King Fahd University of Petroleum and Minerals
Department of Mathematics & Statistics
MATH 503 Syllabus, Term 241

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

Instructor: Dr. Jamal Al-Smail **E-mail:** jamalhas@kfupm.edu.sa
Office Hours:
Sundays & Tuesdays: 3:00 pm – 4:50 pm
Sundays & Tuesdays: 7:40 pm – 8:10 pm
Office: Building 5-407

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, toolboxes, and libraries for data science like NumPy, SciPy, Pandas, SymPy. Topics include data transformation using linear algebra, vector spaces, linear transformations, matrix representations, matrix decompositions (eigenvectors, LU, QR, SVD, Cholesky); multivariate calculus for continuous, convex, and non-convex optimization methods; basic neural network design.
Textbook:
Deisenroth et al, Mathematics for Machine Learning, 2021 (Main reference).

Learning Outcomes:

- Explain the mathematical background to solve data science problems
- Identify the calculus, linear algebra, and optimization topics related to each step of a data science problem
- Apply computational tools in data science problems
- Application of 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:
Group Class Activities (10%), Group Assignments (10%)
Team Projects and Poster Sessions (15%), Two Articles and Presentations (10%),
Attendance (5%), Exam1 (10%), Exam2 (10%), Final Exam (30%)

Attendance: Attendance is a University Requirement. A DN grade is rewarded after accumulating 6 unexcused absences.
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	**Finding Words for Intuition** • 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 2.5-2.6	**Linear Algebra** • Vector Spaces Understanding Solvability of Systems via: • Linear Independence • Basis and Rank
3	2.3	• Solving Systems of Linear Equations **Hands-on Illustration (Elementary Computation)**: Using Numpy and Scipy to • solve linear systems • Check rank of matrix • Illustrate with scipy the challenges in solving rank-deficient problems • Motivate the idea of approximate solution to linear systems
4	**Chapter 2 (cont.)** 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 (**elementary task**) • Use the knowledge of lengths and distances to implement, using Numpy and Scipy, the k Nearest Neighbor algorithm. Goal: Sections 3.1, 3.2, 3.3 and 3.4, which for the basis of kNN classifier.
5-6	3.5-3.9	Orthonormal Basis Orthogonal Complement Inner Product of Functions Orthogonal Projections Rotations **Code Illustration**: • Using Numpy and Scipy to implement least square approximation for fitting data to a straight line. • Apply LinearRegression in scikit learn to same dataset

7	**Chapter 4** 4.1-4.3	**Matrix Decomposition** Determinant and Trace Eigenvalues and Eigenvectors Cholesky Decomposition
8	4.4-4.5	Eigen-decomposition and Diagonalization Singular Value Decomposition **Code Illustration**: • Compute Eigenvalues and Eigenvectors in Numpy • Perform principal component analysis using numpy and compare with Scikit learn • Perform SVD in Numpy • Use SVD to solve column-rank deficient problems (Revisit linear regression for highly-correlated data)
9	**Chapter 5** 5.1-5.3	**Vector Calculus** Differentiation of Univariate Functions Partial Differentiation and Gradient Gradients of Vector-Valued Functions
10	5.4-5.5	Gradients of Matrices Useful Identities for Computing Gradients
11	5.6-5.7	Backpropagation and Automatic Differentiation Higher-Order Derivatives
12	**Chapter 7** 7.1-7.2	**Continuous Optimization** Optimization Using Gradient Descent Constrained Optimization and Lagrange Multipliers **Code Illustration**: • Implement the Gradient Descent iteration in python (Requires Numpy Library and for loops)
13-14	7.3	Convex Optimization
Week 15		**Project Presentation**

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 (mainly Loops, Masks, and Lists); Scikit Learn.

Highlights of Coding Activities:

Weeks	Topics and Related application	Tasks and Possible Goals
3	• Solving Systems of Linear Equations	**Hands-on Illustration (Elementary Computation)**: Using Numpy and Scipy to • solve linear systems • Check rank of matrix • Illustrate with scipy the challenges in solving rank-deficient problems **Goal**: Motivate the idea of approximate solution to linear systems

4	Vector norms, inner products, lengths and distances **Note**: Distance between vectors is the building block of the k Nearest Neighbor algorithm	**Hands-on Illustration**: • Compute matrix and vector norms in Numpy (**elementary task**) • Use the knowledge of lengths and distances to implement, using Numpy and Scipy, the k Nearest Neighbor algorithm. **Goal**: Sections 3.1, 3.2, 3.3 and 3.4, which for the basis of kNN classifier.
6	• Solving Linear system Revisited • Fitting curves to data (formally called Regression)	**Hands-on Illustration**: • Use Numpy and Scipy to implement least square approximation for fitting data to a straight line. • Apply Linear Regression in Scikit learn to same dataset **Goal**: To illustrate some of the materials in chapters 2 and 3 as forming some of the basis for linear regression.
8	• Eigendecomposition and Diagonalization • Singular value decomposition **Applications** **Eigendecomposition** together with projections are vital reducing the dimensionality of high-dimensional data Performing regression on highly-correlated data often lead to solving column-rank deficient problems. We use **SVD**	**Hands-on Illustration**: • Compute Eigenvalues and Eigenvectors in Numpy (Basic Task) • Perform principal component analysis using Numpy and compare with Scikit learn • Perform SVD in Numpy Use SVD to solve column-rank deficient problems (Revisit linear regression for highly-correlated data) **Each of the above tasks is considered a goal by itself**
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 including linear regression (already covered) and Regularization for support vector machine use the gradient descent iteration for speedy approximation	(Requires Numpy Library and for loops)

Important Dates:

- **Exam1**: Week 6. **Exam2**: Week 12.
- **Project Proposal**: Week 8.
- **Article-1:** Week 9
- **Article-2:** Week 11
- **Project Report/Notebook Submission:** Week 13
- **Project Presentations:** Week 14
- **Final Exam**: Posted on the registrar’s website