**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.<br>UM2: To serve as a center for knowledge creation, technological innovation and transfer among academia, industry, and society.<br>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<br>M2: Maintain the quality of teaching and research at international standard<br>M3: Collaborate with stakeholders for planning, statistical analysis and research<br>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<br>b. Contribute to the theoretical and practical development of statistical methods addressing substantive problems<br>c. Organizing workshops, seminars, conferences with stakeholders to improve quality of research and implement the statistical techniques<br>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)<br>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.<br>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. |

| | | |
|---|---|---|
| | | 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. |
| **11.** | **Rationale of the Program:** | 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.<br><br>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.<br><br>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. |
| **12.** | **Structure of the Program:** | 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.<br><br>The total requirement for graduation is 40 credits, distributed over two semesters:<br>• Semester I: 20 credits<br>• Semester II: 20 credits<br><br>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.<br><br>Courses are classified into the following categories:<br>• Core Courses<br>• General Education Courses<br>• Elective or Optional Courses<br><br>This structured approach ensures a comprehensive and rigorous foundation in data science, while maintaining flexibility and adherence to national academic standards. |
| **13.** | **Admission Requirements:** | 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.<br><br>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.<br><br>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 |

| | | |
|---|---|---|
| | | of Advanced Studies through the Dean of the school, for subsequent approval by the Academic Council.<br><br>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.<br><br>A3.2. A SUST faculty member may be admitted to this program with prior approval from the University Authority.<br><br>A3.3 The registration for this program will remain valid for a maximum of 4 (four) semesters.<br><br>A3.4 The period of candidature for this program will remain valid for a maximum of 2 (two) academic years.<br><br>A6.1 Duration: Minimum duration for this degree is 2 (two) semesters.<br><br>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.<br><br>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. |

**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 | 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)<br>Midterm Examination 1 & 2<br>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**

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

**3: Strong** **2: Moderate** **1: Weak**

| Course Learning Outcomes | Fundamental Skill | Social Skill | Thinking Skill | | Personal Skill | |
|---|---|---|---|---|---|---|
| (CO) | 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 | Fundamental Skill | Social Skill | Thinking Skill | | Personal Skill | |
|---|---|---|---|---|---|---|
| (CO) | 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<br>Midterm Examination 1 & 2<br>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:

- o Python Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, Plotly, Dash, Bokeh, Streamlit
- o Web-based Tools (optional): Tableau Public
- o 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<br>Midterm Examination 1 & 2<br>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 | Fundamental Skill | Social Skill | Thinking Skill | | Personal Skill | |
|---|---|---|---|---|---|---|
| (CO) | 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.<br><br>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.