Semester	**2**
Programme	**CHEMISTRY**
Course	**MULTIDISCIPLINARY (AI FOR NATURAL SCIENCES)**
Paper Code	**M1CH250211P**
Credits	**3 (PRACTICAL: 3)**
Hours/week	**3 (PRACTICAL: 3)**
Category: Core/MDC/SEC/VEC	**MDC**
Theory/Practical/Composite	**PRACTICAL**
Number of Modules	**3**
Syllabus	**Module I: Programming Fundamentals (12 Lectures)** 1. Python Basics: •Variables and data types (integers, floats, strings, lists, dictionaries, etc.). •Input and output (e.g., reading and writing files). •Control flow: loops (for, while) and conditional statements (if, elif, else). •Functions and modular programming. 2. Introduction to Jupyter Notebooks: •Setting up and running Jupyter Notebooks. •Markdown for documentation. •Writing and executing Python code interactively. 3. Python Libraries: •NumPy: Arrays, basic mathematical operations, and array manipulation. •Pandas: DataFrames, importing/exporting data, and basic data analysis. •Matplotlib: Creating 2D plots, customizing graphs, and visualizing chemical data. **Module II: Mathematics for Machine Learning (12 Lectures)** 1. Linear Algebra: •Vectors and vector operations. •Matrices, matrix operations, and their properties. •Eigenvalues, eigenvectors, and matrix diagonalization. •Singular value decomposition (SVD) and its applications in dimensionality reduction. 2. Calculus:

	•Differentiation: Derivatives, partial derivatives, and gradients. •Chain rule and backpropagation in neural networks. •Optimization techniques: Gradient descent and its variants. •Taylor series expansion and its role in approximations. 3. Probability and Statistics: •Random variables and probability distributions. •Expected value, variance, and standard deviation. •Bayes' theorem and its applications. •Hypothesis testing and confidence intervals. **Module III: Introduction to Machine Learning and Artificial Intelligence (12 Lectures)** 1. Supervised Learning: • Regression: Linear regression, polynomial regression. • Classification: Logistic regression, decision trees, and support vector machines (SVMs). • Metrics: Mean squared error, accuracy, precision, recall, and F1-score. 2. Unsupervised Learning: • Clustering: K-means clustering, hierarchical clustering. • Dimensionality reduction: Principal component analysis (PCA), t-SNE. 3. Introduction to Neural Networks: • Basics of perceptrons and multi-layer perceptrons (MLPs). • Activation functions: Sigmoid, ReLU, and softmax. • Forward propagation and loss functions.
Course Overview	A course at the ntersection of computer science, advanced mathematics, and artificial intelligence within scientific research. Programming Fundamentals (Module I): Introduces Python programming, focusing on variables, control flow, functions, and the interactive execution of code using Jupyter Notebooks. It emphasizes building data analysis pipelines and visualizing chemical trends using essential libraries like NumPy, Pandas, and Matplotlib. Mathematics for Machine Learning (Module II): Establishes the rigorous mathematical foundation required for predictive modeling. Topics include linear algebra (matrices, eigenvalues, SVD), calculus (partial derivatives, gradient descent, chain rule), and probability/statistics (distributions, Bayes' theorem, hypothesis testing). Introduction to ML and AI (Module III): Transitions from mathematics to algorithmic implementation. It covers supervised learning (linear/polynomial regression, SVMs, decision trees), unsupervised learning (K-means, hierarchical clustering, PCA), and the construction of simple neural networks (MLPs, activation functions, forward propagation) using modern frameworks. 1) Write Python scripts to solve simple problems. 2) Create a basic data analysis pipeline using Pandas.

	3) Plot chemical property trends using Matplotlib. 4) Document code and results using Jupyter Notebooks. 5) Solve linear algebra problems related to data representation and transformations. 6) Compute gradients and optimize simple machine learning cost functions. 7) Analyze datasets using statistical measures and apply probabilistic reasoning to model predictions 8) Implement and evaluate a linear regression model on a dataset. 9) Apply clustering algorithms to chemical datasets (e.g., grouping molecules based on properties). 10) Build a simple neural network using Python libraries such as TensorFlow or PyTorch.
Course Outcomes	Upon successful completion of this practical course, the student will be able to: **CO1 [Python & Modular Programming]:** Develop functional Python scripts using modular programming constructs (functions, control flow) and utilize Jupyter Notebooks with Markdown for interactive coding and documentation. **CO2 [Data Pipelining]:** Perform data import/export, cleaning, and array/DataFrame transformations on scientific datasets using NumPy and Pandas. **CO3 [Graphical Visualization]:** Create and customize 2D plots using Matplotlib to visualize chemical property trends and dataset distributions. **CO4 [Linear Algebra & SVD]:** Solve linear algebra problems involving vector and matrix operations, eigenvalue decomposition, and Singular Value Decomposition (SVD) for data representation and dimensionality reduction. **CO5 [Calculus & Gradient-Based Optimization]:** Compute partial derivatives, gradients, and Taylor series approximations to optimize machine learning cost functions using gradient descent and the chain rule (backpropagation). **CO6 [Statistical Analysis]:** Analyze datasets using statistical measures (expected value, variance) and apply probabilistic reasoning, including Bayes' theorem, hypothesis testing, and confidence intervals, to draw data-driven inferences. **CO7 [Supervised Learning & Model Evaluation]:** Build supervised learning models (linear/polynomial regression, logistic regression, decision trees, SVMs) and evaluate their performance using metrics such as MSE, accuracy, precision, recall, and F1-score. **CO8 [Unsupervised Learning & Dimensionality Reduction]:** Apply clustering algorithms (k-means, hierarchical) and

	dimensionality reduction techniques (PCA, t-SNE) to identify patterns and groupings in chemical/molecular datasets. **CO9 [Neural Network Implementation]:** Construct, train, and evaluate multi-layer perceptrons (MLPs) using activation functions (sigmoid, ReLU, softmax) with Python-based libraries such as TensorFlow or PyTorch.
Program Outcomes	**PO1: Fundamental Knowledge:** Demonstrate a comprehensive understanding of core concepts in chemistry, including organic, inorganic, physical, analytical, and biochemistry. **PO2: Laboratory Skills:** Acquire hands-on experience in laboratory techniques and instrumentation, applying safety protocols and proper methodologies to conduct experiments and analyse results. **PO3: Problem Solving:** Develop critical thinking and problem-solving skills by applying chemical principles to real-world scenarios, including the ability to design experiments and interpret data. **PO4: Research Competency:** Engage in independent and collaborative research projects, employing scientific methods to formulate hypotheses, conduct experiments, and communicate findings effectively. **PO5: Interdisciplinary Applications:** Understand the interdisciplinary nature of chemistry, exploring its applications in fields such as medicine, environmental science, materials science, and nanotechnology. **PO6: Communication Skills:** Effectively communicate scientific information, both verbally and in written form, to diverse audiences, demonstrating the ability to present complex ideas clearly and concisely. **PO7: Ethics and Sustainability:** Recognize the ethical implications and responsibilities of scientific research and practice in chemistry, emphasizing the importance of sustainability and environmental stewardship in chemical practices.

CO #	Core Curriculum Domain	Bloom’s Taxonomy (K-Level)	Mapped POs
CO1	Python & Jupyter Fundamentals	**K3** (Apply)	PO2, PO6
CO2	NumPy & Pandas Data Manipulation	**K3** (Apply), **K4** (Analyze)	PO2, PO5
CO3	Matplotlib Visualization	**K3** (Apply), **K6** (Create)	PO2, PO6
CO4	Linear Algebra & SVD	**K3** (Apply)	PO3, PO5
CO5	Calculus & Gradient Optimization	**K4** (Analyze)	PO3, PO5
CO6	Probability & Hypothesis Testing	**K4** (Analyze), **K5** (Evaluate)	PO3, PO4
CO7	Supervised Learning & Model Evaluation	**K4** (Analyze), **K5** (Evaluate)	PO3, PO4
CO8	Unsupervised Learning & Dimensionality Reduction	**K4** (Analyze), **K5** (Evaluate)	PO3, PO5
CO9	Neural Network Implementation	**K6** (Create)	PO2, PO4

Reading/Reference Lists	Books: 1) Python Crash Course: A Hands-On, Project-Based Introduction to Programming by Eric Matthes. 2) Automate the Boring Stuff with Python by Al Sweigart. 3) Mathematics for Machine Learning by Marc Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. 4) Linear Algebra and Its Applications by Gilbert Strang. 5) All of Statistics: A Concise Course in Statistical Inference by Larry Wasserman. 6) Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aur´elien G´eron. 7) Pattern Recognition and Machine Learning by Christopher M. Bishop. Online Resources: 1) https://www.learnpython.org/ (Free, interactive Python tutorials). 2) https://realpython.com/ (Comprehensive Python tutorials). 3) https://www.khanacademy.org/math/linear-algebra (Linear algebra tutorials). 4) https://www.probabilitycourse.com/ (Comprehensive probability and statistics guide). 5) https://www.coursera.org/learn/mathematics-for-machine-learning (Coursera course on Mathematics for Machine Learning). 6) https://scikit-learn.org/ (Documentation and tutorials for Scikit-learn). 7) https://www.coursera.org/learn/machine-learning (Andrew Ng's Machine Learning course). 8) https://www.tensorflow.org/tutorials (TensorFlow tutorials for beginners).
Evaluation	Practical: 50 Internal: 50 (CA: 45; Other form of Assessment: 2; Attendance: 3)