MATH 503 Syllabus, Term 251

Department of Mathematics
King Fahd University of Petroleum and Minerals

Course Information

Code:	MATH 503
Title:	Mathematics for Data Science
Credit Hours:	3-0-3
Prerequisite:	Graduate Standing

Instructor Information

Instructor:	Dr. Muhammad Imran Qureshi
E-mail:	imran.qureshi@kfupm.edu.sa
Office:	Building 5-405

Course Objectives

- Review selected topics from multivariate calculus, linear algebra, and optimization related to data science.
- Introduce data scientific software, toolboxes, and libraries.
- Solve problems in linear algebra and optimization topics related to data science.
- Application of mathematical topics to basic neural network design.

Course Description

Selected topics from linear algebra, multivariate calculus, and optimization for Data Science with an emphasis on the implementation using numerical and symbolic software, tool boxes, and libraries for data science like NumPy, SciPy, Pandas, and SymPy. Topics include:

- Data transformation using linear algebra.
- Vector spaces, linear transformations, matrix representations, and decompositions (eigenvectors, LU, QR, SVD, Cholesky).

- Multivariate calculus for continuous, convex, and non-convex optimization methods.
- Basic neural network design.

Textbook

Deisenroth, Faisal and Ong., *Mathematics for Machine Learning*, 2021 (Main reference).

Learning Outcomes

- Explain the mathematical background to solve data science problems.
- Identify calculus, linear algebra, and optimization topics related to each step of a data science problem.
- Apply computational tools in data science problems.
- Apply mathematical tools to neural network design.

References

1. Charu C. Aggarwal, *Linear Algebra and Optimization for Machine Learning*, 2020.
2. Thomas Nield, *Essential Math for Data Science*, 2022.

Grading Policy

Component	**Percentage**
Group Class Activities	10%
Group Assignments	10%
Projects and Presentations	25%
Attendance	5%
Mid Term	20%
Final Exam	30%

Table 1: Grading Policy Breakdown

Attendance

Attendance is a University Requirement. A DN grade is awarded after accumulating 6 unexcused absences. You can have two unexcused absences without penalty. You loose 1.25 points for each unexcused absence from the third absence onwards.

Academic Integrity

All KFUPM policies regarding ethics apply to this course.

Course Outline

Weeks	Reference	Topics
1	Chapter 1, Chapter 2 (2.1–2.2)	• Picture of Data Analytics Math Machine Learning • Data as vectors/matrices • Linear Algebra: Matrices and algebra of matrices • Systems of Linear Equations with a brief motivation (Linear Regression case study)
2	Chapter 2 (2.4–2.6)	• Linear Algebra: Vector Spaces • Understanding solvability of systems via Linear Independence, Basis, and Rank
3	Chapter 2 (2.3)	• Solving Systems of Linear Equations • Hands-on Illustration (Elementary Computation): – Solve linear systems using NumPy and SciPy – Check the rank of a matrix – Address challenges in solving rank-deficient problems – Motivate the idea of approximate solutions to linear systems

4	Chapter 2 (cont.), Chapter 3 (3.1–3.4)	• Linear Mappings, Analytic Geometry, Norms, Inner Products, Lengths and Distances, Angles, and Orthogonality
		• Code Illustration:
		  – Compute matrix and vector norms in NumPy
		  – Implement the k-Nearest Neighbor algorithm using NumPy and SciPy

5–6	Chapter 3 (3.5–3.9)	• Orthonormal Basis, Orthogonal Complement, Inner Product of Functions, Orthogonal Projections, Rotations
		• Code Illustration:
		  – Use NumPy and SciPy to implement least square approximation for fitting data to a straight line
		  – Apply LinearRegression in scikit-learn to the same dataset

7–8	Chapter 4 (4.1–4.5)	• Matrix Decomposition: Determinant and Trace, Eigenvalues and Eigenvectors, Cholesky Decomposition, Eigen-decomposition and Diagonalization, Singular Value Decomposition (SVD) • Code Illustration: – Compute Eigenvalues and Eigenvectors in NumPy – Perform PCA using NumPy and compare with scikit-learn – Solve column-rank deficient problems using SVD
9–10	Chapter 5 (5.1–5.5)	• Vector Calculus: Differentiation of Univariate Functions, Partial Differentiation and Gradients, Gradients of Vector-Valued Functions, Gradients of Matrices, Useful Identities for Computing Gradients
11	Chapter 5 (5.6–5.7)	• Backpropagation and Automatic Differentiation • Higher-Order Derivatives
12	Chapter 7 (7.1–7.2)	• Continuous Optimization: Gradient Descent, Constrained Optimization, and Lagrange Multipliers • Code Illustration: – Implement the Gradient Descent iteration in Python

13–14	Chapter 7 (7.3)	• Convex Optimization
15	—	• Project Presentations

Active Learning and Class Activities

Python programming language will be used to implement computational tasks in this course. Frequently used packages include:

- NumPy
- SciPy
- Core Python programming constructs (e.g., Loops, Masks, Lists)
- scikit-learn

Week(s)	**Topics and Related Applications**	**Tasks and Possible Goals**
3	Solving Systems of Linear Equations	Hands-on Illustration: • Use NumPy and SciPy to solve linear systems. • Check the rank of a matrix. • Address rank-deficient problems using SciPy. Goal: Motivate the idea of approximate solutions to linear systems.
4	Vector Norms, Inner Products, Lengths, and Distances	Hands-on Illustration: • Compute matrix and vector norms in NumPy. • Use distances to implement the k-Nearest Neighbor algorithm in NumPy and SciPy. Goal: Foundations for the kNN classifier.

6	Solving Linear Systems Revisited and Fitting Curves to Data (Regression)	Hands-on Illustration: • Implement least square approximation for fitting data to a straight line using NumPy and SciPy. • Apply Linear Regression in scikit-learn on the same dataset. Goal: Illustrate material in Chapters 2 and 3 as a foundation for linear regression.
8	Eigen-decomposition, Diagonalization, and Singular Value Decomposition (SVD)	Hands-on Illustration: • Compute Eigenvalues and Eigenvectors in NumPy. • Perform PCA using NumPy and compare with scikit-learn. • Use SVD to solve column-rank deficient problems (e.g., linear regression with highly correlated data). Goal: Highlight Eigen-decomposition and SVD as tools for dimensionality reduction and solving rank-deficient problems.
12	Optimization Using Gradient Descent	Hands-on Illustration: • Implement the Gradient Descent iteration in Python for finding least square solutions. Note: Many machine learning algorithms, such as linear regression and support vector machines, use gradient descent for optimization.