Course Syllabus: CICS 580
Introduction to Numerical Computing with Python

Instructor: Ali Montazeralghaem (montazer@cs.umass.edu)

Course Description: This course is an introduction to computer programming for numerical computing. The course is based on the computer programming language Python and is suitable for students with no programming or numerical computing background who are interested in taking courses in machine learning, natural language processing, or data science. The course will cover fundamental programming, numerical computing, and numerical linear algebra topics, along with the Python libraries that implement the corresponding data structures and algorithms. The course will include hands-on programming assignments and a small project. No prior programming experience is required. Familiarity with undergraduate-level probability, statistics and linear algebra is assumed. 1 credit.

Required Textbook: We will use the following two freely available materials in this course. Other reading materials for some special topics will be provided as needed.

Introduction to Python for Computational Science and Engineering: This text provides a basic overview of python and an introduction to numerical computing issues.
URL: http://www.southampton.ac.uk/~fangohr/training/python/pdfs/Python-for-Computational-Science-and-Engineering.pdf

SciPy Lecture Notes: These organized notes provide an overview of NumPy and SciPy functions as well as a number of useful exercises.
URL: http://www.scipy-lectures.org/

Course Website: The course website will be hosted on Moodle at https://moodle.umass.edu/. The course website will host lecture notes, assignments, and pointers to readings and videos. We will use Piazza (https://piazza.com/) for a course discussion forum.

Grading Plan: Programming assignments (60%). Project (40%).

Project: The project is an opportunity to use Python on your research data, or any publicly available dataset. Small portions of the project will be assigned with each homework assignment, and a 1-2 page write-up and short (~5 min) video presentation will be due at the end of the course synthesizing each component. You are expected to demonstrate that you were able to load your data, perform at least one visualization, and fit a "proof-of-concept" model on the data. The entire project should only require < 30 lines of code, though you are welcome to spend as much time as you like. Novelty and state-of-the-art performance are not required; the purpose is simply to show basic familiarity with key components of the data science pipeline in Python. If you do not have research data to use, you can look at datasets on Kaggle (https://www.kaggle.com/), the UCI data repository (https://archive.ics.uci.edu/ml/index.php), or any other available data source.

Assignments: Assignments will be handed out and turned in on Gradescope. Late assignments will receive a zero (unless otherwise arranged); however, the lowest score among all **submitted** assignments will be dropped.

Regrade Policy: Regrades requests can be submitted via the link on Gradescope up to **one week** from the day grades are returned.

Accommodation Statement: The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services (DS), you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify me within the first two weeks of the semester so that we may make appropriate arrangements.

Academic Honesty Statement: Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent (http://www.umass.edu/dean_students/codeofconduct/acadhonesty/).

Course Schedule: Lecture videos will be uploaded twice per week for the first six weeks of the semester. We schedule a live demo and Q/A session each week, which are optional to attend. This schedule may change as the semester progresses.

Week 1: Introduction to Python

- Data types and Assignment
- Basic data structures (lists, tuples, dictionaries)
- Mathematical and logical operators
- Control flow (looping and branching)
- Functions
- Objects and Classes
- Assignment 1

Week 2: Computing with Arrays (Numpy)

- Multi-Dimensional Arrays
- Array Indexing and Slicing
- Mathematical and Logical Operators for Arrays
- Broadcasting and Array operations
- Assignment 2
- Project step 1: description of your chosen data

Week 3: Computing with Vectors and Matrices (Numpy, Pandas)

- Loading and manipulating datasets in Python, Numpy, and Pandas
- Vectors and matrices
- Basic linear algebra operations (addition, inner/outer products, etc.)
- Advanced linear algebra operations (matrix inversion and decompositions, etc.)
- Assignment 3
- Project step 2: proof that you could load and iterate through your data

Week 4: Data Visualization (Matplotlib)/Computing with Probability (Numpy)

- Counting and histograms
- Scatterplots, line plots, bar charts, pie charts
- Probability mass functions and probability density functions
- Random numbers and simulating random variables
- Assignment 4
- Project step 3: a plot/visualization related to your data

Week 5: Introduction to Modeling (Numpy/Scipy/Scikit-learn)

- K-nearest neighbors
- K-means
- Assignment 5
- Project step 4: build a model or perform some statistical analysis on your data

Week 6: Introduction to Modeling cont'd (Numpy/Scipy/Scikit-learn/PyTorch)/ Documentation and Testing

- Linear and logistic regression
- Code documentation
- Debugging, exceptions
- Deep learning, PyTorch
- Assignment 6
- Project step 5: write-up + video