PMAI LAB

Mathematics • Python • Artificial Intelligence

COURSE OUTLINE

Linear Algebra for Machine Learning

Mathematical Foundations, Python Practice, and Real-World Applications

Course focus Connecting linear-algebra theory with the mathematical structures and computational methods used in modern machine learning.

Dr. Hira Benish

Associate Professor of Mathematics

PMAI LAB | Ten-Lecture Course | Student Learning Resources

1 Course Profile

1.1 Course Introduction

Linear Algebra for Machine Learning connects essential mathematical foundations with practical data-driven applications. Students learn how vector and matrix methods support modern machine-learning models through Python and real-world examples.

1.2 Course Description

This course provides students with a strong foundation in Linear Algebra, emphasizing its applications in machine learning and data science. Topics include vectors, matrices, systems of linear equations, vector spaces, basis, dimension, linear transformations, orthogonality, projections, and least-squares regression. Students will also study eigenvalues, eigenvectors, singular value decomposition (SVD), principal component analysis (PCA), low-rank approximation, and data compression. The course further explores the role of linear algebra in neural networks, embeddings, attention mechanisms, recommendation systems, and gradient-based learning. Python and NumPy will be used throughout for computational practice and real-world case studies. By the end of the course, students will be able to model, analyze, and solve machine-learning problems using appropriate linear algebra methods.

1.3 Course Keywords

Machine Learning	**Vector Spaces**	**Dimensionality Reduction**
Neural Networks	**Data Modeling**	

1.4 Course Learning Outcomes

On successful completion of the course, students will be able to:

1. apply vectors, matrices, linear systems, and transformations to represent machine-learning problems;
2. analyze orthogonality, projections, eigenvalues, SVD, and PCA for approximation and dimensionality reduction;
3. implement linear-algebra methods using Python, NumPy, and relevant machine-learning libraries; and
4. evaluate and communicate solutions to regression, recommendation, compression, and neural-network problems.

Machine-Learning Connection

The course combines mathematical interpretation, hand calculations, verified Python code, interactive activities, and applied case studies so that students can connect theory with market-relevant machine-learning practice.

1.5 Course Structure

Duration	10 lectures
Core tools	Python, NumPy, Matplotlib, and scikit-learn
Learning approach	Mathematical explanation, worked examples, code labs, practice tasks, and case studies
Applied evidence	Retail regression, recommendation systems, PCA, image compression, and neural-network computations

2 Lecture-Wise Course Outline

No.	Lecture topic	Main learning focus
01	**Introduction and Linear Systems**	Data equations, Gaussian elimination, solution structure, and NumPy solving
02	**Vectors, Span, and Feature Spaces**	Vector operations, linear combinations, span, norms, and feature representation
03	**Vector Spaces, Basis, and Dimension**	Subspaces, independence, basis, coordinates, dimension, and rank-nullity
04	**Matrices as Linear Transformations**	Matrix mappings, composition, inverse transformations, and geometric interpretation
05	**Inner Products and Orthogonality**	Dot products, cosine similarity, orthonormal bases, projections, and Gram-Schmidt
06	**Least Squares and Regression**	Normal equations, residuals, pseudoinverse, regression, and regularization
07	**Eigenvalues and Eigenvectors**	Eigenspaces, diagonalization, repeated transformations, and spectral interpretation
08	**SVD, PCA, and Compression**	Singular values, low-rank approximation, dimensionality reduction, and image compression
09	**Neural Networks, Embeddings, and Attention**	Dense layers, learned vectors, attention matrices, gradients, and NumPy practice
10	**Integrated ML Case Studies and Review**	Reliable workflows, solver selection, validation, capstone planning, and course synthesis

2.1 Practical Learning Components

- Python and NumPy code in every lecture.
- Hand calculations and mathematical interpretation.
- Independent practice questions and self-check tasks.
- Interactive case studies based on retail sales and e-commerce recommendation.
- A portfolio-oriented capstone applying at least two course methods.

2.2 Expected Student Output

Students will complete executable notebooks, mathematical solutions, model comparisons, visual interpretations, and concise technical reports. Emphasis is placed on correct dimensions, stable computation, validation, reproducibility, and clear communication of limitations.

Practice Tasks

Suggested final project: Select a realistic dataset and compare two linear-algebra methods. Submit a reproducible notebook, evaluation evidence, visual results, and a one-page interpretation for a technical or business audience.

Course pathway: Vectors and systems → projections and regression → eigenmethods and SVD → neural networks and applied machine learning.