ANDHRA KESARI UNIVERSITY ::ONGOLE

Model Syllabus for 4-Year UG Honours in B.Sc. (Artificial Intelligence) as Major in consonance with Curriculum framework w.e.f. AY 2025-26

COURSE STRUCTURE (for Semester I to VI)

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
I	I	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation Lab	2	1
		2	Mathematical Foundation for AI	4	4
	II	3	Python Programming and Data Structures	3	3
			Python programming and Data Structure Lab	2	1
		4	Artificial & Computational Intelligence	3	3
			Artificial & Computational Intelligence Lab	2	1
II	III	5	Statistical Foundation of AI	3	3
			Statistical Foundation of AI Lab	2	1
		6	DBMS	3	3
			DBMS Lab	2	1
		7	Exploratory Data Analysis & Data Visualization	3	3
			Exploratory Data Analysis & Data Visualization Lab	2	1
	IV	8	Data Science with R	3	3
			Data Science with R Lab	2	1
		9	Foundation of ML & Supervised Learning	3	3
			Foundation of ML & Supervised Learning lab	2	1
		10	Robotics Principles & Embedded systems	3	3
			Robotics Principles & Embedded systems Lab	2	1
III	V	11	Unsupervised & Reinforcement Learning	3	3
			Unsupervised & Reinforcement Learning Lab	2	1

SEMESTER-III

COURSE 5: STATISTICAL FOUNDATIONS FOR AI

Theory | Credits: 3 | 3 hrs/week

Course Objectives

- Develop a strong grasp of statistical concepts and their role in AI modeling.
- Equip students to describe and analyze random variables, expectation, and variance.
- Enable mastery over various discrete and continuous probability distributions relevant to AI.
- Teach methods for evaluating relationships and making predictions through correlation and regression.
- Instill skills for drawing valid conclusions via inference, estimation, and hypothesis testing.
- Introduce students to advanced topics like multivariate and time series analysis, connecting statistical foundations to AI applications.
- Foster data analysis proficiency using Advanced Excel for simulation and model building.

Course Outcomes

- By the end of this course, students will be able to:
- Analyze and interpret random variables using expectation and variance.
- Identify and apply discrete and continuous probability distributions to AI problems.
- Evaluate relationships between variables using correlation and regression analyses.
- Perform statistical inference, estimate parameters, and conduct hypothesis testing.
- Apply multivariate and time series analysis for complex, real-world AI scenarios.
- Utilize Advanced Excel tools for data exploration, visualization, and statistical modeling suited for AI tasks.

Unit 1: Random Variables, Expectation, and Variance

Random variables: definition, types (discrete & continuous), and properties, Probability mass function (PMF) and probability density function (PDF), Cumulative distribution function (CDF), Mathematical expectation (mean), variance, and standard deviation, Moments and moment-generating functions

Unit 2: Probability Distributions

Discrete distributions: Binomial, Poisson, Geometric, Negative Binomial distributions — definitions, properties, and examples

Continuous distributions: Uniform, Normal (Gaussian), Exponential, Gamma distributions — definitions, properties, and applications

Joint, marginal, and conditional distributions, Introduction to Central Limit Theorem

Unit 3: Correlation and Regression

Bivariate data and scatter plots

Correlation: Pearson and Spearman coefficients, interpretation

Simple linear regression: model, estimation, properties, and analysis of variance

Multiple linear regression basics (conceptual understanding)

Residuals and goodness of fit

Unit 4: Statistical Inference, Estimation, and Hypothesis Testing

Population and sample, parameters and statistics, Sampling distributions, Point and interval estimation (confidence intervals), Tests of significance: z-test, t-test, chi-square test, and F-test, p-values and errors (Type I & II), Power of a statistical test

Unit 5: Multivariate and Time Series Analysis

Introduction to multivariate data and covariance matrices, Principal Component Analysis (PCA) basics and dimensionality reduction (conceptual), Introduction to clustering (k-means concept), Time series: Trend, seasonal, cyclical, and irregular components, Autocorrelation and lag, moving averages, forecasting basics

Activities:

Unit 1: Random Variables, Expectation, and Variance

Activities:

Solve problems involving calculation of expectation and variance using real-world datasets.

Advanced Excel exercise: Compute mean, variance, and standard deviation for sample data.

In-class quiz: Define and distinguish discrete/continuous random variables.

Outcome:

Students can identify, classify, and compute key properties (mean, variance) for random variables using formulas and Excel.

Evaluation Method:

Quiz (written/in-class): 5%

Excel assignment submission: 5%

Unit 2: Probability Distributions

Activities:

Practical Excel session: Simulate and visualize binomial, Poisson, and normal distributions.

Group worksheet: Draw and describe PMFs, PDFs, and CDFs for common distributions.

Homework: Problems on marginal and joint distributions.

Outcome:

Students can model and visualize discrete and continuous distributions for AI-relevant scenarios using Excel.

Evaluation Method:

Excel lab practical: 5%

Homework/problem set: 5%

Unit 3: Correlation and Regression

Activities:

Excel activity: Prepare scatter plots, calculate Pearson/Spearman coefficients, apply linear regression tool.

Mini-project: Analyze relationships between variables in an AI dataset (e.g., prediction, feature analysis).

Class discussion: Interpret regression outputs and residuals.

Outcome:

Students can analyze and interpret relationships and predictions using correlation and regression tools in Excel.

Evaluation Method:

Mini-project or report: 8%

Excel lab practical/lab observation: 4%

Unit 4: Statistical Inference, Estimation, and Hypothesis Testing

Activities:

In-class demonstration: Confidence intervals and hypothesis testing in Excel (t-test, z-test, chi-square).

Lab: Conduct sample analysis and interpret p-values, errors.

Group case study: Real or simulated AI problem involving inference.

Outcome:

Students can perform estimation and hypothesis testing, and make statistically valid decisions using Excel analysis.

Evaluation Method:

Written/lab-based test: 8%

Group/case study submission: 4%

Unit 5: Multivariate and Time Series Analysis

Activities:

Excel-based PCA (Principal Component Analysis) on a sample dataset (using add-ins or manual approach).

Time series exercise: Calculate and plot moving averages, trends, and seasonal decomposition.

Presentation: Apply a multivariate or forecasting approach to a real AI dataset.

Outcome:

Students can summarize multivariate relationships and analyze time series patterns using Excel.

Evaluation Method:

Excel analysis/presentation: 8%

Final test/project component: 7%

Recommended Textbooks & References

1. Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists
2. Douglas C. Montgomery & George C. Runger, Applied Statistics and Probability for Engineers
3. D.C. Agarwal, Statistics for Data Science and AI
4. Larry J. Stephens, Excel Data Analysis: Your visual blueprint for analyzing data, statistics, and AI
5. Online: Khan Academy Statistics & Probability, Coursera Statistics for Data Science, Microsoft Excel Help

SEMESTER-III

COURSE 5: STATISTICAL FOUNDATIONS FOR AI

Practical **Credits: 1** **2 hrs/week**

Advanced Excel Lab

1. Random Variable Simulation

   - Simulate and visualize discrete/continuous random variables using Excel functions and Data Analysis Toolpak.

2. Expectation & Variance Calculation

   - Use Excel formulas to compute expected value, variance, and standard deviation from given datasets.

3. Modeling Discrete Probability Distributions

   - Generate and plot Binomial and Poisson distributions; analyze probabilities and mean/variance.

4. Modeling Continuous Distributions

   - Simulate Normal and Exponential distributions; use NORM.DIST, NORM.INV, EXPON.DIST functions.

5. Correlation Analysis

   - Calculate Pearson/Spearman correlation coefficients; visualize with scatter plots and trendlines.

6. Linear Regression in Excel

   - Fit a linear regression model, interpret coefficients, predict new values; use REGRESSION tool.

7. Statistical Inference & Estimation

   - Create confidence intervals using Excel formulas; visualize sampling distributions.

8. Hypothesis Testing

   - Perform z-test, t-test, chi-square tests in Excel; interpret p-values and results.

9. Multivariate Analysis

   - Conduct simple PCA (manual or via add-ins); analyze covariance matrices for multidimensional datasets.

10. Time Series Analysis

    - Organize time series data, create line charts, calculate moving averages, detect trends and seasonality, basic forecasting.

SEMESTER-III

COURSE 6: DATABASE MANAGEMENT SYSTEMS

Theory	Credits: 3	3 hrs/week

Course Objectives

- Understand fundamentals of data, file systems, and database systems architecture.
- Develop skills in conceptual data modeling through ER/EER diagrams.
- Learn relational database principles including keys, constraints, relational algebra, and normalization.
- Gain proficiency in SQL for data definition, manipulation, and basic procedural programming with PL/SQL.
- Understand NoSQL concepts with a focus on MongoDB architecture, CRUD operations, data modeling, indexing, and replication.

Course Outcomes

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

- Explain database concepts, architectures, and compare file-based systems to DBMS.
- Design conceptual data models using ER/EER diagrams and map to relational schemas.
- Apply SQL queries to perform data definition, manipulation, aggregation, and control PL/SQL blocks.
- Demonstrate fundamental NoSQL concepts and perform CRUD and query operations in MongoDB.
- Model data effectively in MongoDB and understand optimization techniques such as indexing and replication.

Unit 1: Overview of Database Management System

Introduction to data, information, database, and DBMS, File-based system and its drawbacks, Database approach and advantages, Classification of Database Management Systems, Various Data Models and Components of DBMS, Three-schema architecture of database, Costs and risks of database approach

Unit 2: Entity-Relationship Model and Relational Model

Introduction to ER Model: entities, attributes, relationships, Classification of entity sets and attributes, Relationship degree and classification, Reducing ER diagram to tables, Enhanced

ER model: generalization, specialization, IS-A relationships, Relational model: CODD Rules, concepts of keys, relational integrity, Relational algebra operations, Functional dependencies and normal forms

Unit 3: Structured Query Language and PL/SQL

Introduction to SQL and data types, Data Definition Language (DDL) and Data Manipulation Language (DML), Selection, projection, join, set operations, Aggregate functions, views, subqueries, Table modification commands, Introduction to PL/SQL: language elements, procedures, functions, triggers

Unit 4: Introduction to NoSQL and MongoDB Basics

What is NoSQL? Features and types of NoSQL databases, CAP theorem and BASE properties, Difference between RDBMS and NoSQL, MongoDB Architecture: database, collection, BSON format, MongoDB Datatypes (String, Number, Date, Boolean, Array, Embedded Docs), Installation and Setup: Mongo shell and GUI, Database and collection management

Unit 5: MongoDB Operations, Data Modeling, and Optimization

CRUD operations: insertOne, find, updateOne, deleteOne, bulk operations, Query operators and working with arrays, Data modeling in MongoDB: embedded vs referenced documents, Aggregation framework: pipelines, stages, operators, Indexing (single, compound, multikey, text indexes), Replication concepts: replica sets, failover, consistency

Activities

Unit 1: Overview of Database Management System

Activity: Study different database models and classify DBMS architectures.

Outcome: Understand database systems fundamentals and architectures.

Evaluation Method: Written quiz on database models and classification; short assignment analyzing file-based vs DBMS approach.

Unit 2: Entity-Relationship Model and Relational Model

Activity: Create ER and EER diagrams for a sample application.

Outcome: Develop skills in conceptual modeling and understand relational integrity and normalization.

Evaluation Method: Project submission of ER/EER diagrams; quiz on relational algebra and normalization.

Unit 3: Structured Query Language and PL/SQL

Activity: Write and execute SQL queries on sample databases; develop PL/SQL procedures and triggers.

Outcome: Ability to manipulate database data and write procedural code blocks.

Evaluation Method: Practical exam on SQL query writing and PL/SQL program development.

Unit 4: Introduction to NoSQL and MongoDB Basics

Activity: Install and configure MongoDB; perform basic CRUD operations in shell/GUI.

Outcome: Understand NoSQL principles and MongoDB architecture.

Evaluation Method: Lab assignments on basic MongoDB commands and database/collection management.

Unit 5: MongoDB Operations, Data Modeling, and Optimization

Activity: Design MongoDB schema using embedded and referenced documents; implement aggregation pipelines and indexing.

Outcome: Model and query document data and optimize performance.

Evaluation Method: Project on MongoDB data modeling and aggregation; written test on indexing and replication concepts.

Recommended Textbooks & References

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw Hill Education.
2. Ramez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", Pearson.
3. Kristina Chodorow, "MongoDB: The Definitive Guide", O'Reilly Media.
4. MongoDB Manual & Official Documentation (https://docs.mongodb.com/)
5. C.J. Date, "An Introduction to Database Systems", Pearson.

SEMESTER-III

COURSE 6: DATABASE MANAGEMENT SYSTEMS

Practical **Credits: 1** **2 hrs/week**

Experiment 1 : Database: Inventory Management

Table 1: Products

Structure:

Column Name	Data Type	Constraints
product_id	INT	PRIMARY KEY
product_name	VARCHAR(50)	NOT NULL
price	DECIMAL(10,2)	CHECK(price > 0)
stock_qty	INT	CHECK(stock_qty >= 0)

Sample Data:

product_id	product_name	price	stock_qty
1	Pen	10.00	100
2	Notebook	50.00	200
3	Stapler	120.00	50
4	Marker	25.00	80
5	File Folder	60.00	150

Table 2: Suppliers

Structure:

Column Name	Data Type	Constraints
supplier_id	INT	PRIMARY KEY
supplier_name	VARCHAR(50)	NOT NULL
contact_no	VARCHAR(20)	UNIQUE
product_id	INT	FOREIGN KEY REFERENCES Products(product_id)

Sample Data:

supplier_id	supplier_name	contact_no	product_id
101	StationeryMart	9876543210	1
102	PaperWorld	9876500000	2
103	OfficeSupplies	9876512345	3
104	MarkerHub	9876522222	4
105	FileDepot	9876533333	5

Section A: DDL (Data Definition Language)

1. Create a database called InventoryDB.
2. Create a table Products and table Suppliers with the specified columns and constraints:

Section B: DML (Data Manipulation Language)

4. Insert at least 5 rows into the Products table.
5. Insert at least 5 rows into the Suppliers table.
6. Update the stock quantity of product 'Pen' to 120.
7. Delete a supplier with a specific supplier_id.
8. Write a query to rename 'Notebook' to 'NoteBook A4'

Section C: DQL (SELECT Queries)

9. Display all records from the Products table.
10. Display only product_name and price of all products.
11. List all products that have a stock quantity less than 100.
12. Show all products between 20 and 100 price range.
13. Find all suppliers whose contact number starts with '98765'.
14. Find the average price of products.
15. Display the total number of products in the inventory.
16. Show the maximum and minimum stock quantities.
17. Count how many suppliers supply each product.
18. Show all products where price > 50 AND stock_qty > 100.
19. Show all products where price < 20 OR stock_qty < 80.
20. Display suppliers whose supplier_name contains the word 'Mart'
21. List all suppliers along with the product they supply (use INNER JOIN).
22. Display suppliers whose name starts with 'S'.
23. Find products whose name has exactly 5 characters
24. Find suppliers who supply products costing more than 100.

Experiment 2 : ONLINE BOOKSTORE DB

An online book store wants to implement a BOOKSTORE DB for managing their online transactions by using the following tables.

Authors Table

Column Name	Data Type	Constraints
author_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
nationality	VARCHAR	NULL allowed

Books Table

Column Name	Data Type	Constraints
book_id	INTEGER	PRIMARY KEY

Title	VARCHAR	NOT NULL
author_id	INTEGER	FOREIGN KEY REFERENCES Authors
publication_year	INTEGER	
Price	DECIMAL	

Customers Table

Column Name	Data Type	Constraints
customer_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
Email	VARCHAR	UNIQUE, NOT NULL
Address	VARCHAR	NOT NULL

Orders Table

Column Name	Data Type	Constraints
order_id	INTEGER	PRIMARY KEY
customer_id	INTEGER	FOREIGN KEY REFERENCES Customers
book_id	INTEGER	FOREIGN KEY REFERENCES Books
order_date	DATE	NOT NULL
quantity	INTEGER	NOT NULL

SAMPLE DATA SET for BOOKSTORE DB

Authors Table

author_id	first_name	last_name	nationality
1	Jane	Austen	British
2	George	Orwell	British
3	Gabriel	Garcia Marquez	Colombian
4	Toni	Morrison	American
5	Mark	Twain	American
6	Harper	Lee	American
7	Fyodor	Dostoevsky	Russian

Books Table

book_id	Title	author_id	publication_year	price
101	Pride and Prejudice	1	1813	12.99
102	1984	2	1949	9.50
103	One Hundred Years of Solitude	3	1967	15.00
104	Beloved	4	1987	11.25
105	Animal Farm	2	1945	8.75
106	Adventures of Huckleberry Finn	5	1884	10.50
107	To Kill a Mockingbird	6	1960	14.00

Customers Table

customer_id	first_name	last_name	Email	address
201	Alice	Smith	alice.s@example.com	12 Oak St, London
202	Bob	Johnson	bob.j@example.com	45 Pine Ave, Oxford
203	Charlie	Brown	charlie.b@example.com	78 Maple Rd, Bristol
204	Diana	Prince	diana.p@example.com	34 Queen St, York
205	Edward	Norton	edward.n@example.com	22 River Ln, Leeds
206	Fiona	Hall	fiona.h@example.com	56 Lake Dr, Bath
207	Greg	Miller	greg.m@example.com	89 Park Ave, Glasgow

Orders Table

order_id	customer_id	book_id	order_date	Quantity
301	201	101	2025-07-20	1
302	202	102	2025-07-21	2
303	201	105	2025-07-22	1
304	203	103	2025-07-23	1
305	204	106	2025-07-24	1
306	205	107	2025-07-25	3
307	206	104	2025-07-26	2

Section A: DDL (Schema Design & Constraints)

1. Write SQL statements to create all 4 tables (Authors, Books, Customers, Orders) with:
    - Primary Keys
    - Foreign Keys
    - Appropriate data types
    - NOT NULL constraints where necessary.

2. Alter the Books table to add a constraint that price must be greater than 0.
3. Add a new column phone_number to the Customers table (VARCHAR(15)) and ensure it is unique.
4. Drop the phone_number column from the Customers table.

Section B: DML (Data Manipulation)

5. Insert at least 7 records for each table (use sample dataset above).
6. Update the price of the book titled *Animal Farm* by increasing it by 10%.
7. Delete all orders made before 2025-07-21.
8. Change the nationality of Gabriel Garcia Marquez to "Latino-American".

Section C: SELECT Queries (Data Querying)

9. List all books published between 1900 and 2000.
10. Find all customers whose email contains "example.com".
11. Retrieve books whose price is between 10 and 15 and published before 1950.
12. Show authors who are either 'British' or 'American'.
13. Find books that have a price less than 10 or are published after 1980.
14. Display all orders placed after 2025-07-22.
15. List all books written by author with author_id = 2.
16. Find customers whose last name starts with B.
17. Show all books with a price NOT between 9 and 13.
18. Display books whose publication_year is in (1813, 1945, 1987).
19. Find authors whose nationality is NOT 'British'.
20. List customers whose address contains the word Park.
21. Show all books sorted by price in descending order.
22. List authors in alphabetical order by last_name.
23. Display orders sorted by order_date (latest first).

Use of Date Functions

24. Show all orders placed in July 2025.
25. Show all orders with an estimated delivery date (5 days after order date).
26. Show customers who placed an order on a weekend.
27. Calculate how many days have passed since the last order was placed.

Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)

28. Count the total number of books in the database.
29. Find the average price of all books.
30. Show the highest-priced book.
31. Count how many orders each customer has placed.
32. Calculate the total sales (price × quantity) for each customer.

GROUP BY and HAVING

33. Count how many books are written by each author.
34. Group orders by customer_id and display total quantity ordered.
35. Show customers who have ordered more than 2 books in total (use HAVING).
36. Find the total number of books sold per author (GROUP BY author).

Experiment 3: EMPLOYEE DB

An enterprise wants to automate its employee management process by implementing an Employee Database.The goal is to replace manual record-keeping with a centralized system that stores employee, department, and project details. Use the following table structures and data set to implement Employee DB.

EmployeeDB - Table Structures

1. Departments Table

Column	Type	Constraints
dept_id	INT	PRIMARY KEY
dept_name	VARCHAR	UNIQUE, NOT NULL
location	VARCHAR	NOT NULL

2. Employees Table

Column	Type	Constraints
emp_id	INT	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
email	VARCHAR	UNIQUE, NOT NULL
phone	VARCHAR	CHECK (phone LIKE '--____')
hire_date	DATE	NOT NULL
job_title	VARCHAR	NOT NULL
salary	DECIMAL	CHECK (salary > 0)
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)
manager_id	INT	FOREIGN KEY REFERENCES Employees(emp_id) (self-referential)

3. Projects Table

Column	Type	Constraints
project_id	INT	PRIMARY KEY
project_name	VARCHAR	NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NULL
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)

4. Employee_Project Table (Many-to-Many)

Column	Type	Constraints
emp_id	INT	FOREIGN KEY REFERENCES Employees(emp_id), PRIMARY KEY(emp_id, project_id)
project_id	INT	FOREIGN KEY REFERENCES Projects(project_id)
hours_allocated	INT	CHECK (hours_allocated > 0)

Sample Data Set

Departments Table

dept_id	dept_name	Location
1	HR	New York
2	IT	San Francisco
3	Finance	Chicago
4	Marketing	Boston
5	Operations	Seattle
6	Legal	Washington D.C.
7	Sales	Dallas
8	R&D	Austin
9	Procurement	Denver
10	Customer Care	Miami

2. Employees Table

emp_id	first_name	last_name	Email	phone	hire_date	job_title	salary	dept_id	manager_id
101	Alice	Johnson	alice.j@corp.com	123-456-7890	2020-03-15	HR Manager	75000	1	NULL
102	Bob	Smith	bob.s@corp.com	234-567-8901	2019-05-20	IT Analyst	65000	2	104
103	Charlie	Brown	charlie.b@corp.com	345-678-9012	2021-01-10	Finance Executive	58000	3	106

104	Diana	Prince	diana.p@corp.com	456-789-0123	2018-07-12	IT Manager	90000	2	NULL
105	Ethan	Hunt	ethan.h@corp.com	567-890-1234	2022-02-25	Marketing Lead	62000	4	NULL
106	Fiona	Hall	fiona.h@corp.com	678-901-2345	2017-11-01	Finance Manager	85000	3	NULL
107	Greg	Miles	greg.m@corp.com	789-012-3456	2023-04-15	IT Support	45000	2	104
108	Hannah	White	hannah.w@corp.com	890-123-4567	2021-09-05	HR Executive	50000	1	101
109	Ian	Scott	ian.s@corp.com	901-234-5678	2020-11-20	Operations Analyst	56000	5	NULL
110	Julia	Adams	julia.a@corp.com	012-345-6789	2019-12-18	Legal Advisor	70000	6	NULL

3. Projects Table

project_id	project_name	start_date	end_date	dept_id
201	Payroll System	2023-01-01	NULL	3
202	Website Upgrade	2023-02-10	NULL	2
203	Recruitment Drive	2023-03-05	NULL	1
204	Ad Campaign	2023-05-20	NULL	4
205	New CRM Tool	2023-04-15	NULL	7
206	Compliance Portal	2023-06-10	NULL	6
207	Inventory System	2023-07-01	NULL	5
208	AI Research	2023-08-05	NULL	8
209	Customer Feedback	2023-09-10	NULL	10
210	Procurement System	2023-10-01	NULL	9

4. Employee_Project Table

emp_id	project_id	hours_allocated
102	202	120

104	202	80
103	201	100
106	201	150
101	203	50
105	204	70
107	202	60
109	207	90
110	206	110
108	203	40

Section A: DDL (Schema Creation & Modification)

1. Write SQL statements to create the above tables with the specified constraints
2. Alter the Employees table to add a column bonus DECIMAL(8,2) with default value0.
3. Drop the column bonus from Employees.

Section B: DML (Insert, Update, Delete)

4. Insert at least 10 rows into Departments, Employees, Projects, and Employee_Project.(use the above data set)
5. Try inserting an employee with a negative salary (should fail due to CHECK constraint).
6. Update the salary of the employee with emp_id = 103 by 15%.
7. Delete an employee record who has resigned (choose any emp_id).
8. Increase all employees' salaries in the IT department by 5%.
9. Change the department of an employee to "Research".(should fail due to FK constraint)

Section C: DQL (Select Queries)

10. List all employees and their details.
11. Show all employees in the "HR" department.
12. Find employees with salaries between 50,000 and 80,000.
13. Retrieve employees hired after 2020.
14. Show employees who are in either the IT or Finance department.
15. Find employees whose email ends with "@corp.com".
16. List all employees with salary > 60,000 AND located in "New York".
17. Display employees in descending order of salary.
18. Count the number of employees in each department.
19. Show the average salary of employees department-wise.
20. Display departments where the average salary is greater than 70,000.
21. Find the number of employees in each project.
22. Display departments with more than 3 employees.
23. Show the sum of all salaries department-wise.
24. List all distinct department IDs from the Employees table.

25. Show employee names with the year they were hired.
26. Show employees grouped by the year of hire.
27. List employees hired in the last 90 days.
28. List the no of years of experience of all the employees

Section D: Joins

29. List all employees with their department names (INNER JOIN).
30. Display all departments along with employees, including those departments without employees (LEFT JOIN).
31. Show employees and the projects they are working on (JOIN 3 tables: Employees, Employee_Project, Projects).
32. List projects along with total hours allocated by employees.
33. Write a query to find employees who are working on more than one project.
34. Show all projects handled by the 'Finance' department.

Experiment 4 : PL/SQL Programming

1. Write a procedure GetEmpInfo that takes emp_id as input and displays name, salary, and department.
2. Write a PL/SQL block that checks if an employee's salary is above 50,000. If yes, print "High Salary" ;Otherwise print "Standard Salary".
3. Write a PL/SQL program to display the top 10 rows in the Emp table based on their job and salary
4. Write a stored procedure GiveBonus that takes department ID and a designation as input, along with a bonus amount, and updates the salary of all employees in that department who have the specified designation by adding the bonus amount to their current salary.
5. Create a trigger to prevent inserting employees with a salary less than 30,000.
6. Create a trigger to avoid any transactions(insert, update, delete) on the EMP table on Saturday & Sunday.

Experiment 5: MongoDB Setup and Basic Operations

- Install, configure MongoDB and Mongo Compass
- Create databases and collections, basic insertOne and insertMany operations

Experiment 6: MongoDB CRUD Operations

- Query documents using find() with filter operators
- Update and delete documents using updateOne(), updateMany(), deleteOne(), deleteMany()

Experiment 7: Data Modeling in MongoDB

- Design embedded document models for student-course system
- Design normalized data model with document references and implement

Experiment 8: Aggregation Framework

- Create aggregation pipelines with $match, $group, $project, $sort

- Use advanced operators like $lookup (joins), $unwind, $bucket

Experiment 9: Indexing and Query Optimization

- Create single field, compound, multikey, and text indexes
- Analyze query performance and optimize indexes

Experiment 10: Replication and Transactions in MongoDB

- Configure replica sets and test failover scenarios
- Implement multi-document transactions

SEMESTER-III

COURSE 7: EXPLORATORY DATA ANALYSIS & DATA VISUALIZATION

Theory	Credits: 3	3 hrs/week

Course Objectives

1. To introduce the concepts, importance, and workflow of exploratory data analysis.
2. To impart practical skills in data preprocessing, cleaning, and manipulation using Python.
3. To equip students with techniques in univariate, bivariate, and multivariate data analysis.
4. To develop proficiency in data visualization and interpretation using standard Python libraries.
5. To enable students to build reports and dashboards to communicate data insights effectively.

Course Outcomes

After completing this course, students will be able to:

1. Explain the importance and process of EDA in the data science pipeline.
2. Perform robust data preprocessing including missing value handling, outlier detection, transformation, and encoding.
3. Manipulate, filter, and reshape datasets using Python libraries like Pandas and NumPy.
4. Conduct data analysis and visualize results using Matplotlib, Seaborn, and Plotly.
5. Design interactive visualizations and dashboards to communicate findings in real-world scenarios.

Unit 1: Introduction to Exploratory Data Analysis (EDA)

Significance and objectives of EDA, Types and scales of data (nominal, ordinal, interval, ratio), Concepts of variability and central tendency, Role of EDA in the data science pipeline, Introduction to data quality and data issues, Differences between EDA, classical, and Bayesian analysis, Basic terminology and concepts

Unit 2: Data Preprocessing and Cleaning

Overview of data preprocessing, Handling missing types of missingness (MCAR, MAR, MNAR), Techniques for missing data imputation (mean, median, mode, forward/backward fill), Detection and treatment of duplicates, Outlier detection methods (Z-score, IQR method),

Data transformation: normalization and standardization, Encoding categorical variables: label encoding, one-hot encoding, Data integration and reduction techniques, Importance of preprocessing for machine learning models

Unit 3: Data Manipulation using Python with Pandas and NumPy

NumPy Array Creation: Use `np.array()`, `np.zeros()`, `np.ones()` to create arrays of various dimensions. **Array Manipulation:** Reshape, transpose, flatten arrays for flexible data formats. Element-wise Arithmetic Operations, Statistical Functions on NumPy Arrays

Indexing & Slicing: Access array elements via slicing, masking, and fancy indexing for selective data analysis.

Pandas DataFrames: Series and Data frames, Create from dictionaries, Series, CSV, Excel, and JSON files using Dataframe or reader functions

Data Frame Manipulation: Indexing & Selection - Label-based and Position-based indexing; conditional filtering with boolean masks, Aggregation & Grouping- Group data by columns with `.groupby()`, summarize using `.sum()`, `.mean()`, `.count()`.

Data Reshaping & Pivoting and stack/unstack hierarchical index data.

Merging & Joining Data Frames

Datetime & Categorical Data: Handle and transform temporal data with `pd.to_datetime()`, categorical conversions with `.astype('category')`.

Unit 4: Basic Data Analysis and Visualization

What is visualization, what is it's importance? Types: histograms, bar charts, Boxplots and violin plots for data distribution, Scatter plots and correlation analysis (Pearson, Spearman), Cross-tabulation and contingency tables, Analyzing relationships: correlation vs causation, Visualizing categorical vs numerical data

Unit 5: Advanced Visualization and Reporting

Principles of effective graphical representation, Introduction to visualization libraries: Matplotlib, Seaborn, Plotly, Advanced plots: heatmaps, pair plots, joint plots, Interactive visualizations and dashboards, Geographical data visualization basics, Storytelling with using visualization for communication, Introduction to reporting tools and best practices

Activities

Unit 1: Introduction to EDA

Activity:

Analyze the Titanic dataset to explore data types, central tendency, and variability.

Outcome:

Develop a foundational understanding of data structures and EDA significance.

Evaluation Method:

Lab notebook submission and oral quiz.

Unit 2: Data Preprocessing and Cleaning

Activity:

Clean a health survey dataset: identify/impute missing values, standardize columns, encode categories, remove duplicates/outliers.

Outcome:

Demonstrate proficiency in data cleaning and preprocessing techniques in Python.

Evaluation Method:

Script submission and report on preprocessing steps with before/after summary.

Unit 3: Data Manipulation using Python

Activity:

Manipulate a retail transaction dataset by indexing, filtering, grouping, merging, and handling date/time columns.

Outcome:

Apply multiple Pandas/NumPy techniques for dataset transformation.

Evaluation Method:

Graded Jupyter notebook, peer code review.

Unit 4: Basic Data Analysis and Visualization

Activity:

Explore demographic data with descriptive statistics, create histograms, boxplots, scatter plots, and cross-tabulations.

Outcome:

Visualize and interpret univariate and bivariate data insights.

Evaluation Method:

Visualization notebook and critical analysis write-up.

Unit 5: Advanced Visualization and Reporting

Activity:

- Build a dashboard on sales or air quality data using Matplotlib/Seaborn/Plotly.
- Incorporate interactive and geospatial visualizations.

Outcome:

- Develop professional dashboards and communicate insights through visual storytelling.

Evaluation Method:

Final project dashboard evaluation and presentation.

Preferred Textbooks

1. Suresh Kumar Mukhiya, Usman Ahmed, "Hands-On Exploratory Data Analysis with Python", Packt Publishing, 2020.
2. Jake Vander Plas, "Python Data Science Handbook: Essential Tools for Working with Data", O'Reilly, 2017.
3. Catherine Marsh, Jane Elliott, "Exploring Data: An Introduction to Data Analysis for Social Scientists", Wiley, 2008.

References

1. Eric Pimpler, "Data Visualization and Exploration with R", GeoSpatial Training Service, 2017.
2. Claus O. Wilke, "Fundamentals of Data Visualization", O'Reilly, 2019.
3. Matthew O. Ward, Georges Grinstein, Daniel Keim, "Interactive Data Visualization: Foundations, Techniques, and Applications", CRC Press, 2015.

SEMESTER-III

COURSE 7: EXPLORATORY DATA ANALYSIS & DATA VISUALIZATION

Practical **Credits: 1** **2 hrs/week**

Note: This lab has to be executed using interactive computing environments like Jupyter Notebook or Google Colab.

1. Importing Data and Basic Exploration
   - Load datasets from CSV/Excel/JSON
   - Use `.head()`, `.info()`, `.describe()`, `.shape` to examine structure and summary statistics
2. Examining Data Types and Missing Values
   - Identify data types for each column
   - Detect and quantify missing values with `.isnull()` and `.sum()`.
3. Handling Missing Data Techniques
   - Impute missing values using mean, median, mode, forward, and backward fill
   - Drop rows/columns with missing data and compare results
4. Dealing With Duplicates and Outliers
   - Find and remove duplicate entries
   - Detect outliers using Z-score or IQR methods
   - Treat or remove outliers and compare distributions
5. Data Transformation and Scaling
   - Normalize, standardize, log-transform columns
   - Encode categorical variables (Label Encoding, One-Hot Encoding)
6. Indexing, Filtering, and Slicing DataFrames
   - Select specific rows/columns using index and Boolean filters
   - Conditional selections and advanced slicing
7. Aggregating, Grouping, and Pivoting Data
   - Use `.groupby()` for aggregate statistics
   - Create pivot tables for categorical summaries
8. Merging and Concatenating Multiple Datasets
   - Join, merge, concatenate DataFrames
   - Handle merging keys and missing arguments
9. Univariate Visualization Techniques
   - Create histograms, boxplots, violin plots for distributions

- Visualize frequency and central tendency

10. Bivariate and Multivariate Visualization
    - Display scatter plots, correlation matrices, pair plots
    - Visualize relationships and patterns
11. Time Series Data Exploration
    - Parse and index datetime columns
    - Visualize trends and seasonality
12. Interactive Visualizations and Dashboards
    - Build simple interactive graphs with Plotly
    - Design a dashboard showing multiple aspects of a dataset

SEMESTER-IV

COURSE 8: DATA SCIENCE WITH R

Theory	Credits: 3	3 hrs/week

Course Objectives

1. Introduce the data science process, lifecycle, and applications in real-world domains.
2. Build proficiency in R programming for data manipulation, exploration, and visualization.
3. Train students in handling structured, unstructured, and time-based data effectively.
4. Familiarize with basic machine learning and statistical modeling using R.
5. Develop awareness of ethical, interpretability, and responsible use of data science.

Course Outcomes

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

1. Explain the Data Science process and perform EDA (Exploratory Data Analysis).
2. Write R programs using variables, functions, loops, and packages for basic analytics.
3. Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr, ggplot2).
4. Build and evaluate basic machine learning models such as regression and clustering.
5. Apply data science techniques to practical case studies.

Unit 1. Introduction to Data Science Process:

Introduction- Definition - Data Science in various fields - Examples - Impact of Data Science - Data Analytics Life Cycle - Data Science Toolkit - Data Scientist - Data Science Team, Exploratory Data Analysis (EDA), Feature Engineering & Data Transformation

Unit 2. Basics of R Programming:

Introduction to R and RStudio, Data Types, Variables, Operators, Control Structures (if, loops, apply), Functions and Packages, Data Input/Output (CSV, Excel, XML, JSON).

Unit 3. Data Handling & Visualization in R:

Data Frames, Lists, Matrices, Data Wrangling with dplyr and tidyr, Handling Missing Data, Working with Date/Time in R. Visualization with ggplot2: grammar of graphics, aesthetics, geometries, scales.

Faceting and layering techniques,Visualizing categorical and numerical data, Customizing and exporting plots

Unit 4. Applications & Case Studies in Data Science:

Simple Linear Regression, Multiple Regression

Model Evaluation Method: Accuracy, Confusion Matrix, ROC.

K-Means Clustering, Text Mining & Word Clouds, Recommender Systems Basics, Ethical Issues in Data Science

Unit 5. Advanced Topics in Data Science with R :

Introduction to Time Series Analysis in R (ARIMA basics)- Concept of time series (trend, seasonality, noise), Time series objects in R (ts, zoo, xts), Plotting and decomposing time series, Stationarity and differencing, Autocorrelation & Partial Autocorrelation (ACF/PACF), AR, MA, ARIMA model basics, Forecasting using forecast package

Creating interactive visualizations with plotly packages-Converting ggplot2 plots to interactive plots

Animations and sliders in plotly

R Shiny: Building interactive web applications-Introduction to Shiny framework, UI and server functions, Reactive expressions and reactivity in Shiny, Input and output widgets (sliders, dropdowns, text), Layouts and dashboard design

Textbooks

1. An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 2nd Edition, 2021
2. R for Data Science, Hadley Wickham and Garrett Grolemund, O'Reilly Media, 2017.

Reference Books

1. The Art of R Programming, Norman Matloff,, No Starch Press, 2011.
2. Modern Applied Statistics with S, W.N. Venables & B.D. Ripley, Springer, 2002.
3. Introduction to Data Science: Data Analysis and Prediction Algorithms with R, Rafael A. Irizarry, CRC Press, 2020.
4. Data Science from Scratch: First Principles with Python (for conceptual clarity only), Joel Grus,

Activities:

Outcome: Explain the Data Science process and perform EDA (Exploratory Data Analysis).

Activity: Use a real-world dataset (e.g., Titanic or COVID data) to:

- Outline the steps of the Data Science workflow
- Perform EDA using summary statistics and visualizations (histograms, boxplots, scatterplots)

Evaluation Method: Presentation and checklist (10-point scale):

- Clear explanation of workflow stages
- Quality of EDA insights
- Use of appropriate plots and summaries

Outcome: Write R programs using variables, functions, loops, and packages for basic analytics.

Activity: Write an R script that:

- Reads a CSV file
- Uses if, for, and while loops
- Defines and calls custom functions with arguments and return values

Evaluation Method: Code review and execution test to verify (10-point scale):

- Correctness of the syntax and logic
- Functionality of control structures
- Output accuracy and modularity

Outcome: Perform data wrangling, cleaning, and visualization with R libraries (dplyr, tidyr, ggplot2).

Activity: Clean a messy dataset using:

- dplyr for filtering, selecting, and mutating
- tidyr for reshaping and handling missing values
- Time-based operations (e.g., filling gaps, formatting dates)

Evaluation Method: Before-and-after comparison (10 point score):

- Completeness of cleaning steps
- Use of appropriate functions
- Handling of missing/time data

Outcome: Implement basic machine learning models and evaluate performance using appropriate metrics and visual tools.

Activity: Build a simple classification model (e.g., logistic regression or decision tree) using R:

- Train/test split
- Predict outcomes
- Evaluate using confusion matrix, accuracy, precision, recall

Evaluation Method: Model report and demo (10 point scale):

- Correct implementation of model
- Use of evaluation metrics

SEMESTER-IV

COURSE 8: DATA SCIENCE WITH R

Practical	Credits: 1	2 hrs/week

List of Practicals:

1. Compute Mean, Median, Mode, Variance, and Standard Deviation
2. Visualize Binomial, Normal, and Poisson Distributions
3. Perform t-test and Chi-Square Test in R
4. Calculate Correlation and Build a Simple Linear Regression Model
5. Conduct Exploratory Data Analysis (EDA) on a Real-World Dataset
6. Apply Feature Engineering: Scaling, Normalization, and Encoding
7. Practice R Programming: Variables, Control Structures, and Functions
8. Read and Write Data from CSV, Excel, JSON, and XML Files
9. Use dplyr and tidyr for Data Wrangling Tasks
10. Handle Missing Data and Detect Outliers
11. Work with Dates and Times in R
12. Visualize Data Using ggplot2 (Bar, Scatter, Histogram, Boxplot)
13. Perform K-Means Clustering and Visualize Clusters
14. Evaluate Models Using Confusion Matrix, Accuracy, and ROC Curve
15. Perform Text Mining and Create a Word Cloud
16. Time Series Forecasting with ARIMA on a real dateset (e.g., monthly airline passengers, stock prices, or temperature data).
17. Create interactive bar, line, and scatter plots using plotly. On a real dataset (e.g., COVID-19 cases, sales data, or student marks).
18. Develop a Shiny app that lets users upload a CSV file.

SEMESTER-IV

COURSE 9: FOUNDATION OF ML & SUPERVISED MACHINE LEARNING

Theory	Credits: 3	3 hrs/week

Course Objectives

- Provide a comprehensive understanding of supervised machine learning algorithms and their theoretical foundations.
- Enable students to formulate machine learning problems and implement algorithms in Python.
- Introduce fundamental learning theory concepts such as hypothesis, VC dimension, PAC learning, Find-S, and candidate elimination.
- Develop practical skills in model evaluation, tuning, and application to real-world datasets.
- Instill ethical awareness and explainability considerations in supervised learning.

Course Outcomes

On successful completion, students will be able to:

- Describe core supervised learning algorithms and their mathematical foundations.
- Implement and evaluate linear and nonlinear supervised learning models using Python.
- Apply learning theory fundamentals to analyze model complexity and generalization.
- Perform model selection, cross-validation, and hyperparameter tuning.
- Develop end-to-end supervised learning solutions for real-life problems.
- Uphold ethical standards and interpret model results critically.

Unit 1: Introduction to Supervised Machine Learning

What is Machine Learning & Supervised Machine Learning ? - Definitions

Machine learning pipeline: data collection, cleaning, splitting, feature selection

Loss functions, cost functions, bias-variance tradeoff

Performance metrics: accuracy, precision, recall, F1 score, ROC-AUC

Introduction to Python libraries (scikit-learn, pandas, matplotlib) for ML

Learning Theory Fundamentals: Hypothesis and hypothesis space, Concept learning: types of concepts, Find-S algorithm, Candidate Elimination algorithm: overview and intuition, VC dimension, PAC (Probably Approximately Correct) learning

Unit 2: Linear Models for Regression and Classification

Introduction to regression problems and applications

Simple linear regression: theory, assumptions, ordinary least squares

Gradient Descent for training linear models,

Multiple and polynomial regression models

Regularization techniques: Ridge, Lasso, Elastic Net

Logistic regression: sigmoid function, decision boundary, training using gradient descent

Evaluation metrics: RMSE, MAE, R^2, confusion matrix

Unit 3: Classification Algorithms

Concept of classification and categorical data, Decision Trees, Naive Bayes algorithm, Instance based learning, K-Nearest Neighbors (KNN), Support Vector Machines (SVM) with kernel tricks, **Evaluation metrics:** confusion matrix, ROC curves, cross-validation

Unit 4: Model Selection, Tuning, and Real-World Applications

Data preprocessing: imputation, scaling, encoding categorical variables, Feature engineering and selection techniques, Train-test split, k-fold cross-validation, Hyperparameter tuning: grid search, random search, Case studies in finance, healthcare, retail, Ethics and explainability in supervised learning

Unit 5: Ensemble Learning Methods

Bagging, boosting, stacking, Random Forests: theory, applications, feature importance , Gradient Boosting Machines (GBM), AdaBoost, XGBoost, Comparing ensembles versus single models

Avoiding overfitting: cross-validation, hyperparameter tuning

Activities:

Unit 1: Introduction and Learning Theory Fundamentals

Activities:

- Lecture and discussion on ML fundamentals and pipeline
- Concept learning exercises: Hypothesis space, Find-S, Candidate Elimination
- Concept quizzes on VC Dimension and PAC Learning
- Python intro: Data loading and basic evaluation metrics

Outcomes:

- Understand ML basics and key learning theory concepts.
- Illustrate concept learning algorithms and hypothesis characterization.

Evaluation Method:

- Quiz on theory concepts: 8%
- Assignment on implementing Find-S algorithm: 7%

Unit 2: Linear Models for Regression and Classification

Activities:

- Coding lab: Linear regression and logistic regression models in Python
- Visualization of model fits and decision boundaries
- Experiment with gradient descent and regularization parameters

Outcomes:

- Implement and interpret linear regression and logistic regression.
- Explain optimization and regularization for generalization.

Evaluation Method:

- Practical lab report: 10%
- Coding assignment (regression/classification): 7%

Unit 3: Classification Algorithms

Activities:

- Implement KNN, Decision Trees, SVM, and Naive Bayes classifiers
- Performance evaluation via confusion matrix, ROC, and cross-validation
- Group discussion on algorithm strengths and weaknesses

Outcomes:

- Train and evaluate various classification algorithms.
- Select appropriate classifiers for varied contexts.

Evaluation Method:

- Lab submission and class presentation: 12%

Unit 4: Model Selection, Tuning, and Applications

Activities:

- Feature engineering workshop and data preprocessing in Python
- Hyperparameter tuning using GridSearch and RandomizedSearch
- Mini project based on real-world dataset with ethical evaluation
- Presentation and peer review of project findings

Outcomes:

- Master data preparation, model tuning, and performance assessment.
- Demonstrate a full ML pipeline with ethical considerations.

Evaluation Method:

- Mini project and presentation: 16%
- Final written exam: 20%

Unit 5: Ensemble Learning Methods

Activities:

- Ensemble methods lab: Random Forest, AdaBoost, and XGBoost implementation
- Hyperparameter tuning exercises and cross-validation applications
- Case study: Comparing ensemble vs base models

Outcomes:

- Develop ensemble models and understand boosting/bagging concepts.
- Apply model tuning to improve performance.

Evaluation Method:

- Practical lab and report: 10%

Recommended Textbooks & References

- Tom M. Mitchell, Machine Learning, McGraw Hill International Edition
- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer
- Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly
- Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, Wiley
- Online: scikit-learn documentation, Coursera Machine Learning courses, Kaggle tutorials

SEMESTER-IV

COURSE 9: FOUNDATION OF ML & SUPERVISED MACHINE LEARNING

Practical **Credits: 1** **2 hrs/week**

Note: This lab has to be executed using interactive computing environments like Jupyter Notebook or Google Colab.

1. Data Loading, Exploration, and Preprocessing
   - Load datasets (e.g., Iris, Boston Housing) using pandas
   - Perform exploratory data analysis: summary statistics, visualization
   - Handle missing data, categorical encoding, feature scaling
2. Implement Simple Linear Regression
   - Fit linear regression to sample data
   - Visualize regression line and residuals
   - Compute evaluation metrics (RMSE, R^2)
3. Gradient Descent from Scratch
   - Implement gradient descent algorithm for linear regression
   - Experiment with learning rate and convergence
4. Logistic Regression for Classification
   - Train logistic regression model on binary classification dataset
   - Evaluate classification metrics: confusion matrix, accuracy, ROC curve
5. Decision Tree Classifier
   - Train decision tree on classification datasets
   - Visualize tree structure
   - Analyze feature importance and pruning effects
6. Naive Bayes Classifier
   - Train Naive Bayes on text or classification datasets
   - Performance comparison with other classifiers
7. K-Nearest Neighbors (KNN) Algorithm
   - Implement KNN classification using scikit-learn
   - Experiment with different values of k
   - Visualize decision boundaries
8. Support Vector Machine (SVM) Classifier

- Train SVM with linear and RBF kernel
- Understand margin and support vectors
- Parameter tuning and performance evaluation

9. Model Evaluation and Hyperparameter Tuning
   - Perform train-test split and K-fold cross-validation
   - Use GridSearchCV or RandomizedSearchCV
   - Evaluate final model performance on unseen data
10. Random Forest and Ensemble Methods
   - Train Random Forest classifier
   - Understand bagging and feature randomness
   - Compare ensemble performance with base classifiers
11. Gradient Boosting Machine (GBM)
   - Train GBM model using popular libraries (XGBoost or sklearn)
   - Visualize feature importance

SEMESTER-IV

COURSE 10: ROBOTICS PRINCIPLES AND EMBEDDED SYSTEMS

Theory	Credits: 3	3 hrs/week

Course Objectives

1. To introduce fundamental concepts of robotics including robot classification, kinematics, and mechanisms.
2. To explain the roles and types of actuators, drives, and sensors, including piezoelectric technologies.
3. To develop an understanding of robot control systems, feedback mechanisms, and intelligent control methods.
4. To expose students to embedded systems, focusing on microcontrollers and interfacing with hardware components.
5. To provide practical experience in Arduino programming and hardware interfacing for robotics applications.

Course Outcomes

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

1. Classify different types of robots and explain their kinematic principles and degrees of freedom.
2. Describe actuators, drives, and sensor technologies used in robotics, including piezoelectric actuators and sensors.
3. Design and analyze control systems for robotic joints with feedback and intelligent control strategies.
4. Understand embedded systems concepts and program microcontroller-based systems (Arduino) for robotic applications.
5. Develop and implement practical robotics solutions using Arduino hardware and software tools.

Unit 1: Introduction to Robotics

Classification of Robots, Components and Characteristics of Robots, Applications of Robotics, Degrees of Freedom: Definition, importance, calculation, examples

Fundamentals of Kinematics and Position Analysis, Robots as Mechanisms, Matrix Representation and Transformation Matrices, Forward and Inverse Kinematics

Unit 2: Actuators and Drives

Actuating Systems: Features, selection criteria, **Actuators:** Types- electric, hydraulic, pneumatic, **Drives:** Types and characteristics-electric, hydraulic, pneumatic, stepper, servo; comparison and usage, **Piezoelectric actuators:** Principle, applications in microrobotics, advantages (precision, quick response, energy harvesting) , Use of Reduction Gears, Actuating Devices and Control

Unit 3: Grippers and Sensors

Grippers: Types (mechanical, magnetic, vacuum, adhesive), applications, Criteria for gripper selection, Sensor characteristics, Description of Vision, Force, Proximity, Tilt Sensors, Piezoelectric sensors (Working principle, parts, & applications), types (force, vibration, pressure), applications of robotics, Principles and practical applications of key sensor types

Unit 4: Robot Controls

Point-to-Point and Continuous Path Control, Intelligent Robot Control Methods, Control Systems for Robot Joints, Control Actions, **Feedback Devices:** Concepts and significance, Integration of sensors in feedback and control loops (e.g., vibration damping, tactile feedback, health monitoring), Closed-loop vs. Open-loop Control Systems

Unit 5: Embedded Systems

Embedded Systems: Definition, characteristics, and significance in robotics, Microcontrollers and Microprocessors, Arduino microcontroller architecture and applications, Interfacing sensors, actuators, drives, and grippers with embedded systems, Embedded programming essentials (C/C++, input/output, analog/digital signals), Embedded communication protocols (serial, I2C basics), Integration of sensor data in embedded applications.

Unit 1: Introduction to Robotics

Activity: Analyze and classify various types of robots; calculate degrees of freedom for given robot models.

Outcome: Students will understand robot types, components, and the significance of degrees of freedom.

Evaluation Method: Written assignments and quizzes on robot classification and kinematics concepts.

Unit 2: Actuators and Drives

Activity: Investigate characteristics of electric, hydraulic, pneumatic actuators and drives; study piezoelectric actuator demonstrations.

Outcome: Ability to compare different actuators and drives and understand piezoelectric actuator applications.

Evaluation Method: Lab demonstration reports and presentation on actuators.

Unit 3: Grippers and Sensors

Activity: Examine different types of grippers; conduct hands-on experience with piezoelectric, proximity, force, and vision sensors.

Outcome: Understand gripper types, sensor characteristics, and apply piezoelectric sensing principles.

Evaluation Method: Practical sensor interfacing reports and sensor data analysis.

Unit 4: Robot Controls

Activity: Implement point-to-point and continuous path control algorithms; integrate sensor feedback for closed-loop control.

Outcome: Develop understanding of control methods and feedback systems in robotics.

Evaluation Method: Lab assignments implementing control algorithms on robotic joints; tests on control theory.

Unit 5: Embedded Systems

Activity: Program Arduino microcontrollers to interface sensors, actuators, and drives; develop embedded control applications.

Outcome: Ability to design embedded systems and program Arduino for real-world robotics applications.

Evaluation Method: Project work based on Arduino robotics applications, demonstration and code review.

Recommended Textbooks & References

1. "Robotics: Mechanics and Control" by Guruprasad K. Eastern Economy Edition
2. "Modern Robotics: Mechanics, Planning, and Control" by Kevin M. Lynch and Frank C. Park, Cambridge University Press.

3. "Introduction to Autonomous Mobile Robots" by Roland Siegwart et al. (Free e-Resource)
4. "Arduino Programming in 24 Hours" by Richard Blum, Sams Teach Yourself
5. "Springer Handbook of Robotics" edited by Bruno Siciliano and Oussama Khatib.

SEMESTER-IV

COURSE 10: ROBOTICS PRINCIPLES AND EMBEDDED SYSTEMS

Practical **Credits: 1** **2 hrs/week**

Note: This lab has to be completed using Arduino Micro controller and Arduino programming.

1. Introduction to Arduino and Basic Programming
   - Setting up Arduino IDE
   - Blinking an LED — basic digital output
2. Reading Digital Inputs
   - Interfacing and reading button presses/switches
3. Interfacing Analog Sensors
   - Reading values from a potentiometer or light-dependent resistor (LDR)
4. Ultrasonic Distance Sensor Interfacing
   - Measuring distance using HC-SR04 sensor and displaying data
5. Servo Motor Control
   - Controlling servo motor angles for robotic joint simulation
6. DC Motor and Motor Driver Interfacing
   - Running a DC motor with L298N motor driver
7. Stepper Motor Control
   - Basic control of stepper motors for precise movements
8. Line Following Robot Implementation
   - Using IR sensors to detect and follow a line on the ground
9. Obstacle Avoidance Robot
   - Using ultrasonic sensors and motor control to avoid obstacles
10. Force/Tactile Sensor Interfacing (including Piezo sensors)
    - Reading piezoelectric sensor signals as force or vibration inputs
11. Temperature Sensor Interfacing
    - Using sensors like LM35 or DHT11 for environmental sensing
12. Bluetooth Module (HC-05) Controlled Robot
    - Controlling robot movement wirelessly using smartphone commands
13. Servo-Driven Robotic Gripper
    - Building and controlling a simple gripper mechanism
14. PWM Motor Speed Control
    - Controlling motor speed using Pulse Width Modulation techniques