UNIVERSITY SCHOOL
OF
INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAMME STRUCTURE

M.TECH. COMPUTER SCIENCE AND ENGINEERING

SPECILIZATION : DATA SCIENCE

2025-2027

GAUTAM BUDDHA UNIVERSITY

GAUTAM BUDH NAGAR, GREATER NOIDA, UP, INDIA

SEMESTER I

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CS521	Advanced Data Base Management System	3	0	0	3	CC1
2	CS523	Advanced Data Structure and Algorithm	3	1	0	4	CC2
3	CS525	Research Techniques in ICT	3	0	0	3	CC3
4	CS527	Python Programming	3	0	0	3	CC4
5	ES415	Energy and Environment	3	0	0	3	OE1 /AECC
6	CD581	Advanced Data Base Management System Lab	0	0	4	2	CC-L1
7	CD583	Python Programming Lab	0	0	4	2	CC-L2 / SEC
8	GP	General Proficiency		Non Credit			
		Total Hours and Credits	**15**	**1**	**8**	**20**	

SEMESTER II

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CD522	Statistical Foundations for Data Science	3	0	0	3	CC5/SEC
2	CS524	Deep Learning	3	0	0	3	CC6
3	CD526	MLOPs	3	0	0	3	CC7
5	CS528	Blockchain Technology	3	1	0	4	CC8
		Elective-1	3	0	0	3	E1 / DSE
8	CD582	Statistical Foundations for Data Science Lab	0	0	4	2	CC-L3 / SEC
9	CS584	Deep Learning Lab	0	0	4	2	CC-L4
10	GP	General Proficiency		Non Credit			
		Total Hours and Credits	**15**	**1**	**8**	**20**	

ELECTIVES FROM DCSE

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CD530	Natural Language Processing	3	0	0	3	E1
2	CD532	Data Acquisation and Production	3	0	0	3	
3	CD534	Predictive Analytics and Data Visualization	3	0	0	3	
4	CD536	Data Science for Business Analytics	3	0	0	3	
5	CS538	Parallel and High Performance Computing	3	0	0	3	

* Semester 1 is approved in 33rd BoS and Semester 2 is approved in 35th BoS.

SEMESTER 1

ADVANCED DATABASE MANAGEMENT SYSTEM			
Course Code:	**CS521**	**Course Credits:**	**3**
Course Category:	**CC1**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of database design
2 A general understanding of database ,design and dependency
3 Understanding of different types of databases
4 Knowledge of databases on the internet
5 Application on enhanced database

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of ER diagram and UML class diagram.
2 Ability to apply functionality and Normalization on relational database.
3 Understand and fetch data from object oriented, parallel and distributed databases.
4 Use XML and understand unstructured data
5 Implement concept and deduction of enhanced database on different applications

UNIT I INTRODUCTION TO DATABASE DESIGN

Entities, Attributes, Entity Sets, Relationships, Key Constraints, Participation Constraints, Weak Entities, UML Class Diagrams, Subclasses, Super classes, Inheritance, Specialization, Generalization, Constraints and Characteristics of Specialization and Generalization Hierarchies, Modeling of UNION Types Using Categories, Representing Specialization and Generalization In UML Class Diagrams, Data Abstraction, Knowledge Representation and Ontology Concepts.

UNIT II DATABASES DESIGN THEORY

Problems Caused by Redundancy, Decompositions, Problems Related to Decomposition, Reasoning About FD's, FIRST, SECOND, THIRD Normal Form, BCNF, Forth Normal Form, Lossless Join Decomposition, Dependency Preserving Decomposition, Schema Refinement in DataBase Design, Multi Valued Dependencies.

UNIT III OBJECT- ORIENTED, PARALLEL AND DISTRIBUTED DATABASES

Overview of Object-Oriented Concepts, Object Identity, Object Structure, Type Constructor, Encapsulation of Operations, Methods and Persistence; Architectures For Parallel Databases, Parallel Query Evaluation, Parallelizing Individual Operations, Sorting Joins, Distributed Database Concepts, Data Fragmentation, Replication and Allocation Techniques for Distributed Database Design, Query Processing in Distributed Databases, Concurrency Control and Recovery in Distributed Databases.

UNIT IV DATABASES ON THE WEB AND SEMI–STRUCTURED DATA

Web interface, XML, structure of XML data, querying XML data, storage of XML data, XML applications, semi-structured data model, indexes for text data.

UNIT V ENHANCED DATA MODELS FOR ADVANCED APPLICATIONS

Active database concepts, temporal database concepts, spatial databases: concept and architecture, deductive databases and query processing, mobile databases, Geographic Information Systems (GIS).

Text Books:

1. Elmasri and Navathe, Fundamentals of Database Systems,
2. Ramakrishnan and Gehrke, Database Management Systems,

References Books:

1. Korth, Silberschatz, Sudarshan, Database System Concepts,
2. Rob and Coronel, Database Systems: Design, Implementation and Management,
3. Date and Longman, Introduction to Database Systems,

ADVANCED DATA STRUCTURE AND ALGORITHM			
Course Code:	**CS523**	**Course Credits:**	**4**
Course Category:	**CC2**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 01**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Understand the appropriate data structures, ADT libraries, and use it to design algorithms for a specific problem.
2 Be capable of solving problems using abstraction techniques.
3 Be able to choose appropriate algorithms for a specific problem.
4 Be able to analyze algorithms in terms of their efficiency and correctness.
5 Understanding the recent developments in the area of algorithm design.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Design and analyze programming problem statements.
2 Choose appropriate data structures and algorithms for a specific problem.
3 Understand the necessary mathematical abstraction to solve problems.
4 Come up with analysis of efficiency and proofs of correctness.
5 Comprehend and select algorithm design approaches in a problem specific manner.

UNIT I INTRODUCTION

Review of Basic Concepts: Abstract data types, Data structures, Algorithms, Big-Oh, Small-Oh, Omega, Small-Omega and Theta Notations, finding time complexity of programs, **Recurrence Relations:** Solving Recurrence Relations, Substitution Method, Master Theorem.

UNIT II HASHING

Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing, Recent Trends in Hashing.

UNIT III TREES & GRAPH

Trees: Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees, Minimum Spanning Tree (MST), Kruskal's Algorithm and Prim's Algorithm, Applications to MST.

Graph: Graph, Breadth First Search, Depth First Search, Shortest path in edge-weighted case (Dijkstra's), Bellman Ford Algorithms, Topological Sorting.

UNIT IV SELECTED TOPICS

Strassen's Matrix Multiplication, Greedy method VS Dynamic Programming, Job sequencing with deadlines, Fractional Knapsack Problem, 0/1 Knapsack Problem, Travelling Salesman Problem, Huffman coding, Pre order, Post order, Inorder traversal, Postfix to infix notation, Infix to Postfix notation.

UNIT V LINEAR PROGRAMMING & RECENT TRENDS

Linear Programming: Geometry of the feasibility region and Simplex algorithm

NP-completeness: Examples, proof of NP-hardness and NP-completeness.

Recent Trends: Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.

Text Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. Algorithms Unlocked: Thomas H. Cormen.
3. The Algorithm Design Manual, Steven S. Skiena.

References Books:

1. Algorithms: Robert Sedgewick and Kevin Wayne.
2. Advanced Data Structures: Peter Brass.

RESEARCH TECHNIQUES IN ICT			
Course Code:	**CS 525**	**Course Credits:**	**3**
Course Category:	**CC3**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To get the understanding about what is research.			
2 To have the knowledge about research methodology.			
3 Awareness of methodology to be followed for report and paper writing.			
4 Familiarization to different models and algorithms during research.			
5 Conversant with various simulation and soft computing techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Acquainted with procedure for carrying out the research and its importance.			
2 Can identify and apply the appropriate research techniques according to domain of research.			
3 Recognize models and algorithms required for their study.			
4 Know about research paper writing and publication procedure.			
5 Will get the direction about different simulation packages.			

UNIT I INTRODUCTION TO RESEARCH TECHNIQUES

Meaning of research, objectives of research, motivation in research, types of research, characteristics and prerequisites of research, significance of research, research process, sources of research problem, criteria of identifying the problem, necessity of defining the problem, errors in selecting research problem, technique involved in defining the problem, report and paper writing.

UNIT II DATA ANALYSIS AND STATISTICAL TECHNIQUES

Data and their analyses, quantitative methods and techniques, Measure of central tendency, measures of variation, frequency distribution, analysis of variance, methods, Correlation analysis, regression analysis, time series and forecasting, introduction to discriminant analysis, factor analysis, cluster analysis, conjoint analysis, probability distribution, binomial distribution, poisson distribution, uniform distribution, exponential distribution, and normal distribution, sampling methods, test of hypothesis.

UNIT III MATHEMATICAL MODELING

Steps of modeling, operations research models like queuing theory, stochastic processes, application of models, conceptual framework development and validation techniques, optimization techniques.

UNIT IV ALGORITHMIC RESEARCH
Algorithmic research problems, types of algorithmic research, types of solution procedure, steps of algorithm development, steps of algorithmic research, design of experiments.

UNIT V SIMULATION AND SOFT COMPUTING TECHNIQUES
Introduction to soft computing, artificial neural network, genetic algorithm, fuzzy logic and their applications, tools of soft computing, need for simulation, types of simulation, simulation language, fitting the problem to simulation study, simulation models, output analysis, data simulation packages like MATLAB, NS2, ANSYS, Cadence.

Text books:
1. Research Methodology: Methods and Techniques, C.R. Kothari

Reference Books:
1. Research Methodologies, R. Panneerselvam, Prentice Hall, 2007.
2. Research in Education, Best John V. and James V Kahn, Wiley eastern, 2005.
3. Elements of Educational Research, Sukhia, S.P., P.V. Mehrotra, and R.N. Mehrotra, PHI publication, 2003.
4. Methodology of Research Education, K. Setia, IEEE publication, 2004.
5. Research methodology, Methods and Techniques, Kothari, C.R., 2000.

PYTHON PROGRAMMING			
Course Code:	CS527	Course Credits:	3
Course Category:	CC4	Course (U / P)	P
Course Year (U / P):	1P	Course Semester (U / P):	2P
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3

COURSE OBJECTIVES

1. To learn and understand Python programming basics and paradigm
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling and database connectivity.
5. To learn and understand database connectivity in the python programming language.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. To read and write simple Python programs.
2. To develop Python programs with conditionals and loops.
3. To define Python functions and to use Python data structures –- lists, tuples, dictionaries
4. To do input/output with files in Python
5. To do searching ,sorting and merging in Python

UNIT I Introduction:

The Programming Cycle for Python , Python IDE, Interacting with Python Programs, Elements of Python, Type Conversion. Basics: Expressions, Assignment Statement, Arithmetic Operators, Operator Precedence, Boolean Expression.

UNIT II Conditionals:

Conditional statement in Python (if-else statement, its working and execution), Nested-if statement and Elif statement in Python, Expression Evaluation & Float Representation. Loops: Purpose and working of loops , While loop including its working, For Loop , Nested Loops , Break and Continue.

UNIT III Function:

Parts of A Function , Execution of A Function , Keyword and Default Arguments ,Scope Rules. Strings : Length of the string and perform Concatenation and Repeat operations in it. Indexing and Slicing of Strings. Python Data Structure : Tuples , Unpacking Sequences , Lists , Mutable Sequences, List Comprehension, Sets, Dictionaries Higher Order Functions: Treat functions as first-class Objects, Lambda Expressions

UNIT IV File I/O :

File input and output operations in Python Programming Exceptions and Assertions Modules : Introduction, Importing Modules, Abstract Data Types: Abstract data types and ADT interface in Python Programming. Classes : Class definition and other operations in the classes , Special Methods (such as _init_, _str_, comparison methods and Arithmetic methods etc.) , Class Example, Inheritance, Inheritance and OOP.

UNIT V Iterators & Recursion:

Recursive Fibonacci , Tower Of Hanoi Search : Simple Search and Estimating Search Time , Binary Search and Estimating Binary Search Time Sorting & Merging: Selection Sort , Merge List , Merge Sort , Higher Order Sort

Text Books:

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist'', 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016

2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.

ENERGY AND ENVIRONMENT			
Course Code:	**ES415**	**Course Credits:**	3
Course Category:	**OE1**	**Course (U / P)**	P
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**40+00**	**End Sem. Exam Hours:**	3
COURSE OBJECTIVES			
To provide in-depth knowledge of renewable and non-renewable energy resources, their harnessing techniques and energy-environment issues.			
COURSE OUTCOMES			
The knowledge so gathered could be utilized to meet the challenges of energy vis-a-vis environmental security.			

Sun as Source of Energy- Nature of its radiation, solar radiation and its spectral characteristics; Conventional energy sources (coal, oil, biomass and natural gas), Non-conventional energy sources (hydro-electric power, tidal, wind, geothermal, solar, nuclear magneto-hydrodynamic power MHD); Energy use pattern in India and parts of world, Energy security

Fossil Fuels: Classification, composition, physico-chemical characteristics; Calorific value – gross and net; Energy content of coal, petroleum and natural gas, shale oil, coal bed methane, gas hydrates

Concept of Green Energy; Principles of generation of hydro-power, tidal energy, ocean thermal energy conversion, wind power, geothermal energy, solar energy (solar collectors, photo-voltaic modules, solar ponds)

Nuclear Energy - Fission and fusion, Nuclear fuels, Nuclear reactor - principles and types; Mechanism of radiation action on living organisms - Stochastic and Non-stochastic effects, delayed effects; Radioactivity from nuclear reactors, fuel processing and radioactive waste, hazards related to power plants

Bioenergy: Types, importance, methods of energy production from biomass

Environmental Implications of Energy Use; CO_2 emission and atmosphere -scenario in developed and developing world (and India), Global warming, Radiative forcing, Impacts of large scale exploitation of solar, wind, hydro, nuclear and bio-energy sources; National Solar Mission, National Mission for Enhanced Energy Efficiency, case studies.

Text Books:

1. Fay, J.A. and Golomb, D.S. 2011. Energy and the Environment, Oxford University Press, New Delhi.
2. Iqbal, M. 1983. An Introduction to Solar Radiation. Academic Press, New York.
3. Kaushika, N.D. and Kaushik, K. 2004. Energy, Ecology and Environment: A Technological

ADVANCED DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**CS581**	**Course Credits:**	**2**
Course Category:	**CC-L1**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
. To explore the features of a Database Management Systems			
. To interface a database with front end tools			
. To understand the internals of a database system			
. To provide a strong foundation in advanced database concepts from an industry perspective.			
. To learn query processing and transaction management concepts for object-relational database and distributed database			
COURSE OUTCOME			
At the end of the course the students should be able to:			
1. Develop and apply critical thinking skills.			
2. Design and present Lab as well as project reports			
3. Apply appropriate methods for the analysis of raw data			
4. Perform logical troubleshooting as and when required.			
5. Verify and implement the concepts and theory learnt in class.			

List of Experiments:

1. Introduction to MySQL, PostgreSql, Microsoft Sqlsoftwares.
2. An exercise of data types in PostGresql& Data Definition Language Commands
3. Exercise on Data Manipulation Language and Transaction Control Commands using PostgreSql.
4. Exercise on Types of Data Constraints using PostgreSql.
5. Exercise on JOINS (Single-Table) Using Normalization
6. Exercise on JOINS (Multiple-Table) Using Normalization
7. Exercise on GROUP BY/ORDER BY Clause and Date Arithmetic using PostgreSql.
8. Exercise on different Functions (Aggregate, Math and String)
9. Exercise on different types of sub queries
10. Procedures, View and Triggers

PYTHON PROGRAMMING LAB			
Course Code:	**CS583**	**Course Credits:**	2
Course Category:	**CC-L2**	**Course (U / P)**	U / P
Course Year (U / P):	1P	**Course Semester (U / P):**	2P
No. of Lab (Hrs/Week) / Total No. of Lab	03 / 10	**End Sem. Exam Hours:**	3

COURSE OBJECTIVES
1. Interpret the use of procedural statements like assignments, conditional statements, loops and function calls.
2. Infer the supported data structures like lists, dictionaries and tuples in Python
3. Illustrate the application of matrices and regular expressions in building the Python programs.
4.Discover the use of external modules in creating excel files and navigating the file systems.
5. Describe the need for Object-oriented programming concepts in Python.

COURSE OUTCOME
At the end of the course the students should be able to:
1 To write, test, and debug simple Python programs.
2 To implement Python programs with conditionals and loops.
3 Use functions for structuring Python programs.
4 Represent compound data using Python lists, tuples, dictionaries
5 Read and write data from/to files in Python.

List of Experiments:

1. Write a python program find the maximum of a list of numbers.
2. Write a python program to perform Matrix Multiplication.
3. Write a python program first n prime number
4. Write a python program selection sort.
5. Write a python program to compute the GCD of two numbers.
6. Write a python program to find the most frequent words in a text file.
7. Write a Python program to create a scientific calculator
8. Write a Python program to print all the Disarium numbers between 1 and 100.
9. Write a Python program to encrypt the text using Caesar Cipher technique. Display the encrypted text. Prompt the user for input and the shift pattern.
10. Write a Python program to construct a linked list. Prompt the user for input. Remove any duplicate numbers from the linked list.

SEMESTER 2

STATISTICAL FOUNDATIONS FOR DATA SCIENCE			
Course Code:	CD522	Course Credits:	3
Course Category:	CC	Course (U/ P)	IP/P
Course Year (U/ P):	4IP/1P	Course Semester (U/ P):	8IP/P
No. of Lectures + Tutorials (Hrs/Week):	3	Mid Sem. Exam Hours:	1.5
Total No. of Lectures (L +T):	45	End Sem. Exam Hours:	3

COURSE OBJECTIVES
1. Understand probability axioms and apply discrete (Binomial, Poisson) and continuous (Normal) distributions to analyse and summarize datasets.
2. Build and interpret simple and multiple linear regression models while evaluating their significance using SSR, MSE, and R-squared metrics.
3. Implement Gradient Descent and Stochastic Gradient Descent techniques to efficiently estimate intercepts and slopes in regression problems.
4. Develop classification models using Logistic Regression and Naive Bayes (Gaussian and Multinomial) while comparing their strengths and limitations.
5. Assess model effectiveness using performance metrics such as confusion matrices, sensitivity, specificity, and ROC curves for data-driven decision-making.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Apply probability concepts including axioms, conditional probability, and independence to solve uncertainty-based data science problems.
2. Analyse and summarize datasets using measures of central tendency, histograms, and identify suitable discrete and continuous probability distributions such as Binomial, Poisson, and Normal distribution.
3. Build and interpret regression models (simple and multiple linear regression) and evaluate them using SSR, MSE, R^2, and statistical significance using p-values.
4. Implement optimization techniques such as gradient descent and stochastic gradient descent for estimating model parameters in regression problems.
5. Develop classification models using Logistic Regression and Naive Bayes approaches, and compare their assumptions, strengths, and limitations.
6. Evaluate model performance using confusion matrix and metrics like sensitivity, specificity, precision, recall, ROC curve, and Precision–Recall curve to support data-driven decisions.

Unit 1: Probability: Concept of Uncertainty, Axioms and rules of probability, Conditional probability and independence. Measures of central tendency: Mean, Median, Mode, Histograms, Calculating Probabilities Step-by-Step; Probability Distributions, Discrete Probability Distributions: Binomial Distribution, Poisson Distribution, Continuous Probability Distributions: Normal (Gaussian) Distribution and other Continuous Probability Distributions.

Unit 2:

Sum of the Squared Residuals: Residuals, The Sum of Squared Residuals (SSR); Mean Squared Error (MSE), R squared, Calculating R^2 with the Mean Squared Error (MSE), p-values. Linear Regression, Fitting a Line to Data, p-values for Linear Regression and R2, Multiple Linear Regression, Beyond Linear Regression.

Unit 3: Gradient Descent: Gradient Descent, Estimates the intercept and the slope, Residual, SSR; Gradient Descent for One Parameter, Gradient Descent Optimizing for Two or More Parameters, Multiple (Partial) Derivatives of the SSR, Gradient Descent for Two Parameters, Stochastic Gradient Descent.

Unit 4: Logistic Regression, Logistic classification, Probability vs. Likelihood, Logistic Regression: Weaknesses.

Naive Bayes, Multinomial Naive Bayes, Multinomial Naive Bayes vs. Gaussian Naive Bayes, Gaussian Naive Bayes.

Unit 5: Assessing Model Performance: Confusion Matrices, Sensitivity and Specificity, Precision and Recall, True Positive Rate and False Positive Rate, Receiver Operator Curves (ROCs), Precision Recall Graphs.

Text Books:

1. Walpole, Ronald E., Probability and Statistics for Engineers and Scientists
2. D.C. Agarwal, Statistics for Data Science and AI
3. Larry J. Stephens, Excel Data Analysis: Your visual blueprint for analysing data, statistics, and AI

DEEP LEARNING			
Course Code:	CS524	**Course Credits:**	3
Course Category:	CC	**Course (U / P)**	IP / P
Course Year (U / P):	4IP / 1P	**Course Semester (U / P):**	8IP / 2P
No. of Lectures + Tutorials (Hrs/Week):	3	**Mid Sem. Exam Hours:**	1.5
Total No. of Lectures (L + T):	45	**End Sem. Exam Hours:**	3

COURSE OBJECTIVES

1. Gain a historical perspective of Deep Learning and its foundations. Become familiar with basic principles of AI toward problem-solving, inference, perception, knowledge representation and learning.
2. Experience Deep Learning Algorithms and development various AI tools.
3. Explore the current scope, potential, limitations, and implications of Deep learning.
4. Analysis on the effect of deep learning practices in software systems.
5. Evaluation of deep learning algorithms

COURSE OUTCOME

At the end of the course the students should be able to:

1. Demonstrate knowledge of the building blocks of Deep Learning system.
2. Analyze and formalize the problem as a state space, graph, design heuristics, and select different search or game-based techniques to solve them.
3. Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing
4. Attain the capability to represent various real-life problem domains using logic-based techniques and use this to perform inference or planning.
5. To evaluate and compare deep learning models across different application domain.

UNIT I INTRODUCTION TO DEEP LEARNING

Introduction to machine learning - Linear models (SVMs and Perceptron's, logistic regression)- Introduction to Neural Nets: What are a shallow network computes- Training a network: loss functions, back propagation and stochastic gradient descent- Neural networks as universal function approximates.

UNIT II INTRODUCTION TO DEEP LEARNING

History of Deep Learning- A Probabilistic Theory of Deep Learning- Back propagation and regularization, batch normalization- VC Dimension and Neural Nets-Deep Vs Shallow Networks Convolutional Networks- Generative Adversarial Networks (GAN), Semi-supervised Learning.

UNIT III DIMENTIONALITY REDUCTION

Linear (PCA, LDA) and manifolds, metric learning - Auto encoders and dimensionality reduction in networks - Introduction to Convnet - Architectures – AlexNet, VGG, Inception, ResNet - Training a Convnet: weights initialization, batch normalization, hyper parameter optimization.

UNIT IV OPTIMIZATION AND GENERALIZATION IN DEEP LEARNING

Optimization in deep learning– Non-convex optimization for deep networks- Stochastic Optimization Generalization in neural networks- Spatial Transformer Networks- Recurrent networks, LSTM Recurrent Neural Network Language Models- Word-Level RNNs & Deep Reinforcement Learning - Computational & Artificial Neuroscience.

UNIT V APPLICATIONS OF DEEP LEARNING

Image-net- Detection-Audio WaveNet-Natural Language Processing Word2Vec - Joint Detection Bio-Informatics- Face Recognition- Scene Understanding- Gathering Image Captions.

Textbooks:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
2. Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
3. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

MLOPs			
Course Code:	**CD526**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P):**	**IP/P**
Course Year (U / P):	**4IP / 1P**	**Course Semester (U / P):**	**8IP / 2P**
No. of Lectures + Tutorials (Hrs/Week):	**03**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 To understand the fundamentals of MLOps.
2 To learn end-to-end ML pipeline design and deployment.
3 To gain knowledge of model monitoring and maintenance.
4 To work with popular MLOps tools and frameworks.
5 To apply MLOps practices for scalable ML solutions.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Explain the ML model lifecycle in production.
2 Build and automate ML pipelines.
3 Deploy ML models using MLOps tools.
4 Monitor and optimize deployed models.
5 Develop scalable and reliable ML systems.

UNIT 1: INTRODUCTION TO MLOPS

Overview of MLOps and its significance: Key challenges in deploying and managing ML models in production, Comparison of traditional software development, DevOps and MLOps, Key components of MLOps, MLOps workflow, Landscape of MLOps tools and technologies.

UNIT II: ML PIPELINES & DATA MANAGEMENT

Overview of data engineering tools and practices, Data management for ML models and Feature Stores. Motivation, role in ML and Generative AI applications, benefits for MLOps, ML pipeline automation.

UNIT III: CI/CD FOR ML MODELS

Use of version control systems like Git for model development, automated testing & validation, model delivery strategies

UNIT IV: MONITORING AND PERFORMANCE OPTIMIZATION

Techniques for monitoring model performance in production, Logging and error tracking for ML systems, Performance optimization and scaling strategies

UNIT V: ML SECURITY AND ML GOVERNANCE

Overview of Security considerations for ML, field of MLSecOps, tooling options. ML Governance: Overview of model explainability and ethical considerations in ML deployments, tracking bias

Text Books:

1. Building Machine Learning Powered Applications: Going from Idea to Product - Emmanuel Ameisen - O'Reilly publication
2. Reliable Machine Learning: Applying SRE Principles to ML in Production - Chen, Murphy, Parisa - O'Reilly publication
3. Machine Learning Engineering in Action - Ben Wilson - O'Reilly publication
4. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems - Martin Kleppmann - O'Reilly publication

BLOCKCHAIN TECHNOLOGY			
Course Code:	**CS528**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**IP / P**
Course Year (U / P):	**4IP / 1P**	**Course Semester (U / P):**	**8IP / 2P**
No. of Lectures + Tutorials (Hrs/Week):	**3 +1**	**Mid Sem. Exam Hours:**	**1.5**
Total No. of Lectures (L + T):	**45+15**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To get acquainted with the concept of Distributed ledger system and Blockchain.			
2. To learn the concepts of consensus and mining in Blockchain through the Bitcoin network.			
3. To understand Ethereum and develop-deploy smart contracts using different tools and frameworks.			
4. understand permissioned Blockchain and explore Hyperledger Fabric.			
5. To understand different types of crypto assets.			
COURSE OUTCOME			
At the end of the course the students should be able to:			
1. Describe the basic and fundamental concept of Blockchain and Distributed Ledger Technology.			
2. Interpret the knowledge of the Bitcoin network, nodes, keys, wallets and transactions			
3. Implement smart contracts in Ethereum using different development frameworks.			
4. Develop simple applications using Solidity language on Ethereum platform.			
5. Interpret different Crypto assets and Crypto currencies			

UNIT I FUNDAMENTALS OF BLOCKCHAIN TECHNOLOGY

Blockchain – Definition, architecture, elements of blockchain, benefits and limitations, types of blockchain. Consensus – definition, types, consensus in blockchain.

Decentralization – Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization.

UNIT II FUNDAMENTALS OF CRYPTOGRAPHY

Introduction to Cryptography, Symmetric cryptography – AES. Asymmetric cryptography – RSA. Elliptic curve cryptography, Digital signatures – RSA digital signature algorithms. Secure Hash Algorithms – SHA-256. Applications of cryptographic hash functions – Merkle trees, Distributed hash tables.

UNIT III CONSENSUS ALGORITHMS AND BITCOIN

Consensus Algorithms, Crash fault-tolerance (CFT) algorithms – Paxos, Raft. Byzantine fault-tolerance (BFT) algorithms – Practical Byzantine Fault Tolerance (PBFT), Proof of work (PoW), Proof of stake (PoS), Types of PoS. Bitcoin – Definition, Cryptographic keys – Private keys, public keys, addresses. Transactions – Lifecycle, coinbase transactions, transaction validation. Blockchain – The genesis block. Mining – Tasks of miners, mining algorithm, hash rate. Wallets – Types of wallets.

UNIT IV SMART CONTRACTS AND USE CASES

Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts. Decentralization terminology – Decentralized applications, Decentralized Autonomous Organizations. Use cases of Blockchain technology – Government, Health care, Finance, Supply chain management. Blockchain and allied technologies – Blockchain and Cloud Computing, Blockchain and Artificial Intelligence.

UNIT V ETHEREUM AND SOLIDITY

Ethereum – The Ethereum network. Components of the Ethereum ecosystem – Keys and addresses, Accounts, Transactions and messages. The Ethereum Virtual Machine, Blocks and blockchain. The Solidity language – The layout of a Solidity source code, Structure of a smart contract, variables, data types, control structures, events, inheritance, libraries, functions, error handling. Smart contracts Case study: Voting, Auction.

Text Book

1. Imran Bashir, Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, Packt Publishing, Third edition, 2020. Mastering Bitcoin Unlocking Digital Cryptocurrencies, Andreas M. Antonopoulos, O'Reilly Media

Reference Books:

1. Ritesh Modi, Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain, Packt Publishing, First edition, 2018.

2. Kumar Saurabh, Ashutosh Saxena, Blockchain Technology: Concepts and Applications, First Edition, Wiley Publications, First edition, 2020.

3. Chandramouli Subramanian, Asha A George, et al, Blockchain Technology, Universities Press (India) Pvt. Ltd, First edition, August 2020.

5. Blockchain Technology: Concepts and Applications, Kumar Saurabh and Ashutosh Saxena, Wiley.

Statistical Foundations for Data Science			
Course Code:	CD582	Course Credit:	2
Course Category:	CC-L	Course (U/ P)	IP/P
Course Year (U/ P):	4IP/1P	Course Semester (U/ P):	8IP/2P
No. of Lectures + Tutorials (Hrs/Week):	4	Mid Sem. Exam Hours:	NIL
Total No. of Lectures (L +T):	10	End Sem. Exam Hours:	2

COURSE OBJECTIVES
1. To analyse datasets using descriptive statistics (mean, median, mode) and evaluate the impact of outliers on data summaries.
2. To calculate conditional probabilities and simulate discrete and continuous statistical distributions (Binomial, Poisson, and Normal) using Python libraries.
3. To implement and interpret Simple and Multiple Linear Regression models to predict values and calculate error metrics like SSR and MSE.
4. To understand the iterative process of parameter optimization by manually implementing the Gradient Descent algorithm.
5. To develop classification models using Logistic Regression and Naive Bayes algorithms and assess their performance through evaluation metrics like Confusion Matrix, ROC curves, and F1-score.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Use Python to summarize datasets and interpret the robustness of statistical measures against outliers.
2. Binomial, Poisson, and Normal distributions to solve probability-based problems.
3. Develop predictive models using Linear Regression and analyze coefficients and R-squared values to determine model fit.
4. Apply Gradient Descent to optimize cost functions and compare manually tuned parameters with standard library results.
5. Execute and evaluate classification algorithms using confusion matrices, ROC curves, and F1-scores to measure model precision and recall.

Experiment 1: To compute Mean, Median, Mode and study how outliers affect mean and median.

Theory

- Mean: $x^{-} = \frac{\Sigma x_i}{n}$
- Median: middle value after sorting
- Mode: most frequent value
- Mean is **highly affected** by outliers; median is **more robust**.

Algorithm

1. Create a list of marks.
2. Compute mean, median, mode.
3. Add outliers (very low or very high marks).
4. Recompute mean, median, mode.
5. Compare "before vs after" and visualize with histogram.

Experiment 2: A university collected data about students' exam results and gender as shown below:

Result	**Gender**
Pass	Male
Fail	Male
Pass	Female
Pass	Male
Fail	Female
Pass	Female
Pass	Male
Fail	Male

Write a Python program to calculate the conditional probability that a student has passed the exam given that the student is female.

That is, compute:

$$P(Pass|Female)$$

$$P(Pass|Female) = \frac{P(Pass \cap Female)}{P(Female)}$$

Using the same dataset, write a Python program to calculate the conditional probability that a student has passed the exam given that the student is male.

That is, compute:

$$P(Pass|Male)$$

Experiment 3: Binomial and Poisson Distribution

To simulate and visualize Binomial and Poisson distributions using Python.

Tasks: Simulate coin toss experiment (Binomial), Simulate number of calls per minute (Poisson), Plot histograms, Compare theoretical and simulated mean, Tools: numpy, matplotlib

Experiment 4: Normal Distribution and Z-Score

To study Gaussian distribution and calculate probabilities using Z-score: Generate dataset following normal distribution, Plot histogram with normal curve, Compute mean, standard deviation, Calculate Z-score and probability, Tools: numpy, scipy, matplotlib

Experiment 5: Simple Linear Regression from Scratch

To implement simple linear regression manually: Create dataset (hours studied vs marks), Compute slope and intercept, predict values, Calculate SSR and MSE, Plot regression line, Tools: numpy, matplotlib

Experiment 6: Multiple Linear Regression

To implement multiple linear regression using sklearn: Load dataset, Train regression model, Compute R^2 and MSE,Tools: pandas, sklearn

Experiment 7: Gradient Descent Implementation

To implement gradient descent for linear regression: Define cost function, Update parameters iteratively, Plot cost vs iterations, Compare with sklearn result, Tools: numpy, matplotlib

Experiment 8: Logistic Regression Classification

To build a logistic regression classifier, Load dataset, Train logistic regression model, predict class labels, Evaluate accuracy

Experiment 9: Naive Bayes Classification

To implement Gaussian and Multinomial Naive Bayes, Train Naive Bayes model, Compare Gaussian vs Multinomial, Evaluate accuracy

Experiment 10: Model Evaluation Metrics

To evaluate classification model performance: Generate confusion matrix, Compute precision, recall, F1-score, Plot ROC curve and Precision-Recall curve, Tools: sklearn, matplotlib

DEEP LEARNING LAB			
Course Code:	**CS584**	**Course Credits:**	2
Course Category:	**CC-L**	**Course (U / P)**	**IP / P**
Course Year (U / P):	**4 IP / 1P**	**Course Semester (U / P):**	**8IP / 2P**
No. of Labs(Hrs/Week):	**04**	**Mid Sem. Exam Hours:**	**NIL**
Total No. of Labs	**10**	**End Sem. Exam Hours:**	**2**

COURSE OBJECTIVES

1. Provide hands-on experience with deep learning frameworks and tools.
2. Implement and experiment with neural network models for real-world problems
3. Understand training, optimization, and evaluation of deep learning models.
4. Apply deep learning techniques in domains such as computer vision, NLP, and time-series analysis.
5. Encourage research-oriented thinking and reproducible experimentation.

COURSE OUTCOMES

At the end of the course the students should be able to understand:

1. Implement basic to advanced neural network architectures using Python-based frameworks.
2. Train, tune, and evaluate deep learning models on benchmark datasets.
3. Apply CNNs, RNNs, and Transformer-based models to domain-specific problems.
4. Analyze model performance using appropriate metrics and visualization tools
5. Develop mini-projects and research prototypes using deep learning techniques.

Software and Tools:

- Programming Language: Python
- Frameworks/Libraries: TensorFlow, Keras, PyTorch
- Supporting Libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, OpenCV, NLTK / SpaCy
- Platforms: Jupyter Notebook, Google Colab (optional)

List of Practical:

1. Python & DL Environment Setup: NumPy, Pandas, Matplotlib, TensorFlow/Keras installation, dataset preprocessing.
2. Single Layer Perceptron: Implementation from scratch and logical gate classification.
3. Multi-Layer Perceptron (MLP): Implementation from scratch and logical gate classification.
4. Backpropagation Algorithm: ANN model training and performance visualization.
5. Activation & Loss Functions: Comparison of Sigmoid, ReLU, Tanh, Softmax, MSE, Cross-Entropy.
6. Convolutional Neural Network (CNN): MNIST digit classification.
7. Image Classification using CNN: CIFAR-10 dataset with Dropout and Batch Normalization.
8. Sequence modeling or text classification using RNN.
9. Hyperparameter Tuning: Learning rate and optimizer comparison (SGD, Adam, RMSprop).
10. Recurrent Neural Network (RNN): Sequence prediction implementation.
11. LSTM Implementation: Time series forecasting and future value prediction.
12. Autoencoder: Dimensionality reduction and reconstruction analysis.
13. Transfer Learning: Fine-tuning pre-trained models (ResNet) on a custom dataset.

ELECTIVES

ELECTIVES

Natural Language Processing			
Course Code:	CD530	**Course Credits:**	3
Course Category:	E1	**Course (U / P)**	IP/P
Course Year (U / P):	4IP/2P	**Course Semester (U / P):**	8IP/2P
No. of Lectures (Hrs/Week):	03	**Mid Sem. Exam Hours:**	1.5

COURSE OBJECTIVES
1. To learn the fundamentals of Natural language Processing and its algorithm
2. To understand machine translation and applications of NLP
3. Basic understanding of deep learning models for NLP
4. To understand and implement deep learning architectures like RNN, LSTM, and GRU for solving NLP tasks
5. To explore speech processing, text-to-speech, multimodal language systems, and ethical aspects of speech and language technologies.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Learn the fundamentals of Natural language Processing and its algorithm
2. Understand machine translation and applications of NLP
3. Provide basic understanding of deep learning models for NLP
4. Apply the concept of NLP in the real domain.
5. Implement basic speech recognition and multimodal NLP techniques and address ethical concerns in voice-based AI systems

UNIT -I Fundamentals of Natural Language Processing
Introduction to NLP: Characteristics of Natural Language, Language structure, Sentence Structure, Language analyzer, Lexicon, word formation, Morphology, syntax analysis (parsing), semantics, ambiguity, pragmatics and discourse.

UNIT- II Text Processing and Representation Techniques
NLP Algorithms: Understanding Corpus and data attributes, Corpus Formats CSV, JSON, XML, LibSVM, Operations on Text Corpus, Tokenisation, stop words, Term Frequency Inverse Document Frequency (TF-IDF), Text Analysis and word embedding using word2vec, doc2vec, GLoVe, Bag-of- words (BoW).

UNIT-III Machine Translation and NLP Applications
Machine Translation and Applications of NLP: Introduction to Machine Translation (MT), Approaches, Structure of Anusaraka: an Interlingua based MT system, Example/Analogy based MT, Word/phrase based MT, Neural MT. Applications of NLP: Sentiment analysis, chatbots, conversational models (Question Answering system) for Digital Assistants

UNIT- IV Deep Learning for Natural Language Processing
Deep learning models for NLP: Neural Net based NLP models: Study of Convolutional Neural Network(CNN), Recurrent Neural Network(RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit(GRU) using Natural Language Toolkit (NLTK)

UNIT V: Speech and Multimodal Processing in NLP

Fundamentals of Speech Recognition, Speech Synthesis and Text-to-Speech (TTS), Audio signal processing for NLP, Speaker identification and verification, Multimodal NLP: Text, Speech, and Vision Integration, NLP for low-resource languages, Ethical considerations in speech and language technologies

Text Books:

1. Daniel Jurafsky, James H. Martin,"Speech and Language Processing: An Introduction to
   Natural Language Processing", Computational Linguistics and Speech, Pearson Publication, 2014.
2. Thanaki, Jalaj, "Python natural language processing". Packet Publishing Ltd, 2017.

DATA ACQUISITION AND PRODUCTION			
Course Code:	CD532	Course Credits:	3
Course Category:	E1	Course (U / P)	IP / P
Course Year (U / P):	4 IP / 1P	Course Semester (U / P):	8IP / 2P
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1.5
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3

COURSE OBJECTIVES

1. Provide strong foundations in data acquisition pipelines
2. Enable students to collect, ingest, validate, and produce data at scale
3. Train students on industry-standard tools and architectures
4. Train students on industry-standard tools and architectures
5. Prepare students for data engineering, data science, ML research, and PhD-level work

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Design efficient data acquisition architectures
2. Collect structured, semi-structured, and unstructured data
3. Build scalable ETL/ELT pipelines
4. Ensure data quality, governance, and reproducibility
5. Produce analytics-ready and ML-ready datasets

UNIT 1 DATA ACQUISITION FUNDAMENTALS & METHODS

Data lifecycle management; Data granularity and sampling techniques (Simple Random, Stratified); Sources of Data: Web Scraping and Crawling (BeautifulSoup, Scrapy, Selenium); Interacting with Web APIs (REST, GraphQL, Rate limiting, Authentication); Log files parsing; Sensor/IoT data ingestion (MQTT basics); Open Data Standards; Legal and Ethical aspects of data acquisition (GDPR, bias in data collection).

UNIT II DATA INGESTION & STORAGE ARCHITECTURES

Ingestion: Batch vs. Real-time streaming ingestion; Message Queues (Apache Kafka, RabbitMQ) for decoupling data producers; Storage: Polyglot persistence; SQL (Relational) vs. NoSQL (Document, Columnar, Key-Value, Graph); Time-series databases (InfluxDB/Prometheus); Concept of Data Lakes vs. Data Warehouses; HDFS and Cloud Storage principles (S3/GCS buckets).

UNIT III DATA PROCESSING & TRANSFORMATION (ETL/ELT)

Data Wrangling; Handling missing data, outliers, and inconsistencies; Data Integration patterns; ETL Pipelines: Constructing Extract-Transform-Load pipelines; Orchestration tools (Apache Airflow/Prefect); Data Serialization formats (JSON, Avro, Parquet, Protobuf); Normalization and Denormalization strategies for analytics; Feature Engineering pipeline (Automated feature extraction).

UNIT IV BIG DATA PROCESSING & DISTRIBUTED SYSTEMS

Introduction to Distributed Computing (MapReduce paradigm); Apache Spark for data processing (Resilient Distributed Datasets, DataFrames); Stream processing (Spark Streaming/Flink); Data ineage and Provenance; Scalability challenges in data production; CAP Theorem and consistency nodels in distributed data production.

UNIT V DATA PRODUCTION, QUALITY & GOVERNANCE

Data Production: Serving data as APIs (FastAPI/Flask); Data Version Control (DVC); Metadata management; Data Quality: Dimensions of data quality (Accuracy, Completeness, Consistency); automated data validation (Great Expectations); Data Privacy techniques (Anonymization, Differential Privacy); Introduction to DataOps and MLOps workflows.

Textbooks:

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media. (Essential for Industry & Research)
2. Andreas François Vermeulen, Practical Data Science: A Guide to Building the Technology Stack for Turning Data Lakes into Business Assets, APress.
3. W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media.
4. Joe Reis & Matt Housley, Fundamentals of Data Engineering, O'Reilly Media.

PREDICTIVE ANALYTICS AND DATA VISUALIZATION			
Course Code:	CD534	Course Credits:	3
Course Category:	E1	Course (U/P):	IP / P
Course Year (U / P):	4 IP / 1P	Course Semester (U / P):	8IP / 2P
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1.5
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3

COURSE OBJECTIVES
1. To understand the theoretical foundations of supervised and unsupervised learning techniques used in predictive analytics.
2. To develop and evaluate predictive models while addressing issues such as bias–variance trade-off and model complexity.
3. To apply predictive analytics models to real-world datasets.
4. To comprehend the principles of data visualization including data preprocessing, visual encoding, semiology of graphics, and visualization pipelines.
5. To design effective visualization solutions for variety of data using appropriate visualization techniques and tools.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Analyze predictive analytics problems and identify suitable supervised or unsupervised learning techniques
2. Build, validate, and compare predictive models using regression, classification, ensemble methods, and model selection techniques
3. Implement machine learning algorithms such as SVMs, neural networks, trees, clustering, and PCA
4. Apply principles of data pre-processing and visualization foundations to explore and interpret datasets
5. Design advanced visualizations for spatial, multivariate, hierarchical, network, and text data using modern visualization techniques

UNIT – I INTRODUCTION

Overview of Supervised Learning, Regression and Classification, Predictive Framework, Linear Regression, Least Square, Multiple Regression, Subset selection, Linear Discriminant Analysis, Logistic Regression. *Model Assessment and Selection:* Bias, Variance, and model complexity, Bias-variance trade off, Optimism of the training error rate, Effective number of parameters, Bayesian approach and BIC, Cross- validation, Boot strap methods, conditional or expected test error.

UNIT – II PREDICTIVE ANALYTICS

K-nearest Neighbor, Support Vector Machines, Perceptron Learning. *Neural Networks:* Feed-forward Network, Backward Propagation. Regression and Classification trees. Ensemble methods: Bagging, Boosting and Stacking models, Random Forest. *Unsupervised Learning:* Association rules, Cluster analysis, Principal Component Analysis.

UNIT – III DATA VISUALIZATION

Data Foundation: Types of Data, Structure within and between Records, Data Preprocessing, Data Sets. *Visualization Foundation:* Visualization stages, Semiology of Graphical Symbols, Eight Visual Variables, Taxonomies, Visualization Process, Pseudo code Conventions, Scatter plot.

UNIT – IV VISUALIZATION TECHNIQUES

Spatial Data: One-Dimensional Data, Two-Dimensional Data, Three-Dimensional Data, Dynamic Data, Combining Techniques. *Geospatial Data:* Visualizing Spatial Data, Visualization of Point Data, Visualization of Line Data, Visualization of Area Data, Other Issues in Geospatial Data Visualization. *Multivariate Data:* Point-Based Techniques, Line- Based Techniques, Region-Based Techniques, Combinations of Techniques.

UNIT – V ADVANCED DATA VISUALIZATION

Trees Displaying: Hierarchical Structures. *Graphs and Networks:* Displaying Arbitrary Graphs/Networks. *Text and Document Visualization:* Introduction, Levels of Text Representations, The Vector Space Model, Single Document Visualizations, Document Collection Visualizations, Extended Text Visualizations. Introduction to advanced topics and libraries in data visualization.

Text Books:

1. Ivo D. Dinov, "Data Science and Predictive Analytics", Springer Cham, 2018.
2. Jack Dougherty and Ilya Ilyankou, "Hands-On Data Visualization: Interactive Storytelling from Spreadsheets to Code", O'Reilly Media, 2021.
3. Vijay Kotu and Bala Deshpande, "Predictive Analytics and Data Mining", Morgan Kaufmann, 2014.
4. Thomas W. Miller, "Modelling Techniques in Predictive Analytics with Python and R", Pearson Education, 2018.

DATA SCIENCE FOR BUSINESS ANALYTICS

Course Code:	CD 536	Course Credits:	3
Course Category:	E1	Course (U / P)	IP / P
Course Year (U / P):	4 IP / 1P	Course Semester (U / P):	8IP / 2P
No. of Lectures + Tutorials(Hrs./Week):	03 + 00	Mid Sem. Exam Hours:	1.5
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3

COURSE OBJECTIVES

1. To make students understand and make inferences based on relations found in the sample, to relations in the population.
2. To Understand basic concepts in Excel
3. Design effective data visualizations in order to provide new insights into a research question or communicate information to the viewer.
4. Find and select appropriate data that can be used to create a visualization that answers a particular research question.
5. For each individual statistical test students should be able to understand how it works, for what data and design it is appropriate and how results should be interpreted.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Understand the fundamentals of Data Science and Excel
2. Key concepts in data science, including tools, approaches, and application scenarios
3. State-of-the-art tools to build data-science applications for different types of data, including text and CSVdata
4. Work with frequency distribution, mean, covariance, serial correlation, multi-collinearity, conditional probability etc.
5. Analyse data using Sampling Distribution, t-distribution, F-distribution, Chi-Square distribution etc.

UNIT I INTRODUCTION

Introduction to Data Science, Evolution of Data Science, Data Science Roles, Stages in a Data Science Project, Applications of Data Science in various fields, Data Security Issues. A description of the purpose and application of Excel.

UNIT II DATA COLLECTION AND DATA PRE-PROCESSING

Data Collection Strategies, Data Pre-Processing Overview, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization, Techniques for sorting and filtering data, including controlling the order of precedence in a sort, advanced filters.

UNIT III EXPLORATORY DATA ANALYTICS

Math's basic addition, subtraction, multiplication, division, multi brackets, Powers, Rounding, data set. Totals and counts – SUM(), COUNT() and COUNTA(). Other statistics. Conditional totals and counts. Advanced conditional sums, SUMPRODUCT and array formulas, Change the date system, format, or two digit year interpretation, Descriptive Statistics Mean, Standard Deviation, Cell referencing and naming, Creating named ranges, managing named ranges, Calculations with named ranges, and Automating processes with named ranges.

UNIT IV MODEL DEVELOPMENT

Simple and Multiple Regression, Model Evaluation using Visualization, Residual Plot, Distribution Plot, Polynomial Regression and Pipelines, Measures for In-sample Evaluation, Prediction and Decision Making.

UNIT V MODEL EVALUATION

Generalization Error, Out of Sample Evaluation Metrics, Cross Validation, Over fitting, Under Fitting and Model Selection, Prediction by using Ridge Regression, Testing Multiple Parameters by using Grid Search.

Text Books:

1. Jojo Moolayil, "Smarter Decisions : The Intersection of IoT and Data Science", PACKT, 2016.
2. Cathy O'Neil and Rachel Schutt , "Doing Data Science", O'Reilly, 2015.
3. David Dietrich, Barry Heller, Beibei Yang, "Data Science and Big data Analytics", EMC 2013

PARALLEL AND HIGH PERFORMANCE COMPUTING			
Course Code:	CS538	Course Credits:	3
Course Category:	E1	Course (U/ P)	IP /P
Course Year (U / P):	4 IP / 1P	Course Semester (U / P):	8IP / 2P
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1.5
Total No. of Lectures (L +T):	45+ 00	End Sem. Exam Hours:	3

COURSE OBJECTIVES
1. Introduce the fundamental concepts of parallel and high-performance computing, including concurrency, parallelism, scalability, and performance limitations.
2. To develop the ability to analyse performance bottlenecks in parallel programs using profiling tools, memory hierarchy concepts, and efficiency metrics.
3. To enable students to identify parallelism in applications and design efficient parallel algorithms using CPU-based optimization and vectorization techniques.
4. To provide hands-on experience in developing and optimizing shared-memory parallel programs using the OpenMP programming model with appropriate scheduling and synchronization strategies.
5. To familiarize students with distributed-memory and accelerator-based computing by implementing parallel programs using MPI and GPU programming models and evaluating their scalability and performance.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Explain the fundamental concepts of parallel and high-performance computing, including concurrency, scalability, and performance limitations.
2. Analyse performance bottlenecks in parallel programs using profiling techniques, memory hierarchy concepts, and efficiency metrics.
3. Design parallel solutions by identifying parallelism in applications and applying suitable parallel algorithms and CPU-based optimization techniques.
4. Develop and optimize shared-memory parallel programs using OpenMP, applying scheduling, synchronization, and vectorization strategies for improved performance.
5. Implement distributed and accelerator-based parallel programs using MPI and GPU programming models, and evaluate their scalability and performance.

UNIT 1 INTRODUCTION TO PARALLEL AND HIGH PERFORMANCE COMPUTING

Need for parallel computing, real-world applications of parallel and high-performance computing, concurrency vs parallelism, scalability basics, Amdahl's Law and motivation for high-performance computing

UNIT II PERFORMANCE ANALYSIS

Performance bottlenecks in parallel programs, profiling techniques for performance analysis, performance limits and efficiency metrics, data layout and memory hierarchy, cache effects on program performance

IT III PARALLEL PROGRAM DESIGN AND CPU PARALLELISM

nning parallel computing projects, identifying parallel opportunities in programs, parallel orithms and common computation patterns, vectorization concepts, SIMD architecture and speedup

NIT IV SHARED MEMORY PROGRAMMING

oop transformations for performance optimization, compiler support for vectorization, OpenMP rogramming model, OpenMP performance optimization techniques, scheduling, reduction, ynchronization overhead

UNIT V DISTRIBUTED MEMORY PARALLELISM AND GPU COMPUTING

Practical multicore performance tuning, need for MPI and distributed high-performance computing, MPI programming basics, point-to-point communication, collective communication operations, performance considerations in MPI, communication overhead, scalability, GPU architecture and computational advantages, GPU programming model, GPU execution concepts

Text Books:

1. Parallel and High Performance Computing, Robert Robey and Yuliana Zamora, 2021.
2. Introduction to High Performance Computing for Scientists and Engineers, Georg Hager, Gerhard Wellein, 2011.
3. Introduction to parallel computing (2nd edition). Pearson Addison Wesley, 2003.