Savitribai Phule Pune University

(Formerly University of Pune)

Four Year Degree Program in Bachelor of Science (B.Sc)

With

Major: ARTIFICIAL INTELLIGENCE (AI) and MACHINE LEARNING (ML)

(Faculty of Science and Technology)

Syllabi for

(For Colleges Affiliated to Savitribai Phule Pune University)

Choice Based Credit System (CBCS) Syllabus
Under National Education Policy (NEP)

To be implemented from Academic Year 2024-2025

Syllabus Structure as per NEP Guidelines
B.Sc. (Artificial Intelligence & Machine Learning)
TY (Level 5.5) SEMESTER V (from the A.Y. 2026-27)

Course Type	Course code	Course Name	Credits		Teaching Scheme Hrs/Week		Examination Scheme and Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Mandatory	AIML-301-MJ-T	Artificial Intelligence - II	2	--	2	--	15	35	50
	AIML-302-MJ-T	Supervised Machine Learning Techniques	2	--	2	--	15	35	50
	AIML-303-MJ-T	Cybersecurity Concepts and Artificial Intelligence	2	--	2	--	15	35	50
	AIML-304-MJ-T	Prompt Engineering	2	--	2	--	15	35	50
	AIML-305-MJ-P	Practical based on AIML-301-MJ-T	--	2	--	4	15	35	50
	AIML-306-MJ-P	Practical based on AIML-302-MJ-T	--	2	--	4	15	35	50
Major Elective	AIML-310-MJ-T	Data Preprocessing and Visualization	2	--	2	--	15	35	50
	AIML-311-MJ-P	Practical based on AIML-310-MJ-T	--	2	--	4	15	35	50
	OR								
	AIML-312-MJ-T	Vibe Coding	2	--	2	--	15	35	50
	AIML-313-MJ-P	Practical based on AIML-312-MJ-T	--	2	--	4	15	35	50
	OR								
	AIML-314-MJ	DAA - II	2	--	2	--	15	35	50
	AIML-315-MJ-P	Practical based on AIML314MJ	--	2	--	4	15	35	50
VSC	AIML-321-VSC-P	Design Thinking	--	2	--	4	15	35	50
FP / OJT/ CEP	AIML-331-FP	Project	--	2	--	4	15	35	50
Minor	AIML-341-MN-T	Calculus for Machine Learning	2	--	2	--	15	35	50
		Total	**12**	**10**	**12**	**20**			**550**

Syllabus Structure as per NEP Guidelines
B.Sc. (Artificial Intelligence & Machine Learning)
TY (Level 5.5) SEMESTER VI (from the A. Y. from 2026-27)

Course Type	Course code	Course Name	Credits		Teaching Scheme Hrs/Week		Examination Scheme and Marks		
			TH	PR	TH	PR	CE	EE	Total
Major Mandatory	AIML-351-MJ-T	Introduction to NLP	2	--	2	--	15	35	50
	AIML-352-MJ-T	Unsupervised ML techniques	2	--	2	--	15	35	50
	AIML-353-MJ-T	Computer Vision	2	--	2	--	15	35	50
	AIML-354-MJ-T	Software Testing	2	--	2	--	15	35	50
	AIML-355-MJ-P	Practical based on AIML-351-MJ-T	--	2	--	4	15	35	50
	AIML-356-MJ-P	Practical based on AIML-352-MJ-T	--	2	--	4	15	35	50
Major Elective	AIML-360-MJ-T	Business Intelligence	2	--	2	--	15	35	50
	AIML-361-MJ-P	Practical based on AIML-360-MJ-T	--	2	--	4	15	35	50
	OR								
	AIML-362-MJ-T	AI assisted software Development	2	--	2	--	15	35	50
	AIML-363-MJ-P	Practical based on AIML-362-MJ-T	--	2	--	4	15	35	50
	OR								
	AIML-364-MJ-T	Game Theory	2	--	2	--	15	35	50
	AIML-365-MJ-P	Practical based on AIML-364-MJ-T	--	2	--	4	15	35	50
VSC	AIML-371-VSC-P	Database Technologies		2		4	15	35	50
FP / OJT/ CEP	AIML-381-OJT	On Job Training	--	4	--	8	30	70	100
		Total	**10**	**12**	**10**	**24**			**550**

Detailed Syllabus

B.Sc (AI & ML)

Semester V

(Third Year)

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-301-MJ-T: Artificial Intelligence - II

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Artificial Intelligence -I, Python Programming, OOP

Objectives:
The course is designed to teach:

- Bayes' rule to quantify uncertainty in real-world domains.
- Representation of uncertain knowledge using joint distributions, Naive Bayes models, and Bayesian networks, and to perform exact inference for decision support.
- implementation and interpretation of temporal models such as Hidden Markov Models.
- beliefs and preferences under uncertainty, define utility functions, and evaluate decision networks, including multi-attribute trade-offs and value of information.
- algorithms for Markov Decision Processes, Bandit problems, and Partially Observable MDPs, applying them to real-world decision-making scenarios.

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Quantifying Uncertainty** - Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use, Naive Bayes Models and types, Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Exact Inference in Bayesian Networks	**8**	**1,2**
2	**Probabilistic Reasoning over Time:** Time and Uncertainty, Inference in Temporal Models, Hidden Markov Models (HMMs), Kalman Filters, Dynamic Bayesian Networks	**6**	**1,2**
3	**Making Simple Decisions-** Combining Beliefs and Desires under Uncertainty, The Basis of Utility Theory, Utility Functions, Multi-attribute, Utility Functions, Decision Networks, The Value of Information, Unknown Preferences.	**8**	**1,2,3**
4	**Making Complex Decisions**- Sequential Decision Problems, Introduction to Markov Decision Process (MDP), Algorithms for MDPs, Bandit Problems, Partially Observable MDPs, Algorithms for Solving POMDPs.	**8**	**1,2,3**

Course Outcomes:
After competing this course, a student will be able to:

1. Demonstrate the use of conditional probability in decision making
2. Apply Naïve Base models in classification problems
3. Explain Bayesian belief Networks and their applications in inferencing

4. Describe HMMs and apply them in applications in uncertain domains that change with time
5. Design simple utility functions and apply them in in real world cases
6. Define and demonstrate the use of MDP and POMDPs

Text Books:

1. Russell, S., Norvig, P. (2021), *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson Education. (ISBN-13: 978-0-13-461099.,
2. E.Rich, K. Knight, S.B. Nair, *Artificial Intelligence*, McGraw Hill, Third Edition, ISBN 978007008770.

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-302-MJ-T: Supervised Machine Learning Techniques

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 20 End Sem Evaluation: 30	Total Lectures: 30

Prerequisites: Basic knowledge of Python programming, Understanding of probability and statistics, Fundamentals of linear algebra, Prior exposure to correlation and simple linear regression

Objectives:

The course is designed to:

- introduce the fundamental concepts of Machine Learning to first-time learners
- explain the working principles of major supervised learning algorithms
- help develop the ability to choose appropriate supervised models for real-world problems
- prepare students for advanced courses in Unsupervised Learning, Deep Learning, NLP, and Computer Vision

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Introduction to Machine Learning:** Meaning and evolution of Machine Learning, Artificial Intelligence vs Machine Learning vs Deep Learning, Types of Machine Learning: Supervised, Unsupervised, Reinforcement, Supervised Learning paradigm: labeled data, training and testing, Machine Learning workflow: data collection, preprocessing, model building, evaluation, Challenges in Machine Learning: overfitting, underfitting, bias-variance tradeoff, Applications of supervised learning in industry	**06**	**1,2,5**
2	**Regression Techniques:** Regression as a supervised learning technique, Linear Regression for prediction problems, Cost function and concept of loss minimization, Multiple Linear Regression (conceptual understanding), Assumptions of regression models, Model evaluation: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R^2 score, Interpretation of regression models in real-world applications	**06**	**1,2,3,4**
3	**Neural Networks for Supervised Learning:** Motivation for Neural Networks: Introduction, Architectures – feedforward and Feedback, TLNs, limitations of linear models, linear separability, Perceptron model and learning process, Activation functions: Step, Sigmoid, ReLU, Neural Networks for classification and regression Advantages and limitations of Neural Networks	**06**	**1,3,4,5**
4	**Tree-Based and Instance-Based Learning:** Decision Trees as supervised learning models, Decision Tree structure: root, nodes, branches, leaves, ID3	**06**	**1,2,3,5**

	algorithm and concept of information gain, Overfitting in Decision Trees and basic pruning idea, k-Nearest Neighbors (kNN): working principle, Distance measures and choice of k, Comparison of Decision Trees and kNN		
5	**Margin-Based Learning and Model Evaluation:** Support Vector Machines (SVM): intuition and motivation, Hyperplane, margin, and support vectors, Linear SVM for classification, Kernel trick, Classification performance metrics: Accuracy, Precision, Recall, F1-Score, Confusion matrix and interpretation, Model selection and comparison in supervised learning	**06**	**1,2,3,4 , 5**

Course Outcomes:

After competing this course, a student will be able to:

CO1: Explain the basic concepts and workflow of Machine Learning
CO2: Distinguish between different supervised learning approaches
CO3: Apply regression and classification models conceptually to real-world problems
CO4: Interpret model performance using suitable evaluation techniques
CO5: Build a strong foundation for advanced AI & ML subjects

Reference Books

1. Tom M. Mitchell, *Machine Learning*, McGraw Hill, 1997.
2. Edouard Duchesnay, Tommy Lofstedt, Feki Younes, *Statistics and Machine Learning in Python*, 2021. (Free available) (https://hal.science/hal-03038776v3/file/StatisticsMachineLearningPython.pdf)
3. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning: With Applications in Python,* Springer, 2023.
4. Dipanjan Sarkar, Raghav Bali, Tushar Sharma, *Practical Machine Learning with Python*, APress, 2018, ISBN: 978-1-4842-3207-1
5. Sebastian Raschka and Vahid Mirjalili, *Machine Learning with Python*, Packt Publishing, 2017.
6. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, O'Reilly Media, 2nd Edition, 2019.
7. Ethem Alpaydin, *Introduction to Machine Learning*, MIT Press, 4th Edition, 2020.

Online Learning Resources (SWAYAM / NPTEL)

1. Supervised Learning — SWAYAM² (SNEHA, CDOE). Link: https://online-degree.swayam2.ac.in/mri24_01_d03_s2_el06/preview
2. Introduction to Machine Learning — NPTEL / SWAYAM (IIT Kharagpur). Link: https://onlinecourses.nptel.ac.in/noc21_cs85/preview
3. Introduction to Machine Learning — NPTEL / SWAYAM (IIT Madras). Link: https://onlinecourses.nptel.ac.in/noc20_cs29/preview

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-303-MJ-T: Cybersecurity Concepts and Artificial Intelligence

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Pre-requisites:
Basic Programming Knowledge, Preliminary concepts in AI, Algorithms and data structures.

Objectives:
The course is designed to:
1. Introduce foundational cybersecurity principles and threats.
2. Explore AI techniques applied to cybersecurity.
3. Analyze ethical, legal, and societal implications of AI in security.
4. Equip learners with theoretical knowledge for advanced study or research.

Contents:

Unit	Description	Number of Lectures
1	**Foundations of Cybersecurity -** Cybersecurity principles: confidentiality, integrity, availability (CIA triad), Threats and vulnerabilities: malware, phishing, social engineering, Security models and policies, Basics of cryptography: symmetric/asymmetric encryption, hashing.	**8**
2	**Artificial Intelligence Basics for Security-** Introduction to AI: intelligent agents, machine learning paradigm, AI techniques relevant to security: classification, clustering, anomaly detection, Neural networks and deep learning in intrusion detection, Case studies: AI in spam filtering, fraud detection.	**8**
3	**AI-Driven Cybersecurity Systems-** Intrusion Detection Systems (IDS) and AI, Predictive analytics for threat intelligence, AI in malware analysis and detection, Limitations of AI in cybersecurity (adversarial attacks, bias).	**8**
4	**Ethical, Legal, and Future Perspectives-** Ethical issues in AI and cybersecurity, Privacy concerns and data protection laws, AI governance frameworks in security, Future trends: autonomous defense systems, quantum-safe AI security.	**6**

Course Outcomes:
After competing this course, a student will be able to:
CO1: explain fundamental cybersecurity concepts and identify common threats.
CO2: describe AI techniques and relate them to cybersecurity applications
CO3: analyze AI-driven security systems and evaluate their effectiveness.
CO4: critically evaluate ethical/legal aspects and propose future directions in AI-based cybersecurity.

Text Books:

1. Tim Rains, *Cybersecurity Strategy for the AI-Driven Era*, Packt Publishing, 3rd Edition, March 2026, ISBN: 9781806028573.
2. Clarence Chio & David Freeman, *Machine Learning and Security: Protecting Systems with Data and Algorithms*, O'Reilly Media, 1st Edition, 2018, ISBN: 9781491979907.

3. Petar Radanliev, *AI Security: The Most Dangerous Cyber-Attacks on Artificial Intelligence*, 1st ed., Springer Nature, 2026, ISBN: 9783032281050.
4. Leslie F. Sikos, *Artificial Intelligence in Cybersecurity*, Springer, 1st Edition, 2025, Free download: https://cisomarketplace.com/uploads/aiincybersecurity.pdf?srsltid=AfmBOorjO86WenNAfm3Aw5vk5CO20_FLd4V5ZAK9yDMCLfCedAshC5Zk

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-304-MJ-T: Prompt Engineering

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 20 End Sem Evaluation: 30	Total Lectures: 30

Prerequisites: Fundamentals of Artificial Intelligence, Basics of Communication and Language, Computer fundamentals.

Objectives:
The course is designed to-

- understand fundamentals of Large Language Models (LLMs)
- design and develop clear, precise, and effective prompts for domain specific prompt solutions
- apply prompt engineering patterns and frameworks
- control AI behavior and output formats
- build reusable and scalable prompt systems and evaluate and optimize prompt performance
- apply ethical, safe, and responsible prompting practices
- integrate Prompt engineering into tools and applications

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Introduction to Prompt Engineering:** Definition, importance, applications in AI, Introduction to AI, NLP, LLM, GPT, tokens, **Hands-on:** Explore basic prompts on ChatGPT/Copilot/ any other; observe differences in outputs	6	1,2
2	**Fundamentals of Prompt Design:** Elements/Components of a prompt - understanding prompt elements like instructions, context, input data, and output indicators. Key principles for designing clear, specific, and effective prompts, techniques for writing clear instructions, using delimiters, and specifying output formats, prompt design best practices. Prompt health checklists, hands-on practice in crafting basic prompts for common tasks like summarization, Q&A, and classification, Prompt design Vs Prompt Engineering.	8	2,3
3	**Prompt types and Patterns-** **Types of Prompts:** Instructional, role-based, conversational, multi-modal prompts, compare outputs of 3 prompt styles on the same query. **Prompt Structures:** Templates, chaining, hierarchical prompts, Design prompts for generating a poem, a business email, and a lesson plan, Strategies for writing better prompts. **Iterative Refinement:** Testing, feedback loops, prompt tuning, Refine a poorly performing prompt through 3 iterations.	8	2,3

	Evaluation Metrics: Accuracy, relevance, creativity, bias, Evaluate AI responses for bias; suggest refinements. **Ethical Considerations:** Concept of hallucinations, bias, misinformation, responsible use **Case study:** Prompts on cultural/social topics, analyze ethical risk.		
4	**Prompt Structures-** **Advanced Techniques:** Zero-shot, One-, few- and multi-shot, providing examples within the prompt to guide the model's response, Chain-of-Thought (CoT), Zero-shot CoT. **Prompting:** Encouraging the model to think step-by-step for better reasoning, Specialized Techniques: Includes Tree-of-Thought (ToT), ReAct, Self-Consistency, Generated Knowledge, and Step-back prompting, Prompt Patterns- Applying established patterns like the Persona Pattern or the Flipped Interaction Pattern, implement few-shot prompts for summarization and problem-solving **Domain-Specific Prompting:** education, healthcare, business, coding, create prompts tailored to 3 different domains. **Tools & Platforms:** Copilot, ChatGPT, Bard, Claude – strengths & limitations, demo: test same prompt across 2 platforms; compare outputs.	**8**	**1,2**

Course Outcomes:

CO1: Explain the working principles, capabilities, and limitations of Large Language Models (LLMs) used in prompt-based systems.

CO2: Design clear, structured, and goal-oriented prompts to obtain accurate and relevant outputs from AI models.

CO3: Apply standard prompt engineering techniques such as zero-shot, few-shot, role-based, and step-by-step prompting for diverse tasks.

CO4: Develop domain-specific prompt solutions for applications in areas such as programming, business, research, or creative content

CO5: Analyze and mitigate common issues such as hallucinations, bias, and ambiguity in AI-generated outputs.

CO6: Control AI responses by specifying constraints, tone, format, and evaluation criteria within prompts.

CO7: Construct reusable prompt templates and multi-step prompt workflows for scalable AI interactions.

CO8 Evaluate and optimize prompt effectiveness using qualitative and quantitative performance measures.

CO9: Integrate prompt engineering techniques into AI tools, applications, or automated workflows.

Text Books:

1. Lee Boonstra, *Prompt Engineering,* Google, Sept 2024 (free download: https://www.gptaiflow.com/assets/files/2025-01-18-pdf-1-TechAI-Goolge-whitepaper_Prompt%20Engineering_v4-af36dcc7a49bb7269a58b1c9b89a8ae1.pdf).
2. *Prompt Engineering Playbook (Beta v3),* GovTech Data Science & AI Division (free download version available).

3. José Alberto Hernández, Javier Conde et.al., *ChatGPT Learning prompt engineering with 100+ examples*, 2024 (free version - https://oa.upm.es/84328/1/book_english_version.pdf)
4. K, Matthew, *Prompt Engineering for beginners: Master AI, Prompt Engineering, ChatGPT, Claude, and More to Work Smarter and Earn Faster*

Reference Books:

1. Adriano Damiao, *Mastering AI Prompt Engineering: The Ultimate Guide for ChatGPT Users*, September 2024.

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-305-MJ-P: Practical based on AIML-301-MJ-T
(Artificial Intelligence - II)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Pre-requisites: Basic Programming Knowledge in Python, Fundamentals of Data Structures, Basic statistics, AI - I

Objectives:
The course is designed:

1. To familiarize students with the application of basic probability notation in computational contexts.
2. To enable learners to perform inference using joint distributions and test independence assumptions.
3. To apply Bayes' rule and Naive Bayes models for classification tasks.
4. To construct and interpret Bayesian networks and perform exact inference.
5. To explore temporal probabilistic models such as HMMs.
6. To understand and apply utility theory for decision-making under uncertainty.
7. To integrate theoretical knowledge with hands-on coding exercises for real-world uncertain environments.

Contents:

Unit	Description
1	Implementing Basic Probability Notation in Python
2	Inference Using Full Joint Distributions
3	Testing Independence in Probability Models
4,5	Bayes' Rule Application in Medical Diagnosis
6	Naive Bayes Classifier for Text Classification
7	Comparing Types of Naive Bayes Models - Experiment with Gaussian, Multinomial, and Bernoulli Naive Bayes on different datasets.
8	Representing Knowledge in an Uncertain Domain - Model uncertain knowledge (e.g., weather prediction) using probability distributions.
9	Constructing Bayesian Networks Build a Bayesian network for a simple domain (e.g., student performance prediction).
10	Exact Inference in Bayesian Networks - Perform variable elimination or enumeration to compute posterior probabilities.
11	Inference in Temporal Models - Simulate temporal probability updates using simple Markov chains.
12, 13	Hidden Markov Models (HMMs) for Speech Recognition - Implement a basic HMM for sequence labeling tasks.
14, 15	Utility Functions and Decision Networks - Define utility functions and solve decision problems using decision networks.

Note: A Lab book shall be prepared related to the practicals that a student needs to perform.

Course Outcomes:

After competing this course, a student will be able to:

CO1: Apply Bayes' theorem and Naive Bayes models to solve classification problems.
CO2: Construct Bayesian networks and perform exact inference for reasoning under uncertainty.
CO3: Implement temporal models like HMMs for reasoning over time.
CO4: Define and use utility functions to make rational decisions under uncertainty.
CO5: Design decision networks and compute the value of information in uncertain domains.
CO6: Simulate sequential decision-making using MDPs and explore algorithms for solving them.

Text Books:

1. Russell, S., Norvig, P. (2021), *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson Education. ISBN-13: 978-0-13-461099.
2. E.Rich, K. Knight, S.B. Nair, *Artificial Intelligence*, McGraw Hill, Third Edition, ISBN: 978007008770.
3. Allen B. Downey, *Think Bayes: Bayesian Statistics in Python*, O'Reilly Media, 2nd Edition, 2021, ISBN: 978-1492089469
4. Cameron Davidson-Pilon, *Probabilistic Programming & Bayesian Methods for Hackers,* Addison-Wesley, 1st Edition, 2015, ISBN: 978-0133902839
5. Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective (with Python examples)*, MIT Press, 1st Edition, 2012, ISBN: 978-0262018029
6. Sebastian Raschka & Vahid Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python*, scikit-learn, and TensorFlow 2, Packt Publishing 3rd Edition, 2019, ISBN: 978-1789955750
7. Ankur A. Patel, *Hands-On Markov Models with Python*, Packt Publishing, 1st Edition, 2018.

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-306-MJ-P: Practical based on AIML-302-MJ-T
(Supervised Machine Learning Techniques)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Pre-requisites: Python programming, Basic understanding of data structures, Familiarity with NumPy and Pandas

Objectives:
The course is designed:

- To provide hands-on experience with supervised machine learning algorithms
- To understand the complete machine learning workflow from data preprocessing to model evaluation
- To implement regression and classification models using standard ML libraries
- To analyze and compare supervised learning models for real-world datasets

Data Repositories

- Kaggle Datasets – https://www.kaggle.com/datasets
- UCI Machine Learning Repository – https://archive.ics.uci.edu
- Govt. of India data sets - https://indiaai.gov.in/datasets
- Scikit-learn Built-in Datasets

Contents:

Unit	Description
1	Program to understand the Machine Learning workflow including dataset loading, feature label separation, and train–test splitting using Python.
2	Program to perform data preprocessing including handling missing values, feature scaling, and encoding categorical variables.
3	Program to perform Exploratory Data Analysis (EDA) and visualize relationships among variables using statistical plots.
4	Program to implement Simple Linear Regression for predicting continuous values and visualize the regression line.
5	Program to implement Multiple Linear Regression and evaluate the model using MSE, RMSE, and R^2 score.
6	Program to verify the assumptions of Linear Regression and interpret regression coefficients.
7	Program to implement a Perceptron classifier for binary classification problems.
8	Program to design and train a Feedforward Neural Network with a single hidden layer for classification.
9	Program to study the effect of different activation functions on the performance of a Neural Network.
10	Program to implement a Decision Tree classifier and visualize the tree structure.
11	Program to analyze overfitting in Decision Trees by controlling tree depth and pruning parameters.
12	Program to implement k-Nearest Neighbors (kNN) classifier and study the effect of different values of k.
13	Program to evaluate classification models using confusion matrix, accuracy, precision, recall, and F1-score

14	Program to evaluate classification models using confusion matrix, accuracy, precision, recall, and F1-score
15	Program to compare multiple supervised learning models and select the best model based on performance metrics.

Course Outcomes:

After competing this course, a student will be able to:

CO1: Implement supervised machine learning algorithms using Python
CO2: Perform data preprocessing and feature engineering for ML tasks
CO3: Build and evaluate regression and classification models
CO4: Compare multiple supervised models and select appropriate models for given problems
CO5: Apply supervised learning techniques to real-world datasets

Text Books:

1. Sebastian Raschka and Vahid Mirjalili, *Machine Learning with Python*, Packt Publishing, 2017.
2. Dipanjan Sarkar, Raghav Bali, Tushar Sharma, *Practical Machine Learning with Python*, APress, 2018, ISBN: 978-1-4842-3207-1
3. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, O'Reilly Media, 2nd Edition, 2019.
4. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning: With Applications in Python*, Springer, 2023.
5. Tom M. Mitchell, *Machine Learning*, McGraw Hill, 1997.

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-310-MJ-T: Data Preprocessing and Visualization

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Python Programming, Basic Statistics

Objectives:
The course is designed to teach to:

1. introduce the critical concepts of data quality and the necessity of preprocessing in the AI/ML lifecycle.
2. teach robust statistical techniques for identifying and handling missing data and outliers.
3. equip students with mathematical methods for feature scaling, transformation, and encoding.
4. demonstrate the necessity and mathematics behind dimensionality reduction techniques.
5. establish a strong foundation in the cognitive and aesthetic principles of data visualization.
6. familiarize students with industry-standard libraries for crafting effective static and interactive visual narratives.

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Foundations of Data Cleaning & Quality** **Introduction to Data Quality:** Definition and significance of data quality, Dimensions of data quality: accuracy, completeness, consistency, validity, timeliness, uniqueness, Role of data cleaning in AI/ML lifecycle, Types of data (Structured vs. Unstructured), garbage-in-garbage-out (GIGO) principle, data integration. **Handling Missing Data:** Mechanisms of missing data (MCAR, MAR, MNAR). Imputation techniques: Mean, Median, Mode, and algorithmic imputation (KNN). **Outlier Detection:** Understanding outliers. Statistical detection methods: Z-score, Interquartile Range (IQR), Winsorization, and Isolation Forests.	**8**	T1, W1
2	**Feature Engineering & Dimensionality Reduction** **Data Transformation:** Normalization (Min-Max scaling), Standardization (Z-score scaling), Robust scaling, and Log transformations. **Encoding Categorical Data:** Ordinal vs. Nominal data. Label Encoding, One-Hot Encoding, Dummy variable trap, and Target Encoding. **Dimensionality Reduction:** The curse of dimensionality. Mathematical intuition and application of Principal Component Analysis (PCA).	**8**	T1
3	**Grammar of Graphics & Static Visualization**	**7**	T1, T2, W1

	Principles of Good Visualization: Edward Tufte's principles, maximizing the data-ink ratio, avoiding visual clutter, and accessible color theory. **Univariate Analysis:** Best practices for Histograms, Bar charts, Pie charts (and when to avoid them), and Box plots. **Bivariate/Multivariate Analysis:** Scatter plots, Line graphs, Heatmaps, and Pair plots. **Python Ecosystem:** Architectural overview of Matplotlib (Figure and Axes objects) and Seaborn.		
4	**Interactive Visualization & Storytelling** **Interactive Graphics:** The cognitive benefits of interactivity (Zooming, Panning, Filtering, Hover tooltips). **Advanced Tools:** Introduction to the Plotly ecosystem and Bokeh architectures. **Geospatial Data:** Basics of visualizing coordinate and mapping data (Choropleth maps, point maps). **Data Storytelling:** Constructing user-centric dashboards, narrative flows, and presenting data to non-technical stakeholders.	**7**	T2, W1

Course Outcomes:

After competing this course, a student will be able to:

CO1: explain the importance of data quality and the role of preprocessing in AI pipelines.
CO2: apply appropriate statistical techniques to clean noisy datasets and handle missing values.
CO3: transform and encode numerical and categorical data to optimize machine learning models.
CO4: analyze the impact of dimensionality reduction on complex, high-dimensional datasets.
CO5: evaluate the suitability of various chart types and design principles for specific data distributions.
CO6: design interactive visual narratives and dashboards to communicate complex data insights effectively.

Text Book(s)

1. Jake VanderPlas, *Python Data Science Handbook,* O'Reilly, Second Edition, 2022 ISBN: 978-1098121228
2. Claus O. Wilke, *Fundamentals of Data Visualization,* O'Reilly, First Edition, 2019, ISBN: 978-1492031086

Web Reference(s)

1. https://inria.github.io/scikit-learn-mooc/

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-311-MJ-P- Lab course on AIML-310-MJ-T
(Data Preprocessing and Visualization)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Pre-requisites: Basic Python programming

Objectives:
The course is designed to teach:
1. To provide hands-on experience in parsing and structurally organizing raw data from multiple file formats.
2. To train students in writing efficient Python scripts to autonomously clean datasets and treat anomalies.
3. To develop programmatic skills for executing feature engineering, scaling, and categorical encoding workflows.
4. To enable the practical, code-based implementation of dimensionality reduction on high-dimensional data.
5. To foster skills in conducting exhaustive, programmatic Exploratory Data Analysis (EDA).
6. To synthesize preprocessing and visualization skills by building and deploying interactive web dashboards.

Contents:

Unit	Description
1,2,3	**Data Wrangling and Cleaning** • **Data Ingestion & Pandas Basics:** Importing CSV, JSON, and SQL data; exploring DataFrames; basic querying. • **Missing Value Imputation** Implementing SimpleImputer, comparing mean/median imputation vs. KNN imputation on model accuracy. • **Outlier Detection & Treatment:** Scripting Z-score and IQR thresholds to filter or cap extreme values in a noisy dataset.)
4,5,6	**Feature Engineering & Dimensionality Reduction** • **Data Scaling and Normalization:** Writing custom transformers for Min-Max, Standard, and Robust Scaling. • **Categorical Encoding:** Handling high-cardinality categorical variables using One-hot encoding and frequency encoding. • **Dimensionality Reduction:** Implementing PCA using Scikit-Learn; visualizing cumulative explained variance ratios to select optimal components.
7,8,9	**Exploratory Data Analysis (EDA)** • **Univariate EDA:** Scripting automated reports to check the distribution, skewness, and kurtosis of individual variables. • **Bivariate/Multivariate EDA:** Generating correlation matrices, cross-tabulations, and evaluating feature relationships. • **EDA Case Study:** Performing an end-to-end exploratory analysis on a specific domain dataset (e.g., Healthcare diagnostics or E-commerce sales).
10,11,	**Static Data Visualization**

12	• **Matplotlib Mastery:** Customizing axes, ticks, legends, annotations, and creating complex multi-plot grids using subplots. • **Statistical Plotting with Seaborn:** Generating Violin plots, swarm plots, and regression plots (lmplot) to visualize statistical relationships. • **Applied Design Principles:** Refactoring a "bad" visualization into a "good" one by applying Tufte's data-ink ratio and accessible color palettes.
13,14, 15	**Interactive Visualization & Dashboarding** • **Interactive Visualizations:** Building interactive scatter plots, line charts, and 3D surface plots using Plotly Express. • **Geospatial Mapping:** Plotting geographical data utilizing Folium or Plotly to create interactive Choropleth maps. • **Dashboard Creation:** Building and serving a local web dashboard using Streamlit or Dash, integrating data uploading, preprocessing toggles, and reactive visualizations.

Note: A Lab book shall be prepared related to the practicals that a student needs to perform.

Course Outcomes:

After competing this course, a student will be able to:

CO1: Write Python scripts to ingest, parse, and structure raw data from diverse sources.
CO2: Implement algorithmic solutions to impute missing data and remove outliers systematically.
CO3: Execute automated feature engineering workflows, including scaling, encoding, and PCA.
CO4: Conduct comprehensive Exploratory Data Analysis (EDA) to extract patterns from domain-specific datasets.
CO5: Construct advanced, publication-ready static plots using Matplotlib and Seaborn.
CO6: Develop and deploy interactive data visualization dashboards using modern web frameworks

Text Book(s)

1. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly, Second Edition, 2022 ISBN: 978-1098121228
2. Claus O. Wilke, *Fundamentals of Data Visualization*, O'Reilly, First Edition, 2019, ISBN: 978-1492031086

Web Reference(s)

1. https://inria.github.io/scikit-learn-mooc/

T.Y. B.Sc. (AI and ML)
Semester – V
Major Elective
AIML-312-MJ-T: Vibe Coding

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Prerequisites: Basic programming knowledge (Python/JavaScript), interest in creative coding or digital art.

Objectives:
The course is designed to:

1. Introduce students to the concept of vibe coding as a fusion of programming, creativity, and interactive media.
2. Develop skills in using coding environments for music, visuals, and real-time interaction.
3. Encourage experimentation and creative expression through coding.
4. Analyze the role of vibe coding in education, entertainment, and digital art.
5. Create interactive projects that integrate sound, visuals, and user input.

Contents:

Unit	Contents	Number of lectures	Text Book
1	Introduction- Concept and philosophy of vibe coding. Overview of creative coding platforms (Sonic Pi, Processing, p5.js, Hydra). Applications in education, art, and entertainment.	**6**	**1,2**
2	Coding for Sound and Music - Basics of digital sound synthesis. Live coding music with Sonic Pi / SuperCollider. Rhythm, melody, and harmony generation through code. Create a simple music track using vibe coding.	**8**	**1,2**
3	Coding for Visuals and Interaction- Generative art with Processing/p5.js (or any other). Real-time visuals with Hydra. User interaction: keyboard, mouse, and sensor inputs. Build an interactive visualizer responding to sound or user input.	**8**	**1,2,3**
4	Integrative Projects and Ethics - Combining sound, visuals, and interactivity. Collaborative vibe coding sessions (live performances). Ethical considerations: authorship, originality, and digital rights. A Mini-project like Create a vibe coding performance or any other related to AI use or installation.	**8**	**1,2,3**

After competing this course, a student will be able to:
Course Outcomes:

CO1: Understand the fundamentals of vibe coding and its applications.
CO2: Apply coding tools to generate music, visuals, and interactive experiences.
CO3: Analyze vibe coding projects to identify strengths, limitations, and creative potential.

CO4: Evaluate the effectiveness of vibe coding in educational and artistic contexts.
CO5: Create original vibe coding projects integrating sound, visuals, and interactivity.

Text books:

1. Casey Reas & Ben Fry, *Processing: A Programming Handbook for Visual Designers and Artists,* MIT Press, second ed., 2015, ISBN: 978-0262028288
2. Hartmut Bohnacker et al., Generative Design: *Visualize, Program, and Create with JavaScript in p5.js*, Princeton Architectural Press, 2018, ISBN: 978-161689758
3. Scott Wilson, David Cottle, Nick Collins, *The SuperCollider Book*, The MIT Press, 2011, ISBN: 978-0262232692

Online References:

1. Online resources: Sonic Pi tutorials, p5.js reference, Hydra live coding documentation.
2. https://sonic-pi.net/tutorial.html
3. Sam Aaron, Sonic Pi: Live & Coding, https://www.youtube.com/watch?reload=9&v=9fUdnTBYOWE

T.Y. B.Sc. (AI and ML)
Semester – V
Major Elective
AIML-313-MJ-P: Practical based on AIML-312-MJ-T
(Vibe Coding)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Prerequisites: Prerequisites: Basic programming knowledge (Python/JavaScript), interest in creative coding or digital art.

Objectives:
The course is designed to teach:

- installation, configuration, and workspace setup
- develop comparative understanding of different vibe coding tools
- investigate real-world applications of vibe coding
- apply coding to create rhythmic sound structures and explore melodic structures and parameter manipulation
- critically evaluate algorithmic vs human-coded outputs.
- apply coding to produce algorithmic visual art.
- apply cumulative skills to design an immersive, interactive experience

Contents:

Unit	Description
1	Install and configure Sonic Pi / Processing / p5.js or any other, and run a simple "Hello World" vibe coding sketch.
2	Explore at least two vibe coding platforms (e.g., Sonic Pi vs Hydra) and compare their features.
3	Document the applications of vibe coding in education, art, and entertainment with examples (quiz, calculator etc.)
4	Write a Sonic Pi script to generate a simple rhythm pattern (e.g., drum loop).
5	Create a melody using code, experimenting with tempo and pitch variations.
6	Combine rhythm and melody to produce a short-coded music track
7	Analyze AI-generated music vs manually coded music for creativity and structure
8	Use Processing/p5.js to create generative art (e.g., geometric patterns).
9	Build an interactive sketch where visuals respond to mouse or keyboard input.
10	Create a sound-reactive visualizer (visuals change based on audio input).
11	Evaluate the effectiveness of different vibe coding tools for interactive visuals and Write a reflective essay on ethical issues in vibe coding
12-15	Develop a mini-project combining sound and visuals, conduct a collaborative live vibe coding performance with peers, final project: create an interactive installation

Course Outcomes:
After competing this course, a student will be able to:

CO1: set up the IDE, manage projects, and troubleshoot environment issues.
CO2: produce a comparative report highlighting strengths, limitations, and use cases of platforms.
CO3: compile a documented portfolio of case studies showing interdisciplinary relevance.
CO4: Generate a functioning rhythm loop and produce coded melodies with variations in tempo and pitch, showing creative experimentation

CO5: deliver a coded track that demonstrates musical layering and synchronization.
CO6: generate visual outputs showcasing randomness, iteration, and design principles.
CO7: deliver a multimedia project demonstrating cross-modal creativity.
CO8: produce essays articulating ethical challenges and proposing responsible practices.
CO9: present a final installation project demonstrating creativity, technical mastery, and audience engagement, complete progressive hands-on projects culminating in a capstone installation, showcasing both technical and artistic mastery.

Text books:

1. Casey Reas & Ben Fry, *Processing: A Programming Handbook for Visual Designers and Artists,* MIT Press, second ed., 2015, ISBN: 978-0262028288
2. Hartmut Bohnacker et al., Generative Design: *Visualize, Program, and Create with JavaScript in p5.js*, Princeton Architectural Press, 2018, ISBN: 978-161689758
3. Scott Wilson, David Cottle, Nick Collins, *The SuperCollider Book*, The MIT Press, 2011, ISBN: 978-0262232692

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-314-MJ-T: Design and Analysis of Algorithms - II

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Design and analysis of Algorithms - I

Objectives:
The course is designed:

- To understand advanced algorithmic techniques beyond greedy and dynamic programming.
- To explore backtracking and branch & bound for combinatorial problems.
- To analyze randomized algorithms and their probabilistic guarantees.
- To study computational complexity classes (P, NP, NP-complete, NP-hard).
- To learn approximation algorithms for intractable problems.

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Backtracking-** Introduction to backtracking, general strategy, comparison with brute force, N-Queens problem – recursive formulation, Graph coloring problem, Hamiltonian cycle problem, Sudoku solver using backtracking, Complexity analysis of backtracking algorithms.	**9**	**1,2**
2	**Branch and Bound-** Introduction to Branch & Bound, bounding functions, 0/1 Knapsack problem using B&B, Traveling Salesman Problem (TSP), Comparison of backtracking vs branch & bound, Efficiency analysis and limitations.	**9**	**1,2**
3	**Randomized algorithms-** Introduction, Types of Randomized Algorithms – Las Vegas, Monte Carlo, Approximate Median, Randomized Quick Sort using Las Vegas method, shortest enclosing circle (for a set of points), min Cut (in graphs)	**8**	**1,2,3**
4	**P & NP-** Introduction to complexity classes (P, NP), Polynomial time reductions, NP-complete problems – SAT, Clique, Vertex Cover, NP-hard problems and implications, Is P = NP? Current research directions.	**4**	**1,2,3**

Course Outcomes:
After competing this course, a student will be able to:

CO1: Apply backtracking and branch & bound to solve complex search problems.
CO2: Design randomized algorithms and analyze expected performance.
CO3: Differentiate between P, NP, NP-complete, and NP-hard problems.
CO4: Apply approximation techniques to NP-hard problems.
CO5: Evaluate algorithmic efficiency using theoretical and experimental analysis.

Text books:

1. Ellis Horowitz, Satraj Sahni and Rajasekharam, *Fundamentals of Computer Algorithms,* Galgotia publications Pvt. Ltd. Silicon Press, 2007
2. Parag Himanshu Dave and Himanshu Bhalchandra Dave, *Design and Analysis Algorithms,* Publisher: Pearson
3. M.T.Goodrich and R.Tomassia, *Algorithm Design: Foundations, Analysis and Internet examples,* John wiley and sons.
4. Levitin, *The Design and Analysis of Algorithms*, 3rd Edition, Pearson, 2012.
5. Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms,* MIT Press, Third Edition, 2009
6. Motwani and Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
7. Mitzenmacher and Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2nd edition, 2017

T.Y. B.Sc. (AI and ML)
Semester – V
Elective
AIML-315-MJ-P: Practical based on AIML-314-MJ-T
(AIML-314-MJ-T: Design and Analysis of Algorithms – II)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Prerequisites: Python Programming, OOP

Objectives:
The course is designed to teach:

- Use of Backtracking strategy in implementing solution to a few real-world problems.
- Applications of branch and bound strategy in various relevant applications
- Use of randomized algorithms to implement a solution to sorting and rndom selection type of problems.

Contents:

Unit	Description
1, 2	Backtracking - N-Queens solver
3, 4	Graph coloring using backtracking.
5, 6	Sudoku solver
7, 8	0/1 Knapsack using B&B.
9,10	Traveling Salesman Problem using B&B.
11,12	Randomized QuickSort implementation.
13,14, 15	Randomized selection algorithm.

Course Outcomes:
After competing this course, a student will be able to:

CO1: Apply backtracking strategy to solve complex search problems.
CO2: Develop randomized algorithms and analyze expected performance.
CO3: Use branch & bound strategy to computationally solve TSP and 0/1 Knapsack problems

Text books:

1. Ellis Horowitz, Satraj Sahni and Rajasekharam, *Fundamentals of Computer Algorithms,* Galgotia publications Pvt. Ltd. Silicon Press, 2007
2. Parag Himanshu Dave and Himanshu Bhalchandra Dave, *Design and Analysis Algorithms,* Publisher: Pearson

T.Y. B.Sc. (AI and ML)
Semester – V
VSC
AIML-321-VSC-P: Design Thinking

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Objectives:
The course is designed to teach:

- Empathy-driven techniques (interviews, observations, empathy maps, personas) to identify user pain points and latent needs.
- Methods to frame problems effectively by writing Point of View (POV) statements and "How Might We" questions.
- Generating innovative solutions using brainstorming, SCAMPER, mind mapping, and evaluation matrices.
- Development of low- and high-fidelity prototypes to visualize and test ideas.
- Methods to test and refine prototypes through structured feedback loops and iteration cycles.
- Methods to collaborate in mini design sprints to tackle real-world challenges in campus or community contexts.

Contents:

Unit	Description
1	Definition, history, and importance of design thinking, Comparison with traditional problem-solving approaches, Case studies of design thinking in business, education, and social innovation- Discuss - Smart Attendance System Using Face Recognition, AI-Based Waste Segregation System, Student Mental-Health Monitoring System, Crop disease detection for farmers, Clean water in rural India, AI-enabled traffic congestion prediction, Recommendation system for local libraries, Designing your life or any other
2,3	Empathize – Understanding Users- Techniques: interviews, observations, empathy maps, personas, identifying pain points and latent needs, Hands-on activity: empathy mapping exercise for the case study selected above (in 1,2), Students observe and interview 3–4 users in a real environment (e.g., campus, café) to identify pain points. Develop 2–3 user personas based on collected data; present empathy maps.
4,5	Framing the Problem- Problem statements, Point of View (POV), and "How Might We" questions, Converting insights into actionable design challenges, Activity: reframing problems with multiple perspectives. Write clear POV statements and "How Might We" questions for identified challenges. Analyze a real-world design thinking case (IDEO, Apple, etc.) and present insights.
6-9	Ideate – Generating Solutions - Brainstorming, SCAMPER, mind mapping, and lateral thinking, Divergent vs. convergent thinking, Activity: group ideation session with evaluation matrix. Conduct group brainstorming using SCAMPER/mind mapping; shortlist top 5 ideas. Apply evaluation matrix to compare ideas based on feasibility, desirability, viability. Create storyboards for selected ideas to visualize user journeys.
10-12	Prototype – Building to Think - Low-fidelity vs. high-fidelity prototypes, Storyboarding, paper prototyping, mock-ups, Activity: rapid prototyping

	Build low-fidelity prototypes using paper, sketches, or simple mock-ups. Use tools (Figma, Canva, PowerPoint) to create digital prototypes.
13-15	Test – Iteration and Feedback - User testing methods, feedback loops, and iteration cycles, Importance of failure and learning, Activity: testing prototypes with peers and refining solutions. Conduct peer testing of prototypes; collect structured feedback. Refine prototypes based on feedback; document changes. Mini Sprint Design - Run a 1-day sprint: Empathize → Define → Ideate → Prototype → Test on a small challenge. Apply design thinking to a campus/community issue (e.g., waste management, library use), Collaborative project: teams tackle a larger problem, prepare presentation + prototype.

Course Outcomes:

After competing this course, a student will be able to:

- CO1: explain design thinking principles and analyze case studies to identify innovation impact.
- CO2: conduct interviews/observations, create empathy maps, and develop user personas based on real data.
- CO3: formulate clear problem statements, write POVs, and generate HMW questions to reframe challenges.
- CO4: generate multiple ideas, apply SCAMPER/mind maps, shortlist top 5 ideas, and create storyboards for user journeys.
- CO5: build prototypes (paper + digital tools like Figma/Canva), and visualize solutions through storyboarding.
- CO6: test prototypes with peers, collect feedback, refine solutions, and present final prototypes in collaborative projects.

Text books:

1. Brown, T, *Change by Design: How Design Thinking Creates New Alternatives for Business and Society*, Harper Business, 2009.
2. Kelley, T., & Kelley, D. (2013). Creative Confidence: Unleashing the Creative
3. Potential Within Us All. Crown Business.
4. Liedtka, J., & Ogilvie, T. (2011). Designing for Growth: A Design Thinking Toolkit for Managers. Columbia University Press.
5. Plattner, H., Meinel, C., & Leifer, L. (2011). Design Thinking: Understand – Improve – Apply. Springer.
6. Cross, N. (2011). Design Thinking: Understanding How Designers Think and Work. Berg Publishers

T.Y. B.Sc. (AI and ML)
Semester – V
AIML-331-FP: Field Project

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Hours: 60

Objectives:
The course is designed:

- To articulate the relationship between classroom theories and their practical application in a specific real-world context.
- To help students progressively build higher-order thinking skills—moving from basic observation to the creation of new solutions.
- To teach the methods to understand all aspects of a real-life problem and deconstruct complex field observations to identify underlying patterns, anomalies, or systemic issues
- To teach the methods to maintain a Real-Time journal, triangulate data, expect the unexpected, and improve communication skills.

Guidelines to Students:

1. Students should select a field related to the major subject.
2. Students should complete minimum of 60 hours on the Field Project.
3. A field report should be maintained by the student of the activities / work done each day as part of the field work.
4. The field activities report (Reflective Journal) should be signed by the mentor / guide on a weekly basis.
5. Student must submit original copy of this field report to the college before the ESE.
6. Decision of the mentor/guide shall be final.
7. Student should present a certificate of completion of the field project.
8. Student must be present for the internal evaluations.
9. A presentation of the work done and outcomes should be prepared as part of ESE evaluation.

Assessment Criteria:

Component	**Criteria**
Proposal	Problem clarity, objectives, feasibility
Field Work	Data collection rigor, engagement, ethics
Interim Review	Progress, methodology refinement
Final Report	Structure, analysis, originality, insights
Presentation/Viva	Clarity, confidence, defense of findings
Reflective Journal	Depth of reflection, learning evidence

Course Outcomes:
After competing this course, a student will be able to:

CO1: Identify and articulate a real-world problem relevant to their discipline.
CO2: Design and implement a field-based project plan with clear objectives and methodology.
CO3: Collect, interpret, and analyze primary/secondary data from the field.
CO4: Integrate findings into actionable insights or solutions.
CO5: Present project outcomes effectively through written reports and oral presentations.
CO6: Exhibit teamwork, ethical responsibility, and reflective practice

T.Y. B.Sc. (AI and ML)
Semester – V
Minor
AIML-341-MN-T: Calculus for Machine Learning

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Basic knowledge of algebra, Familiarity with graphs and functions, Introductory understanding of Machine Learning concepts

Objectives:
The course is designed:

1. To introduce fundamental calculus concepts required for machine learning.
2. To develop intuitive understanding of change, slope, and optimization.
3. To connect calculus concepts with training of ML models.
4. To prepare students for optimization techniques used in ML and DL.

Contents:

Unit	Contents	Number of lectures	Reference Book
1	**Introduction to Calculus for Machine Learning:** What is Machine Learning , Role of mathematics in ML, Why calculus is required in ML model training, Real-life applications of calculus in ML (image recognition, speech recognition, recommendation systems), Basic mathematical concepts: variables, constants, functions and graphs, Linear and non-linear functions, Concept of change in real-life systems , Types of functions used in ML: linear, quadratic, polynomial, exponential, logarithmic, Graphical understanding of functions: plotting, interpretation, increasing and decreasing functions	**08**	**1,2**
2	**Limits and Continuity:** Meaning and intuitive understanding of limits, Left-hand and right-hand limits, Simple numerical examples, meaning of continuity, Continuous and discontinuous functions, Real-life interpretation of continuity, Importance of continuity in ML (continuous loss functions, smooth optimization), Simple limit problems: polynomial and rational functions, limits using direct substitution	**07**	**1,3**
3	**Differentiation and Applications in ML:** Concept of derivative, Derivative as rate of change, Slope of a curve, Tangent to a curve, Basic rules of differentiation: constant rule, power rule, sum and difference rule, simple chain rule, Common derivatives used in ML (polynomial, exponential, logarithmic functions), Applications of derivatives in ML-gradient and slope interpretation, error minimization, loss functions, Introduction to Gradient Descent – Simple numerical problems	**08**	**1,2,3**
4	**Optimization for Machine Learning:** Introduction to optimization, Need for optimization in ML, Concept of minimum and maximum of functions, Cost and loss	**07**	**2,3**

	functions, Mean Squared Error (MSE), Visualizing loss curves, Gradient Descent: intuition, learning rate, iterative update rule, visual understanding, Simple one-variable optimization problems, ML-based real-life examples		

Course Outcomes:

After competing this course, a student will be able to:

CO1: Explain the role of calculus in machine learning models.
CO2: Interpret functions and graphs used in ML.
CO3: Apply basic differentiation techniques.
CO4: Explain optimization concepts such as loss minimization and gradient descent.
CO5: Relate calculus concepts to real-world ML applications.

Reference Book:

1. Gilbert Strang, *Calculus*, Wellesley-Cambridge Press.
2. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, *Mathematics for Machine Learning*, Cambridge University Press.
3. James Stewart, *Calculus: Early Transcendentals*, Cengage Learning.

Online Learning Resources:

1. Mathematics for Machine Learning, Prof. Balaraman Ravindran, IIT Madras, Link: https://onlinecourses.nptel.ac.in
2. Introduction to Machine Learning, Prof. Sudeshna Sarkar, IIT Kharagpur, Link: https://onlinecourses.nptel.ac.in

Detailed Syllabus

B.Sc (AI & ML)

Semester VI

(Third Year)

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-351-MJ-T: Introduction to Natural Language Processing (NLP)

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Artificial Intelligence -I, Python Programming, OOP

Objectives:
The course is designed to:
- Introduce fundamental concepts of Natural Language Processing (NLP).
- Familiarize students with text preprocessing, linguistic features, and representation techniques.
- Provide exposure to classical NLP tasks (tokenization, POS tagging, parsing, sentiment analysis).
- Introduce modern approaches using embeddings and neural models.
- Enable students to implement basic NLP applications with Python libraries.

Contents:

Unit	Contents	Number of lectures	Text Book
1	Foundations of NLP- Introduction to NLP, applications, challenges, text vs speech, linguistic basics- Collocations, use of Statistical Inference: n-gram Models over Sparse Data, Word Sense Disambiguation, Lexical Acquisition, use and importance of Corpus, Entropy – Joint and conditional, the noisy channel model Text Preprocessing- Tokenization, stemming, lemmatization - Synsets and Hypernyms, stopword removal, normalization, regex	**10**	**1,2**
2	Morphology and Syntax- POS tagging, parsing, chunking, dependency trees. Semantic Analysis- Named Entity Recognition (NER), Word Sense Disambiguation, semantic similarity.	**10**	**1,2**
3	Text Representation- Bag-of-Words, TF-IDF, Word embeddings (Word2Vec, GloVe), contextual embeddings, Building your own corpus	**6**	**1,2**
4	Dependency Parsing, Constituency Parsing Classical NLP Applications- Sentiment analysis, text classification, topic modelling, Recent trends and applications of NLP.	**4**	**1,3**

Course Outcomes:
After competing this course, a student will be able to:
CO1: Explain core NLP concepts and challenges (ambiguity, morphology, syntax, semantics).
CO2: Apply preprocessing techniques (tokenization, stemming, lemmatization, stopword removal).
CO3: Use vectorization techniques (Bag-of-Words, TF-IDF, Word2Vec).

CO4: Build simple applications such as sentiment analysis, text classification, and chatbot prototypes.

Text books:

1. Chris Manning, Hinrich Schütze, Foundations *of Statistical Natural Language Processing*, MIT Press, 1999, ISBN: 9780262303798.
2. Dan Jurafsky & James H. Martin, *Speech and Language Processing*, 3rd Edition, Draft, 2026 (https://web.stanford.edu/~jurafsky/slp3/ed3book_jan26.pdf) (PPTS: https://web.stanford.edu/~jurafsky/slp3/)
3. Pushpak Bhattacharyya & Aditya Joshi, *Natural Language Processing*, Wiley, 2023, ISBN: 9789357462389.
4. Steven Bird, Ewan Klein, Edward Loper, *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*, O'Reilly, 2009, http://www.nltk.org/book/

Reference Books:

1. Christopher Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999, ISBN 0-262-13360-1.
2. Palash Goyal, Sumit Pandey, Karan Jain, *Deep Learning for Natural Language Processing*, APress, 2018, ISBN: 978-1-4842-3685-7 (https://ia801805.us.archive.org/14/items/deep-learning-collection-pdf/Apress%20-%20Deep%20Learning%20for%20Natural%20Language%20Processing%20%282018%29.pdf)

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-352-MJ-T: Unsupervised Machine Learning

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Understanding of Machine Learning fundamentals, Knowledge of probability, statistics, and linear algebra, Basic familiarity with data preprocessing concepts.

Objectives:

- To understand learning from unlabeled data.
- To discover hidden patterns and structures in datasets.
- To apply clustering and association techniques.
- To prepare students for advanced analytics, Deep Learning, CV, and NLP.

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Unsupervised Learning Foundations:** Review of Machine Learning paradigms, Difference between supervised and unsupervised learning, Nature of unlabeled data, Similarity and distance measures, Feature representation, Applications of unsupervised learning, Challenges in unsupervised ML	**07**	**1,3,5**
2	**Clustering Techniques:** Clustering as a learning task, K-Means clustering, Hierarchical clustering, Density-based clustering (DBSCAN), Cluster evaluation measures, Choosing the number of clusters	**08**	**1,2,4,5**
3	**Dimensionality Reduction Techniques:** Curse of dimensionality, Feature reduction vs feature selection, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Manifold learning (t-SNE / UMAP), Visualization of high-dimensional data	**08**	**1,2,3**
4	**Advanced Unsupervised Learning & Applications:** Association rule mining (Apriori), Anomaly and outlier detection, Topic modelling, Autoencoders, Density estimation, Graph-based unsupervised learning, Role of unsupervised learning in modern AI systems	**07**	**2,3,4,5**

Course Outcomes:

After competing this course, a student will be able to:

CO1: Explain the need and challenges of unsupervised learning
CO2: Apply clustering techniques to real-world data
CO3: Perform dimensionality reduction for feature analysis
CO4: Extract association rules from large datasets
CO5: Use unsupervised learning as a preprocessing step for advanced AI systems

Text Books:

1. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning*, Springer, 2nd Edition, 2009.
3. Ethem Alpaydin, *Introduction to Machine Learning*, MIT Press, 4th Edition, 2020.
4. Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 3rd Edition, 2011.
5. Sebastian Raschka and Vahid Mirjalili**,** *Machine Learning with Python*, Packt Publishing, 2017.

Online Learning Resources:

1. **Unsupervised Learning** — SWAYAM² (SNEHA, CDOE)
   Link: https://online-degree.swayam2.ac.in
2. **Data Mining** — NPTEL / SWAYAM (IIT Kharagpur)
   Link: https://onlinecourses.nptel.ac.in
3. **Pattern Recognition and Applications** — NPTEL / SWAYAM (IIT Madras)
   Link: https://onlinecourses.nptel.ac.in
4. **Introduction to Machine Learning** — NPTEL / SWAYAM (IIT Madras)
   Link: https://onlinecourses.nptel.ac.in
5. **StatQuest with Josh Starmer:** https://www.youtube.com/c/joshstarmer

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-353-MJ-T: Human Computer Interface (HCI)

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Basic knowledge of computer fundamentals, Understanding of software systems and applications, Familiarity with web or mobile applications (as a user), Basic awareness of Artificial Intelligence applications**.**

Objectives:
The course is designed:

- To teach the fundamentals of Human–Computer Interaction
- To study user-centered interface design principles
- To analyze usability, accessibility, and user experience in intelligent systems
- To design effective interfaces for AI-driven applications

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Introduction to Human Computer Interaction:** Introduction to HCI, Evolution of Human–Computer Interaction, Importance of HCI in modern software and AI systems, Components of HCI: Human, Computer, Interaction, Goals of HCI: usability, efficiency, safety, learnability, User-centered design approach, Role of HCI in AI-based systems (chatbots, recommender systems, smart assistants), Case studies of successful and failed interface designs	**08**	**1,2**
2	**Human Factors and Cognitive Aspects:** Human information processing model, Memory, perception, and attention, Cognitive models in interface design, Mental models and user expectations, Ergonomics and human factors, Human errors and interface design Designing for different user groups	**07**	**1,3**
3	**Interface Design and Interaction Styles:** Principles of interface design, Interaction styles: command line, menu, form, GUI, touch, voice, Interface design standards and guidelines, UI components: buttons, forms, dialogs, navigation, Prototyping and wireframing concepts Design patterns in interface development, HCI design for mobile and web applications	**08**	**2,3**
4	**Usability, Evaluation and Future Interfaces:** Concept of usability, Usability engineering lifecycle, Heuristic evaluation, Usability testing methods, Accessibility and inclusive design, Ethical issues in HCI, Future of HCI: AR/VR, gesture interfaces, conversational interfaces, brain–computer interfaces	**07**	**2,4**

Course Outcomes:

After competing this course, a student will be able to:

CO1: Understand human factors in computer system design
CO2: Design user-friendly interfaces for software and AI systems
CO3: Evaluate usability of interactive applications
CO4: Apply HCI principles in AI, ML, and intelligent product development

Text Books:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, Pearson Education.
2. Jenny Preece, Yvonne Rogers, Helen Sharp, *Interaction Design: Beyond Human-Computer Interaction*, Wiley.
3. Wilbert O. Galitz, *The Essential Guide to User Interface Design*, Wiley.
4. Steve Krug, *Don't Make Me Think: A Common Sense Approach to Web Usability*, New Riders.

Online Learning Resources:

1. **Human–Computer Interaction** — *SWAYAM (IISc Bangalore)*
   Link: https://onlinecourses.swayam2.ac.in/cec23_cs35/preview
2. **Introduction to Human–Computer Interaction** — *NPTEL / SWAYAM (IIT Bombay)*
   Link: https://onlinecourses.nptel.ac.in/noc20_cs21/preview

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-354-MJ-T: Software testing

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Basic Software Engineering Paradigms, Basic Software Design and Project management, any programming language (OOP preferred)

Objectives:

The course is designed to teach:

1. Understand the fundamental concepts of software testing.
2. Plan and design test cases, generate test report.
3. To understand Defect management.

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Introduction to Software testing-** Basics of Software testing, Testing Objectives, Testing Principles, Testing fundamentals, Activities of a Test Engineer, Software Testing Limitations and Terminology, Coverage Criteria for Testing	**6**	**1,2**
2	**Testing Methods -** White Box Testing, Types of white box testing, Black Box Testing, Types of black box testing. **Software Testing Strategies -** Software Testing Process, Unit Testing, Integration- Top-down, Bottom up System Testing, Acceptance Testing (alpha, Beta testing)	**6**	**1,3**
3	**Test Templates, and test case creation -** Entry criteria Exit Criteria, Test scenario template, Test case template Test plan, Design test case for given application, Design test cases in Excel, Prepare test report for test cases, Demo software testing tools for Web Automation (like Selenium), Mobile Testing Tools (Espresso), API & Performance Testing Tools (JMeter)	**10**	**2,3**
4	**Defect Management -** Defect life Cycle, Defect Classification, Defect Report, Defect management, Finding defects, Logging defects, Tracking and managing defects Test Automation: Introduction and challenges, Use of AI in software Testing	**8**	**1,3**

Course Outcomes:

After competing this course, a student will be able to:

CO1: Understand fundamental concepts, principles, and objectives of software testing.
CO2: Analyze software requirements to design appropriate test plans, test cases, and test scenarios.
CO3: Apply various software testing techniques.
CO4: Understand different levels of testing strategies.

CO5: design basic test cases.
CO6: Identify, report, and manage software defects.

Text books:

1. Roger S. Pressman, *Software Engineering – A Practitioners Approach,* 7thEdition, Tata McGraw Hill, 2009, ISBN: 9339212088.
2. Srinivasan Desikan and Gopalaswami Ramesh, *Software Testing Principles and practices*, Pearson Education, 6ht ed., 2008, ISBN: 9788131785706.
3. Effective Methods of Software Testing – William E Perry, 3rd Edition, Wiley Publishing Inc.

Reference books:

1. Adam Leon Smith, James Davenport et.al, *Artificial Intelligence and Software Testing,* BCS, 2022, ISBN: 9781780175768.
2. Paul Ammann, Jeff Offutt, *Introduction to Software Testing,* Cambridge University Press, 2008, ISBN: 9780521880381.
3. Arnon Axelrod*, Complete Guide to Test Automation: Techniques, Practices, and Patterns for Building and Maintaining Effective Software Projects*, APress, 2018, ISBN: 978-1484238318

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-355-MJ-P: Practical based on AIML-351-MJ-T
(Introduction to NLP)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Prerequisites: Basic Maths and Stats, Python Programming

Objectives:
The course is designed to teach:

- learn the fundamentals of natural language processing
- understand the role of semantics of sentences and pragmatic.
- Understand the importance of corpus and applications of NER
- Methods to classify text.

Contents:

Unit	Description
1	Compare text vs speech datasets. Explore the nltk package, Load and inspect a sample corpus (e.g., Brown, Reuters).
2,3	Read a PDF of English text, for the text compute word frequencies, sentence lengths, vocabulary size. Visualize Zipf's law of distribution. Compare formal vs informal text corpora.
3, 4	For given text (as a case) explore the functions of nltk/spaCy/sikit-learn that handle ambiguity, polysemy, sarcasm, Hands-on: Identify challenges in tweets vs news articles.
5	For given text, implement word, sentence, and subword tokenization. Apply regex-based normalization.
6	Compare stemming vs lemmatization outputs. Evaluate impact on classification accuracy. Point out the trade-offs in preprocessing using library functions.
7,8	Build semantic networks using hypernyms. Explore WordNet synsets.
9,10	Build a project to implement text analysis skills - Implement POS tagging using NLTK & spaCy. Perform noun phrase chunking.
11,12	Train NER on custom dataset. Evaluate precision/recall.
13-15	Build BoW and TF-IDF models. Apply to document classification. Train Word2Vec on a small corpus.

Note: A Lab book shall be prepared related to the practicals that a student needs to perform.

Course Outcomes:
After competing this course, a student will be able to:

CO1: Implement classical NLP tasks using libraries like NLTK, spaCy, and scikit-learn.
CO2: Build simple applications such as sentiment analysis and text classification
CO3: Students understand corpus importance and NLP pipeline basics.

Text books:

1. Dr. Goonjan Jain & Dr. Kanika Garg, *Mastering Natural Language Processing using Python*, S Chand and Sons, 2025, ISBN: 978-9349290624.
2. Lewis Tunstall, Leandro von Werra, Thomas Wolf, *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*, O'Reilly, 2022, ISBN: 9781098136789.

3. Jacob Perkins, *Python Text Processing with NLTK 2.0 Cookbook*, Packt Publishing. (https://karczmarczuk.users.greyc.fr/TEACH/TAL/Doc/python_text_processing_with_nltk_2.0_cookbook.pdf)

Reference Websites:

1. https://pypi.org/project/spacy/
2. https://www.nltk.org/
3. https://www.datacamp.com/tutorial/text-analytics-beginners-nltk
4. https://huggingface.co/

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-356-MJ-P: Practical based on AIML-352-MJ-T
(Unsupervised Machine Learning)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Pre-requisites: Working knowledge of Python programming, Familiarity with NumPy, Pandas, and Scikit-learn, Understanding of data preprocessing and feature representation

Objectives:
The course is designed to teach:

- hands-on experience with unsupervised learning techniques
- exploring hidden patterns and structures in unlabeled datasets
- implementation of clustering, dimensionality reduction, and association mining algorithms
- methods to analyze high-dimensional data using unsupervised learning methods

Data Repositories

- Kaggle Datasets – https://www.kaggle.com/datasets
- UCI Machine Learning Repository – https://archive.ics.uci.edu
- Govt. of India data sets - https://indiaai.gov.in/datasets
- Scikit-learn Built-in Datasets

Contents:

Unit	Description
1	Program to load an unlabeled dataset and study the nature of unlabeled data using basic exploratory analysis.
2	Program to compute similarity and distance measures for an unlabeled dataset.
3	Program to perform feature representation and normalization for unsupervised learning tasks.
4	Program to implement K-Means clustering for grouping unlabeled data.
5	Program to analyze clustering results for different values of K in K-Means clustering.
6	Program to implement Hierarchical clustering for unlabeled data.
7	Program to implement Density-Based clustering (DBSCAN) for identifying clusters and noise points.
8	Program to study the effect of the curse of dimensionality on high-dimensional data.
9	Program to apply Principal Component Analysis (PCA) for dimensionality reduction.
10	Program to apply Independent Component Analysis (ICA) for feature extraction.
11	Program to visualize high-dimensional data using manifold learning techniques.
12	Program to implement the Apriori algorithm for discovering frequent itemsets from transaction data.
13	Program to generate and analyze association rules using support, confidence, and lift measures.
14	Program to perform anomaly and outlier detection on unlabeled data.
15	Program to implement a basic autoencoder for unsupervised feature learning.

Course Outcomes:
After competing this course, a student will be able to:

CO1: Implement unsupervised learning algorithms using Python
CO2: Perform clustering and interpret grouping results
CO3: Apply dimensionality reduction techniques for feature analysis
CO4: Extract association rules from large datasets
CO5: Use unsupervised learning methods for exploratory data analysis

Text Books:

- Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 3rd Edition, 2011.
- Ethem Alpaydin, *Introduction to Machine Learning*, MIT Press, 4th Edition, 2020.
- Sebastian Raschka and Vahid Mirjalili, *Machine Learning with Python*, Packt Publishing, 2017.

T.Y. B.Sc. (AI and ML)
Semester – VI
Major Elective
AIML-360-MJ-T: Business Intelligence

Prerequisites: Fundamentals of Data and Information, Basic Mathematics and Statistics, Spreadsheet Skills, Logical and Analytical Thinking.

Objectives:

The course is designed to teach students:

- understand the fundamentals of Business Intelligence
- learn data preparation using Power BI
- design interactive reports and dashboards
- analyze and communicate insights effectively

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Introduction to Business Intelligence and Power BI:** Business Intelligence concepts, Components of BI, Introduction to Power BI, Power BI Architecture, Power BI Desktop Interface, Power BI Workflow.	6	1
2	**Data Sources and Data Preparation in Power BI:** Connecting to data sources (Excel, CSV), Power Query Editor, Data profiling, Handling missing values, Removing duplicates, Data type transformation	6	1
3	**Data Modeling and DAX Basics:** Data modeling concepts, Relationships, Cardinality, Calculated columns, Measures, Basic DAX functions (SUM, AVERAGE, COUNT, IF)	6	1
4	**Data Visualization in Power BI:** Charts and visuals (Bar, Line, Pie, Table, Card), Filters and slicers, Formatting visuals, Best practices for visualization	6	2
5	**Reports, Dashboards, and Publishing:** Report creation, Interactive dashboards, Drill-down and drill-through, Sharing and publishing reports, Power BI Service overview, Case studies	6	2

Course Outcomes:

After completing this course, students will be able to:

CO1: Understand Business Intelligence concepts and Power BI architecture
CO2: Prepare and clean data using Power BI
CO3: Create data models and basic DAX measures
CO4: Design effective reports and dashboards
CO5: Publish and communicate insights using Power BI

Text Book(s)

1. Russo, M., & Ferrari, A., *Introducing Microsoft Power BI*, Microsoft Press.
2. Few, S., *Show Me the Numbers*, Analytics Press.

Reference Books

1. Gil Raviv, *Collect, Combine, and Transform Data Using Power Query in Power BI*, Microsoft Press.
2. Cairo, A., *The Functional Art*, New Riders.

T.Y. B.Sc. (AI and ML)
Semester – VI
Major Elective
AIML-361-MJ-P: Practical based on AIML-360-MJ-T
(Business Intelligence)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Pre-requisites: Fundamentals of Data and Information, Spreadsheet Skills, Basic Computer Skills.

Objectives:
The course is designed to teach students:

- data loading and preparation using Power BI
- data cleaning and transformation techniques using Power Query
- create effective visualizations and analytical reports
- design interactive dashboards and communicate insights through data-driven storytelling

Contents:

Unit	Description
1	Introduction to Business Intelligence and Power BI: Installation, Power BI Desktop interface, Power BI workflow
2	Connecting data sources in Power BI: Excel, CSV files
3	Understanding data model: tables, fields, data types
4	Data cleaning using Power Query: handling missing values, removing duplicates
5	Data transformation: filtering, sorting, renaming columns, data type conversion
6	Creating relationships between tables
7	Creating calculated columns and measures
8, 9	Data visualization using Power BI: Bar chart, Line chart, Pie chart, Table, Card visuals
10	Filters, slicers, and drill-down features
11	Formatting and customization of visuals
12	Designing reports and layout best practices
13	Dashboard creation and interactivity
14,15	Mini Project: Data preparation, report and dashboard development using Power BI

Note: A Lab book shall be prepared related to the practicals that a student needs to perform. Real-world datasets should be used wherever possible.

Course Outcomes:
After competing this course, a student will be able to:
CO1: Load and prepare data using Power BI
CO2: Clean and transform data using Power Query
CO3: Create visualizations and reports
CO4: Design interactive dashboards
CO5: Communicate insights effectively using Power BI

Recommended Tools & Softwares:
Microsoft Power BI Desktop, Tableau, LookerStudio

Text Books:

1. Russo, M., & Ferrari, A., *Introducing Microsoft Power BI*, Microsoft Press.
2. Few, S., *Show Me the Numbers*, Analytics Press.

T.Y. B.Sc. (AI and ML)
Semester – VI
Major Elective
AIML-362-MJ-T: AI Assisted Software Development

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Basic programming knowledge (Python/C++), fundamentals of software engineering.

Objectives:
The course is designed to:

- Introduce students to the role of Artificial Intelligence in modern software development workflows.
- Familiarize students with AI-assisted tools for coding, debugging, documentation, and project management.
- Develop the ability to critically analyze the strengths, limitations, and ethical implications of AI in software engineering.
- Encourage hands-on practice with AI tools to enhance productivity and creativity in software development.
- Prepare students for future industry practices by integrating AI into software pipelines and project workflows.

Contents:

Unit	Contents	Number of lectures	Text Book
1	**Introduction to AI in Software Development**- Evolution of software development and role of AI, Overview of AI-assisted tools (Copilot, ChatGPT, Tabnine, CodeWhisperer, Antigravity, Gemini etc), Benefits and challenges of AI integration.	**4**	**1,2**
2	**AI-Assisted Coding and Debugging -** Code generation using AI tools (examples in Python/Java/VS Code), AI-assisted debugging and error detection, Automated documentation and test case generation.	**12**	**1,2**
3	**Software Engineering Practices with AI-** AI in requirement analysis, design patterns, and architecture suggestions, AI-assisted project management (task tracking, agile support), Integration of AI tools with IDEs and CI/CD pipelines, Introduction to Robotic Process Automation (RPA), Case studies of industry adoption.	**8**	**1,2**
4	**Ethics, Interpretability, and Future Trends -** Ethical considerations: plagiarism, bias, intellectual property, Interpretability and trust in AI-generated code, Productivity vs dependency debate, Future of AI in software engineering (autonomous coding agents).	**4**	**1,2**

Course Outcomes:
After competing this course, a student will be able to:

CO1: Understand the applications of AI in software development and its impact on industry practices.
CO2: Apply AI-assisted tools for code generation, debugging, documentation, and requirement analysis.
CO3: Analyze AI-generated solutions to evaluate efficiency, correctness, and adaptability across programming languages.
CO4: Evaluate ethical, interpretability, and productivity aspects of AI-assisted development.
CO5: Create small-scale projects integrating AI-assisted tools into software pipelines, demonstrating practical competence.

Text books:

1. Meir Kalech, Rui Abreu, Mark Last, *Artificial Intelligence Methods For Software Engineering*, World Scientific, 2021, ISBN-9811239932, 9789811239939.
2. Sergio Pereira, *Generative AI for Software Development*, O'Reilly, 2025, ISBN: 9781098162269

Reference Books:

1. Research papers & online resources: IEEE Software, ACM Transactions on Software Engineering, GitHub Copilot documentation, AWS CodeWhisperer docs.

T.Y. B.Sc. (AI and ML)
Semester – VI
Major Elective
AIML-363-MJ-P: Practical based on AIML-362-MJ-T
(AI Assisted Software Development)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Prerequisites: Basic programming knowledge (Python/C++), fundamentals of software engineering.

Objectives:
The course is designed to:

- Introduce students to the role of Artificial Intelligence in modern software development workflows.
- Familiarize students with AI-assisted tools for coding, debugging, documentation, and project management.
- Develop the ability to critically analyze the strengths, limitations, and ethical implications of AI in software engineering.
- Encourage hands-on practice with AI tools to enhance productivity and creativity in software development.
- Prepare students for future industry practices by integrating AI into software pipelines and project workflows.

Contents:

Unit	Description
1	Explore GitHub Copilot/ChatGPT to generate a simple Python function (e.g., factorial, Fibonacci etc.).
2	Compare AI-generated code with manually written code for efficiency and readability
3	Document the advantages and limitations of at least two AI-assisted coding tools.
4	Use an AI tool to generate test cases for a given program (e.g., calculator app).
5	Debug a faulty piece of code using AI suggestions and explain corrections.
6	Generate automated documentation for a small project (e.g., student record system).
7	Compare AI-generated solutions for the same problem across different languages (Python vs Java or any other language).
8	Use AI to suggest design patterns for a small E-commerce application.
9	Integrate AI-assisted tools into an IDE (e.g., VS Code with Copilot) and demonstrate workflow.
10	Simulate requirement analysis using AI prompts (e.g., “Generate user stories for a library management system”).
11	Conduct a case study on industry adoption of AI tools (e.g., Microsoft, Amazon, Google)
12	Evaluate plagiarism risks by comparing AI-generated code with open-source repositories.
13	Conduct a debate/report on ethical issues in AI-assisted development (bias, IP rights).
14	Build a mini-project (e.g., to-do list app) using AI-assisted coding, debugging, and documentation
15	Reflective journal: Write a report on personal learning outcomes and future scope of AI in software engineering.

Course Outcomes:

After competing this course, a student will be able to:

- CO1: Understand the applications of AI in software development and its impact on industry practices.
- CO2: Apply AI-assisted tools for code generation, debugging, documentation, and requirement analysis.
- CO3: Analyze AI-generated solutions to evaluate efficiency, correctness, and adaptability across programming languages.
- CO4: Evaluate ethical, interpretability, and productivity aspects of AI-assisted development.
- CO5: Create small-scale projects integrating AI-assisted tools into software pipelines, demonstrating practical competence.

Text books:

1. Meir Kalech, Rui Abreu, Mark Last, *Artificial Intelligence Methods For Software Engineering,* World Scientific, 2021, ISBN-9811239932, 9789811239939.
2. Sergio Pereira, *Generative AI for Software Development,* O'Reilly, 2025, ISBN: 9781098162269

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-364-MJ-T: Game Theory

No. of Credits: 2	Teaching scheme: Theory: 2 hrs/week	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Lectures: 30

Prerequisites: Python Programming, Discrete mathematics and basic probability theory

Objectives:
The course is designed to teach:

- Foundational principles of game theory, strategic game forms and concepts.
- Implement and simulate game-theoretic models under perfect and uncertain information.
- Utility theory concepts used in game theory
- Applying beliefs in game design
- Implementing concepts from bargaining theory in game development.

Contents:

Unit	Contents	Number of lectures	Text Book
1	Introduction- Introduction to game theory, games and solutions, theory of competitive equilibrium, rational behaviour, strategies, payoffs and representing payoff matrices. Normal form games, Zero-sum (min max, max min) vs non-zero-sum games, dominance. Implementing simple 2-player zero-sum games. Pure vs mixed strategies, Rock-Paper-Scissors simulation in Python. Strategic Games- Ordinal games in Strategic form - Game frames and games, Strict and weak dominance, Second-price auction, The pivotal mechanism, Iterated deletion procedures. Nash Equilibrium – definition, significance, mixed, corelated and evolutionary equilibrium. Games with infinite strategy sets – Bayesian Games.	**8**	**1,2**
2	Extensive games with Perfect information- Trees, frames and games, Backward induction, Strategies in perfect-information games General Dynamic Games- Imperfect Information, Strategies, Subgames, Subgame-perfect equilibrium, Games with chance moves Prisoner's Dilemma repeated game simulation. Cooperative vs non-cooperative games. Coalition formation using NetworkX.	**9**	**1,2**
3	Games with Cardinal Payoffs- Expected Utility Theory- Money lotteries and attitudes to risk, The axioms in Expected utility. Interactive Epistemology - Individual knowledge, Interactive knowledge, Common knowledge, Belief, Interactive Belief and Common Beliefs, Belief updation	**9**	**1,2**

	and revision, Harsanyi consistency of beliefs or like-mindedness, agreeing to disagree		
4	Bargaining Games - Bargaining and Game Theory, A Bargaining Game of Alternating Offers, Subgame Perfect Equilibrium, Variations and Extensions	**4**	**1,2**

Course Outcomes:

After competing this course, a student will be able to:

CO1: Narrate the basics of game theory
CO2: Explain the concept of equilibrium and its importance in game theory
CO3: Compare and analyze different strategies (pure vs mixed, cooperative vs non-cooperative)
CO4: Realize the game design strategies under perfect and imperfect information.
CO5: State the concepts in computation of cardinal payoffs in games
CO6: Apply interactive epistemology in games
CO7: Rationalize the strategy of bargaining in game theory

Text books:

1. Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, 2024, ISBN 0-262-65040-1 (free download available)
2. Giacomo Bonanno, *Game theory*, Createspace Independent Publishing Platform, 2nd edition, 2018, ISBN: 978-1985862517.

Reference websites:

1. https://nashpy.readthedocs.io/en/stable/text-book/normal-form-games.html

T.Y. B.Sc. (AI and ML)
Semester – VI
Elective
AIML-365-MJ-P: Practical based on AIML-364-MJ-T
(Game Theory)

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Objectives:
The course is designed to teach:

1. installation of various Python libraries and engines used in game development
2. programmatically represent competitive scenarios using payoff matrices.
3. implementation of complex decision-making logic, such as the Tit-for-Tat strategy in repeated games and coalition formation using NetworkX.
4. design, develop, and deploy a feature-complete, two-player 2D or 3D game. This includes everything from initial asset integration in a modern engine (like Arcade or Panda3D) to final distribution using packaging tools like py2exe.
5. implementation of event-driven programming, handling user inputs, sprite animations, and real-time screen updates.

Contents:

Unit	Description
1	Python setup, Nashpy and NumPy basics, the turtle module, Python Game engines (PyGame Zero/ Pyglet / Arcade/ Wasabi 2D/ Panda 3D or any latest),
2	Best response functions in Python. SciPy optimization for mixed strategies. Repeated games, Tit-for-Tat strategy. (Games such as Hawk Dove Game, Pigs, Matching Pennies using prisoners dilemma)
3, 4	Use of curses, turtle or any other UI Library, PyOpenGL, py2exe
4	Rock-Paper-Scissors simulation, Nash Equilibrium computation
5	Mixed strategy equilibrium (SciPy), Repeated Prisoner's Dilemma
6	Tit-for-Tat strategy implementation, Coalition formation (NetworkX)
7	Tic-tac-Toe game implementation
8, 9	Snakes and ladders games
10	Using any python game engine to develop a 2D / 3D game (any library and any engine can be used) – Game can be 8-puzzle, snake game, angry birds, pacman etc.
11-15	Game development project (game should be other than the one done in assignment 10 and game should be played between 2 players)

Course Outcomes:
After competing this course, a student will be able to:

CO1: Install python libraries for game development
CO2: Use various python libraries for gaming user interface
CO3: Demonstrate the concepts of gaming such as equilibrium, mixed strategy, prisoner's dilemma
CO4: Develop a mini- game using the game theory concepts and translate theoretical game theory concepts into computational models and demonstrate the ability to transition from CLI-based scripts to rich visual environments.
CO5: Develop a feature-complete 2D simulation game (between 2 players)

Text books:

1. Will McGugan, *Beginning Game Development with Python and Pygame: From Novice to Professional*, APress, 2007, ISBN: 978-1-59059-872-6 (Free download available)
2. Sachin Kafle*, Learning Python by Building Games*, Packt Publishing, 2019, ISBN 978-1-78980-298-6.

T.Y. B.Sc. (AI and ML)
Semester – VI
VSC
AIML-371-VSC-P: Database Technologies

No. of Credits: 2	Practicals per week: 1 (4 hrs)	Evaluation Pattern: Continuous Evaluation: 15 End Sem Evaluation: 35	Total Practicals: 15

Pre-requisites: Knowledge of relational databases and SQL, Familiarity with PostgreSQL or any RDBMS, Basic programming knowledge

Objectives:
The course is designed to teach:

1. modern unstructured and semi-structured database technologies.
2. hands-on experience with document-oriented databases using MongoDB.
3. students to work with graph-based data using Neo4j.
4. skills required for selecting suitable databases for real-world applications.
5. database skills with industry requirements for AI and data-driven systems.

Contents:

Unit	Contents
1-5	**Introduction to NoSQL Databases:** Types of NoSQL databases, Document, Key-Value, Column, Graph, JSON and BSON formats, Challenges of relational databases in modern applications Applications and emerging trends of NoSQL databases, MongoDB installation and environment setup
6,7,8	**Working with MongoDB**: Creating databases and collections, Inserting, querying, updating, and deleting documents, Filtering and sorting data, Embedded documents and arrays, Indexing concepts, Aggregation pipeline overview, Data modeling patterns in MongoDB
9,10,11	**Introduction to Graph Databases:** Graph database concepts, Nodes, relationships, and properties, Neo4j installation and setup, Neo4j Browser and Desktop, Creating and managing graph data, Introduction to Cypher query language
12-15	**Graph Querying and Applications:** Pattern matching using Cypher, Relationship traversal and path queries, Graph-based data modeling, Applications of graph databases in social networks, recommendation systems, fraud detection, Comparison of document and graph databases **Vector Databases:** Introduction, ChromaDB, Milvus, Qdrant

Note: A Lab book shall be prepared related to the practicals that a student needs to perform.

Course Outcomes:
After competing this course, a student will be able to:

CO1: Explain the need for unstructured databases and NoSQL systems.
CO2: Perform CRUD and aggregation operations using MongoDB.
CO3: Model and query graph data using Neo4j and Cypher.
CO4: Model and query graph data using Neo4j and Cypher.
CO5: Select appropriate database technologies for scalable AI and data-driven applications.

Text Books:

1. Kristina Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media.

2. Ian Robinson, Jim Webber, Emil Eifrem, *Graph Databases*, O'Reilly Media.
3. Rick Copeland, *MongoDB Applied Design Patterns*, O'Reilly Media.
4. Alira Vexel, *Vector Databases in Practice: Build RAG & AI Search with Qdrant, Milvus & Open-Source Tooling*, ASIN: B0GG6Y5NFV, 2026 (Kindle Ed.)

Reference Books:

1. Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann.
2. Ethem Alpaydin, *Introduction to Machine Learning*, MIT Press.
3. Jim Webber, *Graph Databases: New Opportunities for Connected Data*, O'Reilly Media.

Web Resources:

1. MongoDB Documentation: https://www.mongodb.com/docs/
2. MongoDB University: https://university.mongodb.com
3. Neo4j Documentation: https://neo4j.com/docs/
4. Neo4j GraphAcademy: https://graphacademy.neo4j.com
5. NoSQL Database Concepts: https://www.geeksforgeeks.org/nosql-databases/

T.Y. B.Sc. (AI and ML)
Semester – VI
AIML-381-OJT: On Job Training

No. of Credits: 4	1/2/4/6/8 hrs per week	Evaluation Pattern: Continuous Evaluation: 40 End Sem Evaluation: 60	Total Hours (OJT): 120

Objectives:
The course is designed to:

- teach the difference between academic learning and professional work environments.
- cultivate hands-on experience to develop technical, analytical, and problem-solving skills.
- enhanced communication, teamwork, and interpersonal skills through industry interaction.
- Teach employment of time management and organizational abilities in a professional setting and familiarize students with industry standards, tools, technologies, and workflows and expose students to professional ethics, workplace culture, and discipline.
- cultivate the realities of employment opportunities by building confidence and competence and encourage professional networking and relationship-building within the industry.
- promote a mindset of lifelong learning and adaptability in dynamic industry environments and encourage students to reflect on experiences and integrate lessons into academic growth.

Instructions for OJT-

1. On Job training is *not Industrial Training.*
2. Students will be guided by an industry supervisor and monitored by a faculty coordinator
3. On Job Training needs to be done in industry (IT / ITES (IT Enabled Services) or Manufacturing where ERP/MIS/SAP Departments are there).
4. Students must complete a minimum of 120 hours of industry engagement during the OJT period.
5. While doing OJT students should not miss/bunk college sessions.
6. Students undergoing OJT may allocate 1, 2, 4, or 8 hours per day, or complete the required hours in continuous sessions, subject to industry slot availability.
7. Students are required to provide a letter from the industry certifying their participation in OJT.
8. Attendance must be regular and punctual; any absence should be communicated to both the industry supervisor and the academic coordinator.
9. Students are expected to actively participate in assigned tasks and projects. They should observe industry practices, workflows, and professional standards. Students must respect workplace rules, dress codes, and safety protocols at all times and avoid misuse of company resources and adhere to ethical standards
10. During OJT it is expected that student must able to understand following,
    - Industry type (product based/ service based etc.)
    - The Clients of Industry.
    - Project undergoing
    - Software/tools getting used
    - Coding standards followed.
    - Understanding industry etiquettes.
    - Understand the work culture of the company.
    - If company assigns any Assignments/ small module that they have to Complete.
11. On completion of OJT students have to prepare a report comprising all above and have

to submit along with OJT completion letter.

12. There will be final presentation of OJT which will be assesed by internal and external examiner.

Guidelines for the faculty coordinator:

1. Maintain a weekly record of the progress of the student from the industry supervisor/mentor/SPoC. Sample formats provided below.
2. At the end of the OJT, get a report from the student summarizing activities, skills gained, and reflections on industry exposure.
3. Evaluation of a student may be done based on the following guidelines-
    a. Behaviour, attendance, punctuality and professionalism
    b. Logbook entries and final report quality
    c. Industry feedback on performance and attitude
    d. Learning outcomes achieved

SAMPLE FORMAT OF OJT COMPLETION CERTIFICATE*
(on the LETTER HEAD of the company/organization)

Date:

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./ Ms. _____ student of _____ college studying in T.Y.B.Sc(Computer Science) (VIth semester) has successfully completed his/her On Job Training (OJT) with us for _____ hours. During the period of training he/ she worked under in the following areas.

I. _____
II. _____ (add rows here if required)

He / She has shown special flair for and his/her performance in preparation of the report has been rated as ____ (in 1 to 10 Points/Grade). During the period his/her internship program he/she was punctual and hardworking. I wish Mr. /Ms. _____ every success in his/her career and life.

Signature
Mentor of the Organization

SAMPLE FORMAT of Student Diary (Log)*

Recording Format

Date		Week Task Assigned	Activities Performed	Key Learnings	Signature of Mentor
From	To				

* - can be used as a guideline only, colleges can add or modify the contents of the SAMPLE formats

Course Outcomes:

After competing this course, a student will be able to:

CO1: Apply classroom knowledge to real-world industry problems.

CO2: Gain exposure to tools, technologies, and processes used in the field.

CO3: Develop professional skills such as teamwork, communication, and time management.

CO4: Understand industry expectations and workplace ethics.