

ENTS 669G: Data Mining with Numerical Python

Course Description

This course will provide an introduction to the field of Data Mining, with an emphasis on understanding, manipulating, generating and classifying numerical data. Students will use python extensively to implement algorithms and gain familiarity with data mining techniques. In addition to Python 3, Numpy, and Matplotlib, other python packages with applications to data mining will be explored. Topics will include basic statistics and numerical computation, data acquisition and cleaning, data and dimension reduction, classification, prediction, patterns, and machine learning. ENTS656, ENTS699M, or a good knowledge of Python and Numpy is required. Students who are unsure of their python background should check with the instructor.

Instructor Contact Information

Dr. Michael Dellomo
301.405.1233 (office)
301.728.1864 (cell)
mdellomo@umd.edu

Office hours:

1363 AV Williams Building
Days and times will be announced in class and posted on my website at
www.umd.edu/~mdellomo/advising.html

Required Books

Data Mining: Concepts and Techniques, by Han, Kamber, and Pei

Co-requisite

ENTS656, ENTS699M, or a good knowledge of Python and Numpy is required. Students who are unsure of their python background should check with the instructor.

Course Meeting Times and Location:

See <https://testudo.umd.edu/>

Course Policies

Attendance & participation

Attendance and participation are not required and students will not be graded on it. However, it is highly recommended that students attend lectures to ask questions about the course and to better prepare for the exams.

Assignments

Homework: All assignments must be submitted by deadlines given in class. Each student is expected to complete the assignment on their own, however, it is acceptable for students to compare answers and check each others work. Plagiarism, however, is not permitted. Specifically, copying code or modifying code from any source (other students, online resources, etc.) is strictly forbidden and will be considered a violation of the academic integrity code. It is each student's responsibility that the assignments are submitted in a timely manner so the professor can assess the assignment. In any event, all work must be completed by the end of the course as instructed. Questions about Python (or anything else) can be brought to the instructor.

Academic Integrity

The University of Maryland has a nationally recognized Honor Code, administered by the Office of Student Conduct. This code sets standards for academic integrity for all undergraduate and

graduate students, and you are responsible for upholding these standards in this course. It is very important for you to be aware of the consequences for cheating, fabrication, facilitation and plagiarism. For more information please visit: <https://studentconduct.umd.edu>. Students who engage in academic dishonesty in this course will receive no points for the assignments and will be reported to the Office of Student Conduct for further action. There will be no warnings! Remember, it is not worth it!

Persons with Disabilities

Students with a documented disability should inform the instructor as soon as possible if academic accommodations are needed. Accommodations for individuals with disabilities can be arranged through the Disability Support Service (DSS), a division of the University Counseling Center. Please call 301.314.7682, web: <https://www.counseling.umd.edu/ads/>, or visit Shoemaker Building for more information.

Video Taping, Recording and Photographing

It is against University and Program policy to video tape, record, or photograph lectures unless done in accordance with the procedures for Persons with Disabilities. Lecture material is considered to be copyrighted by the University and unauthorized reproduction is considered to be copyright infringement. The instructor will make available and distribute any necessary material which is too detailed for conventional note taking.

Cell phones

Any use of cell phones is not permitted during class time. Please turn off all cell phones prior to the start of class.

Grading

Students will be given several in-depth python assignments. Scores on these assignments will form a major part of the grade in the class, along with a midterm and final. Course grading will be as follows:

Assignments	200 points	Exact number TBD. (7 are planned but it could vary)
Midterm	100 points	Date Announced in Class
Final	<u>200 points</u>	Date Announced in Class
Total	500 points	

Tentative Course Schedule (will be adjusted as the course progresses) Topics may vary depending on timing, class progression, and subject evolution. Topic order may also change.

- 1) Data mining in the data age with Python
Review of Python and Numpy. Overview of databases, data warehouses, clustering, classifying, outliers, basic statistics
Python: Numpy: timing, error, processing and speed considerations
- 2) Numerical Data
Basic numerical computation principles, error and bounding, good practice for numerical computations.
- 3) Data Attributes and playing with data
Quartiles, variance, data comparisons
Python: manipulating data, using statistics.
- 4) Getting data
Using python to read generate data sets, regression and prediction
Python: get web pages and generate statistics from them
- 5) Similarity and dissimilarity of data
Similarity measures for attributes, cosine similarity
- 6) Preprocessing
Data cleaning, noise, rejection and conflict resolution
Python: processing data, dealing with noise, filtering. Signal processing
- 7) Data reduction
Data transforms, FFT, Wavelet, Principle components, dimension reduction
Python: basic FFT manipulation, Wavelet processing. Comparison and usage
- 8) Data presentation
Histograms, binning, clusters, python routines
- 9) Patterns
Associations, patterns, algorithms
- 10) Classification
Decisions and classification, Bayesian classification (simple)
Python: Naïve Bayesian classifier construction
- 11) Machine learning basics
Matched filtering, training and testing, “bad” data handling
- 12) Neural Networks
Perceptrons, basic networks, training, backpropagation
Python: Neural Network usage with Sci-kit Learn
- 13) Support Vector Machines