

Curriculum for the Graduate Program
Session: 2024-2025
MSc in Data Science (Professional)
Offered by: Department of Statistics (in collaboration with CSE and MAT)
Shahjalal University of Science & Technology (SUST), Sylhet

Part A

1.	Title of the Academic Program:	Graduate Program
2.	Name of the University:	Shahjalal University of Science and Technology, Sylhet
3.	Vision of the University:	To be a leading university of excellence in Science and Technology with a strong national commitment and significant international impact.
4.	Mission of the University:	UM1: To advance learning and knowledge through teaching and research in science and technology. UM2: To serve as a center for knowledge creation, technological innovation and transfer among academia, industry, and society. UM3: To assist in transferring Bangladesh a country with sustainable economic growth and equitable social development.
5.	Name of the Program Offering Entity:	Department of Statistics
6.	Vision of the Department:	Evolving expertise in statistics to serve nationally and internationally
7.	Mission of the Department:	M1: Achieve excellence and expansion of knowledge in Statistics as well as in data science M2: Maintain the quality of teaching and research at international standard M3: Collaborate with stakeholders for planning, statistical analysis and research M4: Promote and tailor research for taking appropriate decision to achieve development goals of the country.
8.	Objectives of the Department:	a. Disseminate fundamental and advance statistical knowledge to adopt and validate data analysis technique b. Contribute to the theoretical and practical development of statistical methods addressing substantive problems c. Organizing workshops, seminars, conferences with stakeholders to improve quality of research and implement the statistical techniques d. Provide adequate and relevant guidelines of statistics to make planning and decision making for achieving Sustainable Development Goals of Bangladesh
9.	Name of the Degree:	MSc in Data Science (Professional) Nature: Master's by Coursework
10.	Description of the Program:	Shahjalal University of Science and Technology (SUST), located in Sylhet, was established in 1986 as the first science and technology university in Bangladesh. The Department of Statistics, under the School of Physical Sciences, is one of the founding departments of the university and was established in 1990. It began its academic journey in the 1991–92 session with a Bachelor of Science (B.Sc.) Honours program in Statistics, followed by the introduction of graduate-level programs starting from the 1994–95 session. In line with the evolving demands of modern education and global standards, the department revised its curriculum and introduced an Outcome-Based Education (OBE) framework beginning in the 2022–2023 academic session. This curriculum was developed in accordance with the Bangladesh National Qualifications Framework (BNQF) for Higher Education and the Semester System Ordinance of the university, with support from the Institutional Quality Assurance Cell (IQAC), SUST.

		<p>The MSc in Data Science (Professional) program represents an advanced academic pursuit designed for students who have successfully completed a bachelor degree. This program aims to provide a comprehensive and interdisciplinary foundation in data science, combining rigorous theoretical learning with hands-on applications. It prepares students to thrive in data-intensive environments by equipping them with the analytical, statistical, and computational tools necessary to solve real-world problems across diverse sectors.</p>
11.	Rationale of the Program:	<p>Data science is reshaping the modern world, driving transformative advancements in business, healthcare, finance, technology, and beyond. As data becomes the new currency of innovation, the demand for professionals skilled in extracting meaningful insights through statistical modeling, machine learning, artificial intelligence (AI), and neural networks (NN) continues to rise. At its core, data science is grounded in statistics, mathematics, and computational science—making proficiency in these disciplines essential.</p> <p>In response to this growing demand, the MSc in Data Science program at Shahjalal University of Science and Technology (SUST) is thoughtfully designed to bridge theory with practical application. The program integrates statistical methodologies, advanced computational techniques, and real-world problem-solving to equip students with the skills needed to address complex data challenges.</p> <p>Graduates of this program will be empowered to drive innovation, support data-driven decision-making, and contribute meaningfully across various industries in the age of AI and Big Data. Upon completion, students are expected to apply their theoretical knowledge and analytical reasoning to make impactful contributions in planning, analysis, and decision-making at both national and global levels.</p>
12.	Structure of the Program:	<p>The MSc in Data Science (Professional) is a coursework-based graduate program designed to be completed in a minimum of 1 year (2 semesters) and a maximum of 2 years (4 semesters). The program will be offered in each academic year with an annual intake of 40 students.</p> <p>The total requirement for graduation is 40 credits, distributed over two semesters:</p> <ul style="list-style-type: none"> • Semester I: 20 credits • Semester II: 20 credits <p>Each semester is of 6 months’ duration, comprising theory, lab, and project-based coursework. Students must complete all courses offered in both semesters to qualify for the degree. The course codes and structure are developed in accordance with the Bangladesh National Qualifications Framework (BNQF) and follow the university’s Semester System Ordinance.</p> <p>Courses are classified into the following categories:</p> <ul style="list-style-type: none"> • Core Courses • General Education Courses • Elective or Optional Courses <p>This structured approach ensures a comprehensive and rigorous foundation in data science, while maintaining flexibility and adherence to national academic standards.</p>
13.	Admission Requirements:	<p>A1.1 Applicants must have a Bachelor’s degree (4 years/3 years or equivalent) with a CGPA of at least 3.0 on a 4.0 scale or a 2nd class equivalent in any academic discipline, with a strong foundation in Mathematics and basic programming knowledge. Moreover, applicants must have a science background at the SSC and HSC levels. Furthermore, applicants must take an admission test (MCQ and/or written). The top 40 students will be selected based on their admission test scores and Bachelor’s program CGPA.</p> <p>A2.1 The candidate for this program must submit the following documents: (i) application in the prescribed form, (ii) academic transcript, and (iii) consent letter from the employer, if applicable.</p> <p>A2.2 After selecting the candidate for this program, the Coordination Committee will forward the recommendation to the Graduate Study Committee, which will then send all the documents mentioned in Clause A2.1 to the Board</p>

		<p>of Advanced Studies through the Dean of the school, for subsequent approval by the Academic Council.</p> <p>A3.1 A student enrolled in this program must register for a minimum of 10 (ten) credits and a maximum of 20 (twenty) credits per semester.</p> <p>A3.2. A SUST faculty member may be admitted to this program with prior approval from the University Authority.</p> <p>A3.3 The registration for this program will remain valid for a maximum of 4 (four) semesters.</p> <p>A3.4 The period of candidature for this program will remain valid for a maximum of 2 (two) academic years.</p> <p>A6.1 Duration: Minimum duration for this degree is 2 (two) semesters.</p> <p>A6.2 Credit Requirement: To attain the Master's by Coursework degree, students are required to fulfill a minimum of 40 credits through coursework instruction including project.</p> <p>A6.3 Course Requirement: Students enrolled in the Master's by Coursework program must complete a minimum of 40 credit hours of instructional coursework. The GSC may recommend required courses, but not more than 12 (twelve) credits, at the graduate and/or undergraduate levels from other disciplines.</p>
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

14. Grading/Evaluation: According to the ordinance of the university

1) Grading Scale and

2) Grades:

Letter Grade and corresponding Grade-Point for a course will be awarded from the roundup marks of individual courses as follows:

Numerical Grade	Letter Grade	Grade Point
80% and above	A+	4.00
75% to less than 80%	A	3.75
70% to less than 75%	A-	3.50
65% to less than 70%	B+	3.25
60% to less than 65%	B	3.00
55% to less than 60%	B-	2.75
50% to less than 55%	C+	2.50
45% to less than 50%	C	2.25
40% to less than 45%	C-	2.00
Less than 40%	F	0.00

3) Grade Point Average (GPA) and Cumulative Grade Point Average (CGPA)

GPA:

Grade Point Average (GPA) is the weighted average of the grade points obtained in all the courses completed by a student in a semester.

CGPA:

Cumulative Grade Point Average (CGPA) of major and second major degrees will be calculated separately by the weighted average of all courses of the previous semesters along with that of the current/present semester. For the calculation of the final CGPA of clearing graduates, if the third digit after the decimal point is nonzero then its previous, that is, the second digit will be incremented by one. A student, if applicable, will also receive a separate CGPA for her/his Second Major courses.

F Grades:

A student will be given an “F” grade if s/he fails or remains absent in the final examination of a registered course. If a student obtains an “F” grade, her/his grade will not be counted for GPA and s/he will have to repeat the course. An “F” grade will be in her/his record, and s/he will not be eligible for distinction, award, and scholarship of the university.

4) Incomplete courses

If a student has an incomplete course(s), s/he has to register such an incomplete course(s) from preceding semesters before registering courses from current or successive semesters. If an incomplete course is not available or offered in the running semesters, the student shall take such course(s) when it is available or offered.

5) Course Withdrawal

A student can withdraw a course by a written application to the Controller of Examinations through the Head of the discipline two weeks before the examination start. The Controller of Examinations will send the revised registration list to the disciplines before the commencement of semester final examination. There will be no record of the course in transcript if the course is withdrawn.

6) Repetition

If a student has to repeat a failed or incomplete course and that course is not available/offered any more, the discipline may allow him/her to take an equivalent course from the current curriculum. For clearing graduates, if any incomplete course is not available/offered in the running semester, the discipline may suggest a suitable/equivalent course to complete the credit requirement so required for the degree.

15. Attendance /class performance/marks distribution

15.1 Distribution of Marks (Theory Courses):

The marks of a given course will be as follows:

S. No.	Component	Marks (%)
1.	Class Attendance	10%
2.	Class performance (Quizzes/MCQ/fill in the gap/report writing/presentation/Assignments)	10%
3.	Mid-Semester Examinations	20%
4.	Final Examination	60%

Note: A student must obtain at least 25% of marks allocated to final examination to pass the course.

15.2 Class participation:

The marks for class participation will be as follows:

Attendance Percentage	≥ 95	90 – < 95	85 – < 90	80 – < 85	75 – < 80	70 – < 75	65 – < 70	60 – < 65	50 – < 60
Marks	10	9	8	7	6	5	4	3	0

16. Graduate Attributes (based on need assessment)

Code	Graduate Attributes	Domain
GA 01	Demonstrate advanced knowledge of statistical, computational, and machine learning techniques.	Fundamental skills
GA 02	Apply interdisciplinary knowledge from fields such as computer science, mathematics, and domain-specific areas in solving data-driven problems.	Fundamental skills
GA 03	Effectively communicate complex data science concepts, insights, and findings to both technical and non-technical audiences using appropriate tools and platforms.	Social skills
GA 04	Understand with the application of data science solutions in the context of social impact, ethical responsibility, and environmental sustainability at local and global levels.	Social skills
GA 05	Analyze alongwith the interpretation of large-scale, complex data using advanced statistical and algorithmic methods to support data-driven decisions.	Thinking skills
GA 06	Demonstrate problem-solving and decision-making skills in uncertain, real-world data scenarios using appropriate modeling and computational techniques.	Thinking skills
GA 07	Engage in continuous learning to adapt evolving technologies and methods in data science for academic and professional growth.	Personal skills
GA 08	Exhibit ethical reasoning, responsible AI practices, and accountability in handling sensitive data and conducting research projects.	Personal skills

17. Program Educational Objectives (PEOs):

PEO1	To provide advanced and interdisciplinary knowledge in data science, integrating statistics, machine learning, artificial intelligence, and computational techniques.
PEO2	To promote innovative thinking and critical analysis for solving complex, real-world problems using data-driven approaches.
PEO3	To guide students in designing and implementing advanced research methodologies in emerging areas of data science, including big data analytics, deep learning, and responsible AI.
PEO4	To foster leadership, collaboration, and communication skills that support data-driven decision-making in diverse organizational and societal contexts.
PEO5	To prepare students for continuous professional development to contribute in national and global advancement through ethical and impactful data science practices.

18. Program Learning Outcomes (POs): After successful completion of the program, the graduates are expected to come up with the ability to-

POs	Program Learning Outcomes	Domain
PO1	Apply advanced data science methods—including statistical analysis, machine learning, and computational modeling—to solve real-world problems across various domains.	Fundamental Skills
PO2	Utilize data-driven approaches to address complex societal issues and contribute to evidence-based policy and decision-making at national and global levels.	Social Skills
PO3	Analyze alongwith the interpretation of large-scale and complex datasets using modern data science tools and techniques.	Thinking Skills
PO4	Tackle multifaceted problems in academic, industrial, and technological settings by integrating critical thinking and analytical reasoning.	Thinking Skills
PO5	Conduct independent and collaborative research on contemporary issues in data science, while upholding ethical, legal, and professional standards.	Personal Skills
PO6	Demonstrate a commitment to lifelong learning and professional growth, and exhibit leadership in data-driven innovation.	Personal Skills

19. Mapping mission of the university with PEOs

PEOs	UM1	UM2	UM 3
PEO 1	√	√	
PEO 2	√	√	
PEO 3	√	√	
PEO 4		√	√
PEO 5		√	√

20. Mapping POs with the PEOs

	PEO 1	PEO 2	PEO 3	PEO 4	PEO 5
PO 1	√	√	√		
PO 2				√	√
PO 3		√	√		
PO 4		√	√		
PO 5			√	√	√
PO 6			√	√	√

21. Mapping courses with the POs

Courses	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6
STA 0542 5181	√	√	√			

MAT 0541 5102	√	√	√			
STA 0542 5182	√	√	√	√		
STA 0542 5183	√	√	√	√		√
STA 0542 5184	√	√	√	√		√
CSE 0613 5185	√	√	√		√	
CSE 0613 5186						
STA 0542 5187	√	√	√		√	
STA 0542 5188						
STA 0542 5189	√	√		√	√	√
STA 0542 5281	√	√		√	√	√
MAT 0541 5203	√	√	√	√	√	√
MAT 0541 5204						
CSE 0612 5283						
CSE 0612 5284	√	√	√	√		
CSE 0612 5286	√	√	√	√		
CSE 0613 5287	√	√	√	√	√	
CSE 0613 5288						
STA 0542 5290	√	√	√			
STA 0542 5292	√	√	√	√	√	√

STA: Statistics; GED: General Education; SPS: School of Physical Sciences; 0542 indicates Statistical courses;

Semester 1 (20 Credits)

Code (odd theory & even lab)	Course Title	Credits (Theory)	Credits (Lab)	Teacher	Course Category
STA 0542 5181	Probability and Statistical Methods	3	0	STA	Core
MAT 0541 5102	Mathematics for Data Science	2	0	MAT	GED
STA 0542 5182	Advanced Programming with Python	0	2	CSE	Core
STA 0542 5183	Artificial Intelligence and Machine Learning	3	0	STA	Core
STA 0542 5184	Artificial Intelligence and Machine Learning Lab	0	1.5	STA	Core
CSE 0613 5185	Advanced Data Structures and Algorithms	2	0	CSE	Core
CSE 0613 5186	Advanced Data Structures and Algorithms Lab	0	1.5	CSE	Core
STA 0542 5187	High-Dimensional Data Learning and Visualization	2	0	STA	Core
STA 0542 5188	High-Dimensional Data Learning and Visualization Lab	0	1.5	STA	Core
STA 0542 5189	Project-I	0	1.5	STA	Core
Total		12	8		

Semester 2 (20 Credits)

Code (odd theory & even lab)	Course Title	Credits (Theory)	Credits (Lab)	Teacher	Course Category
STA 0542 5281	Statistical Inference and Modeling	2	0	STA	Core
MAT 0541 5203	Optimization and Numerical Analysis	2	0	MAT	GED
MAT 0541 5204	Optimization and Numerical Analysis Lab	0	1	MAT	GED
CSE 0612 5283	Database Management for Data Science	3	0	CSE	Core
CSE 0612 5284	Database Management for Data Science Lab	0	1.5	CSE	Core
CSE 0612 5286	Cloud-based Big Data Analytics	0	3	CSE	Core
CSE 0613 5287	Deep Learning	3	0	CSE/STA	Core
CSE 0613 5288	Deep Learning Lab	0	1.5	CSE/STA	Core
STA 0542 5290	Professional Skills and Data Science Ethics	0	1	STA	Core
STA 0542 5292	Project-II	0	2	STA/CSE/MAT	Core
Total		10	10		

Detail Course Contents

Semester 1

Code	Course Title	Credits (Theory)	Teacher	Course Category
STA 0542 5181	Probability and Statistical Methods	3	STA	Core

Rationale:

The Probability and Statistical Methods course is designed to provide students with essential statistical knowledge and analytical skills, which form the cornerstone of data science. The course introduces core principles of statistics and probability, and their application using R/Python programming, offering students the theoretical and practical foundation necessary for analyzing data, making inferences, and interpreting results in the field of data science. This course is crucial for students pursuing advanced topics in data science, machine learning, and artificial intelligence.

Course Objectives:

By the end of the course, students will:

1. Understand and apply the fundamental principles of probability theory.
2. Obtain knowledge on probability distribution and sampling distribution.
3. Understand the basic concepts of statistics and its role in data analysis.
4. Gain knowledge of different types of variables and methods for summarizing data.
5. Be able to compute and interpret descriptive statistics.
6. Learn about different sampling techniques and their application in data science.
7. Develop skills in inferential statistics, including hypothesis testing and confidence intervals.
8. Utilize R/Python programming for performing data analysis and statistical computations.

Course Content:

1. Basic Concepts of Statistics and Presentation of Data: Population, sample, data and variable, Types of data and variables (qualitative and quantitative), Scales of measurement (nominal, ordinal, interval, ratio), Organizing data: classification, tabulation and frequency distribution, Presenting data through graphs and charts: Bar charts, pie charts, line diagram, histograms, boxplots, and scatterplots, Choosing the right graph for different data types. **2. Descriptive Statistics:** Measures of central tendency, Measures of dispersion, Moments, Skewness, Kurtosis, and Measures of Association. **3. Probability:** Basic probability concepts, Random variables and probability distributions (Discrete and Continuous), Mathematical expectation, Joint, conditional, and marginal probability, Bayes' theorem, Knowledge on Binomial, Poisson, Negative Binomial, Normal, Exponential, Gamma, and Beta distributions. **4. Sampling and Sampling Distribution:** Concepts related to survey and sampling, Types of sampling techniques: Simple random sampling, stratified sampling, systematic sampling, cluster sampling, multi-stage sampling. **5. Theory of Statistics:** Sampling distribution (t, F, Chi-square), Central Limit Theorem, Point and interval estimation (Confidence Intervals), Basic hypothesis testing: z-test, t-tests, chi-square tests, and F-test. **6. Correlation and Regression Analysis, Analysis of Variance (ANOVA) and Design of Experiment:** Types of correlation, Correlation coefficient, Rank correlation, Simple and Multiple Regression models with inferential procedures, Basic concepts of ANOVA and Design of Experiments (e.g., CRD, RBD, LSD etc.).

Course Learning Outcomes (COs):

By the end of this course, students will be able to:

CO1: Describe with the application of basic statistical concepts, including frequency distributions and visualize data through various graphs.

CO2: Determine the different measures of central tendency, dispersion, skewness and kurtosis, simple correlation and regression, along with their merits and demerits.

CO3: Apply basic probability concepts including probability distributions in the context of data science and conduct hypothesis testing to derive meaningful inferences from data.

CO4: Implement various sampling techniques for data collection and analysis of data by using correlation, regression and ANOVA including design of experiments

CO5: Apply statistical methods using R/Python.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2			
CO2	3		2			
CO3	3		2			
CO4	3		2			
CO5	3		2			

Mapping COs with Teaching-Learning & Assessment Strategy:

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures, case studies, interactive discussions, projector and Software (R/Python)	Quiz/ Assignment/ Presentation (Individual/group) Midterm Examination 1 & 2 Semester-end examination
CO2		
CO3		
CO4		
CO5		

Textbook References:

1. Newbold, P., Carlson, W. L., & Thorne, B. (2013). *Statistics for business and economics* (8th ed.). Pearson.
2. Witte, R. S., & Witte, J. S. (2017). *Statistics* (11th ed.). Wiley.
3. Moore, D. S., McCabe, G. P., & Craig, B. A. (2020). *Introduction to the practice of statistics* (9th ed.). Macmillan Learning.
4. Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2016). *Probability and statistics for engineers and scientists* (9th ed.). Pearson.
5. Montgomery, D. C. (2017). *Design and analysis of experiments* (9th ed.). Wiley.

Code	Course Title	Credits (Theory)	Teacher	Course Category
MAT 0541 5102	Mathematics for Data Science	2	MAT	GED

Rationale:

This course introduces essential mathematical tools required for data science, focusing on linear algebra and calculus concepts that underpin algorithms, data modeling, and machine learning techniques.

Course Objectives:

By the end of this course, student will:

1. Develop strong foundation in linear algebra for representing and transforming data.
2. Gain the ability to apply calculus concepts in model development and analysis.
3. Be able to apply mathematical reasoning to data science problems.

Course Content:

1. Linear Algebra: Vectors and vector spaces: concepts, operations, basis, dimension, Matrices and matrix operations: addition, multiplication, inverse, rank, determinants, Systems of linear equations: Gaussian elimination, consistency and solution types, Eigenvalues and eigenvectors, diagonalization, singular value decomposition (SVD), Applications in data science: Principal Component Analysis (PCA), dimensionality reduction. **2. Calculus:** Functions, limits, and continuity, Differentiation: rules, higher-order derivatives,

Multivariable calculus: partial derivatives, gradients, Hessian matrix, Integration: definite and indefinite integrals, basic techniques, Applications in data science: gradient-based transformations, behavior of functions.

Course Learning Outcomes (COs):

By the end of this course, students will be able to:

CO1: Understand and apply linear algebra concepts such as vectors and matrices.

CO2: Analyze and solve systems of linear equations using matrix operations.

CO3: Perform differential and integral calculus operations relevant to data modeling.

CO4: Interpret multivariable calculus results for applications in data science.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2		2			
CO2	3		3	1		
CO3	3		2	2		
CO4	2		3	1		

Mapping COs with Teaching-Learning & Assessment Strategy

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures with matrix manipulation demos	Midterm & Semester-end Examinations
CO2	Hands-on problem-solving sessions	Assignments, Quizzes
CO3	Practical examples and illustrations from data science	Semester-end Examination
CO4	Lecture with illustrative applications	Assignments, Midterm Exam

Textbook References:

1. Lay, D. C., Lay, S. R., & McDonald, J. J. (2015). *Linear algebra and its applications* (5th ed.). Pearson.
2. Strang, G. (2016). *Introduction to linear algebra* (5th ed.). Wellesley-Cambridge Press.
3. Kreyszig, E. (2011). *Advanced engineering mathematics* (10th ed.). Wiley.
4. Stewart, J. (2015). *Calculus: Early transcendentals* (8th ed.). Cengage Learning.
5. Rogawski, J., & Adams, C. (2015). *Multivariable calculus* (3rd ed.). W. H. Freeman.

Code	Course Title	Credits (Lab)	Teacher	Course Category
STA 0542 5182	Advanced Programming with Python	2	CSE/STA	Core

Rationale:

This course provides hands-on training in programming languages crucial for data science, with a particular focus on Python. It is designed to equip students with the skills necessary to perform data analysis, manipulation, and visualization tasks using Python, a key tool in the field of data science.

Course Objectives:

By the end of the course, students will:

1. Gain proficiency in Python programming, with an emphasis on applications in data science.
2. Implement practical data analysis workflows using Python tools and libraries.
3. Optimize code performance and efficiency for large-scale data science applications.

4. Visualize and interpret data using Python's visualization libraries.
5. Apply statistical models and perform regression analysis using Python.
6. Use simulation techniques and hypothesis testing to analyze real-world datasets.
7. Develop a deeper understanding of experimental designs and their applications in data science.

Course Content:

The course begins with an **Overview of Python**, providing an introduction to Python, comparing its syntax with other programming languages, and highlighting its applications in data science. Students will then explore **Basic Operations & Data Structures**, covering essential Python structures such as lists, tuples, dictionaries, arrays, and data frames. They will learn techniques for data import, export, and manipulation. The course will continue with **Data Management & Processing**, focusing on handling missing values, cleaning data, grouping, filtering, merging, and reshaping data for analysis. Students will also delve into **Loops, Conditional Execution & Functions**, learning how to implement iteration methods, conditional statements, and user-defined functions to enhance the efficiency of their coding. In the **Summary Statistics & Data Visualization Techniques** section, students will compute measures of central tendency and dispersion while mastering data visualization techniques such as XY plots, bar charts, histograms, and box plots. **Statistical Models & Regression Analysis** will cover the development and evaluation of regression models, including linear, logistic, and polynomial regression, ANOVA, model diagnostics, and comparisons. **Fitting Distributions & Simulation** introduces probability distributions, hypothesis testing, Monte Carlo simulations, and applying these techniques to real-world case studies. The course concludes with an examination of **Analysis of Experimental Designs**, where students will study randomized block designs, treatment comparisons, non-orthogonal designs, and split-plot analysis for data-driven research. Throughout the course, students will also be introduced to important **Data Science Libraries and Frameworks** in Python, such as NumPy, Pandas, Matplotlib, and SciPy.

Course Learning Outcomes (COs):

By the end of this course, students will be able to:

- CO1:** Gain proficiency in Python programming.
- CO2:** Implement data analysis workflows using Python.
- CO3:** Optimize performance in data science applications.
- CO4:** Visualize and analyze data using Python.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	2	2			
CO2	3	2	2	2	2	
CO3	3	3	3	2	2	
CO4	2	2	3			

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy:

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures and coding assignments	Midterm Examination, Project
CO2	Hands-on labs and projects	Project-based assessment, Semester-end examination
CO3	Coding exercises and optimization tasks	Project-based assessment
CO4	Practical data visualization sessions	Project, Semester-end examination

Textbook References:

1. McKinney, W. (2018). *Python for data analysis*. O'Reilly Media.
2. Molin, S. (2021). *Hands-on data analysis with pandas*. Packt Publishing.
3. Grus, J. (2019). *Data science from scratch*. O'Reilly Media.
4. VanderPlas, J. (2016). *Python data science handbook*. O'Reilly Media.
5. Yau, N. (2016). *Practical data science with Python*. No Starch Press.
6. Sweigart, A. (2019). *Automate the boring stuff with Python*. No Starch Press.

Code	Course Title	Credits (Theory)	Teacher	Course Category
STA 0542 5183	Artificial Intelligence and Machine Learning	3	STA	Core

Rationale:

This course provides an introduction to machine learning algorithms and their applications in data analysis.

Course Objectives:

By the end of the course, students will:

1. Understand foundational concepts of AI and ML, their applications, and ethical considerations.
2. Apply supervised and unsupervised learning techniques to solve real-world problems.
3. Utilize ensemble methods, model optimization, and regularization strategies for improved performance.
4. Evaluate and interpret model performance using standard metrics and validation techniques.
5. Implement machine learning models in Python through hands-on projects and case studies.

Course Content:

1. Introduction to Artificial Intelligence and Machine Learning: Fundamental concepts of artificial intelligence, history and evolution of AI, key applications such as natural language processing, computer vision, robotics, and expert systems, ethical and societal implications of AI, distinctions and overlap between AI and machine learning, Fundamental concepts of machine learning, supervised, semi-supervised, and unsupervised learning, reinforcement learning, exploring real-world applications, and an introduction to Python programming for machine learning. **2. Supervised Learning:** Techniques including linear regression, logistic regression, decision trees, random forest, k-nearest neighbors (k-NN), and support vector machines (SVM). **3. Ensemble Learning and Model Optimization:** Methods such as bagging (e.g., Random Forest), boosting (e.g., Gradient Boost, AdaBoost, XGBoost), hyperparameter tuning, cross-validation, and regularization techniques like Lasso and Ridge Regression, hybrid learning. **4. Unsupervised Learning:** Approaches including clustering methods (K-Means, hierarchical clustering, DBSCAN), dimensionality reduction techniques (PCA, t-SNE), and an introduction to recommender systems. **5. Model Evaluation and Practical Implementation:** Understanding performance metrics (accuracy, precision, recall, F1-score), addressing the bias-variance tradeoff, strategies for handling overfitting, and hands-on projects utilizing Python libraries such as Scikit-learn, NumPy, and Pandas.

Course Learning Outcomes (COs):

By the end of the course, students will be expected to:

- CO1:** Understand key concepts and algorithms in machine learning.
- CO2:** Implement supervised and unsupervised learning models.
- CO3:** Evaluate and assess model performance using various metrics.
- CO4:** Gain practical experience in machine learning implementation using Python.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong

2: Moderate

1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	2	2			
CO2	3	3	3	2	2	
CO3	3	2	3	2	2	
CO4	2	2	3	2	3	

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures with case studies and examples	Midterm Examination, Semester-end examination
CO2	Practical implementation and coding labs	Lab assignments, Project
CO3	Hands-on evaluation tasks, discussions	Midterm Examination, Semester-end examination
CO4	Practical coding exercises and problem-solving	Lab project, Semester-end examination

Textbook References:

1. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
3. Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
4. Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
6. Burkov, A. (2019). *The hundred-page machine learning book*. Andriy Burkov.
7. Ng, A. (n.d.). *Machine learning yearning: Technical strategy for AI engineers, in the era of deep learning*. deeplearning.ai. <https://www.deeplearning.ai/machine-learning-yearning/>

Code	Course Title	Credits (Lab)	Teacher	Course Category
STA 0542 5184	Artificial Intelligence and Machine Learning Lab	1.5	STA	Core

Rationale:

This lab course aims to provide practical experience in implementing machine learning algorithms, model optimization, and real-world applications using Python. Students will develop hands-on skills in supervised, unsupervised learning, ensemble methods, model evaluation, and basic NLP techniques. This course will also introduce students to essential machine learning libraries like Scikit-learn, NumPy, and Pandas for data preprocessing and analysis.

Course Objectives:

By the end of this lab course, students will be able to:

1. Understand and apply the fundamental concepts of machine learning and its real-world applications.
2. Implement and evaluate supervised and unsupervised learning algorithms.
3. Optimize machine learning models using ensemble methods and hyperparameter tuning.
4. Evaluate model performance using key metrics and address issues like overfitting.
5. Apply basic natural language processing (NLP) techniques to text data.

Lab Content:

1. Introduction to AI and ML Tools: Students install and set up the Python programming environment, Jupyter Notebook, and essential libraries such as NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow/Keras. They practice writing Python scripts for data preprocessing, manipulation, and visualization. This module includes explicit hands-on exercises distinguishing supervised and unsupervised learning tasks using real-world datasets, providing a clear understanding of foundational machine learning concepts. **2. Supervised Learning Practicals:** Students implement linear regression and logistic regression models, build and visualize decision trees, train random forest classifiers, apply k-nearest neighbors (k-NN) algorithms, and use support vector machines (SVM). Special attention is given to exploring the impact of different kernel functions on SVM performance through targeted experiments. **3. Ensemble Learning & Model Tuning:** Students implement bagging techniques such as Random Forest, boosting methods including AdaBoost, Gradient Boosting, and XGBoost, and perform hyperparameter tuning using GridSearchCV and RandomizedSearchCV. Regularization techniques like Lasso and Ridge Regression are applied to handle overfitting, and cross-validation is used for model assessment. Additionally, hybrid learning approaches combining multiple models or techniques are introduced and practiced through mini-projects. **4. Unsupervised Learning Practicals:** Students apply clustering methods including K-Means, hierarchical clustering, and DBSCAN, perform dimensionality reduction using PCA and t-SNE, and build basic recommender systems using collaborative filtering based on user-item interaction data. **5. AI Fundamentals and Mini Projects:** This module offers hands-on experience with basic AI algorithms such as A* search and Minimax, constructing rule-based expert systems, creating chatbots using NLTK or spaCy, and exploring computer vision

fundamentals with OpenCV or TensorFlow. Students complete mini-projects like Titanic survival prediction, Iris species classification, customer segmentation, and sentiment analysis. **6. Model Evaluation and Reporting:** Students learn to evaluate models using performance metrics such as accuracy, confusion matrix, precision, recall, F1-score, and ROC-AUC. They analyze bias-variance tradeoffs, recognize overfitting and underfitting issues, and apply appropriate regularization. The course concludes with project report preparation, presentation skills, and best practices in coding standards, documentation, and version control using Git and GitHub.

Course Learning Outcomes (COs):

By the end of the course, students will be able to:

CO1: Implement supervised machine learning algorithms (linear regression, logistic regression, k-NN, SVM) for classification and regression tasks.

CO2: Apply unsupervised learning techniques such as clustering, dimensionality reduction, and recommender systems.

CO3: Optimize machine learning models using ensemble learning, hyperparameter tuning, and cross-validation techniques.

CO4: Evaluate model performance using appropriate metrics and understand the bias-variance tradeoff.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3	1	3	2		
CO2	3	1	3	2		
CO3	3		3	3		
CO4	3		3	3	2	

Mapping COs with Teaching-Learning & Assessment Strategy:

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
CO1	Demonstrations, hands-on lab sessions using Python (Scikit-learn), supervised learning implementation exercises	Weekly lab assignments, Midterm practical exam
CO2	Guided coding practice with clustering and PCA methods, exploratory data analysis using real datasets	Lab reports, Practical assignments, Mini-projects
CO3	Code walkthroughs on ensemble methods, parameter tuning via GridSearchCV and RandomizedSearchCV, and model comparison tasks	Weekly lab tasks, Final project, Viva-voce
CO4	Evaluation exercises with metrics (accuracy, precision, recall, F1-score, ROC-AUC), discussion on overfitting/underfitting	Report writing, Quizzes, Final project presentation

Reference Books:

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in Python/R*. Springer
2. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc."
3. Bishop, C. M. (2006). *Pattern recognition and machine learning by Christopher M. Bishop* (Vol. 400). Berlin, Germany:: Springer Science+ Business Media, LLC.
4. Raschka, S., & Mirjalili, V. (2017). Python machine learning second edition. *Birmingham, England: Packt Publishing*.

Code	Course Title	Credits (Theory)	Teacher	Course Category
CSE 0613 5185	Advanced Data Structures and Algorithms	2	CSE	Core

Rationale:

This course introduces fundamental data structures and algorithms used in data science for problem-solving and efficiency.

Course Objectives:

By the end of the course, students will:

1. Understand the fundamental concepts of data structures, abstract data types (ADTs), and algorithm design techniques.
2. Analyze the efficiency of algorithms using time and space complexity, including Big-O notation and performance cases.
3. Study and evaluate classical data structures such as arrays, linked lists, stacks, queues, trees, heaps, hash tables, and graphs.
4. Apply algorithmic strategies such as recursion, divide-and-conquer, dynamic programming, and greedy methods to solve computational problems.
5. Design and analyze sorting and searching algorithms with respect to correctness and efficiency.
6. Explore advanced topics such as graph traversal, shortest path algorithms, spanning trees, and optimization problems like the Travelling Salesman Problem (TSP).

Course Content:

1. Introduction to Data Structures and Algorithms: Basic concepts, abstract data types (ADTs), objects, and methods. **2. Fundamental Data Structures:** Arrays, linked lists, stacks, queues, priority queues, heaps, and hash tables. **3. Trees and Graphs:** Binary search trees (BST), AVL trees, red-black trees, n-ary trees, graph representations, minimum spanning trees, and shortest path algorithms (Dijkstra’s Algorithm). **4. Sorting and Searching:** Bubble sort, quicksort, merge sort, radix sort, binary search, and sequential search. **5. Algorithmic Complexity & Analysis:** Big-O notation, best/worst/average case analysis, recursion, and divide-and-conquer strategies. **6. Dynamic Programming & Greedy Algorithms:** Fibonacci sequence, binomial coefficients, longest common subsequence, optimal binary search trees, and Huffman encoding. **7. Graph Algorithms:** Depth-first search (DFS), breadth-first search (BFS), minimum spanning trees (Kruskal’s and Prim’s algorithms). **8. Advanced Topics & Applications:** Eulerian and Hamiltonian paths, Travelling Salesman Problem (TSP), and real-world applications in data science.

Course Learning Outcomes (COs):

By the end of the course, students will be expected to:

- CO1:** Understand and apply various data structures in problem-solving.
- CO2:** Analyze and implement sorting and searching algorithms.
- CO3:** Solve complex problems using dynamic programming and greedy algorithms.
- CO4:** Evaluate algorithmic efficiency using Big-O notation.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2		
CO2	3		3	2		
CO3	3		3	3		
CO4	3		3	3		

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures with real-life applications	Midterm Examination, Semester-end examination
CO2	Hands-on coding and algorithm implementation	Coding assignments, Semester-end examination

CO3	Problem-solving and algorithm design exercises	Assignment, Midterm Examination
CO4	Discussions on algorithmic efficiency	Semester-end examination

Textbook References:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
2. Hetland, M. L. (2014). *Python Algorithms: mastering basic algorithms in the Python Language*. Apress.

Code	Course Title	Credits (Lab)	Teacher	Course Category
CSE 0613 5186	Advanced Data Structures and Algorithms Lab	1.5	CSE	Core

Lab Rationale:

This lab course is designed to complement the theoretical course "Data Structures and Algorithms" by providing hands-on experience in implementing and applying the various data structures and algorithms introduced in the lectures. Students will gain practical skills in coding and problem-solving using Python, learning how to efficiently implement and analyze algorithms.

Lab Objectives:

1. To understand and implement basic data structures like arrays, linked lists, stacks, and queues.
2. To develop and apply sorting and searching algorithms.
3. To implement tree and graph algorithms for solving complex problems.
4. To practice analyzing algorithmic complexity using Big-O notation.
5. To solve real-world problems using dynamic programming and greedy algorithms.

Lab Content:

1. **Lab 1: Introduction to Data Structures**
 - o Overview of data structures and abstract data types (ADTs).
 - o Hands-on implementation of arrays, linked lists, and basic operations (insertion, deletion, traversal).
 - o Introduction to Python classes and objects for representing ADTs.
2. **Lab 2: Stacks and Queues**
 - o Implementing stack and queue using arrays and linked lists.
 - o Understanding stack operations: push, pop, peek, and isEmpty.
 - o Implementing queue operations: enqueue, dequeue, isEmpty.
 - o Applications of stacks (e.g., balancing symbols) and queues (e.g., breadth-first search).
3. **Lab 3: Priority Queues and Heaps**
 - o Implementation of priority queues using heaps (min-heap and max-heap).
 - o Hands-on with heap operations: insert, extract-min, and heapify.
 - o Applications: Dijkstra's algorithm for shortest paths.
4. **Lab 4: Hash Tables**
 - o Introduction to hash functions and collision resolution techniques (chaining, open addressing).
 - o Implementing a hash table in Python.
 - o Applications: storing and searching data efficiently.
5. **Lab 5: Trees and Binary Search Trees**
 - o Implementation of binary trees and binary search trees (BST).
 - o Operations: insertion, deletion, and traversal (in-order, pre-order, post-order).

- Balancing trees using AVL trees.
 - Applications: searching, sorting, and range queries.
- 6. Lab 6: Graphs and Graph Representations**
- Representation of graphs using adjacency matrices and adjacency lists.
 - Implementing depth-first search (DFS) and breadth-first search (BFS).
 - Finding minimum spanning trees using Kruskal's and Prim's algorithms.
 - Applications: shortest path problems and network analysis.
- 7. Lab 7: Sorting Algorithms**
- Implementing sorting algorithms: bubble sort, quicksort, merge sort, radix sort.
 - Analyzing the time complexity of each algorithm using Big-O notation.
 - Comparing the performance of different sorting algorithms on random data sets.
- 8. Lab 8: Searching Algorithms**
- Implementing search algorithms: binary search, sequential search.
 - Analyzing the time complexity of search operations.
 - Applications: searching in sorted and unsorted data.
- 9. Lab 9: Dynamic Programming**
- Implementing dynamic programming algorithms: Fibonacci sequence, binomial coefficients, longest common subsequence (LCS).
 - Solving optimization problems using dynamic programming techniques.
- 10. Lab 10: Greedy Algorithms**
- Implementing greedy algorithms: optimal binary search trees, Huffman encoding.
 - Solving real-world problems such as coin change and interval scheduling using greedy methods.
- 11. Lab 11: Advanced Graph Algorithms**
- Implementing advanced graph algorithms: Eulerian and Hamiltonian paths.
 - Solving the Traveling Salesman Problem (TSP) using dynamic programming and brute force methods.
- 12. Lab 12: Algorithm Efficiency and Optimization**
- Evaluating algorithmic performance using Big-O notation.
 - Profiling and optimizing Python code for efficiency.
 - Comparing the performance of different algorithms on large data sets.

Course Learning Outcomes (COs):

By the end of this course, students will be able to:

CO1: Understand and apply various data structures in problem-solving.

CO2: Analyze and implement sorting and searching algorithms.

CO3: Solve complex problems using dynamic programming and greedy algorithms.

CO4: Evaluate algorithmic efficiency using Big-O notation.

Mapping Course Learning Outcomes (COs) with the POs

Course Learning Outcomes (CO)	3: Strong	2: Moderate	1: Weak			
	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		2	2		
CO2	3		3	2		
CO3	3		3	2		
CO4			3	3		

Assessment Strategy:

- **Weekly Lab Assignments:** Each student will complete weekly coding assignments that implement the various data structures and algorithms.
- **Midterm Examination:** A practical exam that tests students' ability to implement and analyze algorithms.
- **Final Project:** A comprehensive project that integrates multiple data structures and algorithms to solve a real-world problem.
- **Lab Reports:** Submission of lab reports after each session detailing the implementation, analysis, and results.

Mapping Course Learning Outcomes (COs) with Lab Activities:

COs	Lab Activities	Assessment Strategy
CO1: Understand and apply various data structures in problem-solving.	Lab 1-6 (Implementation of Arrays, Linked Lists, Stacks, Queues, Trees, Graphs)	Lab Assignments, Final Project
CO2: Analyze and implement sorting and searching algorithms.	Lab 7-8 (Sorting and Searching Algorithms)	Midterm Examination, Lab Reports
CO3: Solve complex problems using dynamic programming and greedy algorithms.	Lab 9-10 (Dynamic Programming, Greedy Algorithms)	Lab Assignments, Final Project
CO4: Evaluate algorithmic efficiency using Big-O notation.	Lab 11-12 (Advanced Graph Algorithms, Algorithm Efficiency)	Lab Reports, Final Project

Textbook References:

1. Thomas H, C., Charles, E., Ronald L, R., & Clifford, S. (2009). Introduction to algorithms third edition.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
3. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). *Data structures and algorithms in Python*. John Wiley & Sons Ltd.

Code	Course Title	Credits (Theory)	Teacher	Course Category
STA 0542 5187	High-Dimensional Data Learning and Visualization	2	STA	Core

Course Rationale:

High-dimensional data, often encountered in fields like genomics, finance, and machine learning, presents unique visualization challenges. This course explores advanced techniques for visualizing and interpreting high-dimensional datasets, emphasizing dimensionality reduction, clustering, and interactive visualization tools.

Course Objectives:

By the end of this course, students will:

1. Understand the challenges and techniques involved in high-dimensional data visualization.
2. Gain proficiency in dimensionality reduction methods such as PCA, t-SNE, and UMAP.
3. Apply clustering and classification techniques to enhance visualization clarity.
4. Create interactive and web-based visualizations using tools like Plotly, Dash, and Streamlit.
5. Communicate insights effectively from high-dimensional data visualizations.

Course Content:

1. Introduction to High-Dimensional Data and Visualization Challenges: definition and characteristics of high-dimensional data, the "curse of dimensionality" and its impact on visualization, common sources of high-

dimensional data (genomics, finance, social networks), **2. Basic Visualization Techniques for High-Dimensional Data:** scatter plots, pair plots, and correlation matrices, heatmaps for relationship analysis, parallel coordinates and 3D visualization techniques, **3. Dimensionality Reduction for Visualization:** Principal Component Analysis (PCA), PLS, CA, t-SNE and UMAP for non-linear dimensionality reduction, visualizing clusters and patterns in reduced dimensions, **4. Clustering and Classification for High-Dimensional Data:** K-means, DBSCAN, and hierarchical clustering, Linear Discriminant Analysis (LDA) and Factor Analysis, enhancing visualization clarity using clustering techniques, **5. Interactive and Web-Based Visualization Techniques:** creating interactive plots with Plotly, Dash, and Bokeh, building web-based dashboards using Streamlit, Tableau and Power BI, integrating Python visualizations into interactive applications, **6. Best Practices for High-Dimensional Data Visualization:** choosing appropriate visualization techniques based on data types, handling missing values and outliers in high-dimensional data, communicating insights effectively through visual storytelling.

Course Learning Outcomes (COs):

By the end of the course, students will be able to:

- CO1: Identify high-dimensional data and the challenges using common sources through visualization techniques.
- CO2: Apply dimensionality reduction techniques (PCA, t-SNE, UMAP).
- CO3: Create and interpret advanced high-dimensional visualizations along with clustering and classification.
- CO4: Develop interactive web-based visualizations for complex datasets.
- CO5: High-dimensional data handling and communicating insights effectively using visualization techniques.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		3	3		
CO2	3		3			
CO3	3		3	2		
CO4			3			
CO5			3	3		

Mapping COs with Teaching-Learning & Assessment Strategy:

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures, interactive discussions, projector, Software (Streamlit/Tableau/Power BI/Python), and Hands-on projects	Quiz/ Assignment/ Presentation (Individual/group)/Projects Midterm Examination 1 & 2 Semester-end examination
CO2		
CO3		
CO4		
CO5		

Textbook References:

1. Healy, K. (2019). *Data visualization: A practical introduction*. Princeton University Press.
2. Hand, D. J., & Taylor, C. C. (1987). *Multivariate analysis: A user’s perspective*. Chapman and Hall.
3. Giraud, C. (2022). *Introduction to high-dimensional data analysis*. CRC Press.
4. Milovanovic, I. (2013). *Python Data Visualization Cookbook*. Packt Publishing.

Code	Course Title	Credits (Lab)	Teacher	Course Category
------	--------------	----------------	---------	-----------------

STA 0542 5188	High-Dimensional Data Learning and Visualization Lab	1.5	STA	Core
------------------	---------------------------------------------------------	-----	-----	------

Course Rationale:

High-dimensional data requires specialized visualization techniques to uncover meaningful patterns and relationships. This lab course offers hands-on experience in applying dimensionality reduction, clustering, and interactive web-based visualization tools using Python, Power BI, and other platforms. Through real-world datasets, students will develop the technical skills needed to effectively visualize and interpret complex high-dimensional information.

Course Objectives:

By the end of this lab course, students will be able to:

- Identify challenges and strategies for high-dimensional data visualization.
- Implement dimensionality reduction techniques such as PCA, t-SNE, and UMAP on real datasets.
- Apply clustering and classification algorithms to enhance visualization.
- Create interactive and web-based visualizations using tools like Plotly, Dash, and Streamlit.
- Effectively communicate insights from complex visualizations.

Lab Course Contents:

Module 1: Introduction to High-Dimensional Data Visualization: explore the "curse of dimensionality" with real datasets, identify appropriate visualization strategies for different types of high-dimensional data, **Module 2: Basic Visualization Techniques:** create scatter plots, pair plots, correlation matrices, generate heatmaps and parallel coordinates plots, hands-on exercise: visualizing customer demographics data, **Module 3: Dimensionality Reduction Applications:** apply PCA and interpret principal components, implement t-SNE and UMAP for complex data visualization, hands-on exercise: visualizing handwritten digit datasets (MNIST), **Module 4: Clustering and Classification Techniques for High-Dimensional Data:** application of various clustering and classification methods to analyze high-dimensional data, use clustering results to improve visualization clarity, hands-on exercise: clustering gene expression datasets, **Module 5: Advanced Classification for Visualization:** implement Linear Discriminant Analysis (LDA), perform Factor Analysis for dimension reduction, hands-on exercise: classifying financial transaction data, **Module 6: Interactive Visualizations with Python and Power BI:** build interactive plots using Plotly and Bokeh, create dynamic dashboards using Dash, hands-on exercise: interactive analysis of social media sentiment data using Python and Power BI, **Module 7: Web-Based Visualization Development:** develop and deploy basic dashboards using Streamlit and Power BI, integrate machine learning models into dashboards, hands-on exercise: streaming live visualization of data such as COVID-19 case data, **Module 8: Best Practices and Visual Storytelling:** choose visualization techniques based on dataset characteristics, handle missing values and outliers in visualizations, communicate insights through compelling visual stories, **Module 9: Capstone Project:** full-cycle project: data cleaning, dimensionality reduction, clustering, interactive visualization, and storytelling, final project presentation and written report.

Software and Tools:

- Python Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, Plotly, Dash, Bokeh, Streamlit
- Web-based Tools (optional): Tableau Public
- Power BI

Course Learning Outcomes (COs):

Upon successful completion of the lab course, students will be able to:

- CO1: Understand and address the challenges of visualizing high-dimensional data using appropriate techniques and exercises.
- CO2: Apply and compare different dimensionality reduction techniques (PCA, t-SNE, UMAP).
- CO3: Create advanced visualizations using clustering and classification methods through various projects.
- CO4: Develop interactive and web-based visualization applications using Streamlit, Python, and Power BI, integrating machine learning methods.
- CO5: Software efficiency, present and communicate findings through effective visual storytelling.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong

2: Moderate

1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1	3		3	2		
CO2	3		3			
CO3			3	2		
CO4	3		3			
CO5			3			

Mapping COs with Teaching-Learning & Assessment Strategy:

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures, interactive discussions, Software (Streamlit/Tableau/Power BI/Python), and Hands-on projects	Quiz/ Lab Assignment/ Presentation (Individual/group)/Projects Midterm Examination 1 & 2 Semester-end examination
CO2		
CO3		
CO4		
CO5		

Textbook References:

1. VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. O'Reilly Media.
2. Murray, D. (2020). *Tableau for Dummies* (2nd ed.). Wiley.
3. Lederer, J.(2021). *Fundamentals of High-Dimensional Statistics: With Exercises and R Labs*. Springer.
4. Dabbas, E. (2021). *Interactive dashboards and data apps with Streamlit and Python*. Packt.

Code	Course Title	Credits (Lab)	Teacher	Course Category
STA 0542 5189	Project-I	1.5	STA	Core

Rationale:

The **Data Science Project-I** course provides students with a real-world opportunity to apply their knowledge of data science theories and techniques to practical datasets. The main goal of the course is for students to complete a data science project from start to finish, culminating in a comprehensive report akin to a thesis. Throughout this course, students will gain hands-on experience in addressing a data science problem, from data collection and cleaning to model building and evaluation. Emphasizing critical thinking, problem-solving, and practical application, students will utilize techniques from machine learning, statistical analysis, and data visualization to derive meaningful insights and make data-driven decisions.

Course Objectives:

Upon completion of this course, students will be able to:

1. Develop a comprehensive understanding of the data science workflow, including problem definition, data collection, cleaning, model implementation, and solution delivery.
2. Enhance their ability to collect, clean, preprocess, and analyze real-world data.
3. Apply appropriate statistical and machine learning techniques to generate insights from complex datasets.
4. Evaluate model performance using various metrics and interpret results to make informed decisions.
5. Effectively present their findings through structured reports and presentations in a professional format.

Course Content:

1. Introduction to Data Science Projects: overview of the data science project lifecycle, understanding project objectives, timelines, and deliverables, defining the problem and scope of the project, identifying project stakeholders and expected outcomes, **2. Data Collection and Preprocessing:** methods for data collection (e.g., web scraping, APIs, and dataset repositories), data cleaning techniques: handling missing values, outliers, and duplicates, feature engineering: creating and selecting relevant features for analysis, data transformation and normalization techniques, **3. Exploratory Data Analysis (EDA):** visualization techniques for understanding data distributions and relationships (e.g., histograms, scatter plots, box plots), summary statistics (mean, median, standard deviation, etc.) and identifying key trends, identifying correlations, patterns, and potential predictive features, **4. Model Selection and Evaluation:** introduction to machine learning algorithms: supervised and unsupervised methods, model selection based on the problem type (classification, regression, clustering, etc.), model evaluation using performance metrics such as accuracy, precision, recall, F1 score, RMSE, and AUC, overfitting, underfitting, and model generalization, **5. Model Implementation and Tuning:** hands-on coding using Python and libraries like scikit-learn, TensorFlow, and Keras, hyperparameter tuning using grid search and random search techniques, cross-validation for model validation and performance evaluation, **6. Results Interpretation and Reporting:** interpreting the results of machine learning models and drawing actionable insights, structuring and writing a comprehensive project report (including methodology, results, analysis, and conclusions), communicating technical results to a non-technical audience via effective data storytelling, **7. Ethics, Data Privacy and Responsible AI:** ethical issues and challenges: understanding ethical dilemmas in data collection, analysis, and decision-making, and the impact of biased algorithms, data privacy and security: ensuring compliance with data protection laws such as GDPR and CCPA, and safeguarding sensitive data through anonymization, encryption, and secure storage, responsible use of data: avoiding bias, ensuring fairness, and making ethical decisions based on data insights, AI accountability and transparency: applying ethical AI frameworks, ensuring transparency, and building trust in AI systems, ethical guidelines and regulations: exploring global standards like GDPR, CCPA, and fairness principles to guide data science projects, **8. Programming with R:** introduction to R: data structures (vectors, matrices, data frames, lists), data import and export in R, data manipulation using dplyr, tidyr, etc., basic statistical functions in R, data visualization using ggplot2.

Course Learning Outcomes (COs):

By the end of the course, students will be able to:

- **CO1:** Define the scope and objectives of a data science project.
- **CO2:** Collect, clean, preprocess, and analyze data for data science applications.
- **CO3:** Implement and evaluate machine learning models and extract meaningful insights.
- **CO4:** Effectively communicate findings and results through structured reports and presentations.
- **CO5:** Demonstrate a commitment to ethical practices, including data privacy and responsible use of data in all stages of the project.

Mapping Course Learning Outcomes (COs) with the POs

3: Strong 2: Moderate 1: Weak

Course Learning Outcomes (CO)	Fundamental Skill	Social Skill	Thinking Skill		Personal Skill	
	PO1	PO2	PO3	PO4	PO5	PO6
CO1			3		3	3
CO2					3	3
CO3		3			3	3
CO4		3			3	3
CO5		3			3	3

Mapping COs with Teaching-Learning & Assessment Strategy:

Course Outcome (CO)	Teaching-Learning Strategy	Assessment Strategy
COs	Lectures, Guided discussions, Planning, Workshop (Streamlit/Tableau/Power BI/Python), Practice coding, and Hands-on projects	Project Proposal, Report, Project Deliverables. Midterm Project Checkpoint.

		Final Project Presentation/Report.
--	--	---------------------------------------

Textbook and References:

- McKinney, W. (2022). *Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media.
- Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (3rd ed.). O'Reilly Media.
- Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media.
- George, N., & George, M. (2023). *Practical data science with Python*. Addison-Wesley.
- Grus, J. (2019). *Data science from scratch: First principles with Python* (2nd ed.). O'Reilly Media.
- Bruce, P., & Bruce, A. (2017). *Practical statistics for data scientists: 50 essential concepts* (1st ed.). O'Reilly Media.
- Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.

Second Semester

Semester 2 (20 Credits)

Code	Course Title	Credits (Theory)	Teacher	Course Category
STA 0542 5281	Statistical Inference and Modeling	2	STA	Core

Course Rationale:

Statistical inference and hypothesis testing form the foundation of data-driven decision-making, while predictive modeling enables forecasting and pattern recognition in data science. This course provides a comprehensive understanding of statistical inference techniques, hypothesis testing methods, and predictive modeling, including generalized linear models, for solving real-world data science problems.

Course Objectives:

By the end of this course, students will be able to:

1. Understand the principles of statistical inference and hypothesis testing.
2. Apply different hypothesis testing methods to real-world data science problems.
3. Develop predictive models using statistical and machine learning approaches.
4. Evaluate and interpret the accuracy and reliability of predictive models.
5. Implement statistical inference and predictive modeling using Python/R.

Course Contents:

1. Introduction to Statistical Inference: concepts of parent distribution, sampling distribution, part of statistical inference, classical and Bayesian approach of statistical inference, **2. Point Estimation:** different methods of point estimation under classical approach, prior and posterior distribution, Bayes estimator, optimum properties of point estimators, ideas about MVBE and UMVUE, **3. Interval Estimation:** construction of confidence and Bayesian interval, pivotal quantity method and large sample method, **4. Parametric Hypothesis Testing:** Z-test and t-test for means and proportions, paired and independent sample tests, F-test for variance comparison, **5. Non-Parametric Hypothesis Testing:** Chi-square test for independence and goodness-of-fit, Mann-Whitney U test, Wilcoxon signed-rank test, Kruskal-Wallis test, **6. Generalized Linear Models (GLMs):** introduction to GLMs, exponential family of distributions, logistic regression for classification problems, Poisson regression for count data, model selection and evaluation for GLMs, **7. Survival Models:** concept of Weibull distribution with IFR, DFR and CFR, censoring and truncation, Cox's PH model, AFT models, **8. Hierarchical and Multi-level Modelling:** hierarchical and multi-level modeling, **9. Bayesian Data Analysis:** Bayesian data analysis, **10. Overfitting:** bias-variance tradeoff, techniques to prevent overfitting, **11. Time Series Predictive Modeling:** basics of time series analysis, moving average, ARIMA, GARCH, or RNN-based forecasting models, exponential smoothing, trend and seasonality detection, **12. Case Studies and Applications:** real-world applications of statistical inference, GLMs, multi-level modeling, time series-predictive modeling and Bayesian modeling in finance, healthcare, marketing, and social sciences.

Course Learning Outcomes (COs):

By the end of the course, students will be able to:

- CO1: Explain statistical inference concepts and their applications in data science.
- CO2: Perform hypothesis testing and interpret results correctly.
- CO3: Develop and evaluate predictive models using statistical and machine learning approaches.
- CO4: Assess model performance and prevent overfitting.
- CO5: Apply Python/R to perform statistical inference and predictive modeling.
- CO6: Understand and apply Generalized Linear Models (GLMs) for various data types.

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures, reading materials	Quizzes, class tests
CO2	Hands-on problem-solving	Assignments, practical exams
CO3	Case studies, real-world applications	Projects, coding exercises

CO4	Statistical model evaluation in Python/R	Lab assignments, final exam
CO5	Implementation of predictive models	Research-based case studies
CO6	Application of GLMs in real-world datasets	Lab exercises, group discussions

Reference Books:

1. Casella, G., & Berger, R. L. (2002). *Statistical Inference*. Cengage Learning.
2. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.
4. Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.
5. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
6. Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. Wiley.
7. Dobson, A. J., & Barnett, A. G. (2008). *An Introduction to Generalized Linear Models*. Chapman and Hall/CRC.

Code	Course Title	Credits (Theory)	Teacher	Course Category
MAT 0541 5203	Optimization and Numerical Analysis	2	MAT	GED

Course Rationale:

This lab course provides hands-on experience in implementing and analyzing optimization and numerical methods. Students will use Python or R to solve real-world problems related to machine learning, data analysis, and computational modeling.

Course Objectives:

- Translate theoretical concepts into computational algorithms.
- Solve practical data science problems using optimization and numerical techniques.
- Write optimized and efficient Python/R code.
- Validate and compare results using multiple algorithms and methods.

Mathematical Foundations of Numerical Analysis: Sources of numerical errors, round-off error, truncation error, floating-point representation, propagation of error, Taylor series expansion for numerical approximation.

Numerical Linear Algebra: Systems of linear equations, Gaussian elimination, LU decomposition, Cholesky decomposition, iterative methods, Jacobi method, Gauss–Seidel method, conditioning of matrices, stability of numerical algorithms.

Eigenvalue and Matrix Decomposition Methods: Eigenvalues and eigenvectors, power method, QR algorithm, singular value decomposition (SVD), applications in data science, principal component analysis (PCA), dimensionality reduction.

Numerical Solution of Nonlinear Equations: Root-finding methods, bisection method, Newton–Raphson method, secant method, convergence of iterative methods, rate of convergence, stability of numerical solutions.

Interpolation and Approximation: Polynomial interpolation, Lagrange interpolation, Newton interpolation, spline interpolation, least squares approximation, applications in data fitting and modeling.

Numerical Differentiation and Integration: Finite difference approximations, numerical differentiation, numerical integration, trapezoidal rule, Simpson’s rule, Gaussian quadrature, applications in probability and simulation.

Unconstrained Optimization: Formulation of optimization problems, first-order optimality conditions, second-order optimality conditions, gradient descent method, stochastic gradient descent (SGD), Newton’s method, applications in machine learning model training.

Constrained Optimization: Linear programming, simplex method, duality theory, Lagrange multipliers, Karush–Kuhn–Tucker (KKT) conditions, applications in resource allocation and portfolio optimization.

Course Learning Outcomes (COs)

By the end of this course, students will be able to:

CO Code	Description
CO1	Explain the sources of numerical errors and analyze the stability and accuracy of numerical algorithms.

CO2	Apply numerical linear algebra techniques such as matrix decomposition and iterative methods to solve systems of equations.
CO3	Implement numerical methods for solving nonlinear equations, interpolation, differentiation, and integration problems.
CO4	Formulate and analyze unconstrained and constrained optimization problems relevant to data science.
CO5	Evaluate the convergence, efficiency, and computational trade-offs of different numerical and optimization algorithms.

Mapping Course Learning Outcomes (COs) with Program Outcomes (POs)

(3 = Strong, 2 = Moderate, 1 = Weak)

CO	Fundamental Skill (PO1)	Social Skill (PO2)	Thinking Skill (PO3)	Personal Skill (PO4)	PO5	PO6
CO1	3		2			
CO2	3		3			
CO3	2		3			
CO4	3		3	1		
CO5	2		3	2		

Textbooks

1. Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). New York, NY: Springer.
Link: [Numerical Optimization – Springer Book Page](#)
This book provides a comprehensive treatment of modern optimization algorithms including gradient methods, Newton methods, and constrained optimization techniques widely used in engineering, machine learning, and data science.
2. Burden, R. L., & Faires, J. D. (2011). *Numerical analysis* (9th ed.). Boston, MA: Cengage Learning.
Link: <https://share.google/ddNXR5bD2bMa3AYre>

Reference Books

1. Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge, UK: Cambridge University Press.
Link: [Convex Optimization – Free Book & Resources](#)
This influential book covers convex sets, duality, interior-point methods, and optimization techniques widely used in machine learning and large-scale data analysis.
2. Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes: The art of scientific computing* (3rd ed.). Cambridge, UK: Cambridge University Press.
Link: <https://numerical.recipes/>

Code	Course Title	Credits (Lab)	Teacher	Course Category
MAT 0541 5204	Optimization and Numerical Analysis Lab	1	MAT	GED

Introduction to Scientific Computing: Python or R programming environment, NumPy or matrix operations, data visualization, basic numerical computing tools.

Solving Systems of Linear Equations: Implementation of Gaussian elimination, LU decomposition, iterative methods, solving large systems of equations, comparison of computational efficiency.

Root-Finding Algorithms: Implementation of bisection method, Newton–Raphson method, secant method, analysis of convergence behavior.

Interpolation and Approximation: Polynomial interpolation, spline fitting, approximation of datasets, smoothing and modeling of data.

Numerical Integration: Implementation of trapezoidal rule, Simpson’s rule, Monte Carlo integration, estimation of integrals for data analysis.

Eigenvalue and SVD Computation: Eigenvalue computation, singular value decomposition, implementation of PCA, dimensionality reduction for datasets.

Optimization Algorithms: Implementation of gradient descent, stochastic gradient descent, optimization of objective functions in machine learning.

Linear Programming Applications: Resource allocation problems, portfolio optimization, solving optimization problems using scipy.optimize or lpSolve.

Mini Project: Application of numerical and optimization techniques to real datasets, implementation of algorithms for recommendation systems, regression optimization, or dimensionality reduction.

Lab Learning Outcomes (COs)

CO Code	Description
CO1	Implement optimization and numerical methods programmatically.
CO2	Apply numerical techniques to practical data science problems.
CO3	Use Python or R to solve mathematical and computational problems effectively.
CO4	Evaluate computational performance, convergence behavior, and algorithm accuracy.
CO5	Demonstrate problem-solving ability using real-world datasets through computational experiments.

Mapping COs with Program Outcomes (POs)

CO	Fundamental Skill (PO1)	Social Skill (PO2)	Thinking Skill (PO3)	Personal Skill (PO4)	PO5	PO6
CO1	3		2			
CO2	2		3	1		
CO3	3		2	2		
CO4	2		3	2		
CO5	3		3	2		

Mapping COs with Teaching–Learning & Assessment

CO	Teaching Strategy	Assessment Strategy
CO1	Hands-on laboratory sessions	Weekly lab reports
CO2	Case-based computational exercises	Code submissions
CO3	Independent coding tasks	Midterm practical test
CO4	Algorithm comparison and benchmarking	Project report
CO5	Mini-project with presentation	Final project evaluation

Code	Course Title	Credits (Theory)	Teacher	Course Category
CSE 0612 5283	Database Management for Data Science	3	CSE	Core

Rationale

This course provides a comprehensive theoretical understanding of database management systems (DBMS) for data science applications. Students will learn relational database concepts, SQL query formulation, data integrity, transaction management, indexing, and optimization strategies. The theory course equips students with the principles needed to design, analyze, and evaluate databases efficiently before implementing them in practical settings.

Course Objectives

By the end of this course, students will be able to:

- Understand and explain the fundamental and advanced concepts of DBMS.
- Design relational database schemas from entity-relationship models.
- Analyze and optimize SQL queries for efficient data retrieval and manipulation.
- Apply principles of data integrity, normalization, and transaction management.
- Evaluate database performance and propose strategies for optimization in real-world applications.

Course Content

Introduction to DBMS Concepts: Overview of database management systems, relational databases, and entity-relationship (ER) models. Mapping ER diagrams to relational models and understanding database schemas.

SQL Query Design: Syntax and structure of basic SQL queries (SELECT, INSERT, UPDATE, DELETE), advanced clauses (JOIN, GROUP BY, HAVING, ORDER BY), subqueries, and techniques for efficient data retrieval.

Data Integrity and Transactions: Principles of enforcing data integrity constraints (primary key, foreign key, unique, and check constraints), understanding ACID properties, and managing database transactions (COMMIT, ROLLBACK, SAVEPOINT).

Indexing and Searching: Concepts of indexes (B-tree, hash indexing), performance optimization for query execution, and evaluating indexing strategies for different data scenarios.

Database Design and Management Principles: Designing normalized databases (1NF, 2NF, 3NF, BCNF), organizing tables, views, and relationships, and understanding theoretical considerations for scalable database structures.

Advanced SQL Techniques: Complex SQL queries including multiple joins, nested queries, subqueries, and aggregate functions. Introduction to query optimization strategies and evaluation of execution plans.

Theoretical Case Studies in Data Science: Analysis of database design and query formulation for typical data science applications, such as customer analytics, inventory management, and recommendation systems.

Course Learning Outcomes (COs)

CO Code	Description
CO1	Explain and apply relational database design principles from ER models.
CO2	Formulate basic and advanced SQL queries for data retrieval and manipulation.
CO3	Analyze and ensure data integrity and consistency through constraints and transaction management.
CO4	Evaluate indexing methods and query optimization strategies to enhance performance.
CO5	Apply normalization and advanced SQL techniques to design scalable, real-world databases.

Mapping COs with Teaching–Learning & Assessment Strategy

CO	Teaching Strategy	Assessment Strategy
CO1	Lectures with ER model examples, case studies	Assignments, quizzes
CO2	SQL query demonstrations, illustrative examples	Written exams, homework problems
CO3	Lecture-based discussion of transactions and constraints	Problem-solving exercises, case analysis
CO4	Analysis of indexing and query plans	Case-based evaluation, exams
CO5	Theoretical projects and case studies	Semester-end examination, assignments

Textbook References

1. Forta, B. (2015). *SQL in 10 Minutes, Sams Teach Yourself* (4th ed.). Pearson. [Link](#)
2. Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems* (3rd ed.). McGraw-Hill. [Link](#)

Reference Books

1. Winand, M. (2012). *SQL Performance Explained*. Markus Winand. [Link](#)
2. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson. [Link](#)

Code	Course Title	Credits (Lab)	Teacher	Course Category
CSE 0612 5284	Database Management for Data Science Lab	1.5	CSE	Core

Rationale:

This lab course provides practical experience in the design, implementation, and management of databases using SQL. Students will develop hands-on skills for building and managing relational databases, writing complex SQL queries, and ensuring data integrity and performance optimization in real-world applications.

Course Objectives:

By the end of the course, students will be able to:

- Understand and apply basic and advanced concepts of database management systems (DBMS).
- Design and implement SQL queries for efficient data retrieval and manipulation.
- Ensure data integrity and consistency using database constraints and transaction management.
- Use advanced SQL techniques to optimize database performance.
- Perform tasks related to database design, management, and programming for real-world applications.

Lab Content:

1. Introduction to DBMS Concepts:
 - Overview of database management systems (DBMS).
 - Implementing relational databases and entity-relationship (ER) models.
 - Mapping ER diagrams to relational models and creating the corresponding database schemas.
2. SQL Query Design:
 - Writing basic SQL queries (SELECT, INSERT, UPDATE, DELETE).
 - Using advanced SQL clauses: JOIN, GROUP BY, HAVING, ORDER BY, and subqueries.
 - Efficient data retrieval using proper query design.
3. Data Integrity and Transactions:
 - Implementing and enforcing data integrity constraints: primary keys, foreign keys, unique constraints, and check constraints.
 - Managing database transactions: COMMIT, ROLLBACK, and SAVEPOINT.

- Ensuring database consistency and reliability in multi-user environments.
4. Indexing and Searching:
 - Creating and using indexes for efficient query performance.
 - Implementing and optimizing database searches.
 - Exploring methods of indexing (B-tree, hash indexing) and their applications in data retrieval.
 5. Database Management Tasks:
 - Creating, opening, and modifying databases.
 - Structuring databases using tables, views, and relationships.
 - Sorting, indexing, and searching data to improve efficiency.
 - Designing user interfaces like customer screens, reports, and menus to interact with the database.
 6. Database Programming:
 - Working with memory variables, constants, operators, and arrays in SQL.
 - Writing SQL stored procedures and functions for data entries, updates, and report generation.
 - Implementing triggers for automated actions within the database.
 7. Advanced SQL Techniques:
 - Learning database normalization techniques: 1NF, 2NF, 3NF, and BCNF.
 - Implementing SQL query optimization techniques for better performance.
 - Writing complex SQL queries involving multiple joins, nested queries, and subqueries.

Course Learning Outcomes (COs):

- CO1: Design and implement relational database schemas from entity-relationship models.
- CO2: Write efficient SQL queries for data retrieval and manipulation.
- CO3: Ensure data integrity and consistency through constraints and transaction management.
- CO4: Apply indexing techniques for faster data retrieval and query optimization.
- CO5: Implement advanced SQL features, including database normalization and complex queries for real-world database applications.

Assessment Strategy:

- CO1: Database design tasks, Lab assignments
- CO2: SQL query writing, Practical exams
- CO3: Transaction management and data integrity exercises, Lab reports
- CO4: Indexing and optimization tasks, Lab assignments
- CO5: Advanced SQL query tasks, Final project (database application development)

Textbook References:

- *SQL in 10 Minutes, Sams Teach Yourself* by Ben Forta
- *Database Management Systems* by Raghu Ramakrishnan and Johannes Gehrke
- *SQL Performance Explained* by Markus Winand

Code	Course Title	Credits (Theory)	Credits (Lab)	Teacher	Course Category
CSE 0612 5286	Cloud-based Big Data Analytics		3	CSE	Core

Rationale:

This course equips students with the knowledge and skills required for handling and analyzing big data using cloud computing platforms. It covers fundamental concepts, tools, and hands-on experience in distributed data processing, storage, and analytics.

Course Objectives:

By the end of this course, students will:

1. Understand fundamental concepts of big data and cloud computing.
2. Learn to use big data technologies such as Hadoop, Spark, and NoSQL databases.
3. Gain hands-on experience with cloud computing platforms (AWS, Azure, Google Cloud).
4. Develop skills in distributed data processing, data wrangling, and ETL workflows.
5. Apply machine learning and data mining techniques to big data.
6. Implement scalable and secure cloud-based big data solutions.

Course Content:

Big Data Concepts & Fundamentals: Characteristics of big data, volume, velocity, variety, veracity, value, challenges, ethics, privacy concerns, data storage techniques, data lakes, data warehousing, distributed systems for big data processing.

Big Data Technologies & Analytics: Hadoop ecosystem, HDFS, MapReduce, YARN, Apache Spark, Spark Core, Spark SQL, Spark Streaming, MLlib, NoSQL databases, MongoDB, Cassandra, HBase, Apache Flink, Dask, data mining techniques, machine learning models, data visualization tools.

Cloud Computing Fundamentals & Platforms: Cloud computing overview, benefits, challenges, applications, cloud service models (IaaS, PaaS, SaaS), deployment models (public, private, hybrid, multi-cloud), virtualization, Docker, Kubernetes, cloud security, cloud storage (object storage, block storage, database management), hands-on AWS, Azure, Google Cloud.

Practical Skills & Applications: Data pipelines, ETL workflows, real-time big data processing, serverless computing, cloud-based AI/ML services, case studies in healthcare, finance, social media, cloud-native application development, scalable solution best practices.

Course Learning Outcomes (COs):

By the end of the course, students will be expected to:

- CO1: Understand big data principles, cloud computing models, and their applications.
- CO2: Work with Hadoop, Spark, and NoSQL databases for big data analytics.
- CO3: Utilize cloud platforms for scalable data storage, processing, and security.
- CO4: Implement machine learning and data mining techniques for big data insights.
- CO5: Design and deploy cloud-based big data solutions for real-world applications.

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures, case studies	Midterm Examination
CO2	Hands-on exercises, lab work	Lab assignments, Project
CO3	Cloud computing workshops	Cloud platform project
CO4	Real-world case studies	Project, Semester-end examination
CO5	Capstone Project, Case Analysis	Final Project, Report Submission

Textbook References:

1. *Hadoop: The Definitive Guide* by Tom White

2. *Learning Spark* by Holden Karau et al.
3. *Cloud Computing: Principles and Paradigms* by Rajkumar Buyya et al.
4. *Big Data: Principles and Best Practices* by James Warren
5. *Data Science on the Google Cloud Platform* by Valliappa Lakshmanan

Code	Course Title	Credits (Theory)	Teacher	Course Category
CSE 0613 5287	Deep Learning	3	CSE	Core

Rationale: This course introduces deep learning models and the optimization techniques necessary for training them. It covers both foundational architectures and state-of-the-art advancements, providing students with a comprehensive understanding of deep learning principles and their applications.

Course Objectives:

By the end of the course, students will:

1. Understand the core architectures of deep learning models.
2. Develop and implement deep learning models for various applications.
3. Optimize deep learning models using state-of-the-art techniques.
4. Utilize deep learning frameworks effectively for model development.
5. Understand theoretical principles behind deep learning optimization.

Course Content:

Deep Learning

Introduction to Deep Learning:

- Basics of machine learning and deep learning
- Importance of deep neural networks
- Applications of deep learning in various domains

Fundamental Neural Network Architectures:

- Feedforward Neural Networks (FNNs): Layers, activation functions, and backpropagation
- Convolutional Neural Networks (CNNs): Convolutional layers, pooling layers, and transfer learning
- Recurrent Neural Networks (RNNs): Sequence modeling, Long Short-Term Memory (LSTMs), and Gated Recurrent Units (GRUs)
- Transformers: Self-attention mechanism, BERT, and applications in NLP

Optimization Techniques:

- Gradient Descent and its Variants: SGD, Momentum, RMSprop, Adam
- Learning Rate Schedules: Step decay, exponential decay, and warm-up strategies
- Hyperparameter Tuning: Grid search, random search, Bayesian optimization
- Regularization Techniques: Dropout, batch normalization, weight decay

Theoretical Foundations:

- Non-Convex Optimization: Challenges in high-dimensional loss surfaces
- Distributed Learning: Training deep models at scale

Computer Vision: Image classification, object detection, image segmentation

Natural Language Processing

- Introduction to NLP: Overview, challenges, and applications in AI.
- Text Preprocessing: Tokenization, stopword removal, stemming, lemmatization, POS tagging.
- Feature Extraction & Representation: Bag-of-Words, TF-IDF, word embeddings (Word2Vec, GloVe, FastText).
- Language Models: N-grams, Hidden Markov Models, statistical language models.
- Text Classification: Sentiment analysis, spam detection, topic modeling (LDA).

- Named Entity Recognition (NER) & Information Extraction: Entity linking, coreference resolution.
- Sequence-to-Sequence Models & Machine Translation: Encoder-decoder models, transformers.
- Deep Learning for NLP: RNNs, LSTMs, GRUs, attention mechanisms, BERT, GPT models.
- Text Summarization & Question Answering: Extractive and abstractive summarization, QA systems.

Artificial Intelligence (AI)

- Introduction to AI and Generative AI: Architecture and functioning of modern generative models like Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Transformer-based models (e.g., GPT), training generative models, prompt engineering, fine-tuning, and evaluating text outputs
- Transformer architecture and autoregressive models (e.g., GPT)
- Prompting and fine-tuning pre-trained generative models
- Applications: Text summarization, translation, chatbot development
- Evaluation metrics for generated content (BLEU, ROUGE, perplexity)
- Ethics and Bias in NLP: Addressing bias in AI models and responsible AI practices.

Course Learning Outcomes (COs):

By the end of the course, students will be expected to:

- CO1: Understand the architectures of deep learning models.
- CO2: Implement deep learning models for various tasks.
- CO3: Optimize deep learning models using various techniques.
- CO4: Utilize deep learning frameworks to develop neural networks.
- CO5: Apply theoretical concepts of optimization to deep learning.

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs Teaching-Learning Strategy	Assessment Strategy
CO1 Lectures with demonstrations	Midterm Examination, Assignment
CO2 Hands-on coding sessions	Project-based assessment
CO3 Problem-solving sessions	Lab assignments, Semester-end examination
CO4 Practical workshops and case studies	Lab project, Final project
CO5 Theoretical discussions, research paper readings	Quiz, Final examination

Textbook References:

- *Deep Learning* by Ian Goodfellow, Yoshua Bengio, Aaron Courville
- *Deep Learning with Python* by François Chollet
- *Neural Networks and Deep Learning* by Michael Nielsen
- *Pattern Recognition and Machine Learning* by Christopher Bishop

Code	Course Title	Credits (Lab)	Teacher	Course Category
CSE 0613 1288	Deep Learning	1.5	CSE	Core

Course Rationale:

This lab course provides students with hands-on experience in building, optimizing, and deploying deep learning models. Through practical exercises and projects using popular frameworks like TensorFlow, Keras, and PyTorch, students will deepen their understanding of deep learning architectures, optimization strategies, and real-world applications in computer vision, NLP, and generative modeling.

Course Objectives:

By the end of this lab course, students will:

1. Implement core architectures of deep learning models (FNNs, CNNs, RNNs, Transformers).

2. Develop deep learning models for vision, text, and generative tasks.
3. Apply optimization techniques to improve model performance.
4. Utilize deep learning frameworks such as TensorFlow, Keras, and PyTorch effectively.
5. Connect theoretical concepts of optimization to practical deep learning workflows.

Course Learning Outcomes (COs):

Upon successful completion of the lab course, students will be able to:

- **CO1:** Construct and understand fundamental deep learning architectures.
- **CO2:** Implement and evaluate deep learning models for different real-world problems.
- **CO3:** Optimize deep learning models using modern optimization techniques.
- **CO4:** Use deep learning frameworks for building and deploying neural networks.
- **CO5:** Integrate theoretical optimization principles into practical deep learning solutions.

Lab Course Contents:

Module 1: Introduction to Deep Learning Tools

- Setting up TensorFlow, Keras, and PyTorch environments
- Loading and preprocessing datasets (MNIST, CIFAR-10, IMDB reviews)
- Hands-on: Build your first neural network for digit recognition

Module 2: Building Fundamental Architectures

- Implementing Feedforward Neural Networks (FNNs)
- Implementing Convolutional Neural Networks (CNNs) for image classification
- Hands-on: Train CNNs for CIFAR-10 image classification

Module 3: Recurrent Neural Networks and Transformers

- Building basic RNNs and LSTMs for sequence data
- Implementing a Transformer-based text classifier
- Hands-on: Sentiment analysis with LSTM and fine-tuning BERT

Module 4: Model Optimization Techniques

- Experimenting with different optimizers: SGD, Adam, RMSprop
- Learning rate schedules and warm-up strategies
- Hyperparameter tuning using random search and grid search
- Hands-on: Optimize a CNN model using various optimizers and learning rate strategies

Module 5: Regularization and Generalization

- Applying dropout, batch normalization, and weight decay
- Preventing overfitting in deep networks
- Hands-on: Regularize a deep model to improve validation performance

Module 6: Practical Applications in Vision and NLP

- Image classification with CNNs
- Text classification and sequence-to-sequence modeling
- Hands-on: Image segmentation with U-Net, text generation with LSTM

Module 7: Generative Models

- Implementing Variational Autoencoders (VAEs)
- Building basic Generative Adversarial Networks (GANs)
- Hands-on: Create a GAN to generate synthetic images

Module 8: Transfer Learning and Deployment

- Fine-tuning pre-trained models (e.g., ResNet, VGG, BERT)
- Model export with ONNX; deploying with TensorRT or Streamlit
- Hands-on: Fine-tune a ResNet model and deploy via a simple web app

Module 9: Capstone Project

- Full-cycle project: Data preparation → Model building → Optimization → Evaluation → Deployment
- Final project presentation and demonstration

Software and Tools:

- **Frameworks:** TensorFlow, Keras, PyTorch
- **Supporting Tools:** ONNX, TensorRT, Streamlit
- **Programming Language:** Python

Code	Course Title	Credits (Lab)	Teacher	Course Category
STA 0542 5290	Professional Skills and Data Science Ethics	1	STA	Core

Rationale:

This course is designed to develop essential professional skills for data science practitioners. It covers effective communication techniques, including presentation skills, viva preparation, professional writing, and critical thinking necessary for success in the data science field. Emphasizing both soft and technical skills, the course prepares students to present their research findings confidently, engage in professional discussions, and succeed in data science career paths.

Course Objectives:

By the end of this course, students will be able to:

- Develop strong verbal and non-verbal communication skills for presenting complex data science concepts.
- Prepare and deliver effective presentations to both technical and non-technical audiences.
- Understand the key aspects of academic and professional writing for data science research.
- Practice proper response strategies for academic viva voce examinations.
- Enhance skills for effective teamwork, leadership, and professional networking.

Course Contents:

1. **Introduction to Professional Skills in Data Sciences:**
 - Importance of communication skills in data science.
 - Overview of soft skills required for success in academia and industry.
 - Role of data scientists in collaborative environments and their impact on decision-making.
2. **Effective Presentation Skills:**
 - Structuring a presentation: Introduction, body, conclusion.
 - Designing visually appealing slides and data visualizations.
 - Tailoring presentations for diverse audiences (technical vs. non-technical).
 - Public speaking tips: Confidence-building, managing nerves, and engaging with the audience.
 - Delivering clear explanations of complex concepts using data science examples.
3. **Viva Preparation:**
 - Understanding the format and expectations of a viva in academia and professional environments.
 - Preparing for common viva questions in data science research.
 - Practice answering questions effectively, defending methodologies, and discussing results.
 - Handling unexpected or challenging questions with confidence.

4. Research and Report Writing:

- Writing clear and concise reports for data science projects.
- Structuring research papers and theses: Abstracts, methodology, analysis, results, and conclusions.
- Writing for different audiences: Research papers, project reports, and industry documentation.
- Avoiding plagiarism and maintaining academic integrity.
- Citations and referencing in data science writing.

5. Critical Thinking and Problem Solving:

- Developing the ability to critically analyze and interpret data.
- Evaluating data science models, results, and the significance of findings.
- Engaging in discussions, debates, and presentations with critical insights.
- Developing logical thinking for problem-solving in data science projects.

6. Professional Networking and Career Development:

- Building a professional presence: LinkedIn profiles, online portfolios, and CV writing.
- Networking strategies for data scientists: Conferences, workshops, and online communities.
- Career development pathways in data science: Industry vs. academia.
- Interview preparation: Common questions and how to approach them in data science roles.

Course Learning Outcomes (COs):

By the end of the course, students will be able to:

- **CO1:** Effectively present and communicate data science concepts to both technical and non-technical audiences.
- **CO2:** Prepare and successfully participate in a viva examination.
- **CO3:** Develop and write professional research papers and reports for academic and industry purposes.
- **CO4:** Exhibit critical thinking and problem-solving skills while presenting and defending data science findings.
- **CO5:** Build professional networks and prepare for career opportunities in the data science field.

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs	Teaching-Learning Strategy	Assessment Strategy
CO1	Lectures, presentation practice sessions	Group presentations, peer feedback
CO2	Mock viva sessions, Q&A practice	Viva mock exam, individual assessments
CO3	Writing workshops, case studies	Research paper review, report writing
CO4	Problem-solving discussions, peer reviews	Critical analysis assignments, peer-reviewed projects
CO5	Networking workshops, career-building activities	Networking event participation, CV review, career interviews

Assessment Strategy:

- **Group Presentations:** Evaluation based on content clarity, presentation skills, and audience engagement.
- **Viva Mock Exam:** Assessment based on how well students can defend their work and answer questions.
- **Research Paper Review:** Assessment based on clarity, structure, writing quality, and data analysis.
- **Critical Analysis and Problem Solving:** Practical assessment of how well students analyze and solve data science problems.
- **Career Development Activities:** Assessment based on CV, LinkedIn profile, and participation in networking events.

Reference Materials:

- *The Data Science Handbook* by Carl Shan, William Chen, Henry Wang, and Max Song
- *Presentation Zen: Simple Ideas on Presentation Design and Delivery* by Garr Reynolds
- *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking* by Foster Provost and Tom Fawcett

Code	Course Title	Credits (Lab)	Teacher	Course Category
STA 0542 5292	Project-II	2	STA/CSE/MAT	Core

(Capstone Project/Internship)

Rationale: This course allows students to conduct original research or work on real-world industry problems to apply and extend their data science knowledge.

Course Content:

- Research proposal development and literature review
- Data collection, AI model development, and experimentation
- Thesis writing and final defense or Capstone Project execution

Course Learning Outcomes (COs):

By the end of the course, students will be expected to:

- **CO1:** Conduct independent research or work on a real-world data science problem.
- **CO2:** Develop and optimize AI models to address specific challenges.
- **CO3:** Communicate research findings effectively.
- **CO4:** Defend their thesis or capstone project in front of a panel.

Mapping Course Learning Outcomes (COs) with Teaching-Learning & Assessment Strategy

COs Teaching-Learning Strategy	Assessment Strategy
CO1 Research guidance, industry interaction	Thesis evaluation
CO2 Model development and experimentation	Project-based assessment
CO3 Thesis writing and presentations	Final thesis, Defense
CO4 Capstone project completion	Presentation, Evaluation

Textbook References:

- Research articles relevant to the student's thesis or project