Syllabus template

Semester: 8	
Programme: Mathematics	
Course: Mathematics for Machine Learning and Artificial Intelligence	
Paper Code:C4MT230842T& C4MT230842P	**Credits: 6**
Hours/week: 3 (Th.) +2(Pr.)	
Category: Core/MDC/SEC/VAC: Core (Elective)	
Theory / Practical / composite: Composite	
No of Modules: Nil	

Course Overview: Mathematics for Machine Learning and Artificial Intelligence

This course equips students with the linear-algebraic, calculus-based, and optimization tools underlying modern machine learning and data science. Building on prerequisites in linear algebra, multivariable calculus, and probability, it integrates matrix factorizations with optimization theory and learning algorithms. The foundation block develops spectral decomposition and singular value decomposition, including SVD's properties and its central application to low-rank approximation. These factorizations are then applied to dimensionality reduction and classification through the derivation of Principal Component Analysis and Linear Discriminant Analysis. The statistics block covers least-squares approximation and minimum-norm solutions, leading to linear and multiple regression, logistic regression, and classification metrics for performance evaluation. Structural matrix theory is completed with polar decomposition, the minimal polynomial, and the Jordan canonical form. The optimization block introduces convex sets and convex functions, unconstrained and constrained convex programming, and the Karush–Kuhn–Tucker optimality conditions. Algorithmic methods — steepest descent, Newton's method, and penalty function techniques — provide the computational machinery for solving optimization problems. The course culminates in Support Vector Machines: error-minimizing linear programming, the idea of duality, and hard- and soft-margin classifiers. By connecting SVD, PCA, regression, convex optimization, and SVMs, the course forms a coherent bridge from pure mathematical structure to applied machine-learning methodology.

The practical course develops hands-on Python skills for linear algebra, calculus, and data analysis using NumPy and pandas, covering vectors, matrices, eigenvalues/eigenvectors, SVD, and low-rank approximation. Students then apply dimensionality-reduction techniques (PCA, LDA) and build supervised models—linear/multiple regression, logistic regression, and SVM —while learning to evaluate performance through standard classification metrics. The course concludes with optimization methods for model tuning, equipping students to implement, interpret, and refine complete machine-learning pipelines in Python

C4MT230842T

Course Outcome: On successful completion of the course a student will be able to do the following:

1.Explain the spectral theorem and singular value decomposition, state their defining properties, and apply these factorizations to construct low-rank approximations of matrices and

data.
2. Derive Principal Component Analysis from first principles and implement PCA and Linear Discriminant Analysis for dimensionality reduction and classification tasks.
3. Solve least-squares and minimum-norm approximation problems; build and interpret linear, multiple, and logistic regression models; and assess classifier performance using appropriate classification metrics.
4. Describe structural matrix decompositions — polar decomposition, the minimal polynomial, and the Jordan canonical form — and use them to analyze the canonical structure and spectral behavior of linear operators.
5. Analyze convex sets and convex functions and their properties; formulate unconstrained and constrained optimization problems, including linear and nonlinear convex programming; and establish necessary and sufficient optimality conditions via the Karush–Kuhn–Tucker (KKT) framework.
6. Critique and select numerical optimization algorithms — steepest descent, Newton's method, and penalty function methods — for given problem classes; develop Support Vector Machines through error-minimizing linear programming and duality; and design hard- and soft-margin classifiers, synthesizing the course's tools into a coherent machine-learning methodology.
C4MT230842P **Course Outcome:** On successful completion of the course a student will be able to do the following:
1.Define the basic data structures and functions of NumPy and pandas for handling vectors, matrices, and tabular data in Python.
2. Explain the concepts of eigenvalues, eigenvectors, and basic calculus operations in the context of linear algebra computations.
3.Apply NumPy and pandas to perform vector/matrix operations, compute eigenvalues/eigenvectors, and execute basic calculus-based computations in Python.
4. Apply Singular Value Decomposition (SVD) and low-rank approximation techniques to reduce dimensionality and approximate data matrices.
5. Apply Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for feature extraction and dimensionality reduction on real datasets.
6. Build linear, multiple, and logistic regression models in Python, interpreting coefficients and assessing goodness-of-fit.
7. Evaluate classification model performance using standard metrics (accuracy, precision, recall, F1-score, confusion matrix).
8. Apply optimization techniques to tune model parameters and improve predictive accuracy.
9. Implement Support Vector Machine (SVM) classifiers on real datasets, selecting appropriate kernels and hyperparameters.
10. Design an end-to-end machine-learning pipeline in Python (e.g., PCA-based feature reduction followed by SVM classification with metric-based evaluation).
Prerequisites:1. Vector and matrices.2. Vector space, subspace, basis, dimension, linear transformation, norms, convex set and function, inner product space and other spaces, orthogonal complement and projection mapping, Gram Schmidt method.3. Eigenvalue and

eigenvector, quadratic form, Rayleigh quotient, positive definiteness. .4. calculus I: Function domain range injective surjective, limit continuity derivative partial derivatives, gradient, directional derivative, calculus 5. Calculus II: Jacobian, Hessian matrix, vector and matrix gradient, chain rule, transformation and chain rule and other properties of Jacobian. 6. Probability, Bayes theorem random variable. Expectation and variance. discrete and continuous probability distribution, joint probability distribution and covariance corelation, regression.				
SYLLABUS				
UNIT/Module	**CONTENT**	**NO OF CLASSES**	**CO Mapping**	**COGNITIVE LEVEL**
	C4MT230842T [Theory]			
I.	1. Spectral decomposition [2] 2.Singular Value decomposition: its properties and applications, low rank approximation [2] 3.Idea of PCA: derivation [3] 4.Linear discriminant analysis [2] 5.Least square approximation and minimum normed solution, linear and multiple regression, logistic regression [3] 6. Classifications of metrics: performance evaluation [1] 7.Polar decomposition [1] 8. Minimal polynomial and Jordan canonical form [1] 9. Convex set and function and its. properties.[2] 10. Introduction to optimisation and NLP: unconstrained optimisation: maxima minima in higher order, constrain optimisation I: linear and nonlinear convex programming, constrain optimisation II: Karush Kuhn Tucker condition KKT [4] 11.Steepest decent method, Newton and penalty function method [2] 12.Introduction to SVM: support vector machine [1] 13.Error minimising LPP [2] 14.Idea of duality [1] 15.Hard and soft margin classifier [3]	**30**	**CO1,** **CO2,** **CO3,** **CO4,** **CO5,** **CO6**	**K1,** **K2,** **K3,** **K4,** **K5,** **K6**
II.	**C4MT230842P [Practical: [using Python]**			

	Practical: [using Python] 1. Introduction to python: vector and matrix, numPy, panda. Eigenvalue and eigenvector. Calculus [3] 2.SVD and low rank approximation [1] 4. PCA [1] 5. Linear discriminant analysis [1] 6. Linear and multiple regression [1] 7. Logistic regression through [1] 8. Classification of metrics: for performance matrix [1] 10. Optimisation using [1] 11.SVM [1]	**22**	**CO1, CO2, CO3, CO4, CO5, CO6, CO7, CO8, CO9, CO10**	**K1, K2, K3, K4, K5, K6**

Text Books

1. The hundred-page machine learning book: Andriy Burkov
2. Mathematics for Machine Learning. Deisenroth, M., Faisal, A. A., & Ong, C. S Cambridge University Press. (2020).

Suggested readings

- Introduction to Machine Learning, by Jeeva Jose, Khanna Book Publishing, 2020.

Web Resources and eLearning Sources: Strang 18.06/18.065 + NPTEL Linear Algebra, StatQuest + Andrew Ng CS229, NPTEL Linear Algebra + MIT 18.06, Boyd Convex Optimization + NPTEL Optimization Techniques, Andrew Ng SVM lectures + StatQuest + LIBSVM.

Evaluation: Theory CIA: **10+5**; Semester Exam: **45; Practical: CIA [38 +2]**

Paper Structure for Theory Semester Exam Module: 4 questions each carrying **10** marks out of 7 questions +**1** question carrying **5** marks out of **2** questions.

Course outcomes (COs) and Cognitive Level Mapping

C4MT230842T

COs	CO Description	Cognitive levels
CO1	**Explain** the spectral theorem and singular value decomposition, state their defining properties, and apply these factorizations to construct low-rank	**K1, K2, K3**

	approximations of matrices and data.	
CO2	**Derive** Principal Component Analysis from first principles and implement PCA and Linear Discriminant Analysis for dimensionality reduction and classification tasks.	**K3, K4**
CO3	**Solve** least-squares and minimum-norm approximation problems; build and interpret linear, multiple, and logistic regression models; and assess classifier performance using appropriate classification metrics.	**K3, K4**
CO4	**Describe** structural matrix decompositions — polar decomposition, the minimal polynomial, and the Jordan canonical form — and use them to analyze the canonical structure and spectral behavior of linear operators.	**K2, K3, K4**
CO5	**Analyze** convex sets and convex functions and their properties; formulate unconstrained and constrained optimization problems, including linear and nonlinear convex programming; and establish necessary and sufficient optimality conditions via the Karush–Kuhn–Tucker (KKT) framework.	**K4, K5**
CO6	**Critique** and select numerical optimization algorithms — steepest descent, Newton's method, and penalty function methods — for given problem classes; develop Support Vector Machines through error-minimizing linear programming and duality; and design hard- and soft-margin classifiers, synthesizing the course's tools into a coherent machine-learning methodology.	**K5, K6**

Course outcomes (COs) and Cognitive Level Mapping

C4MT230842P

COs	CO Description	Cognitive levels
CO1	**Define** the basic data structures and functions of NumPy and pandas for handling vectors, matrices, and tabular data in Python.	**K1**
CO2	**Explain** the concepts of eigenvalues, eigenvectors, and basic calculus operations in the context of linear algebra computations.	**K2**
CO3	**Apply** NumPy and pandas to perform vector/matrix operations, compute eigenvalues/eigenvectors, and	**K3**

	execute basic calculus-based computations in Python.	
CO4	**Apply** Singular Value Decomposition (SVD) and low-rank approximation techniques to reduce dimensionality and approximate data matrices.	**K3**
CO5	**Apply** Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for feature extraction and dimensionality reduction on real datasets.	**K3**
CO6	**Build** linear, multiple, and logistic regression models in Python, interpreting coefficients and assessing goodness-of-fit.	**K3**
CO7	**Evaluate** classification model performance using standard metrics (accuracy, precision, recall, F1-score, confusion matrix).	**K5**
CO8	**Apply** optimization techniques to tune model parameters and improve predictive accuracy.	**K3**
CO9	**Implement** Support Vector Machine (SVM) classifiers on real datasets, selecting appropriate kernels and hyperparameters.	**K3**
CO10	**Design** an end-to-end machine-learning pipeline in Python (e.g., PCA-based feature reduction followed by SVM classification with metric-based evaluation).	**K6**