, Data Types, Conditional Statements: Immutability of variables, define mutable and immutable variables, mutability and type safety, Specifying types for variables, Scala Identifier rules, naming conventions, Scala data types, Boolean types, string type, multiline strings, string operations, string concatenation, string interpolation, length of string, splitting string, extracting part of string, index of character of strings, the ANY type, type casting, Boolean expressions, conditional statement in Scala, nested IF/ELSE statement, pattern matching
UNIT 3	Code Blocks, Functions: Code Blocks in Scala, Why use functions in Scala, understanding functions in Scala, define and invoke a function, functions with multiple parameters, positional parameters, functions with no argument, single-line function, passing function as argument, anonymous function
UNIT 4	Collections: Collections in Scala, Understanding List, list size, convert list to string, iterating over list, map function and collection, foreach, reduce operation, list equality, create set, indexing map, manipulating maps, understanding tuples, indexing tuples, mutable collections, nested collections
UNIT 5	Loops, Packages, Classes and Exceptional Handling: For loop, While loop, Breaking Loop iteration, classes and objects in Scala, Create classes and objects, singleton objects, case classes, equality checks, classes and packages, avoid name space collusion, importing package, fundamental of exception handling, type

	inferences and exception handling, try, catch, finally, Scala built tool (SBT), Compile Scala applications
--	---

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate understanding of Scala fundamentals, REPL usage, and performance benefits.

CO2: Apply variables, data types, conditional logic, and pattern matching in programs.

CO3: Write modular programs using functions, code blocks, and functional arguments.

CO4: Utilize Scala collections, classes, objects, and packages effectively.

CO5: Implement exception handling and build Scala applications using SBT.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. Martin Odersky, Lex Spoon & Bill Venners, *Programming in Scala*, Artima Press, 4th Edition.
2. Cay S. Horstmann, *Scala for the Impatient*, Addison-Wesley, 2nd Edition.
3. Dean Wampler, Alex Payne, *Programming Scala: Scalability = Functional Programming + Objects*, O'Reilly Media, 2nd Edition.

Reference Books:

1. Mark Lewis, *Introduction to Programming and Problem-Solving Using Scala*, CRC Press, 1st Edition.
2. Alvin Alexander, *Scala Cookbook: Recipes for Object-Oriented and Functional Programming*, O'Reilly Media, 2nd Edition.

List of Experiments

1. Getting Started with Scala REPL

Aim: Install Scala, use REPL, run Hello World, explore auto-complete, paste mode, history, and exit commands.

Dataset/Context: Simple Hello World and arithmetic expressions.

Tasks: Setup Scala, run sample programs, use REPL features.

2. Mutable vs Immutable Variables

Aim: Define mutable and immutable variables and explore type safety.

Dataset/Context: Numeric and string variables.

Tasks: Create variables with var and val, type inference, reassign values.

3. Working with Strings

Aim: Perform string operations.

Dataset/Context: User input strings.

Tasks: Concatenation, interpolation, length, splitting, extracting substrings, index of characters.

4.Data Types and Type Casting

Aim: Use Scala data types and casting.

Dataset/Context: Numeric, Boolean, String.

Tasks: Demonstrate ANY type, explicit/implicit casting, Boolean expressions.

5.Conditional Statements & Pattern Matching

Aim: Implement decision-making in Scala.

Dataset/Context: Student marks or grading system.

Tasks: Use if, nested if-else, and match expressions.

6.Code Blocks in Scala

Aim: Understand and use code blocks for grouping logic.

Dataset/Context: Simple arithmetic program.

Tasks: Implement nested blocks, return values from blocks.

7.Defining and Invoking Functions

Aim: Explore functions in Scala.

Dataset/Context: Mathematical functions.

Tasks: Single-line functions, multiple parameters, no-argument functions.

8.Higher-Order and Anonymous Functions

Aim: Use functions as arguments.

Dataset/Context: List operations.

Tasks: Define anonymous functions, pass function as parameter.

9.Working with Lists

Aim: Explore Scala lists.

Dataset/Context: Student names or numbers.

Tasks: Create lists, size, iteration, map, foreach, reduce, equality checks.

10.Sets, Maps, and Tuples

Aim: Work with different Scala collections.

Dataset/Context: Key-value data.

Tasks: Create sets, maps, indexing maps, tuples, and nested collections.

11.Loops in Scala

Aim: Apply iteration constructs.

Dataset/Context: Range of numbers.

Tasks: For loop, while loop, break simulation.

12.Classes and Objects

Aim: Create and use Scala classes.

Dataset/Context: Simple banking or student class.

Tasks: Define classes, instantiate objects, singleton objects.

13.Case Classes & Equality

Aim: Work with case classes.

Dataset/Context: Employee or product dataset.

Tasks: Define case class, compare equality, immutability demonstration.

14.Packages and Imports

Aim: Organize code with packages.

Dataset/Context: Multi-file project.

Tasks: Create packages, import classes, avoid namespace collision.

15.Exception Handling & Scala Build Tool (SBT)

Aim: Handle exceptions and build applications.

Dataset/Context: Division by zero, file I/O.

Tasks: Try-catch-finally, type inference in errors, setup and run project using SBT.

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate understanding of Scala fundamentals, REPL usage, and performance benefits.

CO2: Apply variables, data types, conditional logic, and pattern matching in programs.

CO3: Write modular programs using functions, code blocks, and functional arguments.

CO4: Utilize Scala collections, classes, objects, and packages effectively.

CO5: Implement exception handling and build Scala applications using SBT.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-VI BCO647A	Big Data Analytics	2-0-2 [4]
-------------------	---------------------------	------------------

OJECTIVE:

- To introduce Apache Spark and its ecosystem for big data analytics and distributed computing.
- To understand Spark's core concepts including RDDs, DataFrames, and distributed execution.
- To implement SQL queries and structured data processing using Spark SQL and DataFrames.
- To develop real-time applications using Spark Streaming and various data sources/sinks.
- To apply Spark's MLlib for machine learning, pipelines, and model tuning.

UNIT 1	Apache Spark and Installation: Big Data and Distributing computing at Google, Yahoo, Spark at AMPLab, What is ApacheSpark, Speed, Ease of Use, Modularity, extensibility of Spark, Unified Analytics, Spark SQL, Spark Machine Learning Library (MLLIB), Spark structured streaming, Apache Spark Distributed Execution and concepts, Distributed data and partitions, Who uses Spark for What, Data Science Task, Download Spark, Spark directory and files, Spark application concepts, spark session, Spark UI, build stand-alone application in Spark
UNIT 2	Resilient Distributed Datasets (RDD): Introduction to RDD, RDD Operations, Creating RDDs, Transformations, map, flatMap, filter, union, intersection, substract, distinct, sample, Actions, Working with key/value pair RDD, Data Shuffling
UNIT 3	SQL DataFrames: Spark SQL, SQL Tables and Views, unmanaged and managed tables, create SQL database and tables, create view, reading tables into DataFrame, DataFrameReader, DataFrameWriter, Parquet, JSON, reading JSON file into DataFrame, reading CSV file, reading Avro, ORC, Image file
UNIT 4	Spark Streaming: Evolution of Apache Spark Stream Processing Engine, Micro-batch stream processing, DStreams, philosophy of structured streaming, programming model, Stream DataSource and sink, structured streaming application, streaming DataFrame Operations, joining two streaming DataFrames, working with socket Data Source, Rate Data Source, File Data Source, Kafka Data Source, Custom Data Source, Working with Data Sinks, Kafka Data Sinks, Foreach Data Sinks, Console Data Sinks, Memory Data Sinks, Output modes and Triggers
UNIT 5	Machine Learning with Spark: Spark machine learning library, machine learning pipelines, transformers, Binarizer transformer, Bucketizer transformer,

OneHotEncoder transformer, tokenizer transformer, StopWordsRemover transformer, Estimators, StringIndexer estimator, OneHotEncoderEstimator, Word2Vec Estimator, StandardScaler Estimator, Pipeline, ML Persistence, Model Tuning, Model Hyperparameter, Cross Validator, TrainValidationSplit
--

Course Outcome (CO):

At the ends of this course students will have:

- CO1:** Explain Spark’s architecture, execution model, and ecosystem for big data.
- CO2:** Implement RDD-based transformations and actions for distributed data processing.
- CO3:** Work with Spark SQL and DataFrames to manage structured and semi-structured data.
- CO4:** Build real-time stream processing applications using Spark Structured Streaming.
- CO5:** Apply Spark MLlib for machine learning workflows, pipelines, and model validation.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee, *Learning Spark: Lightning-Fast Big Data Analysis*, O’Reilly Media, 2nd Edition.
2. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, *Learning Spark*, O’Reilly Media, 1st Edition.
3. Bill Chambers, Matei Zaharia, *Spark: The Definitive Guide*, O’Reilly Media, 1st Edition.

Reference Books:

1. Tomasz Drabas, Denny Lee, *Learning PySpark*, Packt Publishing, 1st Edition.
2. Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills, *Advanced Analytics with Spark*, O’Reilly Media, 2nd Edition.

List of Experiments

1. Introduction to Apache Spark

- Aim: Install Spark and explore its directory structure.
- Tasks: Download Spark, configure environment, explore Spark UI, run a sample Spark job.

2. Spark Shell and Spark Session

- Aim: Work with Spark REPL and SparkSession.
- Tasks: Launch Spark shell, create SparkSession, load a text file, count words, explore application logs.

3. Creating RDDs and Basic Transformations

- Aim: Learn RDD creation and transformation operations.
- Tasks: Create RDDs from collections/files, apply map, flatMap, filter, distinct, and union.

4. RDD Actions and Key/Value RDDs

- Aim: Use RDD actions and key-value pair operations.
- Tasks: Apply reduce, count, collect, saveAsTextFile, perform groupByKey, reduceByKey, and join.

5. RDD Data Shuffling

- Aim: Understand data movement during transformations.
- Tasks: Perform shuffle-heavy operations, analyze job DAG in Spark UI, measure performance impact.

6. Working with Spark SQL

- Aim: Create and query SQL tables/views.
- Tasks: Create databases, managed/unmanaged tables, run queries, create temporary and permanent views.

7. DataFrames – Read and Write

- Aim: Use DataFrameReader and DataFrameWriter APIs.
- Tasks: Read/write data in Parquet, JSON, CSV, Avro, ORC, Image formats.

8. DataFrame Transformations

- Aim: Manipulate DataFrames using Spark SQL functions.
- Tasks: Select, filter, aggregate, join, and group data; convert DataFrame to RDD and back.

9. Structured Streaming Basics

- Aim: Work with streaming data in Spark.
- Tasks: Build a structured streaming app with socket data source, apply transformations, print results.

10. Streaming with Multiple Data Sources

- Aim: Integrate various data sources in structured streaming.
- Tasks: Stream from socket, file, rate, and Kafka; apply transformations and joins.

11. Working with Streaming Data Sinks

- Aim: Output structured streaming data.
- Tasks: Write streaming data to console, memory, Kafka, and foreach sinks.

12. Output Modes and Triggers in Streaming

- Aim: Configure streaming outputs.
- Tasks: Implement append, complete, and update output modes with different triggers.

13. Machine Learning with MLlib – Transformers

- Aim: Apply MLlib transformers.
- Tasks: Use Binarizer, Bucketizer, OneHotEncoder, Tokenizer, and StopWordsRemover.

14. Machine Learning with MLlib – Estimators

- Aim: Use MLlib estimators for feature engineering.
- Tasks: Apply StringIndexer, OneHotEncoderEstimator, Word2Vec, and StandardScaler.

15. MLlib Pipelines and Model Tuning

- Aim: Build ML pipelines and tune models.
- Tasks: Construct pipeline with transformers and estimators, perform hyperparameter tuning using CrossValidator and TrainValidationSplit, persist models.

Course Outcome (CO):

At the ends of this course students will have:

- CO1:** Explain Spark’s architecture, execution model, and ecosystem for big data.
- CO2:** Implement RDD-based transformations and actions for distributed data processing.
- CO3:** Work with Spark SQL and DataFrames to manage structured and semi-structured data.
- CO4:** Build real-time stream processing applications using Spark Structured Streaming.
- CO5:** Apply Spark MLlib for machine learning workflows, pipelines, and model validation.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

SEM-VI BCO582A	Natural Language Processing & Generative AI	2-0-2 [4]
---------------------------	--	------------------

OJECTIVE:

- To understand foundational NLP concepts, linguistic structures, and tasks like sentiment analysis, NER, and POS tagging.
- To explore machine learning and deep learning methods including Naive Bayes, SVM, Decision Trees, RNNs, LSTMs, and Transformers.
- To implement an end-to-end NLP pipeline including data collection, preprocessing, modeling, evaluation, and deployment.
- To apply and compare text representation methods such as Bag of Words, TF-IDF, Word2Vec, GloVe, FastText, and contextual embeddings like BERT and ELMo.
- To evaluate large language models and text generation techniques while addressing issues like hallucination, bias, and toxicity using prompt engineering and decoding strategies.

UNIT 1	Natural Language Processing & Learning Approaches : What is NLP? NLP in the real world: assistants, search, chatbots, translation, Key NLP tasks: sentiment analysis, NER, POS tagging, summarization, Linguistic Foundations, Phonemes, Morphemes, Lexemes, Syntax, Semantics, and Context Language ambiguity and complexity Challenges in NLP, Ambiguity in interpretation, Sarcasm, idioms, and figurative language, Language diversity and code-switching, Naive Bayes, Support Vector Machine (SVM), Decision Trees, Deep Learning for NLP Neural Networks, RNNs, LSTMs, GRUs, Convolutional Neural Networks for text, Introduction to Transformers (BERT, GPT), Transfer learning in NLP, Limitations of Deep Learning
UNIT 2	NLP Project Pipeline : The NLP Workflow Define task, Collect data, Preprocess, Model, Evaluate, Deploy, Data Acquisition Using public datasets (e.g., Kaggle, Hugging Face), Web scraping for NLP (BeautifulSoup, newspaper3k), Data Augmentation in NLP, Synonym, replacement, Back translation, TF-IDF-based word substitution ,Bigram flipping, Named Entity perturbation, Text Cleaning & Preprocessing, Removing HTML, Unicode normalization, Lowercasing, punctuation, stopword removal, Spelling correction and noise injection, Tokenization & Segmentation Sentence segmentation, Word tokenization (spaCy, Hugging Face Tokenizers),
UNIT 3	Data Processing : Subword methods: BPE, WordPiece, Text Representation, Bag of Words, TF-IDF Word Embeddings: Word2Vec, GloVe, FastText, Contextual Embeddings: BERT, ELMo, N-gram language models, Masked Language Modeling (MLM), Causal Language Modeling (CLM)
UNIT 4	Deep Dive into Transformers and LLM Architectures : Why Transformers? Self-Attention Mechanism, Multi-Head Attention overview, Encoder vs

	Decoder, Positional Encoding, Residual Connections + Layer Norm, Evolution from Transformer → BERT → GPT, BERT Architecture:, Encoder-only, Masked Language Modeling (MLM), GPT Architecture: Decoder-only, Causal Language Modeling (CLM), Fine-tuning Concepts, Fine-tuning vs Pretraining, Why Fine-tuning is important (adapt LLMs to tasks), Introduction to LoRA / PEFT ideas, Evaluating Language Models + Challenges, Perplexity, BLEU, ROUGE, Issues in LLMs: Hallucination, Bias, Computation C
UNIT 5	Text Generation and Evaluation in Generative AI : How AI generates Text, Autoregressive Generation, Conditional Generation, Strategies for Text Generation, Greedy Search, Beam Search, Top-k sampling, Top-p sampling, Evaluating Generated Text, Human evaluation, Fluency, Coherence, Relevance, Automatic metrics, Perplexity, BLEU Score, ROUGE Score, Bias, Hallucinations, and Toxicity in LLMs, Techniques to Control AI Behavior, Prompt Engineering, Conditioning, Safe Decoding Methods

Course Outcome (CO):

At the ends of this course students will have:

CO1: To understand foundational NLP concepts, linguistic structures, and tasks like sentiment analysis, NER, and POS tagging.

CO2: To explore machine learning and deep learning methods including Naive Bayes, SVM, Decision Trees, RNNs, LSTMs, and Transformers.

CO3: To implement an end-to-end NLP pipeline including data collection, preprocessing, modeling, evaluation, and deployment.

CO4: To apply and compare text representation methods such as Bag of Words, TF-IDF, Word2Vec, GloVe, FastText, and contextual embeddings like BERT and ELMo.

CO5: To evaluate large language models and text generation techniques while addressing issues like hallucination, bias, and toxicity using prompt engineering and decoding strategies.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M

CO5	L	M	H	H	M							H	M	H	H
-----	---	---	---	---	---	--	--	--	--	--	--	---	---	---	---

H = Highly Related; M = Medium L = Low

Text Books:

1. *"Speech and Language Processing" by Daniel Jurafsky and James H. Martin, Pearson, 3rd Edition (Draft)*
2. *"Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper, O'Reilly Media, 1st Edition*
3. *"Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, and Karan Jain, Apress, 1st Edition*

Reference Books:

1. *"Foundations of Statistical Natural Language Processing" by Christopher D. Manning and Hinrich Schütze, MIT Press, 1st Edition*
2. *"Transformers for Natural Language Processing" by Denis Rothman, Packt Publishing, 2nd Edition*

BCO583A	Natural Language Processing & Generative AI LAB	0-0-2 [2]
----------------	--	------------------

List of Experiments

1.
 - a. Load and explore a text dataset (e.g., movie reviews or tweets)
 - b. Perform basic text cleaning (lowercasing, punctuation removal)
 - c. Remove stopwords and apply tokenization
2.
 - a. Implement word tokenization using spaCy and NLTK
 - b. Perform sentence segmentation
 - c. Compare different tokenizers
3.
 - a. Apply Part-of-Speech (POS) tagging using spaCy
 - b. Extract nouns and verbs from text
 - c. Visualize POS distributions
4.
 - a. Perform Named Entity Recognition (NER) using spaCy
 - b. Identify and classify named entities
 - c. Annotate and visualize NER results
5.
 - a. Implement text vectorization using Bag of Words
 - b. Use TF-IDF vectorizer on sample corpus
 - c. Compare output dimensions and sparsity
6.
 - a. Train a Naive Bayes classifier for sentiment analysis
 - b. Evaluate using accuracy and confusion matrix
 - c. Analyze misclassified samples
7.
 - a. Build a Support Vector Machine (SVM) for text classification
 - b. Perform feature selection using TF-IDF
 - c. Test model on unseen data
8.
 - a. Build a decision tree for spam detection
 - b. Visualize the tree structure
 - c. Analyze feature importance
9.
 - a. Train Word2Vec embeddings on custom text
 - b. Visualize word similarities using cosine similarity
 - c. Use embeddings for clustering

10. a. Apply pre-trained GloVe embeddings for sentence modeling
b. Compute average word vectors for sentence representation
c. Compare results with TF-IDF
11. a. Implement a simple RNN for sequence classification
b. Train on short text sequences (e.g., emotion detection)
c. Analyze vanishing gradient issues
12. a. Build and train an LSTM network for text classification
b. Use padding and masking for input sequences
c. Evaluate performance with validation accuracy
13. a. Use a pre-trained BERT model for sentence classification using Hugging Face
b. Fine-tune BERT on a custom dataset
c. Evaluate using accuracy and F1 score
14. a. Implement text generation using GPT-2
b. Generate text with different decoding strategies (greedy, top-k, top-p)
c. Evaluate fluency and coherence
15. a. Perform Web Scraping using BeautifulSoup or newspaper3k
b. Extract and clean textual content
c. Store data for NLP preprocessing pipeline

Course Outcome (CO):

At the ends of this course students will have:

CO1: To understand foundational NLP concepts, linguistic structures, and tasks like sentiment analysis, NER, and POS tagging.

CO2: To explore machine learning and deep learning methods including Naive Bayes, SVM, Decision Trees, RNNs, LSTMs, and Transformers.

CO3: To implement an end-to-end NLP pipeline including data collection, preprocessing, modeling, evaluation, and deployment.

CO4: To apply and compare text representation methods such as Bag of Words, TF-IDF, Word2Vec, GloVe, FastText, and contextual embeddings like BERT and ELMo.

CO5: To evaluate large language models and text generation techniques while addressing issues like hallucination, bias, and toxicity using prompt engineering and decoding strategies.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

. SEM-VII BCO248C	Data Visualization	2-0-2 [4]
------------------------------	---------------------------	------------------

OJECTIVE:

- To introduce data handling, data validation, and visualization techniques.
- To understand statistical methods and business intelligence tools for data analysis.
- To perform advanced data manipulation using Excel functions, charts, and Power BI.
- To apply predictive and statistical models such as regression, correlation, and ANOVA.
- To develop interactive dashboards and reports using Tableau for decision-making.

UNIT 1	INTRODUCTION TO DATA HANDLING Overview of Data analysis, Introduction to Data visualization, Working with statistical formulas - Logical and financial functions , Data Validation & data models, Power Map for visualize data , Power BI-Business Intelligence , Data Analysis using statistical methods, Dashboard designing.
UNIT 2	INTRODUCTION TO DATA MANIPULATION USING FUNCTION: Heat Map, Tree Map, Smart Chart, Azure Machine learning , Column Chart, Line Chart , Pie,Bar, Area, Scatter Chart, Data Series, Axes , Chart Sheet , Trendline , Error Bars, Sparklines, Combination Chart, Gauge, Thermometer Chart , Gantt Chart , Pareto Chart etc , Frequency Distribution, Pivot Chart, Slicers
UNIT 3	Tables: Structured References, Table Styles , What-If Analysis: Data Tables, Goal Seek, Quadratic Equation , Transportation Problem, Maximum Flow Problem, Sensitivity Analysis, Histogram, Descriptive, Statistics, Anova, F-Test, t-Test, Moving, Average, Exponential Smoothing Correlation model Regression model, Practical Lab
UNIT 4	TABLEAU SOFTWARE: GETTING STARTED WITH TABLEAU SOFTWARE: What is Tableau? What does the Tableau product suite comprise of? How Does Tableau Work? Tableau Architecture, What is My Tableau Repository? Connecting to Data & Introduction to data source concepts, Understanding the Tableau workspace, Dimensions and Measures, Data Types & Default Properties, Building basic views, Saving and Sharing your work-overview, Practical Lab
UNIT 5	TABLEAU: BUILDING VIEWS (REPORTS): Date Aggregations and Date parts, Cross tab & Tabular charts, Totals & Subtotals, Bar Charts & Stacked Bars, Trend lines, Forecasting, Filters, Context filters, Line Graphs with Date & Without Date, Tree maps, Scatter Plots

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate knowledge of data handling, validation, and visualization basics.

CO2: Apply Excel-based functions and statistical methods for business analytics.

CO3: Analyze data through advanced charting techniques and predictive modeling.

CO4: Design dashboards using Power BI and Tableau for effective visualization.

CO5: Build and share Tableau reports with forecasting, filters, and trend analysis.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. Wayne Winston, *Data Analysis and Business Modeling with Excel*, Microsoft Press.
2. Daniel G. Murray, *Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software*, Wiley.
3. Chris Albright, Wayne Winston, *Business Analytics: Data Analysis and Decision Making*, Cengage Learning.

Reference Books:

1. George Mount, *Advancing into Analytics: From Excel to Python and R*, O'Reilly.
2. Joshua N. Milligan, *Learning Tableau*, Packt Publishing.

BCO649A	Data Visualization LAB	0-0-2 [2]
---------	------------------------	-----------

List of Experiments

1. Introduction to Data Handling & Visualization

- Load datasets, apply logical & financial functions, and perform data validation.

2. Statistical Data Analysis in Excel

- Apply descriptive statistics, correlation, regression, and hypothesis testing (t-test, F-test, ANOVA).

3. Business Intelligence with Power BI

- Import data, clean dataset, create interactive dashboards, and apply filters.

4. Dashboard Designing in Excel

- Build an interactive dashboard using slicers, pivot charts, and conditional formatting.

5. Power Map & Data Models

- Create 3D Power Maps, work with data models, and visualize geographic data.

6. Data Manipulation using Charts

- Create bar, line, pie, area, scatter charts, and explore data series, axes, and trendlines.

7. Advanced Excel Visualization

- Build Heat Maps, Tree Maps, Smart Charts, and use sparklines for trend analysis.

8. Problem Solving using What-If Analysis

- Solve quadratic equations, transportation problems, and perform sensitivity analysis with Goal Seek and Data Tables.

9. Forecasting using Excel

- Implement moving averages, exponential smoothing, trendlines, and error bars for predictive analytics.

10. Frequency Distribution & Histogram

- Build histograms and frequency distribution tables for large datasets.

11. Correlation & Regression Modeling

- Perform linear regression, multiple regression, and correlation analysis in Excel.

12. Getting Started with Tableau

- Connect to data, explore Tableau workspace, understand dimensions/measures, and

save projects.

13. Tableau – Building Basic Views

- Create bar charts, stacked bars, line graphs (date & non-date), and scatter plots.

14. Tableau – Advanced Visualizations

- Implement treemaps, crosstabs, subtotals, filters, and context filters.

15. Tableau – Forecasting & Trend Analysis

- Apply forecasting models, trendlines, and share interactive Tableau dashboards.

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate knowledge of data handling, validation, and visualization basics.

CO2: Apply Excel-based functions and statistical methods for business analytics.

CO3: Analyze data through advanced charting techniques and predictive modeling.

CO4: Design dashboards using Power BI and Tableau for effective visualization.

CO5: Build and share Tableau reports with forecasting, filters, and trend analysis.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

. SEM-VII BCO651A	Devops for web development	2-0-2 [4]
------------------------------	-----------------------------------	------------------

OJECTIVE:

- To understand the principles, drivers, and benefits of DevOps adoption in modern IT.
- To learn DevOps frameworks including CI/CD, source code management, configuration, and release management.
- To implement Continuous Integration using Jenkins, Git, Maven, and automated testing tools.
- To perform Continuous Delivery & Deployment with tools such as Chef, Ansible, and monitoring frameworks.
- To deploy and manage containerized applications using Docker and Kubernetes for cloud-native environments.

UNIT 1	DevOps Infrastructure: What is DevOps, Implement Continuous Integration (CI), Continuous Delivery (CD), and Continuous Delivery (CD), understand Infrastructure as Code (IaC) practices, Business drivers for DevOps adoption, data explosion, cloud computing, Big data, data science and machine learning, in-memory computing, planning DevOps strategy, benefits of DevOps
UNIT 2	DevOps Framework: DevOps process, Source code management, code review, configuration management, build management, Artifacts repository management, release management, test automation, continuous integration, continuous delivery, continuous deployment, routine automation, DevOps maturity life cycle, DevOps Maturity Map, DevOps progression framework, DevOps Maturity checklists, Agile framework
UNIT 3	DevOps – Continuous Integration: Best Practices for CI/CD, Jenkins setup, Git (SCM) integration with Jenkins, Integrating GitHub with Jenkins, Maven (Build) tool integration with Jenkins, Building Jobs with Jenkins, Source Code Review – Gerrit, Installation of Gerrit, Repository Management, Testing with Jenkins
UNIT 4	Delivery and Deployment: Continuous Delivery – Build Pipeline, DevOps continuous Deployment, Chef landscape components, features of Chef, Chef Automate workflow, Features of Ansible, Ansible CMDB, Playbooks, Modules, Inventory, Plugins, Ansible Tower, Ansible Vault, Ansible Galaxy, Monitoring, Aplunk, Nagios Monitoring Tool
UNIT 5	Containerized Applications with Docker and Kubernetes: Installing Docker, Creating Dockerfile, Building and running a container on a local machine, pushing an image to Docker Hub, managing containers with Kubernetes, Technical requirements of Kubernetes, Kubernetes architecture overview,

	Installing Kubernetes on a local machine, Installing Kubernetes Dashboard, Using HELM as package manager, AKS
--	---

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate understanding of DevOps principles, strategy, and business benefits.

CO2: Apply DevOps frameworks for source control, build, test, and release automation.

CO3: Implement CI/CD pipelines using Jenkins, Git, and Maven.

CO4: Manage configuration and deployment automation using Chef, Ansible, and monitoring tools.

CO5: Deploy containerized applications using Docker and Kubernetes for scalability.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course Outcome	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low

Text Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis, *The DevOps Handbook*, IT Revolution Press.
2. Kief Morris, *Infrastructure as Code: Managing Servers in the Cloud*, O'Reilly.
3. Nigel Poulton, *Docker Deep Dive*, Leanpub.

Reference Books:

1. Jez Humble, David Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Addison-Wesley.
2. Brendan Burns, *Kubernetes Up & Running: Dive into the Future of Infrastructure*, O'Reilly.

List of Experiments

1. Introduction to DevOps

- Setup a DevOps-ready environment, explore CI/CD concepts, and understand business drivers for DevOps adoption.

2. Infrastructure as Code (IaC)

- Write and execute simple IaC scripts, provision resources automatically, and manage configuration.

3. Source Code Management with Git

- Create repositories, branches, commits, and perform merges & pull requests.

4. Integrating Git with Jenkins

- Install Jenkins, connect it with GitHub, and automate build triggers.

5. Build Automation with Maven

- Integrate Maven with Jenkins to build and package Java applications.

6. Code Review using Gerrit

- Install Gerrit, configure repositories, and perform collaborative code reviews.

7. Continuous Integration Pipeline

- Build a CI pipeline in Jenkins including build, test, and artifact generation.

8. Continuous Delivery with Jenkins

- Design a Jenkins build pipeline with automated testing and staged deployments.

9. Configuration Management with Chef

- Install Chef, create cookbooks, manage nodes, and automate configuration.

10. Configuration Management with Ansible

- Write playbooks, manage inventories, and deploy applications using Ansible Tower.

11. Monitoring with Nagios and Splunk

- Configure Nagios to monitor system health and set up Splunk dashboards for log monitoring.

12. Docker Basics

- Install Docker, create Dockerfiles, build images, run containers, and push to Docker Hub.

13. Multi-Container Applications with Docker

- Use Docker Compose to define and manage multi-container applications.

14. Kubernetes Basics

- Install Kubernetes locally, explore pods, deployments, services, and manage workloads.

15. Kubernetes Advanced Deployment

- Install Kubernetes Dashboard, deploy apps using HELM, and explore AKS (Azure Kubernetes Service).

Course Outcome (CO):

At the ends of this course students will have:

CO1: Demonstrate understanding of DevOps principles, strategy, and business benefits.

CO2: Apply DevOps frameworks for source control, build, test, and release automation.

CO3: Implement CI/CD pipelines using Jenkins, Git, and Maven.

CO4: Manage configuration and deployment automation using Chef, Ansible, and monitoring tools.

CO5: Deploy containerized applications using Docker and Kubernetes for scalability.

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

<i>Course Outcome</i>	Program Outcome												Program Specific Outcome		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	H	H	M	M	H							L	H	M	
CO2	H	H	M	M	L				L			M	M	H	
CO3	M	H	H	L	M	L						H	M	M	M
CO4	M	H	M	L	H				M			M	L	M	M
CO5	L	M	H	H	M							H	M	H	H

H = Highly Related; M = Medium L = Low