KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DEPARTMENT OF MATHEMATICS

DHAHRAN, SAUDI ARABIA

Term 231 – Fall 2023

MATH 503: Mathematics for Data Science.

Instructor: Ibrahim O. Sarumi

Office: Building 5 – Room 415

Phone: 4197

Email: ibrahim.sarumi@kfupm.edu.sa

Office Hours:

Prerequisite: Graduate Standing

Credit Hours: (3-0-3)

****

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.

Course Material:

My material for this course will be my lecture notes.

Textbook:

Deisenroth etal, Mathematics for Machine Learning, 2021 (Main reference).

References:

1. Charu C. Aggarwal, Linear Algebra and Optimization for Machine Learning, 2020.

2. Thomas Nield, Essential Math for Data Science, 2022

Communication:

For regular announcements, students are advised to check Teams and Blackboard regularly.

Academic Integrity:

All KFUPM policies regarding ethics and academic honesty apply to this course

Attendance: (As per KFUPM policy) **DN** is assigned for 20% (9) unexcused absences and 33% (15) overall absence (excused and unexcused)

Grading:

Activity	Weight
Quizzes	10%
Homework	10%
Projects	20%
Midterm exam	25%
Final Exam	35%

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.

Student 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

Computation Tasks:

Python Programming language will be used for implementing computational tasks in this course. Frequently used package used include:

- Numpy
- Scipy
- Core python programming constructs (mainly Loops, Conditioners and Lists)
- Scikit learn will be used to compare with standard machine learning algorithms

Highlights of Coding tasks

Week	Topics and Related application	Tasks and Possible Goals
Week 3	• Solving Systems of Linear Equations	**Hands-on Illustration (Elementary Computation)**: Using Numpy and Scipy to 1. solve linear systems 2. Check rank of matrix • Illustrate with scipy the challenges in solving rank-deficient problems **Goal**: Motivate the idea of approximate solution to linear systems
Week 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.
Week 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.
Week 8	• Eigendecomposition and Diagonalization • Singular value decomposition	**Hands-on Illustration**: • Compute Eigenvalues and Eigenvectors in Numpy (Basic Task)

	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**	• 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**
Week 12	Optimization Using gradient descent. Note: Many Machine learning algorithms including linear regression (already covered) and Regularization for support vector machine use the gradient descent iteration for speedy approximation	**Hands-on Illustration**: • Implement the Gradient Descent iteration in python for finding least square solutions (Requires Numpy Library and for loops)

SCHEDULE and COVERAGE of MATERIAL

Week No. (Dates)	Reference	Topics
Week 1 Aug. 27 & 29	**Chapter 1** **Chapter 2** 2.2 2.1	**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)
Week 2 Sep. 3 & 5	**Chapter 2** 2.4	**Linear Algebra** • Vector Spaces Understanding Solvability of Systems via:

	2.5 2.6	• Linear Independence • Basis and Rank
Week 3 Sep. 10 & 12	2.3	• Solving Systems of Linear Equations **Hands-on Illustration (Elementary Computation)**: Using Numpy and Scipy to 3. solve linear systems 4. Check rank of matrix • Illustrate with scipy the challenges in solving rank-deficient problems • Motivate the idea of approximate solution to linear systems
Week 4 Sep. 17 & 19	**Chapter 2 (cont.) and Chap 3** 3.1 3.2 3.3 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.
Sep 23: National Day **Sep 24: National Day Holiday** Week 5 Sep. Sep. 26	3.5 3.6 3.7 3.8 3.9	Orthonormal Basis Orthogonal Complement Inner Product of Functions Orthogonal Projections Rotations
Week 6 Oct. 1 & 3		**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 Goal: This illustrates what we have learned in chapters 2 and 3 as forming some of the basis for linear regression.
Week 7 Oct. 8 & 10	**Chapter 4** 4.1 4.2 4.3	**Matrix Decomposition** Determinant and Trace Eigenvalues and Eigenvectors Cholesky Decomposition
Week 8 Oct. 15 & 17	4.4 4.5	Eigendecomposition 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)
Week 9 Oct. 22 & 24	**Chapter 5** 5.1 5.2 5.3	**Vector Calculus** Differentiation of Univariate Functions Partial Differentiation and Gradient Gradients of Vector-Valued Functions
Week 10 Oct. 29 & 31	5.4 5.5	Gradients of Matrices Useful Identities for Computing Gradients
First Progress Report by end of week 10		
Week 11 Nov. 5 & 7	5.6 5.7	Backpropagation and Automatic Differentiation Higher-Order Derivatives
Week 12 Nov. 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)
Midterm Break Nov. 19 - 23		
Week 13 Nov. 26 & 28	7.3	Convex Optimization
Week 14 Dec. 03 & 05		
Final Project report submission		
Week 15 Dec. 10 & 12		**Project Presentation**

Week 16 Dec. 16		