PRESIDENCY UNIVERSITY

Presidency University Act, 2013 of the Karnataka Act No. 41 of 2013 | Established under Section 2(f) of UGC Act, 1956
Approved by AICTE, New Delhi

PRESIDENCY SCHOOL OF INFORMATION SCIENCE

Program Regulations and Curriculum 2026 - 2028

Master of Science (M.Sc) in Data Science

based on Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Table of Contents

PART A – PROGRAM REGULATIONS

1. Vision & Mission of the University and the School / Department

1.1 Vision of the University

To be a Value-driven Global University, excelling beyond peers and creating professionals of integrity and character, having concern and care for society.

1.2 Mission of the University

- Commit to be an innovative and inclusive institution by seeking excellence in teaching, research and knowledge-transfer.
- Pursue research and development and its dissemination to the community, at large.
- Create, sustain and apply learning in an interdisciplinary environment with consideration for ethical, ecological and economic aspects of nation building.
- Provide knowledge-based technological support and services to the industry in its growth and development.
- To impart globally-applicable skill-sets to students through flexible course offerings and support industry's requirement and inculcate a spirit of new-venture creation.

1.3 Vision of Presidency School of Information Science

To be a global centre of excellence in information science and research, fostering innovation and producing professionals with integrity and ethical responsibility.

1.4 Mission of Presidency School of Information Science

- To provide high-quality education in information science, equipping students with strong technical expertise and problem-solving skills.
- To promote research and innovation in information science and technology, addressing real-world challenges through industry collaboration.
- To nurture graduates with strong ethical values and a commitment for lifelong learning for sustained professional growth in the IT sector and allied fields

2. Preamble to the Program Regulations and Curriculum

This is the subset of Academic Regulations and it is to be followed as a requirement for the award of M.Sc. degree.

The Curriculum is designed to take into the factors listed in the Choice Based Credit System (CBCS) with focus on Social Project Based Learning, Industrial Training, and

Internship to enable the students to become eligible and fully equipped for employment in industries, choose higher studies or entrepreneurship.

In exercise of the powers conferred by and in discharge of duties assigned under the relevant provision(s) of the Act, Statutes and Academic Regulations, of the University, the Academic Council hereby makes the following Regulations.

3. Short Title and Applicability

a. These Regulations shall be called the Master of Science Degree Program Regulations and Curriculum 2026-2028
b. These Regulations are subject to, and pursuant to the Academic Regulations.
c. These Regulations shall be applicable to the M.Sc. Degree Programs of the 2026-2028 batch, and to all other M.Sc. Degree Programs which may be introduced in future.
d. These Regulations shall supersede all the earlier M.Sc. Degree Program Regulations and Curriculum, along with all the amendments thereto.
e. These Regulations shall come into force from the Academic Year 2026-2027.

4. Definitions

In these Regulations, unless the context otherwise requires:

a. "Academic Calendar" means the schedule of academic and miscellaneous events as approved by the Vice Chancellor;
b. "Academic Council" means the Academic Council of the University;
c. "Academic Regulations" means the Academic Regulations, of the University;
d. "Academic Term" means a Semester or Summer Term;
e. "Act" means the Presidency University Act, 2013;
f. "AICTE" means All India Council for Technical Education;
g. "Basket" means a group of courses bundled together based on the nature/type of the course;
h. "BOE" means the Board of Examinations of the University;
i. "BOG" means the Board of Governors of the University;
j. "BOM" means the Board of Management of the University;
k. "BOS" means the Board of Studies of a particular Department/Program of Study of the University;
l. "CGPA" means Cumulative Grade Point Average as defined in the Academic Regulations;
m. "Clause" means the duly numbered Clause, with Sub-Clauses included, if any, of these Regulations;
n. "COE" means the Controller of Examinations of the University;
o. "Course In Charge" means the teacher/faculty member responsible for developing and organising the delivery of the Course;
p. "Course Instructor" means the teacher/faculty member responsible for teaching and evaluation of a Course;
q. "Course" means a specific subject usually identified by its Course-code and

Course-title, with specified credits and syllabus/course-description, a set of references, taught by some teacher(s)/course-instructor(s) to a specific class (group of students) during a specific Academic Term;

r. "Curriculum Structure" means the Curriculum governing a specific Degree Program offered by the University, and, includes the set of Baskets of Courses along with minimum credit requirements to be earned under each basket for a degree/degree with specialization/minor/honours in addition to the relevant details of the Courses and Course catalogues (which describes the Course content and other important information about the Course). Any specific requirements for a particular program may be brought into the Curriculum structure of the specific program and relevant approvals should be taken from the BOS and Academic Council at that time.

s. "DAC" means the Departmental Academic Committee of a concerned Department/Program of Study of the University;

t. "Dean" means the Dean / Director of the concerned School;

u. "Degree Program" includes all Degree Programs;

v. "Department" means the Department offering the degree Program(s) / Course(s) / School offering the concerned Degree Programs / other Administrative Offices;

w. "Discipline" means specialization or branch of Master of Science Degree Program;

x. "HOD" means the Head of the concerned Department;

y. "L-T-P-C" means Lecture-Tutorial-Practical-Credit – refers to the teaching – learning periods and the credit associated;

z. "MOOC" means Massive Open Online Courses;

aa. "MOU" means the Memorandum of Understanding;

bb. "NPTEL" means National Program on Technology Enhanced Learning;

cc. "Parent Department" means the department that offers the Degree Program that a student undergoes;

dd. "Program Head" means the administrative head of a particular Degree Program/s;

ee. "Program Regulations" means the M.Sc. Degree Program Regulations and Curriculum, 2026-2028;

ff. "Program" means the Master of Science Degree Program;

gg. "PSIS" means the Presidency School of Information Science;

hh. "Registrar" means the Registrar of the University;

ii. "School" means a constituent institution of the University established for monitoring, supervising and guiding, teaching, training and research activities in broadly related fields of studies;

jj. "Section" means the duly numbered Section, with Clauses included in that Section, of these Regulations;

kk. "SGPA" means the Semester Grade Point Average as defined in the Academic Regulations.

ll. "Statutes" means the Statutes of Presidency University;

mm. "Sub-Clause" means the duly numbered Sub-Clause of these Program Regulations;

nn. "Summer Term" means an additional Academic Term conducted during the summer break (typically in June-July) for a duration of about eight (08) calendar weeks, with a minimum of thirty (30) University teaching days;

oo. "SWAYAM" means Study Webs of Active Learning for Young Aspiring Minds.

pp. "UGC" means University Grants Commission;

qq. "University" means Presidency University, Bengaluru; and

rr. "Vice Chancellor" means the Vice Chancellor of the University.

5. Program Description

The Programme shall be called Master of Science (Data Science), abbreviated as M.Sc. The M.Sc. Degree Program Regulations and Curriculum 2026-2028 are subject to, and, pursuant to the Academic Regulations. These Program Regulations shall be applicable to the ongoing M.Sc. Degree Program of 2026-2028 offered by the Presidency School of Information Science.

5.1 These Program Regulations shall be applicable to other similar programs, which may be introduced in future.

5.2 These Regulations may evolve and get amended or modified or changed through appropriate approvals from the Academic Council, from time to time, and shall be binding on all concerned.

5.3 The effect of periodic amendments or changes in the Program Regulations, on the students admitted in earlier years, shall be dealt with appropriately and carefully, so as to ensure that those students are not subjected to any unfair situation whatsoever, although they are required to conform to these revised Program Regulations, without any undue favour or considerations.

6. Minimum and Maximum Duration

6.1 M.Sc. Degree Program is a Two-Year, Full-Time Semester based program. The minimum duration of the M.Sc. Program is Two (02) years and each year comprises of two academic Semesters (Odd and Even Semesters) and hence the duration of the M.Sc. program is four (04) Semesters.

6.2 A student who for whatever reason is not able to complete the Program within the normal period or the minimum duration (number of years) prescribed for the Program, may be allowed a period of two years beyond the normal period to complete the mandatory minimum credits requirement as prescribed by the concerned Program Regulations and Curriculum. In general, the permissible maximum duration (number of years) for completion of Program is 'N' + 2 years, where 'N' stands for the normal or minimum duration (number of years) for completion of the concerned Program as prescribed by the concerned Program Regulations and Curriculum.

6.3 The time taken by the student to improve Grades/CGPA, and in case of temporary withdrawal/re-joining (Refer to Clause 16.1 of Academic Regulations), shall be counted in the permissible maximum duration for completion of a Program.

6.4 In exceptional circumstances, such as temporary withdrawal for medical exigencies where there is a prolonged hospitalization and/or treatment, as certified through hospital/medical records, women students requiring extended maternity break (certified by registered medical practitioner), and, outstanding sportspersons representing the University/State/India requiring extended time to participate in National/International sports events, a further extension of one (01) year may be granted on the approval of the Academic Council.

6.5 The enrolment of the student who fails to complete the mandatory requirements for the award of the concerned Degree (refer Section 19.0 of Academic Regulations) in the prescribed maximum duration (Clauses 18.1 and 18.2 of Academic Regulations), shall stand terminated and no Degree shall be awarded.

7. Programme Educational Objectives (PEO)

After 3 years of successful completion of the program, the graduates will:

PEO1: Establish themselves as competent Data Science professionals by applying advanced statistical, analytical, and computational techniques to solve complex real-world problems in industry, research, and societal domains.

PEO2: Engage in research, innovation, interdisciplinary learning, and continuous professional development through higher studies, certifications, or entrepreneurial initiatives to adapt to emerging technologies and evolving practices in Data Science.

PEO3: Demonstrate professional integrity, ethical responsibility, and sensitivity to societal, ecological, economic, and data-governance considerations while designing, developing, and deploying data-driven solutions.

8. Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

8.1 Programme Outcomes (PO):

On successful completion of the Program, the students shall be able to:

PO1: Domain Knowledge and Fundamentals: Apply advanced knowledge of mathematics, statistics, computer science, and Data Science principles to analyze, model, and interpret complex datasets.

PO2: Data Analysis and Modeling Skills: Design, develop, and evaluate data-driven models using appropriate statistical methods, machine learning, deep learning, and big data analytics techniques to derive actionable insights.

PO3: Problem Solving and Critical Thinking: Identify, formulate, and solve complex Data Science problems using analytical reasoning, algorithmic thinking, and interdisciplinary approaches.

PO4: Research and Innovation: Conduct independent and collaborative research involving data acquisition, experimentation, evaluation, and scholarly reporting, contributing to innovation and knowledge creation in Data Science.

PO5: Modern Tools and Technologies: Select and effectively use modern Data Science tools, programming frameworks, databases, cloud platforms, and data pipelines for scalable and efficient data-driven solutions.

PO6: Ethics, Professional Responsibility and Sustainability: Apply ethical principles, data privacy regulations, security practices, explainable AI concepts, and sustainability considerations in the development and deployment of data-driven systems.

PO7: Communication and Teamwork: Communicate analytical findings effectively through technical documentation, data visualizations, and presentations, and function efficiently as an individual or as a member of multidisciplinary teams.

PO8: Lifelong Learning and Professional Development: Recognize the need for and engage in continuous learning, professional development, and skill enhancement to adapt to emerging technologies and evolving industry and research requirements.

8.2 Program Specific Outcomes (PSOs):

On successful completion of the program, the graduates shall be able to:

PSO1: Apply advanced statistical, machine learning, and computational techniques to analyze complex datasets and generate validated insights for industry and research applications.

PSO2: Design, implement, and deploy scalable and secure data-driven solutions using modern programming frameworks, cloud platforms, and data engineering tools.

PSO3: Demonstrate ethical, professional, and responsible practices by adhering to data governance, privacy, security, and sustainability standards in data-driven systems.

9. Admission Criteria (as per the concerned Statutory Body)

The University admissions shall be open to all persons irrespective of caste, class, creed, gender or nation. All admissions shall be made on the basis of merit in the qualifying examinations; provided that forty percent of the admissions in all Programs of the University shall be reserved for the students of Karnataka State and admissions shall be made through a Common Entrance Examination conducted by the State Government or its agency and seats shall be allotted as per the merit and reservation policy of the State Government from time to time. The admission criteria to the M.Sc. Program are listed in the following Sub-Clauses:

9.1 An applicant who has successfully completed BCA / Bachelor's Degree in Computer Science Engineering or equivalent Degree, OR Passed B.Sc. / B.Com. / B.A. with Mathematics / Statistics / Computer Applications at 10+2 level.

9.2 Reservation for the SC / ST and other backward Sessions shall be made in accordance with the directives issued by the Government of Karnataka from time to time.

9.3 Admissions are offered to Foreign Nationals and Indians living abroad in accordance with the rules applicable for such admission, issued from time to time, by the Government of India.

9.4 Candidates must fulfil the medical standards required for admission as prescribed by the University.

9.5 If, at any time after admission, it is found that a candidate had not in fact fulfilled all the requirements stipulated in the offer of admission, in any form whatsoever, including possible misinformation and any other falsification, the Registrar shall report the matter to the Board of Management (BoM), recommending revoking the admission of the candidate.

9.6 The decision of the BoM regarding the admissions is final and binding.

10. Transfer Students requirements

Transfer of student(s) from another recognized University to the 2nd year (3rd Semester) of the M.Sc. Program of the University

A student who has completed the 1st Year (i.e., passed in all the Courses / Subjects prescribed for the 1st Year) of the M.Sc., two-Year Degree Program from another recognized University, may be permitted to transfer to the 2nd Year (3rd Semester) of the M.Sc. Program of the University as per the rules and guidelines prescribed in the following Sub-Clauses:

10.1.1 Candidates seeking transfer may be required to complete specified bridge Courses, if any, as prescribed by the University. Such bridge Courses shall not be included in the CGPA computations.

10.1.2 All the existing Regulations and Policies of the University shall be binding on all the students admitted to the Program through the provision of transfer.

10.1.3 The student shall submit the Application for Transfer along with a non-refundable Application Fee (as prescribed by the University from time to time) to the Presidency University no later than the last date of registration of the concerned year/semester for admission to the 2nd Year (3rd Semester) M.Sc. Program commencing on August 1 on the year concerned.

10.1.4 The student shall submit copies of the respective Marks Cards / Grade Sheets / Certificates along with the Application for Transfer.

The transfer may be provided on the condition that the Courses and Credits completed by the concerned student in the 1st Year of the M.Sc. Degree Program from the concerned University, are declared equivalent and acceptable by the Equivalence Committee constituted by the Vice Chancellor for this purpose. Further, the Equivalence Committee may also prescribe the Courses and Credits the concerned students shall

have to mandatorily complete, if admitted to the 2nd Year of the M.Sc. Program of the University.

11. Mandatory Bridge Course for Students from Non-Computer Science Disciplines

Students who have studied Mathematics at the 10+2 level or at the graduation level, but have not undergone Computer Science related courses, are required to undergo the Bridge Courses prescribed by the school before starting of the regular first semester classes to facilitate them with foundational knowledge in computer science.

12. Specific Regulations regarding Assessment and Evaluation (including the Assessment Details of NTCC Courses, Weightages of Continuous Assessment and End Term Examination for various Course Categories)

12.1 The academic performance evaluation of a student in a Course shall be according to the University Letter Grading System based on the class performance distribution in the Course.

12.2 Academic performance evaluation of every registered student in every Course registered by the student is carried out through various components of Assessments spread across the Semester. The nature of components of Continuous Assessments and the weightage given to each component of Continuous Assessments (refer Clause 8.8 of Academic Regulations) shall be clearly defined in the Course Plan for every Course, and approved by the DAC.

12.3 Format of the End-Term examination shall be specified in the Course Plan.

12.4 Grading is the process of rewarding the students for their overall performance in each Course. The University follows the system of Relative Grading with statistical approach to classify the students based on the relative performance of the students registered in the concerned Course except in the following cases:

- Non-Teaching Credit Courses (NTCC)
- Courses with a class strength less than 30

Absolute grading method may be adopted, where necessary with prior approval of concerned DAC.

Grading shall be done at the end of the Academic Term by considering the aggregate performance of the student in all components of Assessments prescribed for the Course. Letter Grade (Clause 8.10 of Academic Regulations) shall be awarded to a student based on her/his overall performance relative to the class performance distribution in the concerned Course. These Letter Grades not only indicate a qualitative assessment of the student's performance but also carry a quantitative (numeric) equivalent called the Grade Point.

12.5 Assessment Components and Weightage

Table 1: Assessment Components and Weightage for different category of Courses		
Nature of Course and Structure	**Evaluation Component**	**Weightage**
Lecture-based Course L component in the L-T-P Structure is predominant (more than 1) (Examples: 3-0-0; 3-0-2; 2-1-0; 2-0-2, 2-0-4 etc.)	Continuous Assessments	50%
	End Term Examination	50%
Lab/Practice-based Course P component in the L-T-P Structure is predominant (Examples: 0-0-4; 1-0-4; 1-0-2; etc.)	Continuous Assessments	75%
	End Term Examination	25%
Skill based Courses like Industry Internship, Major project, Research Dissertation, Integrative Studio, Interdisciplinary Project, Summer / Short Internship, Social Engagement / Field Projects, Portfolio, and such similar Non-Teaching Credit Courses, where the pedagogy does not lend itself to a typical L-T-P structure	Guidelines for the assessment components for the various types of Courses, with recommended weightages, shall be specified in the concerned Program Regulations and Curriculum / Course Plans, as applicable.	

The exact weightages of Evaluation Components shall be clearly specified in the concerned PRC and respective Course Plan.

Normally, for Practice/Skill based Courses, without a defined credit structure (L–T–P) [NTCC], but with assigned Credits (as defined in Clause 5.2 of the Academic Regulations), the method of evaluation shall be based only on Continuous Assessments. The various components of Continuous Assessments, the distribution of weightage among such components, and the method of evaluation/assessment, shall be as decided and indicated in the Course Plan/PRC. The same shall be approved by the respective DAC.

12.6 Minimum Performance Criteria:

12.6.1 Theory only Course and Lab/Practice Embedded Theory Course

A. A student shall satisfy the following minimum performance criteria to be eligible to earn the credits towards the concerned Course:
B. A student must obtain a minimum of 30% of the total marks/weightage assigned to the End Term Examinations in the concerned Course.
C. The student must obtain a minimum of 40% of the AGGREGATE of the marks/weightage of the components of Continuous Assessments, Mid Term Examinations and End Term Examinations in the concerned Course.

12.6.2 Lab/Practice only Course and Project Based Courses

The student must obtain a minimum of 40% of the AGGREGATE of the marks/weightage of all assessment components in the concerned Course.

A student who fails to meet the minimum performance criteria listed above in a Course shall be declared as "Fail" and given "F" Grade in the concerned Course. For theory Courses, the student shall have to re-appear in the "Make-Up Examinations" as scheduled by the University in any subsequent semester, or, re-appear in the End Term Examinations of the same Course when it is scheduled at the end of the following Semester or Summer Term, if offered. The marks obtained in the Continuous Assessments (other than the End Term Examination) shall be carried forward and be included in computing the final grade, if the student secures the minimum requirements (as per the sub-clauses 8.9.1 and 8.9.2 of Academic Regulations) in the "Make-Up Examinations" of the concerned Course. Further, the student has an option to re-register for the Course and clear the same in the summer term/ subsequent semester if he/she wishes to do so, provided the Course is offered.

13. Additional clarifications - Rules and Guidelines for Transfer of Credits from MOOC, etc. – Note: These are covered in Academic Regulations

The University allows students to acquire credits from other Indian or foreign institutions and/or Massive Open Online Course (MOOC) platforms, subject to prior approval. These credits may be transferred and counted toward fulfilling the minimum credit requirements for the award of a degree. The process of transfer of credits is governed by the following rules and guidelines:

13.1 The transfer of credits shall be examined and recommended by the Equivalence Committee (Refer ANNEXURE B of Academic Regulations) and approved by the Dean - Academics.

13.2 Students may earn credits from other Indian or foreign Universities/Institutions with which the University has an MOU, and that MOU shall have specific provisions, rules and guidelines for transfer of credits. These transferred credits shall be counted towards the minimum credit requirements for the award of the degree.

13.3 Students may earn credits by registering for Online Courses offered by *Study Web of Active Learning by Young and Aspiring Minds* (SWAYAM) and *National Program on Technology Enhanced Learning* (NPTEL), or other such recognized Bodies/ Universities/Institutions as approved by the concerned BOS and Academic Council from time to time. The concerned School/Parent Department shall publish/include the approved list of Courses and the rules and guidelines governing such transfer of credits of the concerned Program from time to time. The Rules and Guidelines for the transfer of credits specifically from the Online Courses conducted by SWAYAM/ NPTEL/ other approved MOOCs are as stated in the following Sub-Clauses:

13.3.1 A student may complete SWAYAM/NPTEL/other approved MOOCs as mentioned in Clause 13.2 (as per Academic Regulations) and transfer equivalent credits to partially or fully complete the mandatory credit requirements of Discipline Elective Courses and/or the mandatory credit requirements of Open Elective Courses as prescribed in the concerned Curriculum Structure. However, it is the sole responsibility of the student to complete the mandatory credit requirements of the Discipline Elective Courses and the Open Elective Courses as prescribed by the Curriculum Structure of the concerned Program.

13.3.2 SWAYAM/NPTEL/ other approved MOOCs as mentioned in Clause 13.2 (as per Academic Regulations) shall be approved by the concerned Board of Studies and placed (as Annexures) in the concerned PRC.

13.3.3 Parent Departments may release a list of SWAYAM/NPTEL/other approved MOOCs for Pre-Registration as per schedule in the Academic Calendar or through University Notification to this effect.

13.3.4 Students may Pre-Register for the SWAYAM/NPTEL/other approved MOOCs in the respective Departments and register for the same Courses as per the schedule announced by respective Online Course Offering body/institute/ university.

13.3.5 A student shall request for transfer of credits only from such approved Courses as mentioned in Sub-Clause 13.3.1 above.

13.3.6 SWAYAM/NPTEL/other approved MOOCs Courses are considered for transfer of credits only if the concerned student has successfully completed the SWAYAM/NPTEL/other approved MOOCs and obtained a certificate of successful/satisfactory completion.

13.3.7 A student who has successfully completed the approved SWAYAM/NPTEL/ other approved MOOCs and wants to avail the provision of transfer of equivalent credits, must submit the original Certificate of Completion, or such similar authorized documents to the HOD concerned, with a written request for the transfer of the equivalent credits. On verification of the Certificates/Documents and approval by the HOD concerned, the Course(s) and equivalent Credits shall be forwarded to the COE for processing of results of the concerned Academic Term.

13.3.8 The credit equivalence of the SWAYAM/NPTEL/other approved MOOCs are based on Course durations and/or as recommended by the Course offering body/institute/university. The Credit Equivalence mapped to SWAYAM/ NPTEL approved Courses based on Course durations for transfer of credits is summarised in Table shown below. The Grade will be calculated from the marks received by the Absolute Grading Table 8.11 in the Academic Regulations.

Table 2: Durations and Credit Equivalence for Transfer of Credits from SWAYAM-NPTEL/ other approved MOOC Courses		
Sl. No.	**Course Duration**	**Credit Equivalence**
1	4 Weeks	1 Credit
2	8 Weeks	2 Credits
3	12 Weeks	3 Credits

13.3.9 The maximum permissible number of credits that a student may request for credit transfer from MOOCs shall not exceed 20% of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree.

13.3.10 The University shall not reimburse any fees/expense; a student may incur for the SWAYAM/NPTEL/other approved MOOCs.

13.4 The maximum number of credits that can be transferred by a student shall be limited to forty percent (40%) of the mandatory minimum credit requirements specified by the concerned Program Regulations and Curriculum for the award of the concerned Degree. However, the grades obtained in the Courses transferred from other Institutions/MOOCs, as mentioned in this Section (13), shall not be included in the calculation of the CGPA.

13.5 Mandatory Non-Credit Course Completion Requirements: All mandatory non-credit courses shall be satisfactorily completed by the student as part of the degree requirements. These courses will be evaluated and awarded letter grades based on the following criteria:

- S (Satisfactorily Completed): Awarded when the student successfully completes all prescribed course requirements.
- NC (Not Completed): Awarded when the student fails to meet the prescribed course requirements.

A student receiving an NC grade must reappear for and complete the course in accordance with the guidelines prescribed by the University.

In the case of non-taught and non-credited mandatory courses—where students are advised to undertake learning through MOOC platforms—there shall be a clearly defined Course Catalogue and a corresponding Course Plan. The Course Plan shall outline the assessment components, which will form the basis for evaluation.

PART B: PROGRAM STRUCTURE

14. Structure / Component with Credit Requirements Course Baskets & Minimum Basket wise Credit Requirements

The **M.Sc.** Program Structure (**2026-2028**) totalling 80 credits. Table 3 summarizes the type of baskets, number of courses under each basket and the associated credits that are mandatorily required for the completion of the Degree.

Table 3: M.Sc. 2026–2028
Summary of Mandatory Courses and Minimum Credit Contribution from Various Baskets

Sl. No.	Baskets	Credit Contribution
1	Program Core (PC)	46
2	Elective Courses (EC)	15
3	Project Work (Mini + Major)	16
4	Foundation Courses (FC)	3
	Total Credits	**80**

In the entire Program, the practical and skill-based course component contribute to an extent of approximately 82% out of the total credits of 80 for **M.Sc.** Computer Applications program of Two-year duration.

15. Minimum Total Credit Requirements of Award of Degree

A minimum of 80 credits is required for the award of an **M.Sc.** degree.

16. Other Specific Requirements for Award of Degree, if any, as prescribed by the Statutory Bodies,

16.1 The award of the Degree shall be recommended by the Board of Examinations and approved by the Academic Council and Board of Management of the University**.**

16.2 A student shall be declared to be eligible for the award of the concerned Degree if she/he:

a. Fulfilled the Minimum Credit Requirements and the Minimum Credits requirements under various baskets;
b. Secure a minimum CGPA of 5.0 in the concerned Program at the end of the Semester/Academic Term in which she/he completes all the requirements for the award of the Degree as specified in Sub-Clause 19.2.1 of Academic Regulations**;**
c. No dues to the University, Departments, Hostels, Library, and any other such Centers/ Departments of the University; and
d. No disciplinary action is pending against her/him.

PART C: CURRICULUM STRUCTURE

17.Curriculum Structure – Basket Wise Course List

Table 3.1: List of Program Core

Sl. No.	Course Code	Course Name	L	T	P	C
Program Core						
		Theory Course				
1.	MAT4014	Mathematics for Data Science	3	1	0	4
2.	CSC4201	Data Structures and Algorithms	3	1	0	4
3.	CSC4202	Relational and NoSQL Database Systems	3	0	0	3
		Practical Course				
4.	CSC4203	Cloud Data Platforms and Architecture	2	0	2	3
5.	CSC4204	Python Programming for Data Science	1	0	4	3
6.	CSC4205	Data Structures & Algorithms Lab	0	0	2	1
7.	CSC4206	Database Systems Laboratory	0	0	2	1
8.	CSC4207	R Programming	2	0	2	3
9.	CSC4208	Full Stack Development	2	0	4	4
10.	CSC4209	Applied Machine Learning	2	0	2	3
11.	CSC4210	Data Analysis and Visualization Techniques	2	0	2	3
12.	CSC4211	Web Application Development	1	0	4	3
13.	CSC4212	Deep learning	2	0	4	4
14.	CSC4213	Big Data Analytics	3	0	2	4
15.	CSC4214	Data Pipelines & ETL	2	0	2	3
Total						46

Table 3.2: List of Project Work

Sl. No.	Course Code	Course Name	L	T	P	C
Project Courses						
1	CSC8300	Mini Project (Application / Research Oriented)	0	0	0	4
2	CSC8400	Major Project / Dissertation	0	0	0	12
Total						16

Table 3.3: List of Foundation Courses

Sl. No.	Course Code	Course Name	L	T	P	C
Foundation Courses						
1.	CSC4301	Research Methodology	2	0	0	2
2.	PPS4008	Quantitative Skills and Logical Reasoning	0	0	2	1
Total						3

18. Practical / Skill based Courses – Internships / Thesis / Dissertation / Major Project Work / Portfolio / Mini project

Practical / Skill based Courses like internship, project work, major project, research project / dissertation, and such similar courses, where the pedagogy does not lend itself to a typical L-T-P-C Structure as defined in Clause 5.1 of the Academic Regulations, are simply assigned the number of Credits based on the quantum of work / effort required to fulfill the learning objectives and outcomes prescribed for the concerned Courses. Such courses are referred to as Non-Teaching Credit Courses (NTCC). These Courses are designed to provide students with hands-on experience and skills essential for their professional development. These courses aim to equip students with abilities in problem identification, root cause analysis, problem-solving, innovation, and design thinking through industry exposure and project-based learning. The expected outcomes are first level proficiency in problem solving and design thinking skills to better equip M.Sc. graduates for their professional careers. The method of evaluation and grading for the Practical / Skill based Courses shall be prescribed and approved by the concerned Departmental Academic Committee (refer Annexure A of the Academic Regulations). The same shall be prescribed in the Course Handout.

18.1 Mini Project

A student may opt to do a Project Work for a period of 6-8 weeks in an Industry / Company or academic / research institution or the University Department(s) as an equivalence of Internship during the 3rd Semester as applicable, subject to the following conditions:

18.1.1 The Project Work shall be approved by the concerned HOD and be carried out under the guidance of a faculty member.

18.1.2 The student may do the project work in an Industry / Company or academic / research institution of her / his choice subject to the above-mentioned condition (Sub-Clause 18.1.1). Provided further, that the Industry / Company or academic / research institution offering such project work confirms to the University that the project work will be conducted in accordance with the Program Regulations and requirements of the University.

18.2 Major Project

A student may undergo a Project for a period of 12-14 weeks in an industry / company or academic / research institution in the 4th Semester as applicable, subject to the following conditions:

18.2.1 The Project shall be conducted in accordance with the Project Policy prescribed by the University from time to time.

18.2.2 The selection criteria (minimum CGPA, pass in all Courses as on date, and any other qualifying criteria) as applicable / stipulated by the concerned Industry / Company or academic / research institution for award of the Major Project to a student;

18.2.3 The number of Project available for the concerned Academic Term.

Further, the available number of Major Project shall be awarded to the students by the University on the basis of merit using the CGPA secured by the student. Provided further, the student fulfils the criteria, as applicable, specified by the Industry / Company or academic / research institution providing the Major Project, as stated in Sub-Clause 18.2.2 above.

18.2.4 A student may opt for Project in an Industry / Company or academic / research institution of her / his choice, subject to the condition that the concerned student takes the responsibility to arrange the I Project on her / his own. Provided further, that the Industry / Company or academic / research institution offering such Major Project confirms to the University that the Major Project shall be conducted in accordance with the Program Regulations and Internship Policy of the University.

18.2.5 A student selected for a Project in an industry / company or academic / research institution shall adhere to all the rules and guidelines prescribed in the Major Project Policy of the University.

18.3 Research Project / Dissertation

A student may opt to do a Research Project / Dissertation for a period of 12-14 weeks in an Industry / Company or academic / research institution or the University Department(s) as an equivalence of Major Project, subject to the following conditions:

18.3.1 The Research Project / Dissertation shall be approved by the concerned HOD and be carried out under the guidance of a faculty member.

18.3.2 The student may do the Research Project / Dissertation in an Industry / Company or academic / research institution of her / his choice subject to the above-mentioned condition (Sub-Clause 18.3.1). Provided further, that the Industry / Company or academic / research institution offering such Research Project / Dissertation confirms to the University that the Research Project / Dissertation work will be conducted in accordance with the Program Regulations and requirements of the University.

19. List of Elective Courses under various Specialisations / Stream Basket

Table 3.4: List of Discipline Electives

ELECTIVE BASKET–I: AI & ML

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC4701	Natural Language Processing	3	0	0	3	3	EC
2.	CSC4702	Computer Vision	3	0	0	3	3	EC
3.	CSC4703	Generative AI	3	0	0	3	3	EC
4.	CSC4704	Reinforcement Learning	3	0	0	3	3	EC
5.	CSC4705	Explainable & Responsible AI	3	0	0	3	3	EC

ELECTIVE BASKET–II: DATA SCIENCE & ANALYTICS

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC4706	Time Series Analysis and Forecasting	3	0	0	3	3	EC
2.	CSC4707	Predictive Analytics & Decision Models	3	0	0	3	3	EC
3.	CSC4708	Social Network Analysis	3	0	0	3	3	EC
4.	CSC4709	Geospatial Analytics	3	0	0	3	3	EC
5.	CSC4710	Bioinformatics	3	0	0	3	3	EC

ELECTIVE BASKET–III: CLOUD & DISTRIBUTED SYSTEMS

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC4711	Cloud DevOps	3	0	0	3	3	EC
2.	CSC4712	Blockchain Technologies	3	0	0	3	3	EC
3.	CSC4713	IoT Analytics	3	0	0	3	3	EC
4.	CSC4714	Edge & Fog Computing	3	0	0	3	3	EC
5.	CSC4715	Cyber Security for Data Platforms	3	0	0	3	3	EC

ELECTIVE BASKET–IV: DECISION SCIENCES AND APPLIED ANALYTICS

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC4716	Decision Science and Optimization Techniques	3	0	0	3	3	EC
2.	CSC4717	Operations Research and Supply Chain Analytics	3	0	0	3	3	EC
3.	CSC4718	Financial Analytics and Risk Modelling	3	0	0	3	3	EC
4.	CSC4719	Applied Business Analytics	3	0	0	3	3	EC
5.	CSC4720	Design of Experiments and Statistical Quality Control	3	0	0	3	3	EC

20.List of MOOC (NPTEL) Courses

Presidency University students are given the opportunity to study abroad in International Universities through a selection process coordinated by the Office of International Affairs (OIA). Such selected students need to complete their credits for the semester that they are abroad in the following way:

<table>
<tr><td>20.1.1</td><td>The student needs to study and complete School Core and Program Core Courses in offline mode only.</td></tr>
<tr><td>20.1.2</td><td>Massive Open Online Course (MOOC) courses may be given for Open Elective and Discipline Elective Courses. These courses need to be approved by the concerned BOS and Academic Council from time to time.</td></tr>
<tr><td>20.1.3</td><td>SWAYAM/NPTEL/ other approved MOOCs shall be approved by the concerned Board of Studies and placed in the concerned PRC.</td></tr>
<tr><td>20.1.4</td><td>Student shall register for these courses in the ERP of Presidency University.</td></tr>
<tr><td>20.1.5</td><td>For these MOOC courses faculty coordinators are identified. These faculty should have undergone similar MOOC courses and therefore should be familiar with the mode of class conduction, types of assessments and evaluation procedures.</td></tr>
<tr><td>20.1.6</td><td>Study materials shall be provided to the students as video lectures shared by the MOOCs Coordinator(s), or the students may access the approved MOOCs Portal directly. The mode of class conduction is determined by the MOOCs coordinator(s) as detailed in the Course Catalogue and Course Plan.</td></tr>
<tr><td>20.1.7</td><td>The question paper shall be prepared by the MOOCs coordinator(s).</td></tr>
<tr><td>20.1.8</td><td>Students write the exams in online mode. These exams are scheduled and conducted by the school.</td></tr>
<tr><td>20.1.9</td><td>Results are evaluated by School and given to the Office of the Controller of Examinations (CoE).</td></tr>
<tr><td>20.1.10</td><td>The details of the duration, credits and evaluation are given below:
<table>
<tr><th>Sl.No.</th><th>Duration</th><th>Credits</th><th>Evaluation</th></tr>
<tr><td>1.</td><td>12 weeks</td><td>3</td><td>Continuous Assessment – 50 Marks
Mid Term –50 Marks
End Term-100 Marks</td></tr>
<tr><td>2.</td><td>8 weeks</td><td>2</td><td>Mid Term-50 Marks
End Term-100 Marks</td></tr>
<tr><td>3.</td><td>4 weeks</td><td>1</td><td>End Term-100 Marks</td></tr>
</table>
</td></tr>
</table>

21. List of Open Electives to be offered by the School / Department:

Table 3.5: List of Open Electives

Sl. No.	Course Code	Course Name	L	T	P	C
		Civil Engineering Basket				
1.	CIV5001	Sustainable Smart Cities	3	0	0	3
2.	CIV5002	Systems Design for Sustainability	3	0	0	3
3.	CIV5003	Self-Sustainable Buildings	3	0	0	3
4.	CIV5004	Energy and Buildings	3	0	0	3
		Law Basket				
1.	LAW5001	International Trade Law	3	0	0	3
2.	LAW5002	Law relating to Business Establishment	3	0	0	3
3.	LAW5003	Data Protection Law	3	0	0	3
4.	LAW5004	Law Relating to Consumer Protection	3	0	0	3
5.	LAW5005	Law Relating to Infrastructure Projects	3	0	0	3
		Electronics and Communication Engineering Basket				
1.	ECE5001	Wearable Computing	3	0	0	3
2.	ECE5002	MEMS and Nanotechnology	3	0	0	3
3.	ECE5003	Computer Networks	3	0	0	3
4	ECE5004	Pervasive Computing	3	0	0	3
		Mechanical Engineering Basket				
1	MEC5001	Optimization Techniques	3	0	0	3
2	MEC5002	Industry 4.0	3	0	0	3
3	MEC5003	Six Sigma for Engineers	3	0	0	3
4	MEC5004	Design for Internet of Things	3	0	0	3
5	MEC2003	Supply Chain Management	3	0	0	3
		Management Basket				
1.	MBA3026	Essentials of Leadership	3	0	0	3
2.	MBA3037	Fundamentals of Accounting	3	0	0	3
3.	MBA3038	Sales Techniques	3	0	0	3
4.	MBA3039	Principles of Management	3	0	0	3
		Media Studies Basket				
1	BAJ5001	Media and Entertainment Business	3	0	0	3
2	BAJ5002	TV Journalism and News Management	2	0	2	3
		Research Basket				
1.	RES5001	Research Methodology	3	0	0	3
2.	URE7001	University Research Experience	-	-	-	3
3.	URE7002	University Research Experience	-	-	-	0

22. Recommended Semester Wise Course Structure / Flow including the Programme / Discipline Elective Paths / Options

Semester 1:

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	MAT4014	Mathematics for Data Science	3	1	0	4	4	PC
2.	CSC4201	Data Structures and Algorithms	3	1	0	4	4	PC
3.	CSC4202	Relational and NoSQL Database Systems	3	0	0	3	3	PC
4.	CSC4203	Cloud Data Platforms and Architecture	2	0	2	3	4	PC
5.	CSC4204	Python Programming for Data Science	1	0	4	3	5	PC
6.	CSC4205	Data Structures & Algorithms Lab	0	0	2	1	2	PC
7.	CSC4206	Database Systems Laboratory	0	0	2	1	2	PC
8.	CSC4301	Research Methodology	2	0	0	2	2	FC
9.	PPS4008	Quantitative Skills and Logical Reasoning	0	0	2	1	2	FC
TOTAL			14	2	12	22	28	

Semester 2:

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC4207	R Programming	2	0	2	3	4	PC
2.	CSC4208	Full Stack Development	2	0	4	4	6	PC
3.	CSC4209	Applied Machine Learning	2	0	2	3	4	PC
4.	CSC4210	Data Analysis and Visualization Techniques	2	0	2	3	4	PC
5.	CSC4211	Web Application Development	1	0	4	3	5	PC
6.	-	Elective 1	3	0	0	3	3	EC
7.	-	Elective 2	3	0	0	3	3	EC
TOTAL			15	0	14	22	29	

Semester 3:

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC4212	Deep learning	2	0	4	4	6	PC
2.	CSC4213	Big Data Analytics	3	0	2	4	5	PC
3.	CSC4214	Data Pipelines & ETL	2	0	2	3	4	PC
4.	-	Elective 3	3	0	0	3	3	EC
5.	-	Elective 4	3	0	0	3	3	EC
6.	-	Elective 5	3	0	0	3	3	EC
7.	CSC8300	Mini Project (Application / Research Oriented)	0	0	0	4	–	Project
TOTAL			16	0	8	24	24	

Semester 4:

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1.	CSC8400	Major Project / Dissertation	0	0	0	12	0	Project
TOTAL			0	0	0	12	0	

ELECTIVE BASKET–I: AI & ML

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1	CSC4701	Natural Language Processing	3	0	0	3	3	EC
2	CSC4702	Computer Vision	3	0	0	3	3	EC
3	CSC4703	Generative AI	3	0	0	3	3	EC
4	CSC4704	Reinforcement Learning	3	0	0	3	3	EC
5	CSC4705	Explainable & Responsible AI	3	0	0	3	3	EC

ELECTIVE BASKET–II: DATA SCIENCE & ANALYTICS

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1	CSC4706	Time Series Analysis and Forecasting	3	0	0	3	3	EC
2	CSC4707	Predictive Analytics & Decision Models	3	0	0	3	3	EC
3	CSC4708	Social Network Analysis	3	0	0	3	3	EC
4	CSC4709	Geospatial Analytics	3	0	0	3	3	EC
5	CSC4710	Bioinformatics	3	0	0	3	3	EC

ELECTIVE BASKET–III: CLOUD & DISTRIBUTED SYSTEMS

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1	CSC4711	Cloud DevOps	3	0	0	3	3	EC
2	CSC4712	Blockchain Technologies	3	0	0	3	3	EC
3	CSC4713	IoT Analytics	3	0	0	3	3	EC
4	CSC4714	Edge & Fog Computing	3	0	0	3	3	EC
5	CSC4715	Cyber Security for Data Platforms	3	0	0	3	3	EC

ELECTIVE BASKET–IV: DECISION SCIENCES AND APPLIED ANALYTICS

Sl. No.	Course Code	Course Name	Credit Structure				Contact Hours	Type of Course
			L	T	P	C		
1	CSC4716	Decision Science and Optimization Techniques	3	0	0	3	3	EC
2	CSC4717	Operations Research and Supply Chain Analytics	3	0	0	3	3	EC
3	CSC4718	Financial Analytics and Risk Modelling	3	0	0	3	3	EC
4	CSC4719	Applied Business Analytics	3	0	0	3	3	EC
5	CSC4720	Design of Experiments and Statistical Quality Control	3	0	0	3	3	EC

23.Course Catalogue

Course Catalogue of all Courses Listed including the Courses Offered by other School / Department and Discipline / Programme Electives – Course Code, Course Name, Prerequisite, Anti-requisite, Course Description, Course Outcome, Course Content (with Blooms Level, CO, No. of Contact Hours), Reference Resources.

SEMESTER I
Mathematics for Data Science

Course Code: MAT4014	**Course Title: Mathematics for Data Science** **Type of Course: School Core**	**L-T- P- C**	**3**	**1**	**0**	**4**
Version No.	1.0					
Course Pre-requisites	NIL					
Anti-requisites	NIL					
Course Description	This course introduces the mathematical and statistical foundations of Data Science. It covers linear algebra, descriptive statistics, probability distributions, sampling methods, hypothesis testing, confidence intervals, and ANOVA. Students learn techniques for data analysis, modeling, and interpretation using real-world datasets. The course also introduces Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) for dimensionality reduction. Practical implementation in Python or R enables students to perform statistical analysis, visualization, and data-driven decision-making. The course prepares learners for applications in data science.					
Course Objective	This course aims to provide a strong foundation in linear algebra, probability, and statistics for data analysis and machine learning applications. It develops the ability to analyze, model, and interpret data using statistical and mathematical techniques. The course introduces dimensionality reduction methods such as SVD and PCA and enables students to apply inferential statistical methods for data-driven decision-making. Through Python/R-based practical exercises, students gain hands-on experience in data analysis, visualization, and predictive modelling.					
Course Outcomes	On successful completion of the course the students shall be able to: **CO1:** Apply linear algebra concepts and matrix operations in data science applications. [L3] **CO2:** Employ SVD and PCA techniques for dimensionality reduction and feature extraction. [L3] **CO3:** Apply descriptive statistical methods to summarize and visualize datasets. [L3] **CO4:** Employ probability distributions and Bayes' theorem to solve data-driven problems. [L3] **CO5:** Apply sampling methods, confidence intervals, and hypothesis testing procedures for statistical inference. [L3] **CO6:** Apply regression analysis, ANOVA, and statistical computing techniques to analyze datasets and obtain insights. [L3]					
Course Content:						
Module 1	**Linear Algebra and Matrix Computations for Data Science**	**(14L+5T=19 Sessions)**				

Vector Spaces, Subspaces, Basis and Dimension, Coordinates and Change of Basis, Orthogonal bases and orthogonal projections.
Linear Transformations, The Null Space and Range, Isomorphisms, Matrix Representation, Similarity, Eigenvalues and Eigenvectors, Diagonalization- Diagonalization of Symmetric Matrices.
Singular Value Decomposition: Singular values, computing singular value decomposition, and Introduction to principal component analysis. SVD and PCA are used for dimensionality reduction and feature extraction.

Module 2	**Descriptive Statistics**	**(8L+3T=11 Sessions)**

Measures of Central Tendency, Moments, Skewness and Kurtosis, Correlation - Karl Pearson's coefficient of correlation and rank correlation (with & without repetition, Multiple Correlation - Problems: regression analysis - lines of regression, Multiple regression - Problems. Data interpretation, relationship analysis, and predictive modelling.

Module 3	**Probability Distribution**	**(9L+3T=12 Sessions)**

Introduction to Probability, Conditional Probability and Independence. Bayes' Theorem - Intuition & Applications (Medical Testing, Spam Filtering). Discrete Probability Distributions - The Binomial Distribution, Bernoulli. Continuous Probability Distributions- The Normal Distribution. The Standard Normal Distribution (Z-scores). Statistical inference, risk assessment, and decision-making under uncertainty.

Module 4	**Inferential Statistics**	**(15L+5T =18 Sessions)**

Sampling, population, sample, parameter, statistic, types of sampling techniques, Sampling Distribution, Central Limit Theorem, Hypothesis Testing, Two-Sample Tests, ANOVA: F-test for comparing more than two population means, Confidence Interval Estimation.
Introduction to Sample Surveys: Simple Random Sampling, Stratified Random Sampling, and other sampling schemes; Analysis of Variance (ANOVA); data sanitization and preprocessing; data analysis and interpretation using R or Python.

Targeted Application & Tools that can be used:

- Data representation in high-dimensional spaces, dimensionality reduction, feature extraction, Principal Component Analysis (PCA), image compression, recommendation systems, machine learning, and pattern recognition.
- Data analysis and visualization, correlation and regression analysis, trend identification, predictive modeling, business analytics, and exploratory data analysis.
- Statistical inference, risk assessment, reliability analysis, Bayesian decision-making, uncertainty quantification, medical diagnostics, spam filtering, and predictive analytics.
- Hypothesis testing, confidence interval estimation, ANOVA, survey analysis, sampling design, data sanitization and preprocessing, data-driven decision-making, and machine learning model evaluation.

Software: Python, R Software, Jupyter Notebook, Google Colab, NumPy, Pandas, SciPy, Matplotlib, scikit-learn.

Assignment:

Assignments will involve theoretical problem-solving and computational implementation using suitable programming tools. The assignments may be undertaken individually or in groups, as decided by the instructor.

Text Book

1. G. Strang, Introduction to Linear Algebra, 6th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2023.

2. S. M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 7th ed. Amsterdam, Netherlands: Elsevier, 2021.
3. P. Bruce, A. Bruce, and P. Gedeck, Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2020.
4. J. K. Sharma, Fundamentals of Mathematical Statistics, 9th ed. Noida, Uttar Pradesh, India: Pearson India Education Services Pvt. Ltd., 2024.

References:

1. K. P. Murphy, Probabilistic Machine Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2022.
2. D. C. Lay, S. R. Lay, and J. J. McDonald, Linear Algebra and Its Applications, 6th ed. New York, NY, USA: Pearson, 2021.
3. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.
4. C. M. Bishop and H. Bishop, Deep Learning: Foundations and Concepts. Cham, Switzerland: Springer Nature, 2024.
5. J. K. Sharma, Fundamentals of Mathematical Statistics, 9th ed. Noida, Uttar Pradesh, India: Pearson India Education Services Pvt. Ltd., 2024.
6. B. S. Grewal, Higher Engineering Mathematics, 45th ed. New Delhi, India: Khanna Book Publishing Co. (P) Ltd., 2022.
7. T. M. Cover and J. A. Thomas, Elements of Information Theory, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2024.

E-resources/ Web links:

1. NPTEL – Linear Algebra and Matrix Analysis for Statistics
2. NPTEL – Probability and Statistics
3. MIT OpenCourseWare – Linear Algebra
4. OpenStax – Introductory Statistics
5. https://presiuniv.knimbus.com/user#/home
6. MIT OpenCourseWare – Probabilistic Systems Analysis and Applied Probability
7. NPTEL – Data Analytics with Python

Topics relevant to SKILL DEVELOPMENT: The course focuses on the concepts of calculus and differential equation with reference to specific engineering problems. The course is of both conceptual and analytical type in nature through Problem solving**.** This is attained through the assessment component mentioned in course handout.

Catalogue prepared by	**Dr. V Ramalatha**
Recommended by the Board of Studies on	
Date of Approval by the Academic Council	

CSC4201-Data Structure and Algorithms

Course Code: CSC4201	**Course Title: Data Structure and Algorithms Type of Course: Program Core & Theory**	**L-T-P-C**	3	1	0	4
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course provides an in-depth study of advanced data structures and algorithmic techniques essential for efficient problem solving in computer science. It focuses on abstract data types (ADTs), mathematical foundations, and the selection of appropriate data structures for real-world applications. The course covers advanced topics such as hashing techniques, priority queues, balanced and multiway trees, disjoint sets, and string-matching algorithms. Emphasis is placed on both theoretical understanding and practical implementation using standard libraries and programming tasks. Through analytical problem solving and hands-on exercises, learners develop the ability to design, analyze, and optimize algorithms for complex computational problems.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Advanced Data Structure and Algorithms** and attain **Skill Development** through **Experiential Learning** techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply fundamental concepts of data structures and analyze their role in efficient problem solving. [Apply] **CO2:** Apply hashing techniques including collision resolution methods such as linear probing, quadratic probing, double hashing, and rehashing**.** [Apply] **CO3:** Analyze and implement priority queues using binary heaps and binomial queues for efficient data management. [Analyze] **CO4:** Analyze tree-based data structures such as binary search trees, AVL trees, 2-3 trees, and red-black trees with their operations. [Analyze] **CO5:** Apply disjoint set operations using union-find algorithms with optimizations like path compression. [Apply] **CO6:** Analyze and evaluate string matching algorithms including naive, Rabin-Karp, and Knuth-Morris-Pratt methods. [Analyze]					
Course Content:						
Module 1	**Hashing**					**15 Sessions**
Review of Data Structures – Hashing - General Idea, Hash Function, Separate Chaining, Hash Tables without linked lists: Linear Probing, Quadratic probing, Double Hashing, Rehashing, Hash Tables in the Standard Library, Universal Hashing, Extendible Hashing.						
Module 2	**Priority Queues (Heaps)**					**15 Sessions**
Model, Simple implementations, Binary Heap: Structure Property, Heap Order Property, Basic Heap Operations: insert, delete, Percolate down, Other Heap Operations. Binomial Queues: Binomial Queue Structure, Binomial Queue Operations, Implementation of Binomial Queue, Priority Queues in the Standard Library.						
Module 3	**Tress**					**15 Sessions**

Introduction: **Binary Search Trees** and its operations, **AVL Trees** – Rotations – Operations, **Multiway Search Trees - 2-3 Trees**: Searching for an Element in a 2-3 Tree, inserting a New Element in a 2-3 Tree, Deleting an Element from a 2-3 Tree **Red-Black Trees –** Properties of red-black trees, Rotations, Insertion, Deletion.		
Module 4	**Disjoint Sets and String Matching**	**15 Sessions**
Disjoint Sets – Equivalence relation, Basic Data Structure, Simple Union and Find algorithms, Smart Union and Path compression algorithm. **String Matching –** The naive string-matching algorithm, The Rabin-Karp algorithm, The Knuth-Morris-Pratt algorithm.		
Project Work / Assignment / Case Study 1. Hashing 2. Binary Search Tree, AVL Tree **3.** String Matching		
Textbook(s) T1. "Data Structures and Algorithms Made Easy" – Narasimha Karumanchi, Latest Revised Edition, 2020 T2. "Algorithms" (4th Edition Updated) – Robert Sedgewick & Kevin Wayne, Updated Edition (post-2020 reprint) T3. "Problem Solving with Algorithms and Data Structures Using Python" – Bradley N. Miller & David L. Ranum, Updated Edition, 2021		
References R1. "Grokking Algorithms" (Updated Edition) – Aditya Bhargava, 2021 R2. "Advanced Data Structures" – Peter Brass, Updated Edition (post-2020 reprint) R3. "Hands-On Data Structures and Algorithms with Python" – Dr. Basant Agarwal, 2021		
E-Resources 1. https://visualgo.net/en 2. https://www.geeksforgeeks.org/data-structures/ 3. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms/		

CSC4202- Relational and NoSql DataBase Systems

Course Code: CSC4202	**Course Title: Relational and NoSql Database Systems** **Type of Course: Program Core**	**L-T-P-C**	3	0	0	3
Version No.	**1.0**					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course provides a strong foundation in data management principles and modern database systems, with emphasis on both relational and NoSQL databases. It covers data modeling using the relational model, data manipulation through SQL and database programming, and database development practices including security, integrity, and privacy considerations. The course also introduces NoSQL database concepts, data models, and use cases for handling large-scale and unstructured data. By mastering core concepts and techniques of both relational and non-relational databases, students will be able to design, implement, and manage database solutions to address diverse business and organizational requirements.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Relational Database Management and NoSQL and attain SKILL DEVELOPMENT through Experiential Learning techniques.					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1:** Apply fundamental concepts of database systems including architecture, data models, and transaction management with ACID properties. [Apply] **CO2:** Apply relational database concepts including schema design, keys, relational algebra, and SQL queries for data manipulation. [Apply] **CO3:** Analyze and design ER models and map them to relational schemas for real-world applications. [Analyze] **CO4:** Analyze schema design issues and apply normalization techniques up to Fifth Normal Form to eliminate redundancy and anomalies. [Analyze] **CO5:** Apply MySQL commands and queries for database creation, manipulation, and retrieval operations. [Apply] **CO6:** Analyze and evaluate NoSQL databases such as MongoDB and Cassandra for handling unstructured data and scalable applications. [Analyze]					
Course Content:						
Module 1	**Introduction to Database**					**12 Sessions**
Introduction-Database System, Applications-Advantages of Database approach-Database Architecture-Data Independence-Database Languages- Database Design-Database Model-Data Storage and Querying- Transaction Management-ACID Properties						
Module 2	**Relational Databases**					**11 Sessions**
Introduction to the Relational Model-Structure of Relational Databases-Structure of Relational Databases-Database Schema-Keys-Schema Diagrams-Relational Query Languages-Relational Operations-ER Diagrams-Mapping to ER Model to Relational model -Introduction to SQL -Data Definition-Data Types-Basic Structure of SQL Queries-Operations.						
Module 3	**Schema Design**					**11 Sessions**

Problems in schema design, redundancy and anomalies.Schema refinement: Functional Dependencies, Normalization and forms - First, Second, Third, Dependency Preservation – Boyce/Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Module 4	**MySQL and NoSQL Database**	**11 Sessions**

MySQL Introduction-MySQL Features- Data types- Variables-MySQL Database Creation-Manipulation queries-Select -Built-in functions-MySQL Clauses-Introduction to NoSQL : MongoDB CRUD Operation-Insert- Update-Delete-Query-Indexing-Replication-Using MongoDB with Python-Advanced MongoDB Features – Cassandra: Data Model-Table Operations, CRUD Operations.

Targeted Application & Tools that can be used:
Tools: MYSQL, MongoDB, Apache Cassandra (latest stable version like 4.0+)

Project work/Test:
Mini Project - During the course, students would need to do coding assignments to learn to train and use different database models. Sample coding assignments include:

1. Healthcare Management System
2. University Management System
3. Financial Management System
4. Employee Management System etc

Textbook(s)

T1. "Database System Concepts" (7th Edition) – Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 2020.
T2. "Fundamentals of Database Systems" (7th Edition) – Ramez Elmasri & Shamkant B. Navathe, Updated Edition (post-2020 reprint).
T3. "MongoDB: The Definitive Guide" (3rd Edition) – Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, 2020.

References

R1. "Designing Data-Intensive Applications" – Martin Kleppmann, Updated Edition (post-2020 reprint)
R2. "Learning SQL" (3rd Edition) – Alan Beaulieu, Updated Edition (post-2020 reprint)
R3. "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" – Pramod J. Sadalage & Martin Fowler, Updated Edition (post-2020 reprint)

Web Resources

1. https://dev.mysql.com/doc/
2. https://www.mongodb.com/docs/
3. https://www.postgresql.org/docs/

CSC4203- Cloud Data Platforms and Architecture

Course Code: CSC4203	**Course Title: Cloud Data Platforms and Architecture Type of Course: Program Core Lab Integrated**	**L-T-P-C**	2	0	2	3
Version No.	1.0					
Course pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course provides a hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It dives into all of the details that a student needs to know in order to plan for developing applications on the cloud and what to look for when using applications or services hosted on a cloud.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Cloud Computing and attain Skill Development through Experiential Learning techniques.					
Course Outcomes	**Upon successful completion of the course, the students shall be able to:** **CO1:** Utilize cloud computing concepts and service models (IaaS, PaaS, SaaS) to deploy and manage applications in real-world scenarios. [Apply] **CO2:** Implement virtualization and containerization techniques to create, configure, and manage cloud-based computing environments. [Apply] **CO3:** Employ cloud programming models, development environments, and container technologies to develop cloud-based applications**.** [Apply] **CO4:** Examine cloud infrastructure mechanisms, SLAs, management, and security mechanisms to ensure Quality of Service (QoS) in cloud. [Analyze] **CO5:** Validate cloud cost models, performance metrics, and monitoring tools to ensure efficient and cost-effective cloud infrastructure. [Evaluate] **CO6:** Validate cloud deployment strategies, scalability mechanisms, and fault-tolerance techniques to recommend optimal solutions for enterprise cloud applications. [Evaluate]					
Course Content:						
Module 1	**Introduction to Cloud services**					**15 Sessions**
Evolution of cloud computing, Computing Platforms and Technologies, Historical Developments, Cloud Computing NIST Architecture, Cloud Computing Architecture, IaaS, PaaS, SaaS, Types of Clouds, Cloud Computing Environments.						
Module 2	**Virtualization Techniques**					**15 Sessions**
Basics of Virtualization - Types of Virtualizations, Taxonomy of Virtualization Techniques, Implementation Levels of Virtualization.						
Module 3	**Cloud QoS and Management**					**15 Sessions**
Cloud Infrastructure Mechanisms, SLAs, Specialized Cloud Mechanisms, Cloud Management Mechanisms, Cloud Security Mechanisms						
Module 4	**Application development in the Cloud**					**15 Sessions**
Topics: Programming Models for Cloud Computing - Software Development in Cloud - Service creation environments to develop cloud-based applications. Development						

environments for service development (Demonstration using AWS Cloud); Docker and Containers.

List of Laboratory Tasks

Experiment 1: Understand cloud platforms and set up applications in ZOHO.
Level 1: Create a basic form in Zoho Creator to collect student details (Name, Roll No, Department, Email).
Level 2: Add field validations and make the email field mandatory. Test the form for different inputs
Experiment 2: Differentiate between IaaS, PaaS, and SaaS using online tools.
Level 1: Create a new application in Zoho Creator to manage a library system (Books, Authors, Issued Books).
Level 2: Establish relationships between forms (Books and Authors) using lookup fields
Experiment 3: Differentiate between IaaS, PaaS, and SaaS using online tools.
Level 1: Create a leave management system with a form to apply for leave.
Level 2: Implement an approval workflow with notifications for pending and approved leaves
Experiment 4: Identify key characteristics of Public, Private, Hybrid, and Community clouds.
Level 1: Compare GCP, AWS, and Azure in terms of deployment models.
Level 2: Write a short report outlining the pros/cons of each model.
Experiment 5: Set up and use a virtual machine using VirtualBox or VMware.
Level 1: Install Ubuntu on VirtualBox.
Level 2: Understand ISO mounting, network settings, and guest additions.
Experiment 6: Launch, configure, and manage cloud-based virtual machines.
Level 1: Launch EC2 (AWS) or VM (Azure).
Level 2: Configure security groups, key pairs, and firewall.
Level 3: SSH into the instance and run basic Linux commands.
Level 4: Deploy sample application and access from public.
Experiment 7: Study and explore the global infrastructure of Google Cloud Platform (GCP).
Level 1: Identify and analyze Google Cloud regions, zones, and data center locations using the official Google Cloud website and documentation.
Experiment 8: Access Google Cloud Console and activate Google Cloud Shell.
Level 1: Execute basic Linux commands to perform file and directory operations, view system information, and manage files. Observe and record the output of each command.
Experiment 9: Develop a robust shell script that accepts a domain name or IP address as input and performs multi-layer network diagnostics.
The script should:

- Collect **local network configuration and routing information**
- Resolve DNS using multiple resolvers
- Verify host reachability and analyze latency trends
- Trace and analyze routing hops
- Test TCP/UDP port accessibility
- Validate application-layer service availability
- Capture active socket and connection statistics
- Log all results with timestamps
- Generate a summarized diagnostic report indicating network health

Experiment No 10: Design and simulate a cloud environment using CloudAnalyst by configuring multiple user bases and data centers located in different geographic regions.
Level 1: Analyze application performance metrics such as response time, data center processing time, and overall cost.
Experiment No 11: Creating and Managing Virtual Instances on Cloud
Level 1: Launch EC2 (AWS) or VM (Azure).
Level 2: Configure security groups, key pairs, and firewall.
Level 3: SSH into the instance and run basic Linux commands.

Level 4: Deploy sample application and access from public.
Experiment 12: Create Virtual network on cloud
Level 1: Create and configure a custom virtual network using AWS or Azure.
Level 2: Subnet the network, assign IP ranges, and test connectivity.
Level 3: Launch virtual machines within the virtual network and ensure internal communication
Experiment 13: Implementation of virtual private networks (VPC/VNet) on the cloud.
Level 1: Create and configure a custom virtual network using AWS or Azure.
Level 2: Subnet the network, assign IP ranges, and test connectivity.
Level 3: Launch virtual machines within the virtual network and ensure internal communication
Experiment 14: Use cloud storage services like Amazon S3 / Azure Blob.
Level 1: Create and configure S3 buckets.
Level 2: Upload, retrieve, and delete files using AWS CLI.
Experiment 15: Use AWS Elastic Beanstalk or Google App Engine.
Level 1: Create and deploy a simple Node.js or Python app.
Level 2: Monitor logs and deployment status.

Targeted Applications:
Developing applications on Cloud Platforms via Virtual machines

Cloud Tools:
- CloudSim
- VMWare
- Amazon EC2
- Google Compute Engine
- Microsoft Azure

Project work/Assignment:
1. Automation of performance analysis of students through the Cloud
2. Chatbots development using Cloud resources
3. Blog creation using Cloud computing

Analysis of Case Studies: When deciding to adopt cloud computing architecture, decide if the cloud is right for your requirements (for the application identified).

Text Book(s)
T1. "Cloud Computing: Concepts, Technology & Architecture" – Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Updated Edition (post-2020 reprint)
T2. "Cloud Computing: Theory and Practice" (2nd Edition) – Dan C. Marinescu, Updated Edition (post-2020 reprint)
T3. "Architecting the Cloud: Design Decisions for Cloud Computing Service Models" (2nd Edition) – Michael J. Kavis, 2020

References
R1. "AWS Certified Solutions Architect Study Guide" (3rd Edition) – Ben Piper & David Clinton, 2022
R2. "Docker Deep Dive" – Nigel Poulton, Updated Edition (2023)
R3. "Cloud Native DevOps with Kubernetes" – John Arundel & Justin Domingus, Updated Edition (post-2020 reprint)

Web Resources
1. https://aws.amazon.com/what-is-cloud-computing/
2. https://azure.microsoft.com/en-in/resources/cloud-computing-dictionary/what-is-cloud-computing/
3. https://cloud.google.com/learn/what-is-cloud-computing

CSC4204 - Python Programming for Data Science

Course Code: CSC4204	**Course Title: Python Programming for Data Science** **Type of Course: Theory & Integrated Laboratory**	**L-T-P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	Basic programming knowledge (C / Python / Java), Fundamentals of Statistics, Basic Linear Algebra					
Anti-requisites	**NIL**					
Course Description	This course provides advanced knowledge of Python programming specifically tailored for Data Science applications. It focuses on scientific computing, data manipulation, statistical analysis, data visualization, and machine learning using Python libraries. Students will gain hands-on experience in solving real-world data problems using industry-standard tools and frameworks. Topics include: Advanced Python programming, NumPy, Pandas, Data Cleaning, Exploratory Data Analysis (EDA), Data Visualization (Matplotlib, Seaborn, Plotly), Statistical Analysis, Machine Learning using Scikit-Learn, Introduction to Deep Learning frameworks, and handling large datasets.					
Course Objective	To develop advanced programming skills in Python for data-driven applications. To apply Python libraries for data analysis and visualization. To implement machine learning models using Python frameworks. To analyze and interpret real-world datasets using scientific approaches.					
Course Outcomes	**On successful completion of the course, students shall be able to**: **CO1:** Apply advanced Python programming concepts such as functional programming, iterators, generators, decorators, and API integration for data science applications. **[Apply]** **CO2:** Apply numerical computing techniques using NumPy and perform data manipulation and preprocessing using Pandas. **[Apply]** **CO3:** Analyze datasets using exploratory data analysis techniques and visualize data using Matplotlib, Seaborn, and Plotly. **[Analyze]** **CO4:** Apply statistical methods including descriptive statistics, correlation, and regression analysis for data interpretation. **[Apply]** **CO5:** Apply machine learning algorithms such as regression, classification, and clustering using Python libraries. **[Apply]** **CO6:** Analyze and evaluate machine learning models using performance metrics, cross-validation, and basic deep learning frameworks. **[Analyze / Evaluate]**					
Course Content:						
Module 1	**Advanced Python Programming for Data Science**					**20 sessions (4T+16L)**
Review of Python fundamentals, Functional programming concepts, Lambda, map, filter, reduce, Iterators and Generators, Decorators, Virtual environments and package management, Working with APIs						
Module 2	**Numerical Computing and Data Manipulation**					**15 sessions (3T+12L)**
NumPy arrays and vectorized operations, Broadcasting and indexing, Pandas DataFrames, Data cleaning and preprocessing, Handling missing data Data transformation , Merging, grouping, aggregation						
Module 3	**Data Visualization and Statistical Analysis**					**20 sessions (4T+16L)**

Matplotlib and Seaborn, Interactive visualization using Plotly, Exploratory Data Analysis (EDA), Descriptive statistics, Correlation and regression analysis		
Module 4	**Machine Learning with Python**	**20 sessions (4T+16L)**
Supervised learning (Linear Regression, Logistic Regression, KNN, SVM) Unsupervised learning (K-Means, Hierarchical Clustering), Model evaluation metrics Cross-validation, Feature engineering, Introduction to Deep Learning (TensorFlow / PyTorch basics)		

List of Laboratory Tasks:

Experiment 1: Python Fundamentals Refresher
Level 1: Write Python programs using variables, data types, and control structures.
Level 2: Develop a menu-driven program solving multiple small data problems.

Experiment 2: Functional Programming Basics
Level 1: Implement simple functions using lambda.
Level 2: Solve data transformation problems using map, filter, and reduce.

Experiment 3: Iterators and Generators
Level 1: Create custom iterators and generators.
Level 2: Build a generator pipeline for processing large datasets.

Experiment 4: Decorators in Python
Level 1: Implement basic decorators.
Level 2: Create decorators for logging and performance timing.

Experiment 5: Virtual Environments & Packages
Level 1: Create and manage a virtual environment.
Level 2: Install and use third-party libraries in a project.

Experiment 6: Working with APIs
Level 1: Fetch data using a public API.
Level 2: Parse and store API data for analysis.

Experiment 7: NumPy Array Creation
Level 1: Create and manipulate NumPy arrays.
Level 2: Perform vectorized operations on large datasets.

Experiment 8: NumPy Indexing & Broadcasting
Level 1: Perform indexing and slicing operations.
Level 2: Apply broadcasting for complex computations.

Experiment 9: Pandas DataFrames Basics
Level 1: Create and explore DataFrames.
Level 2: Perform column operations and filtering.

Experiment 10: Data Cleaning
Level 1: Handle missing values in datasets.
Level 2: Apply advanced cleaning techniques and outlier detection.

Experiment 11: Data Transformation
Level 1: Perform sorting and reshaping operations.
Level 2: Apply feature scaling and encoding techniques.

Experiment 12: Data Aggregation
Level 1: Use groupby and aggregation functions.
Level 2: Perform multi-level grouping and pivot tables.

Experiment 13: Data Merging
Level 1: Merge and join datasets.
Level 2: Handle complex joins and data integration tasks.

Experiment 14: Matplotlib Basics
Level 1: Plot basic graphs (line, bar).
Level 2: Customize plots with labels, legends, and styles.

Experiment 15: Seaborn Visualization
Level 1: Create statistical plots.
Level 2: Perform advanced visualization (heatmaps, pairplots).

Experiment 16: Plotly Interactive Visualization
Level 1: Create interactive charts.
Level 2: Build dashboards with interactive components.

Experiment 17: Exploratory Data Analysis (EDA)
Level 1: Perform basic EDA.
Level 2: Generate insights using visualization and statistics.
Experiment 18: Descriptive Statistics
Level 1: Compute mean, median, mode.
Level 2: Analyze variance, skewness, and kurtosis.
Experiment 19: Correlation Analysis
Level 1: Calculate correlation coefficients.
Level 2: Visualize correlation matrices.
Experiment 20: Regression Analysis
Level 1: Implement simple linear regression.
Level 2: Perform multiple regression analysis.
Experiment 21: Classification using Logistic Regression
Level 1: Implement logistic regression.
Level 2: Evaluate model performance.
Experiment 22: K-Nearest Neighbors (KNN)
Level 1: Implement KNN classifier.
Level 2: Optimize K value and evaluate accuracy.
Experiment 23: Support Vector Machines (SVM)
Level 1: Train an SVM model.
Level 2: Tune hyperparameters and evaluate performance.
Experiment 24: K-Means Clustering
Level 1: Implement K-means clustering.
Level 2: Determine optimal clusters using elbow method.
Experiment 25: Hierarchical Clustering
Level 1: Perform hierarchical clustering.
Level 2: Visualize dendrograms.
Experiment 26: Model Evaluation Metrics
Level 1: Compute accuracy, precision, recall.
Level 2: Analyze confusion matrix and ROC curves.
Experiment 27: Cross Validation
Level 1: Apply k-fold cross-validation.
Level 2: Compare multiple models using cross-validation.
Experiment 28: Feature Engineering
Level 1: Create new features from data.
Level 2: Apply feature selection techniques.
Experiment 29: Introduction to Deep Learning
Level 1: Build a simple neural network.
Level 2: Train using TensorFlow/PyTorch.
Experiment 30: Mini Project (End-to-End Data Science)
Level 1: Perform data preprocessing and visualization.
Level 2: Build and evaluate a complete ML model pipeline.

Targeted Application & Tools that can be used:
Jupyter Notebook,Google Colab,Anaconda,NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, TensorFlow / PyTorch, Git & GitHub

Assignment:

1. Perform exploratory data analysis on a COVID dataset and present insights.
2. Build a predictive model for house price prediction.
3. Develop a classification model for spam email detection.
4. Create a dashboard visualizing sales data using Plotly.

Text Book(s)

T1. "Python for Data Analysis" (3rd Edition) – Wes McKinney, 2022.
T2. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (3rd Edition) – Aurélien Géron, 2022.
T3. "Data Science from Scratch: First Principles with Python" (2nd Edition) – Joel Grus, 2021.

References

R1. "Effective Python: 90 Specific Ways to Write Better Python" (2nd Edition Updated) – Brett Slatkin, post-2020 reprint.
R2. "Practical Statistics for Data Scientists" (2nd Edition) – Peter Bruce, Andrew Bruce & Peter Gedeck, 2020.
R3. "Deep Learning with Python" (2nd Edition) – François Chollet, 2021.

Web Resources

1. https://numpy.org/doc/
2. https://pandas.pydata.org/docs/
3. https://scikit-learn.org/stable/

CSC4205-: Data structures and Algorithms Lab

Course Code: CSC4205	**Course Title: Data structures and Algorithms Lab** **Type of Course: Lab**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course provides an in-depth study of advanced data structures and algorithmic techniques essential for efficient problem solving in computer science. It focuses on abstract data types (ADTs), mathematical foundations, and the selection of appropriate data structures for real-world applications. The course covers advanced topics such as hashing techniques, priority queues, balanced and multiway trees, disjoint sets, and string-matching algorithms. Emphasis is placed on both theoretical understanding and practical implementation using standard libraries and programming tasks. Through analytical problem solving and hands-on exercises, learners develop the ability to design, analyze, and optimize algorithms for complex computational problems.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Advanced Data Structure and Algorithms** and attain **Skill Development** through **Experiential Learning** techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply hashing techniques and implement collision resolution methods such as linear probing, quadratic probing, double hashing, and rehashing. [Apply] **CO2:** Apply heap-based data structures by implementing binary heaps and performing priority queue operations. [Apply] **CO3:** Analyze and implement advanced tree structures such as binary search trees, AVL trees, 2-3 trees, and red-black trees. [Analyze] **CO4:** Apply disjoint set operations using union-find algorithms with optimizations like path compression. [Apply] **CO5:** Analyze and implement string matching algorithms including naive, Rabin-Karp, and Knuth-Morris-Pratt methods. [Analyze] **CO6:** Evaluate the performance and efficiency of different data structures and algorithms through practical implementation and testing. [Evaluate]					
List of Laboratory Tasks: **Experiment 1: Program to Implement Hash Table Using Separate Chaining** **Level 1:** Implement basic hash table using separate chaining. **Level 2:** Analyze collision handling efficiency with different hash functions. **Experiment 2: Program to Implement Hash Table Using Linear Probing** **Level 1:** Implement hash table using linear probing. **Level 2:** Compare performance with varying load factors. **Experiment 3: Program to Implement Hash Table Using Quadratic Probing** **Level 1:** Implement hash table using quadratic probing. **Level 2:** Analyze clustering issues and performance improvements.						

Experiment 4: Program to Implement Double Hashing with Rehashing
Level 1: Implement double hashing technique.
Level 2: Extend implementation with rehashing and performance analysis.
Experiment 5: Program to Implement Binary Heap with Insert and Delete Operations
Level 1: Implement heap with insertion and deletion.
Level 2: Analyze time complexity of heap operations.
Experiment 6: Program to Implement Min Heap and Max Heap
Level 1: Implement min heap and max heap structures.
Level 2: Compare behavior and applications of both heaps.
Experiment 7: Program to Implement Priority Queue Using Binary Heap
Level 1: Implement priority queue using binary heap.
Level 2: Analyze efficiency compared to other queue implementations.
Experiment 8: Program to Implement Binomial Queue Operations
Level 1: Implement basic binomial queue operations.
Level 2: Analyze merging efficiency and compare with binary heap.
Experiment 9: Program to Implement Binary Search Tree with Traversals
Level 1: Implement BST with insertion and traversals.
Level 2: Analyze performance for different tree structures.
Experiment 10: Program to Implement AVL Tree with Rotations
Level 1: Implement AVL tree with basic rotations.
Level 2: Analyze balancing efficiency compared to BST.
Experiment 11: Program to Implement 2–3 Tree Insertion and Search
Level 1: Implement insertion and search in 2–3 tree.
Level 2: Analyze tree balancing and height properties.
Experiment 12: Program to Implement Red-Black Tree Insertion and Deletion
Level 1: Implement insertion and deletion in red-black tree.
Level 2: Analyze balancing rules and performance.
Experiment 13: Program to Implement Disjoint Set (Union–Find) with Path Compression
Level 1: Implement union-find operations.
Level 2: Analyze optimization using path compression.
Experiment 14: Program to Implement Naïve String-Matching Algorithm
Level 1: Implement naïve string-matching algorithm.
Level 2: Analyze time complexity with different inputs.
Experiment 15: Program to Implement Knuth–Morris–Pratt (KMP) String Matching Algorithm
Level 1: Implement KMP algorithm.
Level 2: Compare efficiency with naïve string matching.

References/Manual/Software:

1. Open-source implementation of algorithms in multiple languages (Python, C++, Java)
2. Visualgo – Visualizing DSA concepts such as trees, graphs, sorting etc.,
3. Offline/IDE tools: VS Code, code runner, C++ Intellisense

CSC4206-: DATABASE SYSTEMS LABORATORY

Course Code: CSC4206	**Course Title: DATABASE SYSTEMS LABORATORY Type of Course:**	**L-T- P- C**	0	0	2	1
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This laboratory course provides practical experience in designing and implementing databases using MySQL and NoSQL technologies. Students perform schema design, normalization, SQL queries, and transaction management through hands-on exercises. The lab also covers indexing, security considerations, and database programming concepts. Exposure to NoSQL databases enables students to manage semi-structured and unstructured data. By the end of the course, students will be able to develop and manage real-world database applications.					
Course Objective	To enable students to design, implement, and manage relational and NoSQL databases using appropriate tools and techniques for real-world applications.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply SQL commands to create, modify, and manage relational databases and tables. **[Apply]** **CO2:** Apply data manipulation operations such as insertion, deletion, updating, and querying using SQL. **[Apply]** **CO3:** Analyze and implement database queries using joins, subqueries, and aggregate functions. **[Analyze]** **CO4:** Demonstrate the installation and configuration of MongoDB and perform document retrieval and data management using queries, projections, limits and operators. **[Apply]** **CO5:** Apply Cassandra Query Language (CQL) to create keyspaces, tables and perform CRUD operations on NoSQL databases. **[Apply]** **CO6:** Analyze and implement transaction management and constraints to ensure data integrity. **[Analyze]**					

List of Laboratory Tasks

Experiment 1: Mysql Installation and Database Creation

Level 1: Installation of MySql and database creation(including all the constraints like Primary key, not null, check, unique, Referential integrity)

Level 2: Create a database School and table Students with fields: ID, Name, Age, Class, Marks.Insert at least 10 records and display all.

Experiment 2: SQL queries in MySQL to filter and sort data using appropriate conditions and ordering clauses.

Level 1: Creating student tables and using clauses for filter and sort data.Display students with marks > 80 and Sort students by Age in descending order.

Level 2: Find students who scored more than the average marks and List courses not enrolled by any student.

Experiment 3: SQL queries in MySQL to apply aggregate functions and use the GROUP BY clause for data summarization.

Level 1: Create table structures using MySQL data types and apply constraints to MySQL tables.

Level 2: Summarize data:Count total students,Calculate average marks and Show total students per class

Experiment 4: SQL statements in MySQL to perform Data Definition Language (DDL) operations such as creating, altering, and deleting database objects
Level 1: Practicing DDL commands in MySQL.
Level 2: Insert, Select update and delete queries (DML) from within a MySQL table.
Experiment 5: SQL queries in MySQL to implement various SQL clauses for effective data retrieval and manipulation.
Level 1: MySQL Querying Using aggregate functions COUNT and SUM.
Level 2: Filter records based on conditions using WHERE,DISTINCT, GROUPBY and HAVING Clause
Experiment 6: SQL statements in MySQL to perform data modification operations
Level 1: To update and delete records.
Level 2: Increase marks by 5 for all students in class '8A' and Delete students with marks < 40
Experiment 7: SQL queries in MySQL to perform data formatting using appropriate functions for presenting query results.
Level 1: To use functions for formatting dates in MySQL.
Level 2: Build formatting dates, string functions using UPPER(), LOWER(), CONCAT(), SUBSTRING(), LENGTH()
Experiment 8: Mongodb Installation and configuration
Level 1: Installation and Configuration of MongoDB in windows.
Level 2: Create a database and collection in MongoDB.
Experiment 9: Querying documents in mongodb
Level 1: Query documents using different criteria and operators using find() and findOne()
Level 2: Updating and Deleting Documents
Experiment 10: Projection and limiting
Level 1: Use projection to limit fields and limit() to restrict documents in mongoDB
Level 2: Sort documents in ascending or descending order and perform group-based operations using $group.
Experiment 11: Mongodb Operators
Level 1: Practice MongoDB query operators like $in, $and, $or, $exists.
Level 2: Create and view an indexes
Experiment 12: CRUD Operations
Level 1: To insert and multiple documents into a MongoDB collection.
Level 2: Implement CRUD on documents with embedded/nested objects and arrays.
Experiment 13: CASSANDRA installation
Level 1: Installing Cassandra on Local Machine (Windows)
Level 2: Creating Tables in Cassandra
Experiment 14: CASSANDRA CRUD OPERTIONS
Level 1: Create CRUD operations in Cassandra
Level 2: Querying data using SELECT statement
Experiment 15: Implement Transaction Control Language (TCL) commands to manage database transactions and ensure data consistency.
Level 1: Design a transaction management scenario for an online shopping system using COMMIT, ROLLBACK, and SAVEPOINT.
Level 2: Create multiple savepoints in a transaction and demonstrate rolling back to different savepoints.

References/Manual/Software:

1.Abraham Silberschatz, Henry F.Korth,"Database Ssystem Concepts", McGraw HIll,7th Edition, 2020.
2.Elvis C. Foster, Shripad V. Godbole, "Database Systems",Apress,2014
3.Daniel Nichter,Efficient MySQL Performance: Best Practices and Techniques,O'Reilly Media,First Edition,January 2022
4. Michael E Kirshteyn,Mastering NoSQL Database Design: A Comprehensive Guide to Building Scalable, High-Performance, and Flexible Data Systems,April 2024.

CSC4301 - Research Methodology

Course Code: CSC4301	**Course Title: Research Methodology Type of Course: Theory**	**L-T-P-C**	2	0	0	2
Version No.						
Course Pre-requisites	Descriptive statistics, statistical analysis tools (SPSS, R, Python, or MATLAB).					
Anti-requisites	**NIL**					
Course Description	**Essentials of Research Methodology** introduces students to the fundamental principles of conducting systematic research across disciplines. It covers research design, data collection methods, statistical analysis, and ethical considerations. The course equips learners with critical thinking, academic writing, and analytical skills for producing high-quality research. By the end, students will be able to formulate research questions, apply appropriate methodologies, and effectively present their findings.					
Course Objectives	The course aims to develop a strong foundation in research methodology, enabling students to design and conduct systematic research. It enhances skills in data collection, statistical analysis, and academic writing while emphasizing ethical research practices. By the end, students will be able to critically evaluate research, formulate hypotheses, and present their findings effectively.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply fundamental concepts of research methodology including research types, processes, ethics, and problem formulation. [Apply] **CO2:** Apply appropriate data collection methods and sampling techniques for conducting research studies. [Apply] **CO3:** Analyze collected data using statistical tools and software for meaningful interpretation. [Analyze] **CO4:** Apply standard formats and referencing styles to prepare research reports, papers, and proposals. [Apply] **CO5:** Evaluate research findings using hypothesis testing and effectively present results using appropriate visualization techniques. [Evaluate] **CO6:** Create research proposals and innovative solutions by applying critical thinking, digital tools, and knowledge of intellectual property rights. [Create]					
Course Outcome	By the end of the course, students will be able to understand key research methodologies, ethical principles, and data collection techniques. They will apply statistical tools and analytical methods to interpret research findings effectively. Additionally, they will develop well-structured research reports, proposals, and presentations following academic standards.					
Course Content:						
Module 1	**Introduction to Research Methodology**					**8 Sessions**
Introduction to Research Methodology, Meaning, Objectives, and Significance of Research, Types of Research: Fundamental, Applied, Qualitative, and Quantitative, Research Process and Design, Ethics in Research and Plagiarism, Identifying Research Problems and Formulating Research Questions						
Module 2	**Data Collection and Analysis**					**6 Sessions**
Introduction to Data Collection and Analysis, Sources of Data: Primary and Secondary, Sampling Techniques and Sample Size Determination, Methods of Data Collection: Surveys, Interviews, Case Studies, Statistical Tools for Data Analysis (Descriptive and Inferential Statistics), Software Tools for Data Analysis (SPSS, R, Python)						

Module 3	**Report Writing and Presentation**	**8 Sessions**
Introduction to Report Writing and Presentation, Structure of a Research Paper, Thesis, and Dissertation, Referencing Styles: APA, MLA, IEEE, Chicago, Hypothesis Testing and Interpretation of Results, Writing Research Proposals and Funding Applications, Presentation of Research Findings and Use of Visual Aids.		
Module 4	**Skill Development for Research and Innovation**	**6 Sessions**
Skill Development for Research and Innovation, Critical Thinking and Problem-Solving in Research, Digital Tools for Literature Review (Mendeley, Zotero, EndNote), Scientific Writing and Research Paper Formatting (LaTeX, MS Word), Presentation Skills: Visual Representation of Data (Tableau, Excel, Canva), Intellectual Property Rights (IPR) and Patents, Research for Industry and Entrepreneurship (Market Research & Business Analytics)		
Project work/Assignment: Mention the Type of Project /Assignment proposed for this course: Research Proposal Development, Literature Review Report, Data Collection and Analysis, Research Paper Writing and Presentation.		
After completion of each module a programming-based Assignment/Assessment will be conducted. **On completion of Module 2, student will be asked to develop a Project work.**		
Textbook(s) T1. "Research Methodology: Methods and Techniques" (4th Edition) – C.R. Kothari & Gaurav Garg, 2021. T2. "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches" (5th Edition) – John W. Creswell & J. David Creswell, 2023. T3**.** "The Craft of Research" (4th Edition Updated) – Wayne C. Booth, Gregory G. Colomb & Joseph M. Williams, post-2020 reprint.		
References R1. "Practical Research: Planning and Design" (12th Edition) – Paul D. Leedy & Jeanne Ellis Ormrod, 2021. R2. "Doing Your Research Project" (4th Edition) – Judith Bell & Stephen Waters, 2022. R3. "Business Research Methods" (13th Edition) – Donald R. Cooper & Pamela S. Schindler, post-2020 reprint.		
Web Resources 1. https://www.scribbr.com/category/methodology/ 2. https://owl.purdue.edu/owl/research_and_citation/resources.html 3. https://www.statisticshowto.com/		

PPS4008 - Quantitative Skills and Logical Reasoning

Course Code: PPS4008	**Course Title: Quantitative Skills and Logical Reasoning Type of Course:**	**L- T -P- C**	**0**	**0**	**2**	**2**
Version No.	1.0					
Course Pre-requisites	Students should know the basic Mathematics & aptitude along with understanding of English					
Anti-requisites	**NIL**					
Course Description	The objective of this course is to prepare the trainees to tackle the questions on various topics and various difficulty levels based on Quantitative Ability, and Logical Reasoning asked during the placement drives. There will be sufficient focus on building the fundamentals of all the topics, as well as on solving the higher order thinking questions. The focus of this course is to teach the students to not only get to the correct answers, but to get there faster than ever before, which will improve their employability factor.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Aptitude and attain Skill Development through Problem Solving techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply quantitative aptitude concepts such as percentages, ratios, averages, and profit & loss to solve numerical problems. [Apply] **CO2:** Apply mathematical techniques to solve problems related to time & work, time speed & distance, permutation, combination, and probability. [Apply] **CO3:** Analyze data interpretation problems to extract meaningful insights and make logical conclusions. [Analyze] **CO4:** Apply logical reasoning techniques to solve problems involving coding-decoding, blood relations, and directions. [Apply] **CO5:** Analyze complex reasoning problems such as puzzles, number series, and syllogisms for decision making. [Analyze] **CO6:** Evaluate problem-solving strategies to improve accuracy and efficiency in quantitative and logical reasoning tasks. [Evaluate]					
Course Content:						
Module 1	**Quantitative Ability**					**24 Sessions**
Topics: Introduction to Aptitude, Percentages, Profit & Loss, Ratio & Proportion, Averages, Time & Work, Time speed & Distance, Permutation and Combination, Probability, Data Interpretation						
Module 2	**Logical Reasoning**					**21 Sessions**
Topics: Linear & Circular Arrangement Puzzle, Coding & Decoding, Blood Relations, Directions, Number Series, Wrong number series, Visual Reasoning, Critical thinking, Syllogism						
Text Book(s) T1. "Quantitative Aptitude for Competitive Examinations" – R.S. Aggarwal, Updated Edition (post-2020 reprint). T2. "A Modern Approach to Verbal & Non-Verbal Reasoning" – R.S. Aggarwal, Updated Edition (post-2020 reprint).						

T3. "How to Prepare for Quantitative Aptitude for CAT" – Arun Sharma, Latest Edition (post-2020 reprint).
References R1. "Quantitative Aptitude Quantum CAT" – Sarvesh K. Verma, Updated Edition (post-2020 reprint). R2. **"**Logical Reasoning and Data Interpretation for CAT" – Nishit K. Sinha, Updated Edition (post-2020 reprint). R3. "Fast Track Objective Arithmetic" – Rajesh Verma, Updated Edition (post-2020 reprint).
Web References 1. https://www.indiabix.com/aptitude/questions-and-answers/ 2. https://www.geeksforgeeks.org/aptitude-for-placements/ 3. https://www.hitbullseye.com/Quant/

Semester II
CSC4207- R Programming

Course Code: CSC4207	**Course Title: R Programming Type of Course: Theory-Integrated**	**L-T- P- C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course is designed to introduce students to the core concepts of data analytics using the R programming environment. The course begins with fundamental R programming concepts and gradually progresses to more advanced data analytics techniques. By mastering essential data analytics concepts and techniques in R, students will be able to analyze, visualize, and interpret data effectively across a wide range of application domains.					
Course Objective	The course is designed to familiarize the learners with the concepts of R Programming for Data Science and attain Skill Development through Experiential Learning techniques.					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1:** Understand basic R functions pertaining to fundamental data analysis. [Understand] **CO2:** Interpret data using appropriate statistical methods. [Apply] **CO3:** Apply Data transformation techniques to clean, manipulate, and prepare datasets. [Apply] **CO4:** Apply Data visualization techniques using ggplot2 to effectively explore, summarize insights from datasets. [Apply] **CO5:** Apply Linear regression techniques to model relationships between variables. [Apply] **CO6:** Demonstrate the classification techniques to evaluate and interpret predictive models. [Apply]					
Course Content:						
Module 1	**Introduction**					**8 Sessions**
Introduction to R, Overview of data analysis, Working with directory in R, Loading and handling data in R, GGPLOT2 Calls, Data Transformation with Dplyr.						
Module 2	**Exploratory Data Analysis**					**8 Sessions**
Exploring a new dataset, Understanding variables and data types, Anomalies in numerical data, Visualizing relations between variables, Assumptions of Linear Regression, Handling Missing Values, Covariation, Patterns and Models, Importance of Data Visualization, Data Visualization with GGPLOT.						
Module 3	**Regression Analysis**					**7 Sessions**
Introduction, Types of Regression Analysis Models, Linear Regression, Simple Linear Regression, Non-Linear Regression, Regression Analysis with Multiple Variables, Cross Validation, Principal Component Analysis, Factor Analysis.						
Module 4	**Classification**					**7 Sessions**
Introduction, Different types of Classification, Logistic Regression, Support Vector Machines, K-Neatest Neighbours, Naïve Bayes Classifier, Decision Tree Classification, Random Forest Classification, Evaluation.						
Targeted Application & Tools that can be used: **Tools:** RStudio / Google Colab						
List of Laboratory Tasks: **Experiment 1: Introduction to R and RStudio**						

Level 1: Install R and RStudio, perform basic syntax, arithmetic operations, and variable assignments.
Level 2: Develop a simple R script demonstrating data input, processing, and formatted output.
Experiment 2: Data Structures and Data Handling in R
Level 1: Create and manipulate vectors, matrices, lists, and data frames.
Level 2: Perform data exploration and basic preprocessing on a dataset.
Experiment 3: Data Visualization using ggplot2
Level 1: Create basic plots such as scatter, bar, line, and histogram.
Level 2: Customize plots using themes, aesthetics, and faceting for multi-dimensional analysis.
Experiment 4: Data Transformation using dplyr
Level 1: Perform data selection, filtering, and sorting operations.
Level 2: Apply grouping, aggregation, and dataset merging techniques.
Experiment 5: Basic Statistics with R
Level 1: Compute descriptive statistics such as mean, median, and variance.
Level 2: Analyze probability distributions and interpret statistical results.
Experiment 6: Introduction to Machine Learning with R
Level 1: Split dataset into training and testing sets and build a simple model.
Level 2: Evaluate model performance and perform predictions.
Experiment 7: Correlation and Covariance
Level 1: Compute correlation matrix for a dataset.
Level 2: Visualize correlations using plots and interpret relationships (e.g., Iris dataset).
Experiment 8: Simple Linear Regression
Level 1: Build a regression model for a dataset.
Level 2: Analyze model results and interpret coefficients.
Experiment 9: Multiple Linear Regression
Level 1: Implement multiple regression with multiple independent variables.
Level 2: Evaluate model accuracy and interpret relationships among variables.
Experiment 10: Principal Component Analysis (PCA)
Level 1: Perform PCA on a dataset.
Level 2: Analyze dimensionality reduction and visualize principal components.
Experiment 11: k-Nearest Neighbors (kNN) Classification
Level 1: Implement kNN classifier.
Level 2: Tune parameters and evaluate classification accuracy.
Experiment 12: Naïve Bayes Classifier (Performance Evaluation)
Level 1: Implement Naïve Bayes classifier.
Level 2: Evaluate performance using metrics like accuracy and confusion matrix.
Experiment 13: Decision Tree Classifier
Level 1: Build a decision tree model.
Level 2: Evaluate and visualize decision tree performance.
Experiment 14: Naïve Bayes Classification (Advanced Implementation)
Level 1: Develop classification model using Naïve Bayes.
Level 2: Compare performance with other classification algorithms.
Experiment 15: Mini Project – Real-world Data Analysis
Level 1: Perform data cleaning and visualization on a dataset.
Level 2: Build a complete ML pipeline including model implementation, evaluation, and presentation of results.

Project work/Test:

Mini Project - During the course, students would need to do coding assignments to learn to train and use different models. Sample coding assignments include:

1. Exploratory Data Analysis of Student Performance Data
2. Sales Data Analysis and Visualization
3. Exploratory Analysis of Weather Data
4. Visualization of Population Growth Data
5. Employee Salary Analysis using R etc

Text Book(s)

T1. "R for Data Science" (2nd Edition) – Hadley Wickham & Garrett Grolemund, 2023.
T2. "Hands-On Programming with R" – Garrett Grolemund, Updated Edition (post-2020 reprint).
T3. "Practical Data Science with R" – Nina Zumel & John Mount, Updated Edition (post-2020 reprint).

References

R1. "Applied Predictive Modeling" – Max Kuhn & Kjell Johnson, Updated Edition (post-2020 reprint).
R2. "An Introduction to Statistical Learning with Applications in R" – Gareth James et al., Updated Edition (post-2020 reprint).
R3. "Machine Learning with R" – Brett Lantz, Updated Edition (post-2020 reprint).

Resources

1. https://cran.r-project.org/manuals.html
2. https://ggplot2.tidyverse.org/
3. https://dplyr.tidyverse.org/

CSC4208 - Full Stack Development

Course Code: CSC4208	**Course Title: Full Stack Development** **Type of Course: Program Core – Theory Integrated**	**L- T-P- C**	2	0	4	4
Version No.	**1.0**					
Course Pre-requisites	• Programming in Python • Fundamentals of Object-Oriented Programming • Basic knowledge of HTML, CSS, and Databases					
Anti-requisites	**NIL**					
Course Description	This course is designed to provide students with in-depth knowledge and practical skills required for **Full Stack Web Application Development using Python and the Django framework**. It focuses on the complete lifecycle of web application development, covering **frontend design, backend logic, database integration, RESTful services, security, testing, and deployment**.					
Course Objective	The objective of this course is to equip students with comprehensive knowledge and practical skills in **full stack web application development using Python and the Django framework**. The course aims to enable learners to design, develop, and deploy dynamic, secure, and scalable web applications by integrating frontend technologies, backend logic, databases, and RESTful services. It focuses on building strong foundations in Django architecture, database management using Django ORM, API development with Django REST Framework, and modern authentication and security mechanisms. Additionally, the course emphasizes hands-on learning through real-world projects, testing, and deployment practices, thereby preparing students for industry roles in full stack and backend development.					
Course Outcomes	**Upon successful completion of the course the students shall be able to:** **CO1:** Apply Python programming concepts and web development fundamentals including HTML, CSS, JavaScript, and version control for building web applications**.** [Apply] **CO2:** Apply Django framework concepts including MVT architecture, routing, templates, forms, and middleware for developing dynamic web applications. [Apply] **CO3:** Analyze and implement database integration using Django ORM, including model design, relationships, and query optimization. [Analyze] **CO4:** Apply REST principles and develop APIs using Django REST Framework with authentication and security mechanisms. [Apply] **CO5:** Analyze and implement testing, debugging, and logging techniques for ensuring application reliability**.** [Analyze] **CO6:** Apply deployment strategies using Docker, cloud platforms, and CI/CD pipelines for scalable and production-ready applications. [Apply]					
Course Content:						
Module 1	**Python & Web Development Fundamentals**	**[L-6, P-12 = 18**				
Review of Python Programming; Python Syntax and Program Structure; Data Types and Operators; Control Statements; Functions and Modules; Object-Oriented Programming in Python (Classes, Objects, Inheritance, Polymorphism, Encapsulation); Exception Handling; File Handling; Virtual Environments and Package Management using pip; Introduction to Web Application Architecture; Client–Server Model; HTTP and HTTPS Protocols; Basics of HTML5 (Forms, Tables, Semantic Elements); CSS3 (Selectors, Layouts, Flexbox, Media Queries); JavaScript Fundamentals; DOM Manipulation; Introduction to Responsive Web Design; Version Control using Git; GitHub Repository Management.						

Module 2	**Django Framework Fundamentals**	**[L-6, P-12 = 18**
Introduction to Django Framework; Features and Advantages of Django; Django Architecture and MVT Design Pattern; Django Request–Response Cycle; Django Project Creation and App Configuration; Settings and Environment Configuration; URL Mapping and Routing; Function-Based Views and Class-Based Views; Django Templates; Template Language, Filters, and Tags; Template Inheritance; Handling Static Files and Media Files; Django Forms and Model Forms; Form Validation Techniques; Cross-Site Request Forgery (CSRF) Protection; Middleware Concepts; Error Handling and Debugging in Django Applications.		
Module 3	**Database Integration & Django ORM**	**[L-6, P-12 = 18**
Introduction to Relational Databases; Database Design Concepts; Integration of Django with MySQL/PostgreSQL; Django ORM Architecture; Defining Models and Fields; Database Migrations and Schema Evolution; CRUD Operations using Django ORM; QuerySets and Query Optimization; Filtering, Aggregation, and Annotation; Model Relationships (One-to-One, One-to-Many, Many-to-Many); Foreign Keys and Constraints; Admin Interface Customization; Pagination and Search in Django Admin; Handling Transactions; Database Indexing and Performance Optimization.		
Module 4	**RESTful Services, Authentication & Security**	**[L-6, P-12 = 18]**
Introduction to REST Architecture; REST Principles and HTTP Methods; Status Codes and API Design Best Practices; Introduction to Django REST Framework (DRF); Serializers and Model Serializers; API Views, Generic Views, and ViewSets; URL Routing in DRF; CRUD Operations using REST APIs; Authentication Mechanisms in Django and DRF; Session-Based Authentication; Token-Based Authentication; JSON Web Tokens (JWT); User Registration and Login APIs; Role-Based Access Control; Permissions and Throttling; Securing APIs; Handling CORS; Input Validation and Error Handling; Security Best Practices in Django Applications.		
Module 5	**Testing, Deployment & Automation Tools**	**[L-6, P-12 = 18**
Frontend Integration using Bootstrap and JavaScript; Overview of Modern Frontend Frameworks; Unit Testing in Django using TestCase; Writing Tests for Models, Views, and APIs; API Testing using Postman; Debugging and Logging Techniques; Introduction to Docker; Containerization of Django Applications; Environment Variables and Production Settings; Deployment using Gunicorn and Nginx; Cloud Deployment on AWS / Render / PythonAnywhere; Static and Media File Handling in Production; Introduction to CI/CD Pipelines; Project Packaging, Documentation, and Maintenance Practices.		
List Of Laboratory Tasks: **Experiment No. 1:** **Level 1:** Install Python and configure virtual environment for Django development. **Level 2:** Manage project dependencies using pip and virtual environments. **Experiment No. 2:** **Level 1:** Create a Django project and multiple apps. **Level 2:** Analyze project and app structure with proper configuration. **Experiment No. 3:** **Level 1:** Design static web pages using HTML5 and CSS3. **Level 2:** Integrate static pages with Django templates. **Experiment No. 4:** **Level 1:** Implement URL routing and request handling using function-based views. **Level 2:** Develop applications using class-based views. **Experiment No. 5:** **Level 1:** Develop dynamic web pages using Django Template Language. **Level 2:** Implement template inheritance and manage static files. **Experiment No. 6:** **Level 1:** Create Django models and perform database migrations. **Level 2:** Integrate databases such as MySQL/PostgreSQL with Django. **Experiment No. 7:** **Level 1:** Perform CRUD operations using Django ORM. **Level 2:** Implement advanced queries with filtering, aggregation, and relationships.		

Experiment No. 8:
Level 1: Implement Django Forms and Model Forms.
Level 2: Apply server-side validation techniques.
Experiment No. 9:
Level 1: Develop user registration module using Django authentication system.
Level 2: Customize authentication workflows.
Experiment No. 10:
Level 1: Implement user login and logout functionality.
Level 2: Manage session-based authentication and session security.
Experiment No. 11:
Level 1: Design role-based access control for users.
Level 2: Implement permissions and authorization mechanisms.
Experiment No. 12:
Level 1: Use Django Admin interface for data management.
Level 2: Customize admin interface with advanced configurations.
Experiment No. 13:
Level 1: Develop a Student/Employee Management System using Django ORM.
Level 2: Extend system with additional modules and validations.
Experiment No. 14:
Level 1: Implement file upload functionality in Django.
Level 2: Handle media files and storage configurations.
Experiment No. 15:
Level 1: Create RESTful APIs using Django REST Framework.
Level 2: Implement advanced API features and best practices.
Experiment No. 16:
Level 1: Implement API authentication using token-based authentication.
Level 2: Implement JWT-based authentication and security enhancements.
Experiment No. 17:
Level 1: Consume REST APIs using frontend templates.
Level 2: Use JavaScript (AJAX/fetch) for dynamic API consumption.
Experiment No. 18:
Level 1: Perform API testing using Postman.
Level 2: Analyze HTTP responses and debug API issues.
Experiment No. 19:
Level 1: Write unit tests for Django models.
Level 2: Extend testing to views and APIs.
Experiment No. 20:
Level 1: Implement pagination in Django applications.
Level 2: Add search and filtering functionalities.
Experiment No. 21:
Level 1: Apply CSRF protection in Django forms.
Level 2: Implement input validation and security best practices.
Experiment No. 22:
Level 1: Configure production settings for Django applications.
Level 2: Manage environment variables securely.
Experiment No. 23:
Level 1: Deploy Django application on a cloud platform.
Level 2: Optimize deployment with scaling and monitoring.
Experiment No. 24:
Level 1: Containerize a Django application using Docker.
Level 2: Manage multi-container applications and deployment.
Experiment No. 25:
Level 1: Develop an end-to-end Django full stack mini project.
Level 2: Present and evaluate the project with complete deployment and documentation.
Experiment No. 26:
Level 1: Implement middleware in Django for request and response processing.

Level 2: Develop custom middleware for logging, authentication checks, or performance monitoring.
Experiment No. 27:
Level 1: Implement caching in Django using built-in cache framework.
Level 2: Optimize application performance using advanced caching strategies (Redis/Memcached).
Experiment No. 28:
Level 1: Integrate Bootstrap with Django templates for responsive UI design.
Level 2: Customize UI with advanced components and responsive layouts.
Experiment No. 29:
Level 1: Implement logging and debugging techniques in Django applications.
Level 2: Analyze logs and handle exceptions for robust application performance.
Experiment No. 30:
Level 1: Implement email functionality in Django (send emails for registration/reset password).
Level 2: Develop a complete email-based workflow (verification, password reset, notifications).

Text Book(s)

T1. "Django for Professionals" – William S. Vincent, 2020.
T2. "Full Stack Python Security" – Matt Makai, 2021.
T3. "Web Development with Django" (2nd Edition) – Ben Shaw, Saurabh Badhwar, Andrew Bird, 2022.

References

R1. "Two Scoops of Django 3.x" – Daniel Roy Greenfeld & Audrey Roy Greenfeld, Updated Edition (post-2020).
R2. "REST API Design Rulebook" – Mark Masse, Updated Edition (post-2020 reprint).
R3. "Docker Deep Dive" – Nigel Poulton, 2023.

Web Resources

1. https://docs.djangoproject.com/en/stable/
2. https://www.django-rest-framework.org/
3. https://developer.mozilla.org/en-US/docs/Learn

CSC4209- Applied Machine Learning

Course Code: CSC4209	**Course Title: Applied Machine Learning**	**L-T-P-C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	Artificial Intelligence and Machine Learning					
Anti-requisites	**NIL**					
Course Description	This course builds the foundational insight of understanding principles of machine learning process and to explore various ML algorithms and practices to accelerate strategic decision making with data. Overall, this course gives a comprehensive insight to appropriately choose ML algorithms on real time problems to construct an intelligent ML model.					
Course Objective	The objective of the course is to familiarize the learners to explore the competence and comprehend with potential machine learning algorithms and techniques to revolutionize with real-world problems and create prominent solutions (with ML models) to attain Employability Skills through Experiential Learning techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply fundamental learning theories such as PAC learning, VC dimension, and risk minimization techniques in machine learning models. [Apply] **CO2:** Apply ensemble learning methods including Random Forests, boosting techniques, and gradient-based models for improved prediction performance. [Apply] **CO3:** Analyze kernel-based methods such as SVMs, Kernel PCA, and probabilistic models including Bayesian networks and EM algorithm. [Analyze] **CO4:** Apply dimensionality reduction techniques such as LDA, ICA, NMF, and regularization methods for efficient data representation. [Apply] **CO5:** Analyze graph-based machine learning techniques including spectral methods and semi-supervised learning. [Analyze] **CO6:** Evaluate advanced machine learning applications including GNNs, NLP techniques, and domain-specific solutions in healthcare, business, and recommender systems. [Evaluate]					
Course Content:						
Module 1	**Introduction to Learning techniques**					**No. of Sessions:7**
PAC Learning and VC Dimension: Definitions, Intuition, and Applications. Empirical Risk Minimization (ERM) vs. Structural Risk Minimization (SRM). Ensemble Learning: Advanced Bagging and Boosting Techniques (Random Forests, XBoost, LightGBM).						
Module 2	**Kernel Methods**					**No. of Sessions:7**
Kernel Methods: Advanced SVMs with non-linear kernels, Kernel Ridge Regression, Kernel PCA. Latent Semantic Analysis (LSA) and Applications. Bayesian Inference: Probabilistic Graphical Models, Bayesian Belief Networks, Expectation-Maximization (EM) Algorithm, and Gaussian Processes						
Module 3	**Dimensionality Reduction Techniques**					**No. of Sessions:8**
Dimensionality Reduction Techniques: Canonical Correlation Analysis (CCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF). Matrix Factorization and Low-Rank Approximation. Regularization Techniques: L1 and L2 Penalties.						
Module 4	**Applications in Advanced Machine Learning**					**No. of Sessions:8**
Graph-Based Machine Learning: Graph Construction, Laplacian Matrix, Smoothness Constraints, and Label Propagation. Semi-Supervised Learning on Graphs, Spectral						

Methods, Graph Embeddings. Introduction to Graph Neural Networks (GNNs): Architecture, Message Passing, and Applications in Classification and Link Prediction. Implementation of NLP techniques in Machine Learning. Applications of Advanced ML in Recommender Systems, Healthcare, Business Intelligence, and Scientific Domains

List of Laboratory Tasks

Experiment No 1: Empirical Risk Minimization (ERM) Vs Structural Risk Minimazation (SRM)
Level 1: Implementation of Regression/Classification, L1 and L2 regularization
Level 2: Bias Variance Analysis
Experiment No 2: Implementation of Random Forest
Level 1: Comparison of Decision tree
Level 2: Performance evaluation metrics.
Experiment No 3: Implementation of boosting Techiques – XGBoost / LightGBM
Level 1: Model training
Level 2: Hyperparameter tuning and comparison with random Forest
Experiment No 4: Implementation of Support Vector Machines with Non-Linear Kernels
Level 1: RBF and Polynomial Kernels
Level 2: Decision boundary visualization
Experiment No 5: Expectation-Maximization (EM) Algorithm
Level 1: Implementation of Gaussian Mixure Model
Level 2: Cluster convergence analysis.
Experiment No 6: Implementation of Advanced Dimensionality Reduction
Level 1: Canonical Correlation Ananlysis (CCA), Independent Component Analysis (ICA**)**
Level 2: Non-negative Matrix Factorization (NMF)
Experiment No 7: Graph Neural Networks (GNNs)
Level 1: Graph Convolutional Network (GCN).
Level 2: Node Classification or Link Prediction
Experiment No 8: Latent Semantic Analysis (LSA) using SVD
Level 1: Document-term matrix
Level 2: Topic extraction and visualization
Experiment No 9: Kernel Ridge Regression
Level 1: Implementation of Kernel Ridge Regression
Level 2: Comparison with Linear Regression and analysis of regularization effects
Experiment No 10: Bayesian Belief Networks
Level 1: Construct a simple Bayesian Network for a dataset
Level 2: Perform probabilistic inference and analyze conditional dependencies
Experiment No 11: Gaussian Processes
Level 1: Implement Gaussian Process Regression
Level 2: Analyze uncertainty estimation and compare with traditional regression models
Experiment No 12: Linear Discriminant Analysis (LDA)
Level 1: Implement LDA for dimensionality reduction
Level 2: Compare classification performance before and after applying LDA
Experiment No 13: Regularization Techniques
Level 1: Implement L1 (Lasso) and L2 (Ridge) regularization
Level 2: Compare model performance and feature selection capability
Experiment No 14: Graph-Based Semi-Supervised Learning
Level 1: Implement label propagation on graph data
Level 2: Evaluate performance on partially labelled datasets
Experiment No 15: Recommender Systems using Machine Learning
Level 1: Implement collaborative filtering techniques
Level 2: Evaluate recommendations using metrics and compare with content-based methods

Targeted Application & Tools that can be used:

Targeted Applications:

Developing applications on Recommender systems, Business Intelligence & Finance, NLP, Scientific & Engineering Domains, Smart systems & IoT

Tools & Platforms:

- **NumPy, Pandas** – Data handling
- **Scikit-learn** – Classical ML algorithms
- **SciPy** – Scientific computing
- **Statsmodels** – Statistical analysis

Advanced Deep learning Frameworks:

- TensorFlow
- Keras
- PyTorch

Project work/Assignment:

Analysis of Case Studies: Deciding to adopt Machine learning algorithms to solve real world problems.

Text Book(s)

T1. "Machine Learning: A Probabilistic Perspective" – Kevin P. Murphy, Updated Edition (post-2020 reprint).
T2. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (3rd Edition) – Aurélien Géron, 2022.
T3. "Machine Learning Engineering with Python" – Andrew P. McMahon, 2021.

References

R1. "Pattern Recognition and Machine Learning" – Christopher M. Bishop, Updated Edition (post-2020 reprint).
R2. "Deep Learning" – Ian Goodfellow, Yoshua Bengio & Aaron Courville, Updated Edition (post-2020 reprint).
R3. "Graph Representation Learning" – William L. Hamilton, Updated Edition (post-2020 reprint).

Web Resources

1. https://scikit-learn.org/stable/
2. https://pytorch.org/tutorials/
3. https://developers.google.com/machine-learning

CSC4210- Data Analysis and Visualization

Course Code: CSC4210	**Course Title: Data Analysis and Visualization Techniques** **Type of Course: Theory + Lab**	**L-T-P-C**	2	0	2	3
Version No.	1.0					
Course Pre-requisites	Python Programming					
Anti-requisites	NIL					
Course Description	The Course consists of two parts where first Part covers advanced analytics that covers topics necessary to give businesses greater insight into their data than they could ordinarily, and the Second Part covers data visualization concepts. Primary concepts include machine learning, data mining, predictive analytics, location analytics, big data analytics, and location intelligence. Visualization for Time series, Geolocated data, Correlations, connections, Hierarchies, networks, and interactivity.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of **Data Analysis and Visualization Techniques** and attain **Skill Development** through **Experiential Learning** techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply data analytics concepts including data types, feature engineering, and data preparation techniques**.** [Apply] **CO2:** Apply advanced analytical methods including clustering, model selection, and performance evaluation for data-driven decision making. [Apply] **CO3:** Analyze statistical methods and machine learning techniques for extracting insights from datasets. [Analyze] **CO4:** Apply principles of data visualization including perception, interaction, and visualization techniques for different data types. [Apply] **CO5:** Analyze and design effective visualizations using modern tools for time-oriented, network, and geospatial data. [Analyze] **CO6:** Evaluate visualization systems and techniques based on effectiveness, usability, and benchmarking standards. [Evaluate]					
Course Content:						
Module 1	**Data Analytics**					**L-6, P-6 Sessions**
Characteristics and types of data, Types of Analytics, Location Analytics, Working with Geospatial Data, Feature Engineering and Selection, Dimensionality Reduction Techniques, Data Preparation.						
Module 2	**Advanced Analytics**					**L-8, P-8 Sessions**
Statistical methods for Data Analytics, Advance topics in Supervised and Unsupervised Machine Learning: Cluster Analysis, Hyper-Parameter Tuning, Measuring Performance of the Models, Model Selection, Data Mining techniques						
Module 3	**Introduction to Data Visualization**					**L-8, P-8 Sessions**
Fundamentals of Data Visualization, Human Perception, Basic plotting techniques, Interaction concepts, Visualization techniques for Time Oriented data, Visualization techniques Networks. Introduction to Data Visualization Tools.						
Module 4	**Application - Data Visualization**					**L-8, P-8 Sessions**

Designing effective Visualizations, Advanced Visualization Tools, Visualizing Geospatial Data, Document Visualization, Visualization Systems, Evaluating Visualizations, Visualization Benchmarking.

List Of Laboratory Tasks

Experiment 1:
Level 1: Collect real-world datasets and classify them based on structure and type.
Level 2: Analyze datasets based on domain and identify suitable use cases
Experiment 2:
Level 1: Perform exploratory data analysis to identify missing values and summary statistics.
Level 2: Detect outliers and visualize distributions using appropriate plots.
Experiment No. 3:
Level 1: Implement descriptive and diagnostic analytics on a dataset.
Level 2: Apply predictive and prescriptive analytics for decision-making.
Experiment No. 4:
Level 1: Analyze location-based data using latitude and longitude.
Level 2: Perform advanced location analytics and visualize spatial patterns.
Experiment No. 5:
Level 1: Apply feature creation, transformation, and encoding techniques.
Level 2: Implement normalization and evaluate impact on model performance.
Experiment No. 6:
Level 1: Perform dimensionality reduction using PCA.
Level 2: Analyze reduced feature space and interpret results.
Experiment No. 7:
Level 1: Apply statistical measures such as correlation and regression.
Level 2: Perform hypothesis testing and analyze probability distributions.
Experiment No. 8:
Level 1: Build supervised learning models like Linear Regression and k-NN.
Level 2: Implement Decision Trees and compare model performance.
Experiment No. 9:
Level 1: Perform clustering using K-Means algorithm.
Level 2: Apply hierarchical clustering and evaluate cluster quality.
Experiment No. 10:
Level 1: Measure model performance using accuracy and precision.
Level 2: Evaluate models using recall, F1-score, and ROC-AUC.
Experiment No. 11:
Level 1: Apply association rule mining techniques.
Level 2: Implement classification techniques on large datasets.
Experiment No. 12:
Level 1: Create basic visualizations such as line, bar, scatter, and pie charts.
Level 2: Analyze the effect of color, scale, and layout on interpretation.
Experiment No. 13:
Level 1: Visualize time-series data using line plots.
Level 2: Perform trend analysis and pattern identification.
Experiment No. 14:
Level 1: Implement interactive visualizations with basic tools.
Level 2: Add advanced interactivity such as zooming, filtering, and tooltips.
Experiment No. 15:
Level 1: Design dashboards using visualization principles.
Level 2: Evaluate visualization effectiveness and implement GIS-based spatial visualizations.

Project Work / Assignment / Case Study

1. **Problem Solving:** Understand and break down the given real scenario into business problem statement.
2. **Programming:** Implementation of any suitable Modelling Technique.

3. **Mini project**: Choose a real-world problem in any domain and provide the solution by applying any modelling technique along with the suitable visualization tools.

Text Book(s)

T1. "Storytelling with Data" – Cole Nussbaumer Knaflic, Updated Edition (post-2020 reprint)

T2. "Data Visualization: A Practical Introduction" (2nd Edition) – Kieran Healy, 2021

T3. "Python for Data Analysis" (3rd Edition) – Wes McKinney, 2022

References

R1. "Fundamentals of Data Visualization" – Claus O. Wilke, Updated Edition (post-2020 reprint).

R2. "Practical Statistics for Data Scientists" (2nd Edition) – Peter Bruce, Andrew Bruce & Peter Gedeck, 2020.

R3**.** "Interactive Data Visualization for the Web" (2nd Edition Updated) – Scott Murray, post-2020 reprint.

Web Resources

1. https://www.tableau.com/learn

2. https://matplotlib.org/stable/tutorials/index.html

3. https://seaborn.pydata.org/

CSC4211- Web Application Development

Course Code: CSC4211	**Course Title: Web Application Development** **Type of Course: Lab**	**L-T- P- C**	1	0	4	3
Version No.	1.0					
Course Pre-requisites	**HTML, PHP**					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive understanding of Web application development, covering both foundational and advanced web application concepts. Learners will gain hands-on experience with **Java script**, **NodeJS, and ReactJS** for building robust web applications. A blend of theory and intensive programming practice ensures industry-readiness in web application development.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply JavaScript concepts including control structures, functions, arrays, and DOM manipulation for client-side web development. [Apply] **CO2:** Apply server-side development using Node.js and Express framework to handle HTTP requests, routing, and middleware. [Apply] **CO3:** Apply ReactJS fundamentals including components, state, props, and lifecycle methods for building interactive user interfaces. [Apply] **CO4:** Analyze and implement advanced ReactJS concepts such as hooks, routing, and performance optimization techniques. [Analyze] **CO5:** Apply database integration using MongoDB and PostgreSQL for performing CRUD operations in web applications**.** [Apply] **CO6:** Evaluate and develop full-stack web applications by integrating frontend, backend, and database components. [Evaluate]					
Course Content:						
Module 1	**Java Script Core Concepts – CO1**	**15 Sessions (3T + 12P)**				
Introduction – Operators – Control Structures: Selection: if – if-else – switch. Repetition: while – do-while – for – break and continue. Functions: Function Definition – Scope Rules – Recursion. Array: Declaration – Initialization – Growing Arrays – Passing Arrays to Function. Event Handling-DOM-Local Storage. Introduction – Operators – Control Structures: Selection: if – if-else – switch. Repetition: while – do-while – for – break and continue. Functions: Function Definition – Scope Rules – Recursion. Array: Declaration – Initialization – Growing Arrays – Passing Arrays to Function. Event Handling-DOM-Local Storage.						
Module 2	**Server-side JS Framework – CO2**	**15 Sessions (3T + 12P)**				
Node JS: Introduction – Architecture – Features – Creating Web Servers with HTTP Request – Response – Event Handling – GET and POST Methods – Modules – Express: Routing-Middleware-Error Handling.						
Module 3	**ReactJS– Part 1 – CO3**	**15 Sessions (3T + 12P)**				
React: Introduction – Installation –create React app - components – state – props - props validation – state vs props – constructor – Component API – Component Life						

cycle - Forms – controlled and uncontrolled component – Events – conditional rendering.		
Module 4	**ReactJS - Part 2– CO4**	**15 Sessions (3T + 12P)**
List – keys – refs – Fragments - Router – CSS – Animation – Map – Table –Code splitting – Hooks: useState–useEffect– useContext.		
Module 5	**Database– CO5**	**15 Sessions (3T + 12P)**
MongoDB: Basics- Connect to NoSQL Database using Node JS – Implementation of CRUD operations- PostgreSQL-Basics-Schema-Installation and connection-CRUD operations.		

List of Laboratory Tasks:

Experiment No. 1:
Level 1: Simple form validation using JavaScript.
Level 2: Build a GUI based application to perform form validation

Experiment No. 2:
Level 1: Design a webpage to create simple interactive CGPA calculator using Event Handling.
Level 2: Design a web application using HTTP Request and HTTP Response

Experiment No. 3:
Level 1: Develop simple login page by performing event handling using GET and POST method.
Level 2: Build web applications to connect with a database using Java script that perform database manipulations.

Experiment No. 4:
Level 1: Design a simple calculator using 'Modules' in Node.js.
Level 2: Implement a web application based on the MVC design pattern, to create an Employee Registration module using Node.js.

Experiment No. 5:
Level 1: Design a web application to show how a server-side application responds to a client request to a particular endpoint using Routing in Express.
Level 2: Create a User Registration project using Routing in Express.

Experiment No. 6:
Level 1: Design a webpage to maintain personal information using CRUD operations in MongoDB.
Level 2: Design a webpage to maintain personal information using CRUD operations in PostgreSQL.

Experiment No. 7:
Level 1: Design a web application using components and modules in React.
Level 2: Design a web application with routing in React.

Experiment No. 8:
Level 1: Implement various Hooks in ReactJS.

Experiment No. 9:
Level 1: Design a webpage demonstrating DOM manipulation (create, update, delete elements dynamically).
Level 2: Build an interactive task manager application using DOM manipulation and local storage.

Experiment No. 10:
Level 1: Implement JavaScript functions to handle arrays and perform operations like sorting and searching.
Level 2: Develop a dynamic data table with filtering, sorting, and pagination features.

Experiment No. 11:
Level 1: Create a Node.js server to handle file operations (read/write).
Level 2: Develop a RESTful API using Express for file management with proper error handling.

Experiment No. 12:
Level 1: Design a simple React application using props and state management.
Level 2: Build a multi-component React application with state lifting and props validation.
Experiment No. 13:
Level 1: Implement form handling in React using controlled components.
Level 2: Develop a complete form-based application with validation and submission handling.
Experiment No. 14:
Level 1: Connect a Node.js application with MongoDB and perform basic CRUD operations.
Level 2: Develop a full-stack CRUD application integrating React frontend, Node.js backend, and MongoDB database.
Experiment No. 15:
Level 1: Implement basic authentication using sessions in Node.js.
Level 2: Develop a secure authentication system using JWT with role-based access control.

Targeted Application & Tools that can be used:
Node.js, React, MongoDB, PostgreSQL, MySQL.

Project work/Assignment:

1. Build a Library Management System using Java script.
2. Develop a Student Registration Portal using Java Script.
3. Create an Employee Management System using ReactJS.
4. Build a ReactJS based Inventory Management System with RESTful API support.

Topics related to
1. Problem Solving: Develop a full-stack Java web application integrating Java script, NodeJS, and ReactJS to address real-world business challenges such as inventory management or customer tracking.
2. Employability: Simulate industry-level scenarios using tools like Apache Maven and Selenium for automated testing, and apply Spring Boot REST APIs to build scalable web services, enhancing job-readiness in enterprise Java development.

Textbook(s):
T1. "Node.js Design Patterns" (3rd Edition) – Mario Casciaro & Luciano Mammino, 2020.
T2. "Learning React" (2nd Edition) – Alex Banks & Eve Porcello, 2020.
T3. "Full-Stack JavaScript Development with MERN" – Shama Hoque, 2021.

References
R1. "React Cookbook" – David Griffiths & Dawn Griffiths, 2021.
R2. "Web Development with Node and Express" (2nd Edition Updated) – Ethan Brown, post-2020 reprint.
R3. "MongoDB: The Definitive Guide" (3rd Edition) – Shannon Bradshaw et al., 2020.

Web Resources
1. https://developer.mozilla.org/en-US/docs/Web/JavaScript
2. https://react.dev/learn
3. https://nodejs.org/en/docs/

Semester III
CSC4212- Deep Learning

Course Code: CSC4212	**Course Title: Deep Learning Type of Course: Theory & Integrated Laboratory**	**L-T-P- C**	2	0	4	4
Version No.	1.0					
Course Pre-requisites	Artificial Intelligence & Machine learning					
Anti-requisites	**NIL**					
Course Description	The course introduces the core intuitions behind Deep Learning, an advanced branch of Machine Learning involved in the development, implementation and application of Artificial Neural Networks that function by simulating the working principle of human brain. Deep learning algorithms extract layered high-level representations of data in a way that maximizes performance on a given task. The course includes theory and lab components which emphasizes on understanding the implementation and application of deep neural networks in various prominent problem domains like speech recognition, sentiment analysis, recommendations, and computer vision etc. The course facilitates the students to interpret and appreciate the successful application and implementation of deep neural nets in various prediction and classification tasks of ML.					
Course Objective	On successful completion of the course the students shall be able to familiarize the learners with the concepts of **Deep Learning** and attain **Skill Development** through **Participative Learning** techniques.					
Course Outcomes	**On successful completion of this course, the student will be able to:** **CO1:** Apply fundamental deep learning concepts including gradient descent optimization techniques and neural network training. [Apply] **CO2:** Apply feedforward neural networks and perform hyperparameter tuning and regularization techniques. [Apply] **CO3:** Analyze convolutional neural networks for image and text classification tasks. [Analyze] **CO4:** Apply sequence modeling techniques using RNNs, LSTMs, and encoder-decoder architectures. [Apply] **CO5:** Analyze different types of autoencoders and their representational capabilities. [Analyze] **CO6:** Evaluate deep learning models for performance, generalization, and real-world applications. [Evaluate]					
Course Content:						
Module 1	**Introduction to Deep Learning**	**L-6, P-12 Sessions**				
Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive sub-gradient method. Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyper parameter tuning.						
Module 2	**Convolution Neural Networks**	**L-8, P-16 Sessions**				
Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.						
Module 3	**Sequence Modeling**	**L-8, P-16 Sessions**				

Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks, LSTM networks.

Module 4	**Autoencoders**	**L-8, P-16 Sessions**

Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

List of Laboratory Tasks:

Experiment 1: Implement logistic regression classification using gradient descent and plot the cost function over iterations.
Level 1: Apply the concept of gradient descent to optimize logistic regression parameters.
Level 2: Implement the model and visualize convergence behavior using cost function plots.
Experiment 2: Implement logistic regression classification using stochastic gradient descent and compare convergence behavior.
Level 1: Apply stochastic optimization techniques for model training.
Level 2: Implement and analyze convergence differences between batch and stochastic methods.
Experiment 3: Experiment with logistic regression using momentum and adaptive sub-gradient optimization methods.
Level 1: Analyze advanced optimization techniques for improving convergence.
Level 2: Experiment with momentum and adaptive learning strategies to evaluate performance improvements.
Experiment 4: Compute binary cross-entropy loss for a logistic classifier.
Level 1: Apply loss function concepts in classification problems.
Level 2: Implement and validate the correctness of binary cross-entropy computation.
Experiment 5: Implement perceptron learning for Boolean functions (NOT, OR, AND, NOR, NAND).
Level 1: Apply perceptron learning algorithm for simple logical operations.
Level 2: Implement and verify model behavior for linearly separable problems.
Experiment 6: Develop a feed-forward neural network for regression problems with hyperparameter tuning.
Level 1: Apply neural network concepts for regression tasks.
Level 2: Implement and tune hyperparameters to improve model accuracy.
Experiment 7: Implement a feed-forward neural network for binary classification tasks.
Level 1: Apply neural network architecture for classification problems.
Level 2: Implement and evaluate classification performance using suitable metrics.
Experiment 8: Train a multi-class classification neural network using softmax output layer.
Level 1: Apply softmax activation for multi-class classification.
Level 2: Implement and analyze model performance across multiple classes.
Experiment 9: Analyze the effect of learning rate variation on neural network performance.
Level 1: Analyze the impact of learning rate on training stability and convergence.
Level 2: Experiment with multiple learning rates and compare performance metrics.
Experiment 10: Explore the effect of network depth and number of neurons on performance.
Level 1: Analyze how architecture complexity affects learning capability.
Level 2: Experiment with different configurations to identify optimal structure.
Experiment 11: Compare stochastic gradient descent and batch gradient descent for prediction tasks.
Level 1: Analyze differences between optimization techniques.
Level 2: Perform comparative evaluation based on convergence speed and accuracy.

Experiment 12: Build a neural network model for employee attrition prediction.
Level 1: Apply neural networks to real-world prediction problems.
Level 2: Implement and evaluate model performance using real datasets.
Experiment 13: Implement a basic convolutional neural network for image classification.
Level 1: Apply convolutional neural network concepts for image processing.
Level 2: Implement and evaluate classification accuracy on image datasets.
Experiment 14: Study the effect of convolution kernel size on feature extraction.
Level 1: Analyze how kernel size impacts feature detection.
Level 2: Experiment with different kernel sizes and observe feature maps.
Experiment 15: Apply striding and pooling operations to study dimensionality reduction.
Level 1: Apply convolution operations for reducing spatial dimensions.
Level 2: Implement and analyze the effect on output size and performance.
Experiment 16: Compare max pooling and average pooling in convolutional neural networks.
Level 1: Analyze pooling techniques in feature reduction.
Level 2: Perform comparative evaluation on model accuracy and feature retention.
Experiment 17: Implement a CNN for handwritten digit classification using a standard dataset.
Level 1: Apply CNN architecture for classification tasks.
Level 2: Implement and validate model performance on benchmark datasets.
Experiment 18: Visualize feature maps and learned filters in convolution layers.
Level 1: Analyze internal representations learned by CNNs.
Level 2: Implement visualization techniques to interpret learned features.
Experiment 19: Fine-tune a pre-trained CNN model for a custom image dataset.
Level 1: Apply transfer learning concepts using pre-trained models.
Level 2: Implement fine-tuning and evaluate performance improvements.
Experiment 20: Perform image data augmentation for CNN training.
Level 1: Apply data augmentation techniques to improve generalization.
Level 2: Implement augmentation methods and analyze impact on accuracy.
Experiment 21: Classify plant leaf diseases using convolutional networks.
Level 1: Apply CNN models for real-world classification tasks.
Level 2: Implement and evaluate model performance on domain-specific datasets.
Experiment 22: Implement a basic recurrent neural network for sequence prediction.
Level 1: Apply RNN concepts for sequential data modeling.
Level 2: Implement and analyze prediction performance on sequence data.
Experiment 23: Implement a deep recurrent neural network with multiple RNN layers.
Level 1: Apply deep learning concepts to sequential models.
Level 2: Implement stacked RNN architecture and evaluate performance.
Experiment 24: Implement and compare GRU and LSTM networks for sequence modeling.
Level 1: Analyze gated recurrent architectures for sequence learning.
Level 2: Implement and compare performance of GRU and LSTM models.
Experiment 25: Develop an encoder–decoder sequence-to-sequence model.
Level 1: Apply sequence-to-sequence learning concepts.
Level 2: Implement encoder-decoder architecture for sequence tasks.
Experiment 26: Analyze the vanishing gradient problem in long sequences.
Level 1: Analyze training challenges in deep recurrent networks.
Level 2: Experiment and observe gradient behavior across long sequences.

Experiment 27: Compare RNN, LSTM, and bidirectional RNN for sequence prediction tasks.
Level 1: Analyze different recurrent architectures.
Level 2: Perform comparative evaluation on performance metrics.
Experiment 28: Generate text sequences using recurrent neural networks.
Level 1: Apply sequence generation concepts using RNNs.
Level 2: Implement and evaluate generated text quality.
Experiment 29: Implement autoencoders including sparse and denoising variants.
Level 1: Apply unsupervised learning techniques for representation learning.
Level 2: Implement and compare different autoencoder variants.
Experiment 30: Apply autoencoders and GANs for anomaly detection, image compression, and generation tasks.
Level 1: Analyze generative models for unsupervised learning applications.
Level 2: Implement and evaluate models on real-world datasets.

Targeted Application & Tools that can be used:
Python

Assignment:
Assignments are given after completion of each module which the student need to submit within the stipulated deadline.

Text Book(s)
T1. "Deep Learning with Python" (2nd Edition) – François Chollet, 2021.
T2. "Dive into Deep Learning" – Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola, 2023.
T3. **"**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (3rd Edition) – Aurélien Géron, 2022.

References
R1. "Deep Learning" – Ian Goodfellow, Yoshua Bengio & Aaron Courville, Updated Edition (post-2020 reprint).
R2. "Neural Networks and Deep Learning" – Charu C. Aggarwal, Updated Edition (post-2020 reprint).
R3. "Pattern Recognition and Machine Learning" – Christopher M. Bishop, Updated Edition (post-2020 reprint).

Web Resources
1. https://pytorch.org/tutorials/
2. https://www.tensorflow.org/tutorials
3. https://d2l.ai/

CSC4213- Big Data Analytics

Course Code: CSC4213	**Course Title: Big Data Analytics** **Type of Course: Theory & Integrated Laboratory**	**L-T-P- C**	3	0	2	4
Version No.	2.0					
Course Pre-requisites	DDL, DML of SQL Queries and Creation of Class & object, interface, reading & writing a file, control statements in java programming.					
Anti-requisites	**NIL**					
Course Description	This course is designed to provide the fundamental knowledge to equip students being able to handle real world big data problems including the three key resources of Big Data: people, organizations, and sensor. With the advancement of IT storage, processing, computation and sensing technologies, big data has become a novel norm of life.					
Course Objective	The objective of the course is to familiarize the learners with the concepts **Big Data Analytics** and attain **Skill Development** through E**xperiential Learning** techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply fundamental deep learning concepts including gradient descent optimization techniques and neural network training. [Apply] **CO2:** Apply feedforward neural networks and perform hyperparameter tuning and regularization techniques. [Apply] **CO3:** Analyze convolutional neural networks for image and text classification tasks. [Analyze] **CO4:** Apply sequence modelling techniques using RNNs, LSTMs, and encoder-decoder architectures. [Apply] **CO5:** Analyze different types of autoencoders and their representational capabilities. [Analyze] **CO6:** Evaluate deep learning models for performance, generalization, and real-world applications. [Evaluate]					
Course Content:						
Module 1	**Introduction to Big data Analytics**	**20 Hours (T12+ L8)**				
Introduction to Big Data: Basics of Distributed File System, Four Vs, Drivers for Big data, Big data applications, Structured, unstructured, semi-structured and quasi structured data. Big data Challenges-Traditional versus big data approach, The Big Data Technology Landscape: No-SQL The Hadoop: History of Hadoop-Hadoop use cases, The Design of HDFS, Blocks and replication management, Rack awareness, HDFS architecture, HDFS Federation, Name node and data node, Anatomy of File write, Anatomy of File read.						
Module 2	**Hadoop MapReduce Framework**	**19 Hours (T11+ L8)**				
MapReduce: Hadoop Map Reduce paradigm,Map and reduce tasks, Job Tracker and task tracker,Map reduce execution pipeline, Key value pair, Shuffle and sort, Combiner and Partitioner, APIs used toWrite/Read files into/from Hadoop. Hadoop 2.0 Features, Name Node High Availability, YARN Architecture.						
Module 3	**Hive and Hbase Analytical tools**	**18 Hours (T11+ L7)**				
Hive : Apache Hive with Hive Installation, Hive Data Types, Hive Table partitioning, Hive DDL commands, Hive DML commands, and Hive sort by vs. order by, Hive Joining tables, Hive bucketing.						

Hbase: Introduction to HBase and its working architecture- Commands for creation and listing of tables- disabled and is disabled of table - enable and is enabled of table- describing and dropping of table-Put and Get command - delete and delete all command-commands for scan, count, truncate of tables.

Module 4	**Data Analytics with Spark**	**18 Hours (T11+ L7)**

Spark: Introduction to Apache Spark A unified Spark, Who uses Spark and for what?, A Brief History of Spark, Spark version and releases, Storage layers for Spark. Programming with RDDs: RDD Basics, Creating RDDs, RDD Operations, Passing functions to Spark, Common Transformations and Actions, Persistence. Scala: The Basics, Control Structures and functions, Working with arrays, Maps and Tuples.

List of Laboratory Tasks:

Experiment 1: Installation of Hadoop single node cluster using Ubuntu operating system

Level 1: Explain Hadoop concepts.

Level 2: Demonstrate Installation of Hadoop single node cluster

Experiment 2: Imagine you're managing a large-scale data processing task for a company that deals with huge volumes of data. You're responsible for using Hadoop to handle and analyze this data. In this context, you frequently need to execute various Hadoop commands to perform tasks like storing data, querying datasets, and managing distributed storage. Your goal is to efficiently leverage these commands to ensure smooth and effective data processing operations.

Level 1: Describe Hadoop concepts.

Level 2: Demonstrate various Hadoop Commands

Experiment 3: Imagine you're working on a project where you need to analyze a large collection of text documents to determine the frequency of each word. To handle this efficiently, you decide to use a MapReduce approach. In this scenario, you'll use a sample dataset of text documents. Your goal is to implement a MapReduce job that processes these documents to count how many times each word appears across the entire dataset.

Level 1: Describe MapReduce concepts.

Level 2: Demonstrate Word Count analysis

Experiment 4: Imagine you're working for a financial analytics company that wants to optimize its stock analysis process. The company has collected a vast amount of stock market data over the past decade. To efficiently analyze this data and extract valuable insights, you decide to use a MapReduce approach. You start by splitting the data into manageable chunks, then apply the Map function to process each chunk, and finally use the Reduce function to aggregate and summarize the results. This way, you can handle large datasets and perform complex stock analysis quickly and effectively.

Level 1: Describe MapReduce concepts.

Level 2: Demonstrate Stock analysis program.

Experiment 5: "Imagine you're working for a company that needs to understand the web traffic patterns on their site. To do this, you decide to use a sample dataset of web logs. By employing MapReduce, you can efficiently process and analyze this data to uncover insights about user behavior and site performance. How would you approach this scenario?"

Level 1: Describe MapReduce concepts.

Level 2: Demonstrate Web log analysis program

Experiment 6: Imagine you're working for a weather forecasting company that needs to analyze temperature data collected from various weather stations across the country. You have a large dataset with temperature readings from thousands of stations. To make sense of this massive amount of data, you decide to use a MapReduce approach.

Level 1: Describe MapReduce concepts.

Level 2: Demonstrate Temperature analysis program

Experiment 7: You're managing a large dataset of customer transactions and need to analyze it using Hive.
Level 1: Define hive concepts.
Level 2: Demonstrate basic hive commands
Experiment 8: Imagine you're working as a data analyst at a company that deals with large datasets stored in Hadoop. You've been tasked with optimizing queries and extracting valuable insights from these datasets. To achieve this, you need to use advanced Hive commands to efficiently manage and analyze the data.
Level 1: Describe hive concepts.
Level 2: Demonstrate advanced hive commands
Experiment 9: Imagine you're working for a company that maintains a large customer database. You've been tasked with using HBase to perform various operation ns on this database. Here's how you might approach it with some basic HBase commands.
Level 1: Define hbase concepts.
Level 2: Demonstrate basic hbase commands
Experiment 10: Imagine you're working on a large-scale data processing project where you need to efficiently manage and interact with a massive HBase database.
Level 1: Describe hbase concepts.
Level 2: Demonstrate advance hbase commands
Experiment 11: Install, Deploy & configure Apache Spark
Level 1: Define Apache Spark concepts.
Level 2: Demonstrate Install, Deploy & configure Apache Spark
Experiment 12: You're working on a text analysis project for a large publishing company. They've asked you to develop a solution using Apache Spark to help them analyze a collection of text files.
Level 1: Describe Apache spark concepts.
Level 2: Demonstrate count the occurrences words in a given text file program
Experiment 13: Write a MapReduce program to create an index from a collection of documents to allow for fast keyword-based search.
Level 1: Map Function: Emit a key-value pair where the key is a word and the value is the document identifier.
Level 2: Reduce Function: Combine the document identifiers for each word to create an index.
Level 1: Define fast keyword-based search.
Level 2: Demonstrate identifiers for each word to create an index.
Experiment 14: Write a MapReduce program to analyze relationships and interactions in a social network graph (e.g., identifying communities or influential nodes).
Level 1: Map Function: Emit relationships or interactions between users as key-value pairs.
Experiment 15: Reduce Function: Aggregate and analyze the data to find communities, influential users, etc.
Level 1: Demonstrate relationships or interactions between users as key-value pairs.
Level 2: Aggregate and analyze the data to find communities, influential users, etc.

Targeted Application & Tools that can be used:
HDFS – for data storage
Map reduce – Mapping and reducing.
Hive - Database
Hbase – No SQL
Spark – SCALA LANGUAGE

Assignment:
Built-in Functions of hadoop mapreduce framework in java and basics of scala , Industry oriented latest Hadoop ecosystem tool.
Dataset resource link:
https://www.kaggle.com/datasets

Text Book(s)

T1. "Big Data Fundamentals: Concepts, Drivers & Techniques" – Thomas Erl, Wajid Khattak & Paul Buhler, Updated Edition (post-2020 reprint).
T2. "Hadoop: The Definitive Guide" (4th Edition Updated) – Tom White, post-2020 reprint.
T3. "Learning Spark: Lightning-Fast Data Analytics" (2nd Edition) – Jules S. Damji, Brooke Wenig, Tathagata Das & Denny Lee, 2020.

References

R1. "Spark: The Definitive Guide" – Bill Chambers & Matei Zaharia, Updated Edition (post-2020 reprint).
R2. "NoSQL Distilled" – Pramod J. Sadalage & Martin Fowler, Updated Edition (post-2020 reprint).
R3. "HBase: The Definitive Guide" – Lars George, Updated Edition (post-2020 reprint).

Online-Resources

1. https://hadoop.apache.org/docs/
2. https://spark.apache.org/docs/latest/
3. https://cwiki.apache.org/confluence/display/Hive/Home

CSC4214- Data Pipelines & ETL

Course Code: CSC4214	**Course Title: Data Pipelines & ETL** **Type of Course: Theory & Integrated Laboratory**	**L-T-P- C**	2	0	2	3
Version No.	2.0					
Course Pre-requisites	Programming in Python or Java, Databases & SQL fundamentals, Basics of Data Science and Data Structures					
Anti-requisites	**NIL**					
Course Description	This course builds foundational and advanced skills in designing, implementing, and managing data pipelines, the backbone of analytics and data science systems. Students learn how data is extracted from sources, transformed for quality & consistency, and loaded into analytical storage systems using modern tools such as Apache Airflow, Kafka, and batch/streaming frameworks.					
Course Objective	• Understand key concepts of data movement and transformation at scale • Build and deploy robust ETL pipelines both in batch and streaming modes • Orchestrate workflows using open-source tools • Monitor, debug and optimize pipeline performance • Learn best practices for data quality, governance, and metadata management					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Design and document an end-to-end data pipeline architecture for real-world data engineering use cases. [Create] **CO2:** Implement batch and streaming ETL workflows using Python and industry-standard tools. [Analyse] **CO3:** Integrate data pipelines with Big Data ecosystems such as Hadoop and Spark for scalable data processing. [Apply] **CO4:** Develop and manage workflow orchestration using tools like Apache Airflow and implement messaging systems using Apache Kafka. [Create] **CO5:** Evaluate pipeline performance, optimize data processing efficiency, and ensure data quality, governance, and security compliance. [Evaluate] **CO6:** Deploy, monitor, and troubleshoot production-ready data pipelines in distributed or cloud-based environments. [Create]					

Course Content:

Module 1	**Fundamentals of Data Pipelines & ETL**	**15 Hours (T7+ L8)**
Data engineering fundamentals: ingestion, transformation, storage, Batch VS streaming pipelines, ETL concepts, architecture, and processing models, Data Lake & Data Warehouse differentiation, Data quality, governance, and metadata management.		
Module 2	**Workflow Orchestration & Scheduling**	**15 Hours (T7+ L8)**
Workflow orchestration fundamentals, Workflow Orchestration & Scheduling, Apache Airflow: DAG design, sensors, operators, Error handling, retries, triggers, Parameterising and scheduling production workflows, Pipeline monitoring basics		
Module 3	**Real-Time Pipelines & Messaging Systems**	**15 Hours (T7+ L8)**
Streaming pipelines overview, Message queues: Apache Kafka basics, Integrating Kafka with ETL frameworks Real-time transformations, Windowing, backpressure, offset management		

Module 4	Big Data Tools & Advanced Architectures	15 Hours (T7+ L8)

Hadoop ecosystem overview: HDFS, MapReduce, Hive, Spark for ETL tasks, Distributed ETL design patterns, Scalable data processing with Spark SQL / PySpark, Cloud ETL basics (AWS/GCP)

List of Laboratory Tasks:

Experiment 1. Basic ETL using Python
Level 1: Extract data from CSV/Excel files, Perform transformations (cleaning, null handling, type conversion)
Level 2: Load into MySQL/PostgreSQL database
Experiment 2. API Data Extraction Pipeline
Level 1: Extract JSON data from a public REST API
Level 2: Transform into structured tabular format
Level 3: Load into relational database
Experiment 3. Data Cleaning & Validation Framework
Level 1: Implement validation rules (missing values, duplicates, constraints)
Level 2: Generate data quality report, Log rejected records
Experiment 4. Incremental ETL Pipeline
Level 1: Extract only newly added/modified records
Level 2: Implement timestamp-based incremental loading, Compare full load vs incremental load
Experiment 5. Creating a Basic Airflow DAG
Level 1: Install and configure Apache Airflow
Level 2: Create a DAG with PythonOperator, Schedule daily ETL job
Experiment 6. Multi-Task DAG with Dependencies
Level 1: Create DAG with multiple tasks, Implement task dependencies
Level 2: Add retries and failure notifications
Experiment 7. Parameterized & Dynamic DAG
Level 1: Create dynamic tasks
Level 2: Pass parameters to DAG, Use Airflow Variables and XCom
Experiment 8. Monitoring and Logging
Level 1: Monitor DAG execution, Analyze logs
Level 2: Configure email alerts for failures
Experiment 9. Kafka Producer & Consumer
Level 1: Install Kafka
Level 2: Create topic, Build producer and consumer using Python
Experiment 10. Real-Time ETL with Kafka
Level 1: Stream live data (e.g., simulated transactions)
Level 2: Transform data in consumer, Store processed data in database
Experiment 11. Window-Based Stream Processing
Level 1: Implement time-based window aggregation
Level 2: Calculate rolling averages or counts, Store aggregated results
Experiment 12. HDFS Data Ingestion
Level 1: Upload large dataset into HDFS
Level 2: Perform Hive queries, Export results
Experiment 13. PySpark Batch ETL
Level 1: Load large dataset into Spark
Level 2: Perform transformations using Spark SQL, Save results in Parquet format
Experiment 14. Spark Streaming Pipeline
Level 1: Consume data from Kafka
Level 2: Process using Spark Structured Streaming, Store results in HDFS
Experiment 15. End-to-End Data Pipeline Project
Level 1: Students must design and implement a complete pipeline:

- Extract data from multiple sources (CSV/API/Database)
- Perform transformations
- Orchestrate workflow using Airflow
- Integrate Kafka for streaming

- Process using Spark
- Load into Data Warehouse
- Implement logging, monitoring & data validation
- Submit architecture diagram + report

Targeted Application & Tools that can be used:

- Business Intelligence & Data Warehousing
- Real-Time Analytics Systems
- Big Data Processing Systems
- Machine Learning Data Pipelines
- Cloud-Based Data Engineering
- Enterprise Data Integration

Assignment:

To design a complete ETL architecture for a real-world use case.
Problem Statement:

- E-commerce sales analytics
- Healthcare data integration

Text Book(s)

T1. "Fundamentals of Data Engineering" – Joe Reis & Matt Housley, 2022.
T2. "Designing Data-Intensive Applications" – Martin Kleppmann, Updated Edition (post-2020 reprint).
T3. "Streaming Systems: The What, Where, When, and How of Large-Scale Data Processing" – Tyler Akidau et al., Updated Edition (post-2020 reprint).

References

R1. "Data Pipelines Pocket Reference" – James Densmore, 2021.
R2. "Building Data Pipelines with Python" – James Densmore, 2021.
R3. "Spark: The Definitive Guide" – Bill Chambers & Matei Zaharia, Updated Edition (post-2020 reprint).

Resources

1. https://airflow.apache.org/docs/
2. https://kafka.apache.org/documentation/
3. https://spark.apache.org/docs/latest/

CSC8300- Mini Project (Application/Research Oriented)

Course Code: CSC8300	**Course Title: Mini Project(Application/Research Oriented)** **Type of Course: Project Work**	**L- T-P- C**	-	-	-	4
Version No.	**1.0**					
Course Pre-requisites	**Knowledge and Skills related to all the courses studied in previous semesters.**					
Anti-requisites	**NIL**					
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain **Employability Skills** through **Experiential Learning** techniques.					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1:** Apply problem identification techniques to select real-world computing problems addressing local, regional, or global needs. [Apply] **CO2:** Apply appropriate tools, technologies, and methodologies to design and develop solutions for the identified problem. [Apply] **CO3:** Analyze and design experiments or system architectures based on defined standards and specifications. [Analyze] **CO4:** Analyze and interpret experimental results to derive meaningful insights and conclusions. [Analyze] **CO5:** Evaluate project outcomes and communicate findings effectively through technical reports or scholarly publications. [Evaluate] **CO6:** Create innovative solutions by integrating research methodologies and modern technologies to address complex problems. [Create]					

COURSE BROAD SCHEDULE			
Topics	Mark		Rubrics

1. Domain, Technology, Project Scope and Planning	10	5%	Domain and technology (5) Scope and planning (5)
2. Importance and relevance of working technology in solving the problem	10	5%	Importance and relevance of working technology (8) and Identifying the applications (2)
3. Contributions to the project in solving real-world problems.	10	5%	Feature/characteristics/Applications selection and Constraints Identification (5), Analysis and Feature finalization subject to constraints and Design(5).
4. Application or usage of the project and impact on the society	20	10%	Application or usage of the project (10) and impact on the society (10) (SDG)
5. End Term Viva	100	50%	Technical Competency, Scope fulfilment, originality of problem solving, analysis and discussion of results, presentation content (delivery, clarity, supporting material quality), cost and impact analysis.
6. Project Report	10	5%	Compliance of report as per format (10)
7. *Supervisor	10	5%	Individual, Team work, discipline, Project Progression (10)
8. Publication/Patent	20	10%	Research Publication (15) and Patenting the project work (5)
9. Git Hub Repository	10	5%	Uploading the complete coding, Reports, publication proof, patent proof and other necessary documents in Git Hub(10)
Total	200	100%	** For S.No 1-4 , 5 marks should be allotted for communication and documentation skills*

SEMESTER IV
CSC8400- Major Project/Dissertation

Course Code: CSC8400	**Course Title: Major Project/Dissertation** **Type of Course: Project Work**	**L- T-P- C**	-	-	-	12
Version No.	**1.0**					
Course Pre-requisites	Knowledge and skills gained from all courses studied in previous semesters, including those applied in the mini-project.					
Anti-requisites	**NIL**					
Course Description	Students observe science and technology in action, develop an awareness of the method of scientific experimentation, and often get an opportunity to see, study and operate sophisticated and costly equipment. They also learn about the implementation of the principles of management they have learnt in class, when they observe multidisciplinary teams of experts from engineering, science, economics, operations research, and management deal with techno-economic problems at the micro and macro levels. Finally, it enables them to develop and refine their language, communication and inter-personal skills, both by its very nature, and by the various evaluation components, such as seminar, group discussion, project report preparation, etc. The broad-based core education, strong in mathematics and science and rich in analytical tools, provides the foundation necessary for the student to understand properly the nature of real-life problems. The students have options to pursue this course as either Project Work and Dissertation at the university, or Project Work in an Industry/ Company/ Research Laboratory, or Internship Program in an Industry/Company.					
Course Objectives	The objective of the course is to familiarize the learners with the concepts of Professional Practice and attain **Employability Skills** through **Experiential Learning** techniques.					
Course Outcomes	**On successful completion of this course the students shall be able to:** **CO1:** Understand and identify real-world computing problems addressing local, regional, national, or global needs. [Understand] **CO2:** Apply appropriate techniques and modern tools to design and develop solutions for the identified problem. [Apply] **CO3:** Apply experimental design principles based on standards and specifications for validating solutions. [Apply] **CO4:** Analyze experimental results and system performance to derive meaningful conclusions. [Analyze] **CO5:** Evaluate project outcomes and effectively communicate findings through technical reports and scholarly publications. [Evaluate] **CO6:** Create innovative and scalable solutions by integrating research methodologies and advanced technologies. [Create]					

COURSE BROAD SCHEDULE			
Topics	Mark		Rubrics
1. Domain, Technology, Project Scope and Planning	10	5%	Domain and technology (5) Scope and planning (5)

2. Importance and relevance of working technology in solving the problem	10	5%	Importance and relevance of working technology (8) and Identifying the applications (2)
3. Contributions to the project in solving real-world problems.	10	5%	Feature/characteristics/Applications selection and Constraints Identification (5), Analysis and Feature finalization subject to constraints and Design(5).
4. Application or usage of the project and impact on the society	20	10%	Application or usage of the project (10) and impact on the society (10) (SDG)
5. End Term Viva	100	50%	Technical Competency, Scope fulfilment, originality of problem solving, analysis and discussion of results, presentation content (delivery, clarity, supporting material quality), cost and impact analysis.
6. Project Report	10	5%	Compliance of report as per format (10)
7. *Supervisor	10	5%	Individual, Team work, discipline, Project Progression (10)
8. Publication/Patent	20	10%	Research Publication (15) and Patenting the project work (5)
9. Git Hub Repository	10	5%	Uploading the complete coding, Reports, publication proof, patent proof and other necessary documents in Git Hub(10)
Total	200	100%	** For S.No 1-4 , 5 marks should be allotted for communication and documentation skills*

ELECTIVE I AIML BASKET
CSC4701- Natural Language Processing

Course Code: CSC4701	**Course Title: Natural Language Processing** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive introduction to Natural Language Processing (NLP), combining theoretical foundations with hands-on programming. Learners will explore fundamental text preprocessing, word and text representation techniques, sequence labeling with probabilistic models, and practical NLP applications like sentiment analysis, machine translation, and text summarization. The course integrates machine learning concepts such as embeddings, RNNs, LSTMs, and attention mechanisms, preparing students to build real-world NLP systems using modern toolkits and language models.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply NLP preprocessing techniques such as tokenization, stemming, lemmatization, and text normalization to prepare textual data for analysis. [Apply] **CO2:** Analyze word and text representation techniques including Bag-of-Words, TF-IDF, Word2Vec, and N-gram models for extracting meaningful linguistic features. [Analyze] **CO3:** Apply machine learning and deep learning models such as Logistic Regression, Perceptron, RNN, LSTM, and CNN for solving NLP tasks. [Apply] **CO4:** Analyze sequence labeling and probabilistic models including Hidden Markov Models and Viterbi algorithms for structured prediction tasks. [Analyze] **CO5:** Evaluate different NLP techniques for tasks such as Named Entity Recognition, parsing, and language modeling. [Evaluate] **CO6:** Design and develop NLP-based applications such as sentiment analysis, machine translation, and text summarization for real-world problems. [Create]					
Course Content:						
Module 1	**Foundations of Natural Language Processing**					**12 Sessions**
Introduction to Natural Language Processing, terminologies, empirical rules, why NLP is hard, why NLP is useful, NLP Processing pipeline, Corpus Cleaning techniques – word tokenization, sentence tokenization, word frequency distribution, stemming, lemmatization, dictionary, Introduction to Part of Speech Tagging, Textual Pre-Processing techniques – Stop words removal, regular expression, lower case, text standardization.						
Module 2	**Word and Text Representations in NLP**					**13 Sessions**
Word relationships, Word Embeddings techniques- bag of words, TF-iDF, Word2Vec and optimization. Simple N-gram models. Estimating parameters and smoothing. Negative Sampling Evaluating language models. Logistic regression – Sigmoid and Softmax. Perceptron and backpropagation. RNN, LSTM, CNN. Attention. Pre-trained language models. Multilinguality.						

Module 3	**Sequence Labeling and Parsing Techniques**	**10 Sessions**
Sequence Labeling, Hidden Markov Models. Best Emission Probability, Best Forward Probability and Viterbi Decoding Algorithms. Analysis of Viterbi Algorithm. Named Entity Recognition. Constituency Parsing.		
Module 4	**Real-World Applications of NLP**	**10 Sessions**
Application of NLP. Lexical Resource Creation. Machine Translation. Sentiment Analysis. Lexical Simplification. Text Summarization.		

Project work/Assignment:

1. Text Classification App
2. Sentiment Analysis Dashboard
3. Chatbot using RNN or Transformers
4. Named Entity Recognition System
5. Machine Translation App
6. Text Summarizer
7. Resume Parser for HR Systems
8. Topic Modeling on Research Papers
9. Speech-to-Text Processing Pipeline
10. Grammar and Spell Checker

Topics relevant to the development of Employability:

Problem Solving: Text preprocessing, POS tagging, NER, sentiment analysis, and machine translation challenges.
Employability: Building NLP pipelines, deploying models, chatbot development, word embeddings, and real-world applications using Python libraries.

Textbook(s)

T1. Jacob Eisenstein, *Introduction to Natural Language Processing*, MIT Press, 2023.
T2. Steven Bird, Ewan Klein, Edward Loper, *Natural Language Processing with Python (Updated Edition)*, O'Reilly Media, 2021.
T3. Delip Rao, Brian McMahan, *Natural Language Processing with PyTorch*, O'Reilly Media, 2020.

References

R1. Denis Rothman, *Transformers for Natural Language Processing (2nd Edition)*, Packt Publishing, 2021.
R2. Thushan Ganegedara, *Natural Language Processing with TensorFlow (2nd Edition)*, Packt, 2021.
R3. Cole Howard, Hannes Hapke, *Building Machine Learning Pipelines*, O'Reilly Media, 2020. *(Relevant for NLP workflows & deployment)*

Web Resources

1. https://nlp.stanford.edu
2. https://huggingface.co/docs
3. https://ai.google/education

CSC4702- Computer Vision

Course Code: CSC4702	**Course Title: Computer Vision** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course explores the foundations and advancements in digital image processing and computer vision. Learners will gain hands-on experience in analyzing visual data, applying machine learning to vision tasks, and understanding geometric techniques for depth and motion analysis. Emphasis is placed on both theoretical underpinnings and practical implementations across real-world applications like face recognition, object tracking, and scene reconstruction.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply digital image processing techniques such as filtering, edge detection, feature extraction, and PCA for analyzing visual data. [Apply] **CO2:** Apply geometric concepts including image transformations, camera calibration, and stereo vision for 3D scene understanding. [Apply] **CO3:** Analyze machine learning techniques for image classification, object detection, and segmentation in vision-based applications. [Analyze] **CO4:** Evaluate advanced mid-level vision techniques such as stereopsis, structure from motion, and template-based recognition for solving complex vision problems. [Evaluate] **CO5:** Analyze feature extraction methods including SIFT, texture analysis, and edge-based representations for improving visual recognition performance. [Analyze] **CO6:** Evaluate real-world computer vision applications such as face recognition, object tracking, and pattern classification using appropriate models. [Evaluate]					
Course Content:						
Module 1	**Foundations of Digital Image Processing - CO1**					**11 Sessions**
Image Formation, Image Filtering, Edge Detection, Principal Component Analysis, Corner Detection SIFT, Applications: Large Scale Image Search. Correspondence and Pose consistency, finding templates using classifiers, Recognition by relations between templates, Applications - Pattern classification, Face Recognition.						
Module 2	**Geometric Approaches in Computer Vision – CO2**					**12 Sessions**
Image Transformations, Camera Projections, Camera Calibration, Depth from Stereo, Two View Structure from Motion, Object Tracking.						
Module 3	**Machine Learning Techniques for Vision Tasks – CO3**					**11 Sessions**
Introduction to Machine Learning, Image Classification, Object Detection, Semantic Segmentation, Linear filters, Edge detection, Filters and Features, Texture.						
Module 4	**Advanced Topics in Mid-Level Vision – CO4**					**11 Sessions**
The geometry of multiple views - Stereopsis, Affine structure from motion, Correspondence and Pose consistency, finding templates using classifiers, Recognition by relations between templates, Applications - Pattern classification, Face Recognition.						

Project work/Assignment:

1. Image Filtering and Feature Extraction
2. Geometric Transformation and Camera Calibration
3. Machine Learning for Image Classification
4. Project: Face Recognition System Development
5. Project: 3D Reconstruction from 2D Images
6. Project: Object Detection and Segmentation using Machine Learning

Topics related to the development of Employability:

1. Problem Solving: Develop and implement innovative image processing techniques to address challenges in segmentation, feature extraction, and pattern recognition in complex visual datasets.

2. Employability: Design and simulate a complete computer vision application such as a face recognition or object detection system that demonstrates practical, industry-relevant skills.

Textbook(s)

T1. Richard Szeliski, *Computer Vision: Algorithms and Applications (2nd Edition)*, Springer, 2022.

T2. Simon J.D. Prince, *Computer Vision: Models, Learning, and Inference (2nd Edition)*, Cambridge University Press, 2023.

T3. Ethem Alpaydin, *Introduction to Machine Learning (4th Edition)*, MIT Press, 2020. *(Relevant for vision-based ML techniques)*

References

R1. **Adrian Rosebrock**, *Deep Learning for Computer Vision with Python*, PyImageSearch, 2020.

R2. **Rajalingappaa Shanmugamani**, *Deep Learning for Computer Vision (2nd Edition)*, Packt Publishing, 2021.

R3. **Thushan Ganegedara**, *TensorFlow 2 Deep Learning Cookbook*, Packt Publishing, 2021.

Web Resources

1. https://docs.opencv.org
2. https://cs231n.stanford.edu
3. https://ocw.mit.edu

CSC4703- Generative AI

Course Code: CSC4703	**Course Title: Generative AI** **Type of Course: Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	-					
Anti-requisites	**NIL**					
Course Description	This course builds the foundational insight of understanding generative AI models and to explore various architectures, algorithms and practices of Gen AI skills to accelerate strategic decision making with data and deliver cutting-edge products faster with GenAI-augmented software development and leverage Gen AI tools to optimize workflows.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using Experiential LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply foundational concepts of generative models and large language models to understand their architecture and real-world applications. [Apply] **CO2:** Analyze transformer-based architectures and text generation models such as BERT and GPT for developing NLP-based generative applications. [Analyze] **CO3:** Apply LangChain components and retrieval-augmented techniques to build efficient LLM-based applications. [Apply] **CO4:** Analyze generative models such as GANs and VAEs for multimodal data generation including images, text, and audio. [Analyze] **CO5:** Evaluate the limitations of large language models and implement techniques such as prompt engineering and retrieval augmentation to improve performance. [Evaluate] **CO6:** Evaluate generative AI applications including text generation, image synthesis, and speech-to-text systems for practical deployment scenarios. [Evaluate]					
Course Content:						
Module 1	**Introduction to Generative AI-CO1**					**12 Sessions**
Introduction to Generative models: Historical perspective and evolution, Applications, Types of Generative models for different data modalities, Large Language Models (LLMs) – Introduction, evolution, Generative pre-trained transformers (GPT) and its variants, Google DeepMind's, PaLM2, LLaMa and its series of models by Meta AI, Claud and its variants by Anthropic, Prompt Engineering-basic prompting						
Module 2	**Text-based Generative models-CO2**					**11 Sessions**
Text-based Generative models: State-of-the Art models, RNN, LSTM, Transformer Architecture, Transformer based Generative models: BERT, GPT, Training and Fine tuning LLMs for Generative task, Open AI's Pre-trained transformers for Text Generation: ChatGPTs, Limitations of LLMs: Lack of context and Hallucination risks, Techniques to mitigate these limitations: chaining and retrieval augmentation, Workflow of an LLM application.						
Module 3	**Introduction to Lang Chain – CO3**					**11 Sessions**
Introduction to Lang chain: Types, Components, Information retrieval using agents and tools in Lang chain, Retrieval Augmented Language Models (RaLM): Understanding						

Retrieval and vectors: Embeddings, Vector storage, Vector indexing, Vector Libraries, Vector Databases, Chatbot using memory and conversation buffer		
Module 4	**Generative models for other Data modalities- CO4**	**11 Sessions**
Generative Adversarial Networks (GAN): GAN Architecture, GAN variants, Neural Style transfer with GAN, Training GANs and common challenges, GAN applications in image and text generation, Variational Auto Encoders (VAEs) and its variants, Image generation models: Dall-E, MidJourney and stable diffusion: Architecture and components of stable diffusion, Text-to-image Generation, Parameter tuning, Image-to-image generation, Training custom models, In-Painting: Exchanging classes, Multi-modal generative models using Whisper for Audio: Speech-to-Text generation.		
Targeted Application & Tools that can be used: Open AI Generative AI models: GPT 3.5 Turbo, GPT 4.0 vision model, Dall-E 3.0, Lang Chain Framework in Python, Python IDE, Stable Diffusion, Gemini, Hugging Face,.		
Textbook(s) T1. Generative AI with LangChain, Ist Edition by Ben Auffarth, Packt. Inc. ISBN: 978-1-83508-346-8, Decemeber 2023. T2. Generative Deep Learning,2nd Edition by David Foster, O'Reilly Media, Inc. ISBN: 9781098134181, May 2023. T3. Prompt Engineering for Generative AI, by James Phoenix, Mike Taylor, O'Reilly Media, Inc., ISBN:9781098153373, July 2024.		
References R1. Denis Rothman, *Transformers for Natural Language Processing (2nd Edition)*, Packt Publishing, 2021. R2. Sean Moriarity, *Generative AI Tools for Developers*, O'Reilly Media, 2023. R3. Joshua Newlan, *Building Generative AI Applications with Python*, Packt Publishing, 2024.		
Web Resources: 1. https://platform.openai.com/docs 2. https://huggingface.co/docs 3. https://python.langchain.com/docs		

CSC4704- Reinforcement Learning

Course Code: CSC4704	**Course Title: Reinforcement Learning** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	**Machine Learning**					
Anti-requisites	**NIL**					
Course Description	This course introduces the foundational concepts and practical applications of Reinforcement Learning (RL). Learners will explore core elements like agent-environment interactions, rewards, and policies through the lens of Markov Decision Processes (MDP). The course progresses through Monte Carlo methods, Temporal Difference learning (including SARSA and Q-Learning), and culminates with Multi-Armed Bandit (MAB) problems and an introduction to Deep Reinforcement Learning (DRL). Practical implementation using the OpenAI Gym environment is emphasized throughout to build hands-on skills in solving real-world sequential decision-making problems.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using PARTICIPATIVE LEARNING techniques**.**					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1**: Apply the fundamentals of reinforcement learning and Markov Decision Processes to model decision-making problems and compute optimal policies. [Apply] **CO2:** Apply Monte Carlo methods for prediction and control tasks in reinforcement learning environments. [Apply] **CO3:** Analyze temporal difference learning algorithms such as SARSA and Q-learning for policy optimization. [Analyze] **CO4:** Apply multi-armed bandit strategies and deep reinforcement learning techniques such as Deep Q Networks for solving sequential decision problems. [Apply] **CO5:** Analyze and compare reinforcement learning approaches including Dynamic Programming, Monte Carlo, and Temporal Difference methods. [Analyze] **CO6:** Evaluate reinforcement learning techniques for real-world applications such as recommendation systems and decision optimization. [Evaluate]					
Course Content:						
Module 1	**Foundations of Reinforcement Learning and MDPs**	**11 Sessions**				
Elements of RL, Agent, environment Interface, Goals and rewards, RL platforms, Applications of RL, Markov decision process (MDP), RL environment as a MDP, Maths essentials of RL, Policy and its types, episodic and continuous tasks, return and discount factor, fundamental functions of RL – value and Q functions, model-based and model-free learning, types of RL environments, Solving MDP using Bellman Equation, Algorithms for optimal policy using Dynamic Programming -Value iteration and policy iteration, Example : Frozen Lake problem, Limitations and Scope.						
Module 2	**Monte Carlo Methods for Prediction and Control**	**11 Sessions**				
Monte Carlo methods, prediction and control tasks, Monte Carlo prediction: algorithm, types of MC prediction, examples, incremental mean updates, Monte Carlo Control: algorithm, on-policy MC control, MC with epsilon-greedy policy, off-policy MC control. Limitations of MC method.						

Module 3	**Temporal Difference Learning Techniques**	**12 Sessions**
Temporal difference learning: TD Prediction, TD Control: On-policy TD control – SARSA, computing the optimal policy using SARSA, Off-policy TD control – Q learning, computing optimal policy using Q learning, Examples, Difference between SARSA and Q-learning, Comparison of DP, MC and TD methods.		
Module 4	**Multi-Armed Bandits and Introduction to Deep Reinforcement Learning**	**11 Sessions**
Understanding the MAB problem, Various exploration strategies – epsilon-greedy, softmax exploration, upper confidence bound and Thompson sampling, Applications of MAB - finding the best advertisement banner for a web site, Contextual bandits, introduction to Deep Reinforcement Learning (DRL) Algorithm – Deep Q Network (DQN).		
Project work/Assignment: Assignment on MDP Formulation. Comparison Report: SARSA vs. Q-Learning. Exploration Strategy Evaluation. Project: Frozen Lake Navigation using Policy Iteration Project: Train an Agent using Q-Learning to Play CartPole Project: Contextual Multi-Armed Bandit for Ad Recommendation		
Topics related to **1. Problem Solving:** Design a reinforcement learning-based strategy to solve the Frozen Lake navigation problem using Value Iteration and Policy Iteration. **2. Employability:** Simulate real-world decision-making environments using Multi-Armed Bandit algorithms to recommend optimal content or advertisements.		
Textbook(s) T1. Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction (2nd Edition)*, MIT Press, 2020. T2. Phil Winder, *Reinforcement Learning: Industrial Applications of Intelligent Agents*, O'Reilly Media, 2021. T3. Enes Bilgin, *Mastering Reinforcement Learning with Python*, Packt Publishing, 2021.		
References R1. Maxim Lapan, *Deep Reinforcement Learning Hands-On (2nd Edition)*, Packt Publishing, 2020. R2. Sudharsan Ravichandiran, *Deep Reinforcement Learning with Python*, Packt Publishing, 2020. R3. Ivan Gridin, *Deep Reinforcement Learning Algorithms with Python*, Packt Publishing, 2021.		
Web Resources 1. https://spinningup.openai.com 2. https://www.deepmind.com/learning-resources 3. https://gymnasium.farama.org		

CSC4705- Explainable & Responsible

Course Code: CSC4705	**Course Title: Explainable & Responsible AI** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	This course offers a solid foundation in transfer learning and its importance in deep learning. Students learn to build and evaluate CNN models and use pre-trained networks for image classification. The course covers text classification using word embeddings like Word2Vec and GloVe. Learners work with real-world datasets for images and text analysis. The course also explores image colorization using transfer learning techniques.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using PARTICIPATIVE LEARNING techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply transfer learning techniques and pre-trained deep learning models for solving domain-specific problems. [Apply] **CO2:** Analyze deep learning models and datasets for image recognition and classification tasks using transfer learning approaches. [Analyze] **CO3:** Apply text representation and deep learning techniques for document classification and sentiment analysis tasks. [Apply] **CO4:** Analyze deep neural network architectures for image colorization and related computer vision tasks. [Analyze] **CO5:** Evaluate the performance of deep learning and transfer learning models using appropriate metrics and validation techniques. [Evaluate] **CO6:** Evaluate the effectiveness and challenges of applying AI models responsibly across image and text-based applications. [Evaluate]					
Course Content:						
Module 1	**Transfer Learning Fundamentals - CO1**					**10 Sessions**
Introduction – Transfer learning strategies – Transfer learning and deep learning – Deep transfer learning types – Challenges of transfer learning – Need for transfer learning – Building CNN models – Leveraging transfer learning with pre-trained CNN models – Evaluating deep learning models						
Module 2	**Image Recognition and Classification – CO2**					**10 Sessions**
Deep learning based image classification – Benchmarking datasets – State-of-the-art deep image classification models – Image classification and transfer learning – Working with CIFAR-10 – Working with dog breed identification dataset.						
Module 3	**Text Document Categorization – CO3**					**10 Sessions**
Traditional text categorization – Shortcomings of BoW model – Benchmark datasets – Word2vec model – Using gensim – GloVe model – Building a review sentiment classifier – Working with IMDB dataset with word embeddings – Creating document summaries using CNN model – Multiclass classification with CNN model – Visualizing document embeddings.						
Module 4	**Image Colorization: – CO4**					**15 Sessions**
Problem statement – Color images – Building a coloring deep neural network – Preprocessing – Loss function – Encoder –Transfer learning – Feature extraction – Fusion layer – Decoder – Post-processing – Training and results – Challenges.						
Project work/Assignment: 1. Implement transfer learning with pre-trained CNN model.						

2. Implement transfer learning for image classification with CIFAR-10 dataset.
3. Build review sentiment classifier using transfer learning.
4. Create document summaries using transfer learning.

Topics related to

1. Problem Solving: Choose efficient algorithmic or system-level solutions for real-world challenges such as disease classification and traffic sign classification.

2. Employability: Simulation of industry-relevant environments such as real-time medical dataset scenarios to enhance practical skills and job readiness.

Textbook(s)

T1. Denis Rothman, *Artificial Intelligence By Example (2nd Edition)*, Packt Publishing, 2020.

T2. Markus Hofmann, Ralf Klinkenberg, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer, 2021.

T3. Sinan Ozdemir, *Mastering Machine Learning with Python in Six Steps (2nd Edition)*, Packt Publishing, 2021.

References

R1. Thushan Ganegedara, *Deep Learning with TensorFlow 2 (2nd Edition)*, Packt Publishing, 2021.

R2. Adrian Rosebrock, *Deep Learning for Computer Vision with Python*, PyImageSearch, 2020.

R3. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd Edition)*, O'Reilly Media, 2022.

Web References

1. https://christophm.github.io/interpretable-ml-book/
2. https://developers.google.com/machine-learning/crash-course
3. https://captum.ai/

ELECTIVE II DATA SCIENCE & ANALYTICS BASKET

CSC4706- Time Series Analysis and Forecasting

Course Code: CSC4706	**Course Title: Time Series Analysis and Forecasting** **Type of Course: Discipline Elective**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Probability and Statistics, Linear Algebra, Basics of Programming					
Anti-requisites	**NIL**					
Course Description	This course introduces the principles and techniques of time series analysis with a focus on modeling and forecasting real-world temporal data. Students will learn classical and modern approaches for analyzing time-dependent data, including statistical models and forecasting methods widely used in economics, finance, engineering, and data science. Emphasis is placed on understanding data patterns, building forecasting models, and evaluating model performance.					
Course Objective	The objective of this course is to equip learners with the knowledge and analytical skills required to model, analyze, and forecast time-dependent data using appropriate statistical and computational techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply time series analysis techniques to identify components such as trend, seasonality, and autocorrelation in temporal data. [Apply] **CO2:** Apply classical time series models including moving averages, exponential smoothing, and autoregressive models for data modelling. [Apply] **CO3:** Analyze ARIMA and seasonal models for forecasting and handling non-stationary time series data. [Analyze] **CO4:** Apply advanced forecasting techniques such as state space models and regression-based approaches for time series prediction. [Apply] **CO5:** Analyze volatility models such as ARCH and GARCH for modelling financial time series data. [Analyze] **CO6:** Evaluate forecasting models using accuracy measures and diagnostic techniques for real-world applications. [Evaluate]					
Course Content						
Module 1	**Introduction to Time Series Analysis**	**12 Sessions**				
Introduction to Time Series Data, Components of Time Series (Trend, Seasonality, Cyclic, Irregular), Time Series Decomposition, Exploratory Data Analysis, Stationarity, Autocorrelation and Partial Autocorrelation, White Noise Process.						
Module 2	**Classical Time Series Models**	**13 Sessions**				
Moving Average Models, Exponential Smoothing Techniques (Simple, Holt, Holt-Winters), Autoregressive (AR) Models, Moving Average (MA) Models, ARMA Models, Model Identification using ACF and PACF.						
Module 3	**ARIMA and Seasonal Models**	**10 Sessions**				

Stationarity and Differencing, ARIMA Models, Seasonal ARIMA (SARIMA), Model Estimation and Diagnostics, Forecasting using ARIMA Models, Handling Missing Values and Outliers.		
Module 4	**Advanced Forecasting Techniques and Applications**	**10 Sessions**
State Space Models, Kalman Filter (Introduction), Time Series Regression Models, Volatility Models (ARCH, GARCH), Introduction to Machine Learning for Time Series Forecasting, Model Evaluation and Forecast Accuracy Measures.		
Targeted Application & Tools that can be used: Python (NumPy, Pandas, Statsmodels, Scikit-learn), R (forecast package), MS Excel, MATLAB		
Case Study/Assignment: Assignments and case studies mapped to **CO1–CO4**, including: • Time series decomposition and visualization • Forecasting using ARIMA and exponential smoothing • Comparative analysis of forecasting models • Real-world datasets (finance, weather, sales, energy demand)		
Text Book(s) T1. Rob J Hyndman, George Athanasopoulos, *Forecasting: Principles and Practice (3rd Edition)*, OTexts, 2021. T2. Galit Shmueli, Kenneth C. Lichtendahl Jr., *Practical Time Series Forecasting with R (2nd Edition)*, Axelrod Schnall Publishers, 2020. T3. Francesco Corea, *Applied Time Series Analysis and Forecasting*, Springer, 2021.		
References R1. Terence C. Mills, *Applied Time Series Analysis: A Practical Guide to Modeling and Forecasting*, Academic Press, 2020. R2. Ivan Svetunkov, *Forecasting and Analytics with ADAM*, Springer, 2021. R3. Jason Brownlee, *Deep Learning for Time Series Forecasting*, Machine Learning Mastery, 2020.		
Web Resources 1. https://otexts.com/fpp3/ 2. https://www.statsmodels.org 3. https://machinelearningmastery.com/time-series-forecasting/		

CSC4707-: Predictive Analytics & Decision Models

Course Code: CSC4707	**Course Title: Predictive Analytics & Decision Models** **Type of Course: Discipline Elective**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Prerequisites	Probability and Statistics, Machine Learning, Deep Learning					
Anti-requisites	**NIL**					
Course Description	This course provides an in-depth understanding of predictive analytics and decision models used for data-driven decision making. It introduces statistical, machine learning, and optimization techniques for building predictive models from structured and time-series data. Students learn methods for model evaluation, validation, and interpretation to support reliable predictions. The course also covers decision analysis under uncertainty, risk modelling, and optimization-based decision frameworks. Emphasis is placed on practical implementation using modern analytical tools and real-world datasets. By integrating theory with hands-on practice, the course prepares students to solve complex decision problems across various application domains.					
Course Objective	The objectives of this course are to: 1. Provide a strong foundation in the concepts and techniques of predictive analytics and decision modelling. 2. Enable students to develop and evaluate predictive models using statistical and machine learning methods. 3. Introduce time-series forecasting and optimization techniques for decision support under uncertainty. 4. Develop practical skills in implementing predictive and decision models using modern analytical tools. 5. Enhance students' ability to interpret model results and apply them to real-world decision-making problems.					
Course Outcomes	**After successful completion of the course, the student will be able to:** **CO1:** Explain the fundamental concepts of predictive analytics and decision modeling in data-driven environments. [Understand] **CO2:** Apply statistical and machine learning techniques to build predictive models for regression and classification problems. [Apply] **CO3:** Analyze and evaluate predictive models using appropriate performance metrics and validation techniques. [Analyze] **CO4:** Develop time-series forecasting models to analyze trends, seasonality, and temporal patterns in data. [Analyze] **CO5:** Formulate and solve decision-making problems using optimization and decision analysis techniques under uncertainty. [Evaluate] **CO6:** Interpret predictive and decision model outputs and communicate insights effectively for real-world applications. [Create]					
Course Content						
Module 1	**Foundations of Predictive Analytics**	**9 Sessions**				
Introduction to Predictive Analytics: concepts, lifecycle, applications. Data Types, Data Quality, and Preprocessing: Handling missing values, outliers, transformations. Exploratory Data Analysis (EDA): Descriptive stats, visualizations, correlation analysis. Basic Probability & Statistics for Prediction: Bayes' Theorem, distributions, hypothesis						

testing. Introduction to Decision Making Under Uncertainty: Decision Trees, decision tables.		
Module 2	**Regression & Classification Models**	**9 Sessions**
Linear Regression: Simple & multiple regression, assumptions, diagnostics. Logistic Regression: Binary outcomes, odds & log-odds, metrics. Model Evaluation & Validation: Train/test split, cross-validation, ROC/AUC. k-Nearest Neighbors (KNN). Naive Bayes Classification.		
Module 3	**Advanced Machine Learning Models**	**9 Sessions**
Decision Trees & Random Forests. Gradient Boosting Machines: XGBoost, LightGBM, CatBoost (comparisons). Support Vector Machines (SVM). Ensemble Methods: Bagging, boosting, stacking. Model Interpretability: SHAP, LIME, feature importance.		
Module 4	**Time Series & Forecasting Models**	**9 Sessions**
Time Series Fundamentals: Trends, seasonality, stationarity. Classical Models: AR, MA, ARMA, ARIMA, SARIMA. Exponential Smoothing Methods: Holt–Winters, ETS. State-Space Models. Deep Learning for Time Series: LSTM/GRU basics.		
Module 5	**Decision Models & Optimization Techniques**	**9 Sessions**
Decision Analysis: Decision trees (advanced), utility theory. Risk Analysis: Scenario analysis, Monte Carlo simulation. Linear & Integer Optimization: Objective functions, constraints, simplex method. Multi-criteria Decision Making (MCDM): AHP, TOPSIS, PROMETHEE. Reinforcement Learning for Decision Making: Markov decision processes, basic RL concepts.		
Case Study/Assignment: Assignments and case studies mapped to **CO1–CO6**, including: • Analyze a real-life scenario (e.g., healthcare, retail, finance) and explain how predictive analytics supports decision-making processes. • Develop regression and classification models using a given dataset and apply appropriate machine learning techniques. • Compare multiple predictive models using evaluation metrics such as accuracy, precision, recall, ROC-AUC, and RMSE. • Build and analyze a forecasting model for sales, demand, or stock price data and interpret trends and seasonality. • Formulate a decision problem using optimization techniques or decision trees and evaluate outcomes under uncertainty. • Develop a complete predictive and decision model, interpret results, and communicate insights through a report and presentation.		
Textbook(s) T1. **Eric Siegel**, *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die (Revised Edition)*, Wiley, 2020. T2. **John D. Kelleher, Brian Mac Namee, Aoife D'Arcy**, *Fundamentals of Machine Learning for Predictive Data Analytics (2nd Edition)*, MIT Press, 2020. T3. **Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel**, *Data Mining for Business Analytics: Concepts, Techniques, and Applications*, Wiley, 2021.		
References R1. **Aurélien Géron**, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd Edition)*, O'Reilly Media, 2022. R2. **Jason Brownlee**, *Machine Learning Mastery with Python*, Machine Learning Mastery, 2021. R3. **Francesco Corea**, *Machine Learning for Business: Decision Making and Predictive Analytics*, Springer, 2020.		

Web Resources

1. https://www.kaggle.com/learn
2. https://scikit-learn.org/stable/
3. https://developers.google.com/machine-learning

CSC4708- Social Network Analysis

Course Code: CSC4708	**Course Title: Social Network Analysis Type of Course: Discipline Elective**	**L- T-P- C**	3	0	0	3
Version No.	1					
Course Pre-requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	Networks are a fundamental tool for modeling complex social, technological, and biological systems. Coupled with the emergence of online social networks and large-scale data availability in social sciences, this course focuses on the analysis of massive networks which provide many computational, algorithmic, and modeling challenges. The course will cover research on the structure and analysis of such large networks and on models and algorithms that abstract their basic properties. We will explore how to practically analyze large-scale network data and how to reason about it through models for network structure and evolution. Topics covered in this course are how information spreads through society; robustness and fragility of networks; algorithms for the World Wide Web; prediction and recommendation in online social networks; representation learning for large networks; etc.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Information Retrieval and Recommendation Systems and attain SKILL DEVELOPMENT through Participative Learning techniques					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply fundamental concepts of social network analysis and graph theory to model and visualize networked data. [Apply] **CO2:** Analyze network growth models and structural properties such as centrality, assortativity, and clustering in real-world networks. [Analyze] **CO3:** Apply link analysis algorithms such as PageRank and similarity-based methods for ranking and prediction tasks in networks. [Apply] **CO4:** Analyze link prediction techniques using heuristic, probabilistic, and machine learning-based approaches. [Analyze] **CO5:** Apply anomaly detection techniques in static and dynamic networks using machine learning pipelines. [Apply] **CO6:** Evaluate representation learning methods and their effectiveness in network analysis applications and case studies. [Evaluate]					
Course Content:						
Module 1	**Introduction**					**10 Sessions**
Introduction, Applications of SNA, Preliminaries, Three levels of SNA, Historical Development, Graph Visualization Tools						
Module 2	**Network Growth Models**					**10 Sessions**
Network Basics, Node Centrality, Assortativity, Transitivity and Reciprocity, Similarity, Degeneracy, Properties of Real-World Networks, Random Network Model, Ring Lattice Network Model, Watts-Strogratz Model, Preferential Attachment Model, Price's Model, Local-World NNetwork Growth Model, Network Model with Accelerating Growth						
Module 3	**Link Analysis and Prediction**					**13 Sessions**
Applications of Link Analysis, Signed Networks, Strong and Weak Ties, Link Analysis and Algorithms, Page Rank, Personalized Page Rank, DivRank, SImRank, PathSIm, Temporal Changes in Network, Problem Definition, Evaluating Link Prediction Methods, Heuristic Models, Probabilistic Models, Supervised Random Walk, Information-theoretic Model, Latest Trends in Link Prediction						

Module 4	**Anomaly Detection**	**12 Sessions**
Outliers vs. Network-based Anomalies, Challenges, Anomaly Detection in Static Networks and Dynamic Networks, Machine Learning Pipelines, Intuition behind Representation Learning, Benefits of Representation Learning, Criterion of GRL, GRL Pipeline, Representation of Learning Models, Applications and Case Studies		
Assignment: Group assignment, Quiz		
Text Book(s) T1. Ryan A. Rossi, Nesreen K. Ahmed, The Network Data Mining Handbook, Cambridge University Press, 2020. T2. Charu C. Aggarwal, Machine Learning for Graphs, Springer, 2021. T3. Derek Hansen, Ben Shneiderman, Marc A. Smith, Analyzing Social Media Networks with NodeXL (2nd Edition), Morgan Kaufmann, 2020.		
References R1. Albert-László Barabási, The Formula: The Universal Laws of Success, Little, Brown and Company, 2020. (Relevant to network theory concepts) R2. Stanley Wasserman, Katherine Faust, Social Network Analysis: Methods and Applications (Updated Edition), Cambridge University Press, 2021. R3. Matthew O. Jackson, The Human Network: How Your Social Position Determines Your Power, Beliefs, and Behaviors, Pantheon, 2020.		
Web References 1. https://snap.stanford.edu 2. https://networkx.org 3. https://gephi.org		

CSC4709- Geospatial Analytics

Course Code: CSC4709	**Course Name: Geospatial Analytics** **Type of Course: Discipline Elective**	**L- T-P- C**	3	0	0	3
Version No.	1					
Course Pre-requisites	Statistics and Probability, Linear Algebra, Fundamentals of Data Analytics, Basic Database Management Systems, Programming in Python / R, Introductory knowledge of Machine Learning					
Anti-requisites	**NIL**					
Course Description	This course introduces the principles and techniques of Geospatial Analytics, focusing on the acquisition, management, analysis, and visualization of spatial data. It integrates GIS concepts, spatial statistics, remote sensing data, and machine learning to extract actionable insights from geospatial datasets. The course emphasizes real-world applications such as urban planning, environmental monitoring, transportation, public health, and location-based services, equipping data science students with spatial thinking and analytical skills.					
Course Objective	• Introduce core concepts of geospatial data models and spatial thinking. • Develop skills in spatial data processing, visualization, and querying. • Apply spatial statistics and machine learning techniques to geospatial problems. • Analyze real-world geospatial datasets for decision-making and predictive modeling.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain geospatial data types, coordinate reference systems, and spatial data models used in geospatial analytics. [Understand] **CO2**: Acquire, preprocess, and manage spatial datasets from multiple sources using GIS tools and spatial databases. [Apply] **CO3:** Visualize and interpret geospatial data using cartographic principles and interactive visualization techniques. [Apply] **CO4:** Apply spatial statistical methods to analyse spatial patterns, relationships, and dependencies in geospatial data. [Apply] **CO5:** Implement machine learning techniques for geospatial data analysis and predictive modelling. [Apply] **CO6:** Design, evaluate, and communicate geospatial analytics solutions for real-world applications across diverse domains. [Create]					
Course Content:						
Module 1	**Introduction to Geospatial Data and GIS**					**12 Sessions**
Topics: Overview of geospatial analytics, Spatial thinking and geographic information systems (GIS), Types of geospatial data: vector and raster, Coordinate reference systems and map projections, Spatial data formats and standards, Data sources: GPS, remote sensing, open geospatial data, Introduction to GIS software and spatial databases						
Module 2	**Spatial Data Management and Visualization**					**10 Sessions**
Spatial databases and indexing, Geospatial data preprocessing and cleaning, Spatial queries and spatial joins, hematic mapping and cartographic principles, Interactive geospatial visualization, Geospatial data handling using Python/R libraries, Web-based geospatial visualization concepts						
Module 3	**Spatial Analytics and Modeling**					**12 Sessions**

Exploratory spatial data analysis (ESDA), Spatial autocorrelation and spatial heterogeneity, Point pattern analysis, Spatial interpolation techniques, Network and proximity analysis, Spatial regression models, Introduction to geospatial machine learning		
Module 4	**Advanced Applications and Case Studies**	**11 Sessions**
Topics: Remote sensing data analytics, Urban and regional planning applications, Environmental and climate analytics, Transportation and mobility analytics, public health and epidemiological mapping, Location-based services and business analytics, Case studies using real-world geospatial datasets		
Text Book(s) T1. Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind, *Geographic Information Systems and Science (4th Edition)*, Wiley, 2021. T2. Joel Lawhead, *Learning Geospatial Analysis with Python (4th Edition)*, Packt Publishing, 2020. T3. Serge Rey, Dani Arribas-Bel, *Geographic Data Science with Python*, CRC Press, 2022.		
References R1. Margaret L. L. Lee, Martin J. Wong, *Geospatial Data Science Techniques and Applications*, Springer, 2021. R2. Rafael Moreno-Sanchez, Christopher J. Anderson, *Spatial Analysis Methods and Practice*, Esri Press, 2020. R3. Fang Qiu, Benjamin Sridharan, *Geographic Information Systems for Geospatial Intelligence*, Springer, 2020.		
Web Resources 1. https://www.qgis.org 2.https://geopandas.org 3. https://developers.google.com/earth-engine		

CSC4710- Bioinformatics

Course Code: CSC4710	**Course Title: Bioinformatics** **Type of Course: Discipline Elective**	**L- T-P- C**	3	0	0	3
Version No.	2.0					
Course Pre- requisites	**NIL**					
Anti-requisites	**NIL**					
Course Description	This course is designed to provide the knowledge of the concepts related to bioinformatics. The course is aimed at understanding the DNA and Protein sequences and databases. It also deals with Pairwise comparison and calculating the scoring matrix. Further, it focuses on Sequence Alignment techniques, discovering the Motifs in the sequence. Students will also learn the overview of Structural Bioinformatics and Genome sequencing.					
Course Objective	The objective of the course is to familiarize the learners with the concepts of Introduction to Bioinformatics and attain Employability through Participative Learning techniques.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply fundamental concepts of molecular biology and bioinformatics to analyze biological sequences and structures. [Apply] **CO2:** Apply bioinformatics tools and databases for retrieval, storage, and management of genomic and proteomic data. [Apply] **CO3:** Analyze sequence similarity techniques and alignment methods using tools such as BLAST and PSI-BLAST. [Analyze] **CO4:** Apply sequence alignment algorithms such as Needleman-Wunsch and Smith-Waterman for pairwise and multiple sequence analysis. [Apply] **CO5:** Analyze motif discovery techniques and gene prediction models such as HMM and Bayesian approaches. [Analyze] **CO6:** Evaluate bioinformatics methods for DNA sequence analysis and their applications in genomics and proteomics. [Evaluate]					
Course Content:						
Module 1	**Fundamentals of Bioinformatics**					**15 Sessions**
Introduction to Bioinformatics: Introduction to molecular biology, Cell, DNA, RNA, Transcription,Translation, Folding, Gene Structure, Introduction to Bioinformatics, Components and fields of bioinformatics, Omics, basic principles of structural/functional analysis of biological molecules, Biological Data Acquisition, Types of DNA sequences, Genomic DNA ,Mitochondrial DNA,DNA Sequencing tools, Protein sequencing and structure determination methods, Finding Reverse complement of a sequence.						
Module 2	**Genome databases and Sequence Similarity**					**15 Sessions**
Types and classification of genome databases, DNA sequence retrieval system, various DNA and protein sequence file formats, Common sequence file formats; Files for multiple sequence alignment; Files for structural data, Frequent words and k-mers in Text, String Reconstruction problem, Sequence Similarity searching, Sequence Similarity searching tools, NCBI BLAST, PSI BLAST, Significance of sequence alignments, Alignment scores and gap penalties.						
Module 3	**DNA sequence analysis and applications**					**15 Sessions**

Sequence similarity searches and alignment tools, Finding alignment using Needleman Wunsch and Smith- Waterman algorithm, Heuristic Methods of sequence alignment, Pair-wise and multiple sequence alignments, DNA sequence analysis, Motif in protein sequence ,Motif discovery using Gibbs sampling,Motif finding, Gene Prediction models: Hidden Markov model(HMM), Generalized Hidden Markov model(GHMM), Bayesian method.
Targeted Application & Tools that can be used: BLAST, FastA, , ClustalW, MEGA
Project work/Assignment: Each batch of students (self-selected batch mates – up to 4 in a batch) will be allocated case studies/assignments
Textbook(s) T1. Arthur M. Lesk, *Introduction to Bioinformatics (5th Edition)*, Oxford University Press, 2023. T2. Pavel Pevzner, Phillip Compeau, *Bioinformatics Algorithms: An Active Learning Approach (3rd Edition)*, Active Learning Publishers, 2021. T3. Ramsden Jeremy J., *Bioinformatics: An Introduction (2nd Edition)*, Springer, 2020.
References R1. Xiong Jin, *Essential Bioinformatics*, Cambridge University Press, 2020. R2. Claus O. Wilke, *Fundamentals of Data Visualization (Applied to Bioinformatics)*, O'Reilly Media, 2020. R3. Dan E. Krane, Michael L. Raymer, *Fundamental Concepts of Bioinformatics*, Pearson, 2021.
Web References 1. https://www.ncbi.nlm.nih.gov 2. https://blast.ncbi.nlm.nih.gov 3. https://www.ebi.ac.uk

ELECTIVE III CLOUD & DISTRIBUTED SYSTEMS
CSC4711- Cloud DevOps

Course Code: CSC4711	**Course Title: Cloud DevOps** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Basic knowledge of programming (preferably Python) and familiarity with software development concepts. Fundamental understanding of operating systems, computer networks, and basic cloud computing concepts.					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive understanding of Cloud DevOps practices for designing, deploying, and managing scalable applications in cloud environments. It covers cloud computing architecture, CI/CD pipelines, containerization, orchestration, infrastructure as code, and monitoring techniques. The course also introduces MLOps concepts to enable automated deployment and lifecycle management of data science models in modern cloud platforms.					
Course Objective	To equip students with skills to deploy, automate, and monitor scalable data science applications using Cloud DevOps techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply cloud computing concepts including service models, deployment models, and core cloud services to design scalable cloud architectures. [Apply] **CO2:** Apply DevOps principles and CI/CD practices using version control systems and automation tools for efficient software delivery. [Apply] **CO3:** Apply containerization and orchestration techniques using Docker, Kubernetes, and infrastructure automation tools. [Apply] **CO4:** Analyze cloud-based MLOps workflows including model deployment, monitoring, and lifecycle management. [Analyze] **CO5:** Analyze infrastructure as code and configuration management tools for automating cloud resource provisioning. [Analyze] **CO6:** Evaluate cloud DevOps practices for security, scalability, and reliability in real-world applications. [Evaluate]					
Course Content:						
Module 1	**cloud Computing and Architecture**					**11 Sessions**
Fundamentals of Cloud Computing – Characteristics and Evolution – Service Models (IaaS, PaaS, SaaS) – Deployment Models (Public, Private, Hybrid, Multi-Cloud) – Virtualization and Containers – Cloud Architecture Components – Compute, Storage and Networking Services – Identity and Access Management (IAM) – Overview of AWS, Azure and GCP core services						
Module 2	**Devops And Ci/Cd In Cloud**					**11 Sessions**
DevOps Principles and Lifecycle – Agile vs DevOps – Continuous Integration, Continuous Delivery and Continuous Deployment – Version Control using Git and GitHub – Branching Strategies – Build Automation – CI/CD tools (Jenkins, GitHub Actions, Azure DevOps) – Cloud-native CI/CD services (AWS CodePipeline, CodeBuild) – Introduction to GitOps.						
Module 3	**Containerization, Orchestration And Automation**					**12 Sessions**
Containerization Concepts – Docker Architecture – Dockerfile and Image Creation – Docker Compose – Containerizing Python and Data Science applications – Kubernetes Architecture – Pods, Services, Deployments – Scaling and Rolling Updates – Managed Kubernetes services (EKS, AKS, GKE) – Infrastructure as Code concepts – Terraform basics – Configuration Management using Ansible.						

Module 4	**CLOUD DEVOPS FOR DATA SCIENCE (MLOPS)**	**11 Sessions**
Machine Learning Lifecycle in Cloud – ML Pipelines – Model Versioning and Reproducibility – Deployment of ML models using Flask/FastAPI – Containerized Model Deployment – CI/CD for ML workflows – Monitoring and Logging (Prometheus, Grafana, CloudWatch) – Model Drift Detection – Security and Best Practices in Cloud DevOps.		
Project work/Assignment: 1. Cloud Service Models and Architecture Study 2. Dockerizing a Python Application 3. CI/CD Workflow Design 4. Project: Deployment of a Machine Learning Model using Docker and Cloud VM 5. Project: Implementation of CI/CD Pipeline for a Data Science Application 6. Project: Cloud-Based Hosting of a Containerized Web Application		
Textbook(s) T1. Kief Morris, *Infrastructure as Code (2nd Edition)*, O'Reilly Media, 2020. T2. Mikael Krief, *Learning DevOps (2nd Edition)*, Packt Publishing, 2021. T3. John Arundel, Justin Domingus, *Cloud Native DevOps with Kubernetes*, O'Reilly Media, 2022.		
References R1. Mark Richards, Neal Ford, *Fundamentals of Software Architecture (2nd Edition)*, O'Reilly Media, 2024. R2. Josh Arundel, *GitOps and Kubernetes*, O'Reilly Media, 2023. R3. Noah Gift, Alfredo Deza, *Practical MLOps*, O'Reilly Media, 2021.		
Web Resources 1. https://docs.aws.amazon.com 2. https://kubernetes.io/docs 3. https://learn.microsoft.com/azure/devops		

CSC4712- Blockchain Technology

Course Code: CSC4712	**Course Name: Blockchain Technology Type of Course: Theory Course**	**L- T-P- C**	3	0	0	3
Version No.	1					
Course Pre-requisites	Fundamentals of Computer Networks, Data Structures and Algorithms, Discrete Mathematics and Probability, Introductory knowledge of Cryptography (hashing, encryption concepts), Programming experience in Python / Java / JavaScript					
Anti-requisites	**NIL**					
Course Description	This course introduces the fundamental concepts, architectures, and technologies underlying blockchain systems and distributed ledgers. It explores consensus mechanisms, cryptographic primitives, and fault-tolerant distributed computing that enable secure, decentralized applications. Students will study prominent blockchain platforms such as Bitcoin and Ethereum, smart contracts, and emerging privacy-preserving techniques. The course also examines security vulnerabilities, scalability challenges, and real-world applications of blockchain across financial services, supply chains, and governance systems, with a strong analytical perspective suited for data science graduates.					
Course Objective	• Provide a solid understanding of distributed consensus and fault tolerance in adversarial environments, explain the cryptographic foundations that enable secure blockchain protocols. • Analyze the design and working of Bitcoin and Ethereum blockchains. • Develop the ability to design and implement smart contracts. • Examine privacy, security threats, and scalability challenges in blockchain systems. • Evaluate real-world blockchain use cases using data-driven and analytical reasoning.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Explain the principles of distributed record keeping, Byzantine fault tolerance, and consensus mechanisms. [Understand] **CO2:** Apply cryptographic primitives such as hash functions, digital signatures, and zero-knowledge proofs in blockchain contexts. [Apply] **CO3:** Analyse Bitcoin's design, consensus models, and scripting mechanisms. [Analyse] **CO4:** Design, develop, and verify smart contracts using Ethereum and Solidity. [Apply] **CO5:** Evaluate privacy-preserving blockchain technologies and identify security vulnerabilities and attacks. [Apply] **CO6:** Assess blockchain applications across domains such as finance, supply chain management, and government services. [Apply]					
Course Content:						
Module 1	**Introduction to Distributed Systems**					**12 Sessions**
Distributed Systems, need for distributed record keeping , modeling faults and adversaries, byzantine generals problem, consensus algorithms and their scalability problems, nakamoto's concept with blockchain based cryptocurrency, technologies borrowed in blockchain, hash pointers , consensus , byzantine fault, tolerant distributed computing digital cash etc.						
Module 2	**Basic Distributed Computing and Crypto primitives**					**10 Sessions**

Atomic broadcast, consensus, byzantine models of fault tolerance, hash functions, puzzle friendly hash, collison resistant hash, digital signatures, public key crypto, verifiable random functions, zero – knowledge systems.		
Module 3	**Basics of Bitcoin & Ethereum**	**12 Sessions**
Bitcoin blockchain challenges and solutions, proof of work, proof of stake, alternatives to bitcoin consensus, bitcoin scripting language and their use. Ethereum and smart contracts, the turing completeness of smart contract languages and verification challenges, using smart contracts to enforce legal contracts – comparing bitcoin scripting vs. Ethereum smart contracts, writing smart contracts using solidity & javascript.		
Module 4	**Privacy – Security issues in Blockchain**	**11 Sessions**
Pseudo – anonymity vs. Anonymity, zcash and zk – snarks for anonymity preservation, attacks on blockchains: sybil attacks , selfish mining, 51% attacks advent of algorand; sharding based consensus algorithms to prevent these attacks. Case studies: block chain in financial service, healthcare, supply chain management and government services.		
Text Book(s) T1. Arshdeep Bahga, Vijay Madisetti, *Blockchain Applications: A Hands-On Approach (2nd Edition)*, VPT, 2020. T2. Imran Bashir, *Mastering Blockchain (3rd Edition)*, Packt Publishing, 2020. T3. Andreas M. Antonopoulos, Gavin Wood, *Mastering Ethereum*, O'Reilly Media, 2021.		
References R1. Joseph Bonneau et al., *Bitcoin and Cryptocurrency Technologies*, Princeton University Press, 2021. R2. Roger Wattenhofer, *The Science of the Blockchain*, CreateSpace, 2020. R3. Narayan Prusty, *Building Blockchain Projects (2nd Edition)*, Packt Publishing, 2021.		
Web Resources 1. https://bitcoin.org/bitcoin.pdf 2. https://ethereum.org/en/developers/docs 3. https://soliditylang.org/docs		

CSC4713- IoT Analytics

Course Code: CSC4713	**Course Title: IoT Analytics** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites						
Anti-requisites	**NIL**					
Course Description	This course introduces students to the concepts, tools, and techniques required to analyze data generated by Internet of Things (IoT) systems. The course covers IoT architecture, data acquisition, storage technologies, real-time analytics, machine learning methods, and visualization techniques. Students will gain hands-on knowledge to design analytics solutions for IoT applications such as smart cities, healthcare, and industrial automation.					
Course Objective	The objective of the course is EMPLOYABILITY of student by using various IoT analytical techniques.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply IoT fundamentals, architectures, and communication technologies to design data-driven IoT systems. [Apply] **CO2:** Apply IoT data acquisition, preprocessing, and storage techniques using edge, fog, and cloud-based platforms. [Apply] **CO3:** Analyze IoT data characteristics and big data technologies such as Hadoop, Kafka, and Spark for handling large-scale data. [Analyze] **CO4:** Apply machine learning techniques for IoT analytics including classification, clustering, and anomaly detection. [Apply] **CO5:** Analyze real-time stream processing techniques and visualization tools for monitoring IoT data. [Analyze] **CO6:** Evaluate IoT analytics solutions considering security, privacy, and real-world applications such as smart cities and healthcare. [Evaluate]					
Course Content:						
Module 1	**IoT Fundamentals and Data Characteristics**					**12 Sessions**
Introduction to IoT:Overview and evolution of IoT, architecture and components, Sensors, actuators, and embedded systems, IoT ecosystem and enabling technologies. IoT Communication Technologies: IoT communication protocols, Short-range communication (Bluetooth, ZigBee, RFID, NFC),Long-range communication (LoRaWAN, NB-IoT, Cellular IoT).IoT Data Characteristics, Types of IoT data, Streaming vs batch data, Big data challenges in IoT,3Vs of IoT data (Volume, Velocity, Variety), IoT Platforms Overview: Cloud-based IoT platforms, Open-source IoT frameworks.						
Module 2	**IoT Data Acquisition and Storage Technologies**					**13 Sessions**
IoT Data Acquisition: Sensor data collection methods, MQTT protocol and REST APIs, Edge and Fog computing concepts. IoT Data Pre-processing: Data cleaning and noise removal, Handling missing data, Time-series data preparation. IoT Data Storage: Relational vs NoSQL databases, Time-series databases, Distributed data storage concepts. Big Data Technologies for IoT: Hadoop ecosystem overview,Apache Kafka basics, Apache Spark fundamentals.						
Module 3	**IoT Data Analytics Techniques**					**10 Sessions**
Types of IoT Analytics: Descriptive analytics, Predictive analytics, Prescriptive analytics. Machine Learning for IoT: Supervised learning techniques, Unsupervised learning and clustering, Anomaly detection in sensor data, Predictive maintenance concepts. Real-						

Time Stream Analytics: Stream processing concepts, Spark Streaming basics, Event-driven data processing.

Module 4	**IoT Visualization, Security, and Applications**	**10 Sessions**

IoT Data Visualization: Dashboard design principles, Visualization tools (Grafana, Power BI, Tableau), Real-time monitoring systems. IoT Security and Privacy: Security challenges in IoT, Authentication and encryption techniques, Privacy issues and data protection. IoT Analytics Applications: Smart cities, Industrial IoT, Healthcare IoT, Smart agriculture.

Project work/Assignment:

1. Smart Air Quality Monitoring and Prediction System
2. Predictive Maintenance for Industrial Equipment using Sensor Data
3. Smart Energy Consumption Analytics using IoT Smart Meter Data
4. IoT-Based Healthcare Monitoring and Anomaly Detection System
5. Smart Traffic Flow Analysis and Congestion Prediction
6. Smart Agriculture Analytics for Irrigation and Crop Monitoring
7. IoT-Based Smart Home Behavior and Energy Optimization Analytics
8. Real-Time IoT Sensor Anomaly Detection using Stream Data
9. Cold Chain Monitoring and Spoilage Prediction System
10. Smart Waste Management and Route Optimization Analytics

Topics relevant to the development of Employability:

ProblemSolving:

IoT data acquisition, sensor data preprocessing, time-series analysis, anomaly detection, real-time stream analytics, predictive maintenance, and IoT data visualization challenges.

Employability:

Building IoT analytics pipelines, real-time data processing, machine learning model deployment, dashboard development, edge analytics implementation, and developing industry-oriented IoT solutions using Python, cloud platforms, and big data tools.

Textbook(s):

T1. Pethuru Raj, Anupama C. Raman, *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, CRC Press, 2020.
T2. Vijay Madisetti, Arshdeep Bahga, *Internet of Things: A Hands-On Approach (2nd Edition)*, VPT, 2021.
T3. Kai Hwang, Min Chen, *Big Data Analytics for IoT*, Wiley, 2021.

References

R1. Rajkumar Buyya, Amir Vahid Dastjerdi, *Internet of Things: Principles and Paradigms*, Elsevier, 2020.
R2. David Hanes, Gonzalo Salgueiro, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases*, Cisco Press, 2020.
R3. Neha Sharma, Nitin Kumar, *IoT Analytics: Data Processing and Applications*, Springer, 2022.

Web Resources

1. https://iot.eclipse.org
2. https://kafka.apache.org
3. https://spark.apache.org

CSC4714- Edge & Fog Computing

Course Code: CSC4714	**Course Title: Edge & Fog Computing** **Type of Course: Discipline Elective**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	• Fundamentals of computer networks – OSI and TCP/IP models, routing, and switching • Basics of distributed systems and cloud computing concepts • Operating systems concepts – processes, threads, and resource management • Programming fundamentals using C / Python / Java • Basic knowledge of virtualization and container concepts (desirable)					
Anti-requisites	Students who have already completed or are currently registered for courses that substantially overlap in content with Edge Computing, Fog Computing, Advanced Cloud Computing, or IoT Computing Systems shall not be permitted to enroll in this course.					
Course Description	This course introduces the principles, architectures, and technologies of Edge and Fog Computing for supporting latency-sensitive, distributed, and real-time applications. The course emphasizes application and analysis of edge–fog architectures, enabling technologies, data processing mechanisms, resource management strategies, and security models. Students will gain practical exposure to industry-relevant platforms and use cases such as IoT systems, smart infrastructure, and Industry 4.0 applications, enabling them to design and analyse efficient edge–fog computing solutions.					
Course Objective	To enable students to apply and analyse edge and fog computing principles, architectures, platforms, and real-world solutions to support latency-sensitive, bandwidth-efficient, and context-aware applications..					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply the concepts of edge, fog, and cloud computing to design distributed computing architectures for real-time applications. [Apply] **CO2:** Apply enabling technologies such as containerization, virtualization, and orchestration frameworks for edge computing environments. [Apply] **CO3:** Analyze data processing techniques including stream analytics, task offloading, and resource management in edge–fog systems. [Analyze] **CO4:** Apply resource scheduling, load balancing, and QoS techniques to optimize performance in distributed environments. [Apply] **CO5:** Analyze security mechanisms including authentication, authorization, and data privacy in edge and fog computing systems. [Analyze] **CO6:** Evaluate real-world applications and emerging trends such as federated learning, TinyML, and 5G-enabled edge computing. [Evaluate]					
Course Content						
Module 1	**Fundamentals of Edge & Fog Computing**					**12 Sessions**
Evolution of distributed computing – cloud to fog to edge, Motivation for edge and fog computing – latency, bandwidth, scalability, Edge vs fog vs cloud computing – architectural comparison, Layered architecture of edge–fog systems, Edge nodes, fog nodes, and cloud coordination, Communication models and protocols – MQTT, CoAP, HTTP, gRPC, Industry use cases and deployment scenarios						
Module 2	**Enabling Technologies and Platforms**					**11 Sessions**

Virtualization concepts for edge environments, Containerization and microservices at the edge, Edge orchestration frameworks – Kubernetes, K3s, Docker Swarm, Software Defined Networking (SDN) for edge and fog, Network Function Virtualization (NFV), Edge middleware and APIs, Data representation and messaging formats – JSON, CBOR, Protocol Buffers

Module 3	**Data Processing and Resource Management**	**11 Sessions**

Edge data acquisition and preprocessing, Stream processing and real-time analytics, Task offloading strategies between edge, fog, and cloud, Resource discovery and monitoring in fog environments, Scheduling and load balancing techniques, Quality of Service (QoS) and Service Level Agreements (SLA), Performance evaluation and optimization techniques

Module 4	**Security, Applications, and Emerging Trends**	**11 Sessions**

Security challenges in edge and fog computing, Authentication, authorization, and access control mechanisms, Data privacy and trust management, Secure communication protocols for edge devices, Edge and fog computing applications – smart cities, healthcare, Industry 4.0, autonomous systems, Integration of AI and machine learning at the edge, Emerging trends – federated learning, TinyML, 5G-enabled edge computing

Targeted Application:

- Internet of Things (IoT) based real-time monitoring systems
- Smart city infrastructure – traffic control, surveillance, and utilities
- Industry 4.0 and smart manufacturing systems
- Healthcare monitoring and remote diagnostics
- Autonomous and connected vehicles
- Augmented Reality (AR) and Virtual Reality (VR) applications
- Edge-enabled video analytics and computer vision systems

Tools / Platforms

- Edge computing platforms – AWS IoT Greengrass, Azure IoT Edge
- Containerization tools – Docker, Podman
- Orchestration frameworks – Kubernetes, K3s
- Programming languages – Python, Java, Go
- Messaging and communication – MQTT, CoAP, REST APIs
- Data processing frameworks – Apache Kafka, Apache Edgent
- Monitoring and security tools – Prometheus, Grafana, OpenSSL

Case Study/Assignment: Assignments and case studies mapped to CO1–CO6, including:

CO1 – Architecture Analysis

- Comparative case study on Cloud vs Fog vs Edge architectures for a latency-sensitive application
- Analysis of architectural layers and workload placement for a given industry scenario

CO2 – Enabling Technologies

- Assignment on containerizing an edge application using Docker
- Case study on orchestration of edge services using Kubernetes/K3s

CO3 – Edge Data Processing

- Design and implementation of real-time data processing pipelines at the edge
- Case study on stream analytics for IoT sensor data

CO4 – Resource Management

- Assignment on task offloading and scheduling strategies in fog environments
- Analysis of resource utilization and QoS trade-offs in edge–fog systems

CO5 – Security and Privacy

- Case study on security threats and mitigation techniques in edge–fog computing
- Assignment on secure communication and access control mechanisms

CO6 – End-to-End Application Development

- Mini-project to design and deploy an edge–fog based application for smart city / Industry 4.0 / healthcare use cases

- Case study evaluating performance, scalability, and security of the developed solution

Textbook(S)

T1. Zaigham Mahmood, *Fog Computing: Concepts, Frameworks and Applications*, Springer, 2020.
T2. Rajkumar Buyya, Satish Narayana Srirama, *Fog and Edge Computing: Principles and Paradigms*, Wiley, 2020.
T3. Kashif Naseer Qureshi, Anand Paul, *Edge Computing: A Primer*, Springer, 2022.

References

R1. Flavio Bonomi et al., *Fog Computing and Its Role in the Internet of Things*, Cisco Press, 2020.
R2. Sudeep Tanwar, *Edge Computing: From Hype to Reality*, Wiley, 2021.
R3. Kailash Chand Sharma, Divya Gupta, *Edge AI: Convergence of Edge Computing and Artificial Intelligence*, Springer, 2021.

Web references

1. https://www.openfogconsortium.org
2. https://kubernetes.io
3. https://www.ietf.org

CSC4715 – Cyber Security for Data Platforms

Course Code: CSC4715	**Course Title: Cyber Security for Data Platforms** **Type of Course: Discipline Elective**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	• Fundamentals of computer networks • Basics of operating systems • Introduction to databases and SQL • Programming fundamentals (any high-level language) • Basic understanding of cloud computing concepts					
Anti-requisites	• Students who have completed a foundational course in Cyber Security • Students who have completed Information Security or Network Security courses covering similar content • Students who have completed advanced courses in Cloud Security or Database Security					
Course Description	This course focuses on applied security mechanisms, threat modelling, and defence strategies for data platforms including databases, data lakes/warehouses, and cloud data services. Students will learn to apply security tools, detect threats, configure secure environments, and enforce compliance using real-world data technologies.					
Course Objective	To enable students to apply cyber security principles, tools, and techniques for protecting modern data platforms by configuring secure access controls, implementing encryption mechanisms, monitoring security events, and responding to data-centric cyber threats in enterprise and cloud environments.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply cybersecurity fundamentals including CIA triad and data lifecycle security to protect data across storage and processing environments. [Apply] **CO2:** Apply identity and access management techniques such as authentication, authorization models, and cloud IAM for securing data platforms. [Apply] **CO3:** Analyze cryptographic techniques and key management practices for securing data at rest and in transit. [Analyze] **CO4:** Apply encryption mechanisms and secure configurations to safeguard databases, data lakes, and warehouses. [Apply] **CO5:** Analyze monitoring and logging techniques including SIEM integration for detecting and responding to security incidents. [Analyze] **CO6:** Evaluate incident response strategies and security best practices for handling data breaches and ensuring compliance. [Evaluate]					
Course Content						
Module 1	**Foundations of Data Platform Security**					**12 Sessions**
Cyber security fundamentals for data platforms= CIA triad in data storage and processing environments, Data lifecycle security – data at rest – data in transit – data in use, Threat landscape for databases – data lakes – data warehouses, Secure configuration baselines for data platforms, Vulnerability assessment techniques for data systems						
Module 2	**Identity and Access Management for Data Platforms**					**11 Sessions**
Authentication mechanisms for data platforms= Authorization models – RBAC – ABAC – DAC, Principle of least privilege and segregation of duties, Federated identity management – LDAP – SAML – OAuth- Cloud IAM for data platforms – AWS – Azure – GCP, Privileged access management and misuse prevention						

Module 3	**Data Encryption and Key Management**	**11 Sessions**
Cryptography basics for data security, Symmetric and asymmetric encryption techniques, Encryption for data at rest – TDE – disk-level encryption, Encryption for data in transit – SSL – TLS, Key management lifecycle – generation – storage – rotation – revocation, Hardware and cloud-based key management services		
Module 4	**Monitoring – Logging and Incident Response**	**11 Sessions**
Security logging for databases and data platforms, Audit trails and compliance logging, Centralized log management – ELK – Splunk, SIEM integration for data platforms, Security incident detection and analysis, Incident response procedures for data breach scenarios		

Targeted Application:

- Securing enterprise databases and data warehouses
- Protection of cloud-based data platforms and analytics services
- Implementation of access control and encryption in data platforms
- Monitoring and incident response for data breach scenarios
- Compliance enforcement in data-driven organizations
- Security operations for big data and data engineering environments

Tools / Platforms

- Database platforms – MySQL – PostgreSQL – MongoDB
- Cloud data platforms – AWS Redshift – Azure Synapse – Google BigQuery
- Identity and access management tools – AWS IAM – Azure Active Directory
- Encryption and key management services – AWS KMS – Azure Key Vault
- Monitoring and logging tools – ELK Stack – Splunk
- Security information and event management – SIEM platforms
- Vulnerability assessment tools – OpenVAS – Nessus

Case Study/Assignment: Assignments and case studies mapped to CO1–CO6, including:

- Secure configuration of a data platform by applying baseline security controls *(CO1)*
- Threat modeling of a database or cloud data platform to identify vulnerabilities and attack surfaces *(CO2)*
- Design and implementation of identity and access management policies using role-based and least-privilege principles *(CO3)*
- Log analysis and anomaly detection using audit logs from data platforms *(CO4)*
- Implementation of encryption mechanisms for data at rest and data in transit with key management practices *(CO5)*
- Incident response case study involving detection, analysis, and mitigation of a simulated data breach *(CO6)*

Text book(s)

T1. Mark Rhodes-Ousley, *Information Security: The Complete Reference (3rd Edition)*, McGraw Hill, 2020.
T2. Nina Godbole, Sunit Belapure, *Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives (2nd Edition)*, Wiley, 2021.
T3. William Stallings, Lawrie Brown, *Computer Security: Principles and Practice (4th Edition)*, Pearson, 2022.
Textbook references

Reference Books

R1. **Charlie Kaufman, Radia Perlman, Mike Speciner**, *Network Security: Private Communication in a Public World (3rd Edition)*, Pearson, 2020.
R2. **Jason Andress**, *The Basics of Information Security (3rd Edition)*, Syngress, 2021.
R3. **Chris Dotson**, *Practical Cloud Security (2nd Edition)*, O'Reilly Media, 2022.

Web References

1. https://owasp.org
2. https://csrc.nist.gov
3. https://www.splunk.com

ELECTIVE BASKET–IV: DECISION SCIENCES AND APPLIED ANALYTICS
CSC4716 - Decision Science and Optimization Techniques

Course Code: CSC4716	**Course Title: Decision Science and Optimization Techniques** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Basic knowledge of Linear Algebra, Calculus, and Probability Theory. Familiarity with programming (preferably Python/R) and fundamental data analysis concepts.					
Anti-requisites	NIL					
Course Description	This course introduces the principles of decision science and mathematical optimization techniques used for data-driven decision making. It covers linear and non-linear programming, network models, decision analysis under uncertainty, and simulation methods. The course emphasizes practical applications using real-world data analytics problems.					
Course Objective	To understand the fundamental concepts of decision science and mathematical modeling, and apply optimization techniques to solve real-world decision-making problems.					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply decision science concepts and linear programming techniques to formulate and solve optimization problems in business and analytics. [Apply] **CO2:** Apply advanced optimization techniques including integer programming, non-linear programming, and convex optimization for complex decision problems. [Apply] **CO3:** Analyze network optimization and dynamic models such as shortest path, maximum flow, and Markov decision processes for real-world applications. [Analyze] **CO4:** Apply decision analysis techniques including decision trees, Bayesian methods, and simulation models for decision-making under uncertainty. [Apply] **CO5:** Analyze optimization methods such as simplex, duality, and sensitivity analysis for evaluating solution robustness. [Analyze] **CO6:** Evaluate decision models and optimization techniques for solving real-world problems in logistics, resource allocation, and data-driven systems. [Evaluate]					
Course Content:						
Module 1	**Introduction To Decision Science And Linear Programming**					**11 Sessions**
Decision Science Concepts – Role of Analytics in Decision Making – Mathematical Modeling – Formulation of Linear Programming Problems – Graphical Method – Simplex Method – Duality in Linear Programming – Sensitivity Analysis – Applications in Business and Data Analytics.						
Module 2	**Advanced Optimization**					**11 Sessions**
Integer Programming – Binary and Mixed Integer Programming – Branch and Bound Method – Non-Linear Programming – Convex Optimization – KKT Conditions (basic concepts) – Multi-objective Optimization – Applications in Resource Allocation and Machine Learning.						
Module 3	**Network and Dynamic Models**					**11 Sessions**

Network Optimization Models – Shortest Path Problem – Minimum Spanning Tree – Maximum Flow Model – Transportation and Assignment Problems – Dynamic Programming – Markov Decision Processes – Applications in Logistics and Data-driven Systems.

Module 4	**Decision Analysis And Simulation**	**12 Sessions**

Decision Making under Uncertainty – Decision Trees – Bayesian Decision Theory – Utility Theory – Game Theory Basics – Monte Carlo Simulation – Risk Analysis – Simulation using Python – Case Studies in Data Science Applications.

Project work/Assignment:

1. Formulate and solve a Linear Programming problem for profit maximization using Python.
2. Solve a Transportation or Assignment problem and interpret results.
3. Construct and analyze a Decision Tree for a real-world business decision problem.
4. Formulate and solve an Integer Programming problem for resource allocation using Python.
5. Develop and analyze a Shortest Path or Minimum Spanning Tree model for a logistics network
6. Perform a Monte Carlo Simulation to evaluate risk in an investment or business decision problem.

Textbook(s)

T1. Hamdy A. Taha, *Operations Research: An Introduction (10th Edition)*, Pearson, 2021.
T2. Stephen G. Powell, Kenneth R. Baker, *Management Science: The Art of Modeling with Spreadsheets (5th Edition)*, Wiley, 2020.
T3. Dimitris Bertsimas, John N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific (Reprint/Updated Edition), 2021.

References

R1. David G. Luenberger, Yinyu Ye, *Linear and Nonlinear Programming (4th Edition)*, Springer, 2021.
R2. Sheldon M. Ross, *Introduction to Probability Models (12th Edition)*, Academic Press, 2020.
R3. S. R. S. Varadhan, *Stochastic Processes and Decision Models*, Springer, 2021.

Web Resources

1. https://www.solver.com
2. https://ocw.mit.edu
3. https://www.analyticsvidhya.com

CSC4717- Operations Research and Supply Chain Analytics

Course Code: CSC4717	**Course Title: Operations Research and Supply Chain Analytics** **Type of Course: Discipline Elective**	**L-T- P- C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Probability and Statistics / Discrete Mathematics / Basic Optimization					
Anti-requisites	**NIL**					
Course Description	This course introduces students to **Operations Research (OR)** techniques and their applications in **Supply Chain Analytics**. The course focuses on mathematical modeling, optimization methods, and analytical tools used for effective decision-making in logistics, inventory management, transportation, scheduling, and network optimization. Students learn to formulate real-world supply chain problems as quantitative models and solve them using linear programming, network models, inventory theory, and project management techniques. The course emphasizes analytical thinking, optimization-based decision-making, and practical problem-solving skills relevant to modern supply chain systems.					
Course Objective	The objectives of this course are to: • Provide strong foundations in Operations Research methodologies • Enable students to model and solve supply chain optimization problems • Develop analytical skills for data-driven decision-making • Apply OR techniques to logistics, inventory, transportation, and scheduling • Enhance employability through problem-solving and optimization skills					
Course Outcomes	**On successful completion of this course, the students shall be able to:** **CO1:** Apply operations research techniques and linear programming methods to formulate and solve optimization problems in business and supply chain contexts. [Apply] **CO2:** Apply transportation and assignment models to optimize logistics and distribution decisions. [Apply] **CO3:** Analyze inventory models and cost structures to manage stock levels and improve supply chain efficiency. [Analyze] **CO4:** Apply network optimization techniques such as shortest path, minimum spanning tree, and maximum flow in supply chain design. [Apply] **CO5:** Analyze project scheduling techniques including PERT and CPM for efficient project management. [Analyze] **CO6:** Evaluate decision-making models under certainty, risk, and uncertainty for strategic supply chain planning. [Evaluate]					
Course Content:						
Module 1	**Foundations of Operations Research and Linear Programming**					**12 Sessions**
Introduction to Operations Research: evolution, scope, applications, phases of OR study, characteristics and limitations, OR models. Linear Programming Problems (LPP): formulation, assumptions, standard and canonical forms. Graphical solution of LPP (two-variable problems).						

Simplex method, slack, surplus and artificial variables. Sensitivity analysis and managerial interpretation of solutions.		
Module 2	**Transportation and Assignment Models**	**11 Sessions**
Transportation problem: formulation, balanced and unbalanced cases. Initial basic feasible solutions: North-West Corner Rule, Least Cost Method, Vogel's Approximation Method. Optimality test using MODI method. Assignment problem: formulation and Hungarian method. Travelling Salesman Problem (TSP) and applications in logistics and distribution.		
Module 3	**Inventory Models and Supply Chain Network Analytics**	**11 Sessions**
Inventory management concepts and cost structures. Deterministic inventory models: EOQ, EPQ. Inventory models with shortages and quantity discounts. Supply chain network design. Shortest path, minimum spanning tree, and maximum flow models with supply chain applications.		
Module 4	**Project Scheduling and Decision Analytics**	**11 Sessions**
Project management concepts in supply chains. PERT and CPM: network construction, critical path, floats. Project completion time estimation and cost analysis. Time–cost trade-off and project crashing. Decision theory: decision-making under certainty, risk, and uncertainty.		

Project work/Assignment:

1. Linear Programming using Simplex Method
2. Transportation and Assignment problem case studies
3. EOQ and inventory optimization problems
4. Supply chain network optimization models
5. Project scheduling using PERT/CPM

Topics relevant to the development of Employability:

Problem Solving: Formulation and solution of optimization problems in supply chain systems using linear programming, transportation, inventory, and network models. **Employability:** Applying Operations Research techniques for supply chain planning, logistics optimization, inventory control, and decision-making using analytical tools and software.

Textbook(s)

T1. Hamdy A. Taha, *Operations Research: An Introduction*, Pearson Education, 10th Edition, 2022.

T2. Hillier F. S. & Lieberman G. J., *Introduction to Operations Research*, McGraw-Hill, 11th Edition, 2022.

T3. J. K. Sharma, *Operations Research: Theory and Applications*, 6th Edition, 2020.

References

R1. Frederick S. Hillier, Gerald J. Lieberman, *Introduction to Operations Research (11th Edition)*, McGraw-Hill, 2021.

R2. David Simchi-Levi, Philip Kaminsky, *Designing and Managing the Supply Chain (4th Edition)*, McGraw-Hill, 2020.

R3. Winston Wayne L., *Operations Research: Applications and Algorithms (Updated Edition)*, Cengage Learning, 2020.

Web Resources

1. https://ocw.mit.edu
2. https://www.solver.com
3. https://www.informs.org

CSC4718- Financial Analytics and Risk Modelling

Course Code: CSC4718	**Course Title: Financial Analytics and Risk Modelling** **Type of Course: Discipline Elective**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Basic Probability and Statistics, Fundamentals of Finance					
Anti-requisites	**NIL**					
Course Description	This course provides an in-depth understanding of financial analytics techniques and quantitative risk modelling used in modern financial decision-making. It introduces students to financial data analysis, risk measurement, portfolio theory, asset pricing models, and predictive analytics using statistical and computational methods. The course emphasizes analytical thinking and real-world financial problem solving, preparing students for careers in finance, analytics, banking, and risk management.					
Course Objective	The objective of this course is to familiarize learners with financial analytics concepts and risk modelling techniques, enabling them to analyze financial data, quantify risk, and support informed financial decisions using analytical tools and models.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply financial analytics concepts including time value of money, risk, and return for analyzing financial data. [Apply] **CO2:** Apply statistical and econometric techniques such as regression, correlation, and hypothesis testing for financial modeling. [Apply] **CO3:** Analyze portfolio management techniques including diversification, CAPM, and APT for investment decision-making. [Analyze] **CO4:** Apply risk measurement techniques such as Value at Risk (VaR) and Conditional VaR for assessing financial risk. [Apply] **CO5:** Analyze financial risk models including market, credit, and operational risk using simulation and forecasting techniques. [Analyze] **CO6:** Evaluate financial decision-making strategies using derivatives, hedging techniques, and predictive analytics models. [Evaluate]					
Course Content						
Module 1	**Introduction to Financial Analytics**					**12 Sessions**
Introduction to Financial Analytics, Role of Analytics in Finance, Financial Markets and Instruments, Types of Financial Data, Time Value of Money, Returns and Risk, Descriptive Statistics for Financial Data, Introduction to Financial Statements and Ratio Analysis.						
Module 2	**Statistical Analysis and Financial Modelling**					**11 Sessions**
Probability Distributions in Finance, Correlation and Covariance Analysis, Regression Analysis in Finance, Forecasting Financial Time Series, Introduction to Econometric Models, Hypothesis Testing in Financial Decision Making, Volatility Modelling.						
Module 3	**Portfolio Theory and Risk Measurement**					**11 Sessions**
Modern Portfolio Theory, Risk–Return Trade-off, Portfolio Diversification, Capital Asset Pricing Model (CAPM), Arbitrage Pricing Theory (APT), Risk Measures: Variance, Standard Deviation, Beta, Value at Risk (VaR), Conditional VaR.						

Module 4	**Risk Modelling and Financial Decision Analytics**	**11 Sessions**
Market Risk, Credit Risk, Operational Risk, Risk Modelling Techniques, Stress Testing and Scenario Analysis, Monte Carlo Simulation, Financial Derivatives and Risk Hedging, Introduction to Algorithmic and Predictive Financial Analytics.		
Targeted Application & Tools that can be used: MS Excel, Python (NumPy, Pandas, Matplotlib), R, MATLAB, Financial Databases (Yahoo Finance, Quandl)		
Case Study/Assignment: Assignments and case studies mapped to **CO1-CO4**, including: • Portfolio optimization problems • Risk assessment using real financial datasets • VaR calculation and stress testing • Financial forecasting using historical data		
Text Book(s) T1. John C. Hull, *Risk Management and Financial Institutions (5th Edition)*, Wiley, 2021. T2. Frank J. Fabozzi, Sergio M. Focardi, Petter N. Kolm, *Financial Modeling and Valuation: A Practical Guide*, Wiley, 2021. T3. Ruppert David, *Statistics and Data Analysis for Financial Engineering (2nd Edition)*, Springer, 2020.		
References R1. Paul Wilmott, *Paul Wilmott Introduces Quantitative Finance (2nd Edition)*, Wiley, 2020. R2. Philippe Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk (3rd Edition)*, McGraw-Hill, 2021. R3. Ernest P. Chan, *Machine Trading: Deploying Computer Algorithms to Conquer the Markets*, Wiley, 2021.		
Web Resources 1. https://www.investopedia.com 2. https://fred.stlouisfed.org 3. https://www.cfainstitute.org		

CSC4719- Applied Business Analytics

Course Code: CSC4719	**Course Title: Applied Business Analytics** **Type of Course: Discipline Elective**	**L-T-P-C**	3	0	0	3
Version No.	1.0					
Course Pre-requisites	Basic Statistics, Fundamentals of Management, Basics of Programming					
Anti-requisites	**NIL**					
Course Description	This course focuses on the application of analytical techniques to solve real-world business problems. It equips students with the ability to analyze business data, generate actionable insights, and support strategic decision-making using data-driven approaches. The course covers descriptive, predictive, and prescriptive analytics with practical business use cases across marketing, finance, operations, and human resources.					
Course Objective	The objective of this course is to enable learners to apply business analytics tools and techniques for data-driven decision-making and to solve complex business problems using analytical thinking.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply business analytics concepts and data preprocessing techniques to derive insights from business data. [Apply] **CO2:** Analyze descriptive and diagnostic analytics techniques including EDA, visualization, and correlation for business decision-making. [Analyze] **CO3:** Apply predictive analytics models such as regression and classification for solving business problems like demand forecasting and churn prediction. [Apply] **CO4:** Analyze prescriptive analytics techniques including optimization and simulation for supporting business decisions. [Analyze] **CO5:** Apply business intelligence tools and dashboard design principles to communicate insights effectively. [Apply] **CO6:** Evaluate data-driven strategies, ethical considerations, and governance practices in business analytics. [Evaluate]					
Course Content						
Module 1	**Introduction to Business Analytics**					**12 Sessions**
Overview of Business Analytics, Types of Analytics (Descriptive, Diagnostic, Predictive, Prescriptive), Business Data Sources, Data Collection and Cleaning, Data Visualization for Business Insights, Key Performance Indicators (KPIs), Introduction to Business Intelligence.						
Module 2	**Descriptive and Diagnostic Analytics**					**13 Sessions**
Exploratory Data Analysis, Summary Statistics, Data Visualization Techniques, Correlation Analysis, Root Cause Analysis, Dashboard Design, Business Reporting, Case Studies in Marketing and Operations Analytics.						
Module 3	**Predictive Analytics for Business**					**10 Sessions**
Regression Analysis, Classification Techniques, Time Series Forecasting (Overview), Customer Analytics, Demand Forecasting, Churn Prediction, Model Evaluation Metrics, Business Interpretation of Predictive Models.						
Module 4	**Prescriptive Analytics and Decision Support**					**10 Sessions**

Optimization Techniques, Decision Trees for Business Decisions, What-if Analysis, Simulation Models, Risk and Sensitivity Analysis, Introduction to Data-Driven Strategy, Ethics and Governance in Business Analytics.

Targeted Application & Tools that can be used: MS Excel (Advanced), Power BI / Tableau, Python (Pandas, Scikit-learn), R, SQL

Case Study/Assignment: Assignments and case studies mapped to **CO1–CO4**, including:

- Sales and marketing performance analysis
- Customer segmentation and churn analysis
- Demand forecasting and inventory optimization
- Business dashboards and executive reports

Text Book(s)

T1. Albright S. Christian, Winston Wayne L., *Business Analytics: Data Analysis and Decision Making (7th Edition)*, Cengage Learning, 2021.
T2. James Evans, *Business Analytics: Methods, Models, and Decisions (2nd Edition)*, Pearson, 2020.
T3. U Dinesh Kumar, *Business Analytics: The Science of Data-Driven Decision Making*, Wiley, 2020

References

R1. Foster Provost, Tom Fawcett, *Data Science for Business (2nd Edition)*, O'Reilly Media, 2023.
R2. John D. Kelleher, Brendan Tierney, *Data Science*, MIT Press, 2020.
R3. Galit Shmueli, Peter C. Bruce, *Data Mining for Business Analytics*, Wiley, 2021.

Web Resources

1. https://powerbi.microsoft.com
2. https://www.tableau.com/learn
3. https://www.kaggle.com/learn

CSC4720- Design of Experiments and Statistical Quality Control

Course Code: CSC4720	**Course Title: Design of Experiments and Statistical Quality Control** **Type of Course: Discipline Elective**	**L- T-P- C**	3	0	0	3
Version No.	**1.0**					
Course Pre-requisites	Basic Probability and Statistics					
Anti-requisites	**NIL**					
Course Description	This course provides a comprehensive understanding of statistical techniques used for process improvement, quality assurance, and experimental analysis. It focuses on the systematic planning of experiments, analysis of experimental data, and application of statistical quality control tools to monitor, control, and improve processes in manufacturing and service industries.					
Course Objective	To equip students with the ability to design statistically sound experiments, analyze experimental data using appropriate statistical methods, and apply quality control techniques to achieve continuous process improvement and decision-making.					
Course Outcomes	**On successful completion of the course the students shall be able to:** **CO1:** Apply statistical concepts including probability distributions, sampling, and descriptive statistics for quality analysis. [Apply] **CO2:** Apply statistical quality control techniques such as control charts and process capability analysis for monitoring process performance. [Apply] **CO3:** Analyze design of experiments principles including randomization, replication, and blocking for experimental studies. [Analyze] **CO4:** Apply ANOVA and classical experimental designs such as CRD, RBD, and LSD for analyzing experimental data. [Apply] **CO5:** Analyze factorial and fractional factorial designs to study interaction effects and optimize processes. [Analyze] **CO6:** Evaluate advanced DOE techniques such as response surface methodology, Taguchi methods, and Six Sigma for quality improvement. [Evaluate]					
Course Content:						
Module 1	**Introduction to Quality and Statistical Foundations**					**9 Sessions**
Concept of quality and quality characteristics; Quality perspectives and dimensions; Role of statistics in quality engineering; Types of data and measurement scales; Sources of variation in processes; Descriptive statistics: measures of central tendency and dispersion; Probability concepts; Discrete and continuous probability distributions (Binomial, Poisson, Normal); Sampling methods and sampling distributions; Central Limit Theorem.						
Module 2	**Statistical Quality Control**					**9 Sessions**
Introduction to Statistical Quality Control (SQC); Philosophy of process control; Chance and assignable causes of variation; Control charts for variables: $\bar{X}$–R and $\bar{X}$–S charts; Construction and interpretation of control charts; Control charts for attributes: p, np, c, and u charts; Operating characteristic curves; Process capability analysis (Cp, Cpk); Acceptance sampling plans.						
Module 3	**Fundamentals of Design of Experiments**					**9 Sessions**

Need for Design of Experiments; Limitations of one-factor-at-a-time experimentation; Basic principles of DOE: randomization, replication, and blocking; Experimental error and model adequacy checking; Completely Randomized Design (CRD); Randomized Block Design (RBD); Latin Square Design (LSD); Analysis of variance (ANOVA) for single-factor experiments.		
Module 4	**Factorial and Fractional Factorial Designs**	**9 Sessions**
Introduction to factorial experiments; Two-level factorial designs; Main effects and interaction effects; Yates algorithm; ANOVA for factorial experiments; Confounding and alias structure; Fractional factorial designs; Resolution of designs; Screening experiments; Practical industrial examples.		
Module 5	**Advanced DOE and Quality Improvement Techniques**	**9 Sessions**
Response Surface Methodology (RSM); First-order and second-order models; Steepest ascent method; Central composite and Box–Behnken designs; Taguchi approach to robust design; Signal-to-noise ratios; Introduction to Six Sigma and DMAIC methodology; Applications of DOE and SQC through case studies.		
Text Book(s) T1. Douglas C. Montgomery, *Design and Analysis of Experiments (10th Edition)*, Wiley, 2021. T2. Douglas C. Montgomery, *Introduction to Statistical Quality Control (8th Edition)*, Wiley, 2020. T3. Jiju Antony, *Design of Experiments for Engineers and Scientists (2nd Edition)*, Elsevier, 2021.		
References R1. George E. P. Box, J. Stuart Hunter, William G. Hunter, *Statistics for Experimenters (2nd Edition, Updated)*, Wiley, 2020. R2. Dale H. Besterfield, *Quality Control (9th Edition)*, Pearson, 2021. R3. Kai Yang, Basem El-Haik, *Design for Six Sigma (2nd Edition)*, McGraw-Hill, 2020.		
Web References 1. https://www.itl.nist.gov/div898/handbook/ 2. https://asq.org/quality-resources 3. https://www.sixsigmaonline.org		