Spring 2022

DS 3000: Foundations of Data Science

Course Description

Data Science (DS) is an interdisciplinary field concerned with the study and application of systematically extracting knowledge from data and using this knowledge to draw useful and educated conclusions. This course is an introductory DS class focusing on the foundations of DS as an emerging field. The course introduces core modern DS technologies and methods that provide a foundation for subsequent DS classes. Therefore, the focus of this class in on the breadth of various DS skills, rather than the depth of the topics covered. As a skills-based course, DS 3000 will cover the use of Python for DS and will introduce some of the widely-used essential Python libraries, such as NumPy, Pandas, Matplotlib, and Scikit-Learn.

More specifically, this class covers:

- working with tensors and applied linear algebra in standard numerical computing libraries (e.g., NumPy);
- loading, processing and integrating data from a variety of structured and unstructured sources using Python libraries (e.g., pandas)
- visualizing data using basic techniques and tools (e.g., matplotlib/seaborn);
- applying introductory concepts in probability, statistics, and machine learning using Python libraries (e.g., scikit-learn);
- using a standard DS tool (e.g., Jupyter Notebook).

Course Structure & Objectives

This course is fully **asynchronous -- all lectures are included in Canvas.** There will be a **synchronous lab/recitation** and its purpose is to help students with their labs, answer any questions or clarify any questions or concerns. This approach is the best of both worlds providing the flexibility of prerecorded video lectures and/or readings that cover the course material asynchronously followed by synchronous sessions where you apply your understanding to these core concepts through carefully chosen problems and activities which will enable you to solidify your knowledge. Therefore, do not treat the **synchronous lab/recitation** as you would a lecture. It will not teach any new concepts; it is a session that is designed to help you and ensure that you do not feel disconnected from the course (or your colleagues).

Your attendance in the **synchronous labs/recitations** are NOT required but is strongly encouraged, especially for students who do not have a programing background. Ensure that you watch the lectures before joining so that you can fully participate and/or receive assistance with any challenges on the homework.

Objectives

Upon the successful completion of this course, you are expected to:

1. Demonstrate an understanding and appreciation of the breadth of the field of DS,
2. Demonstrate the ability to work confidently with tensors and multidimensional DataFrames,
3. Read in data from a variety of sources,
4. Perform, interpret, and report basic data analyses (visualization, regression, etc.),
5. Work fluently in Pandas,
6. Demonstrate an introductory, conceptual understanding of machine learning,
7. Be able to work with Jupyter Notebooks,
8. Apply DS concepts, tools, and technologies to a life-like DS project,
9. Deliver a presentation describing a DS project.

Prerequisites

DS 2000 or CS 2510 (strong background in programming assumed).

Course Methodology & Assignments

Each week, students are expected to:

1. Review the week's learning objectives
2. Attend the synchronous sessions (this is optional)
3. Complete all assigned readings
4. Complete all lessons for the week
5. Participate in any online discussions and collaborative sessions with TAs
6. Complete and submit all assessments by the due date

Communication

Communication between instructor and students is through:

- E-mail via the Canvas distribution list
- Announcements posted on Canvas
- Notes posted on the MS Teams discussion channel
- Private email exchanges
- MS Teams or Zoom for private communication

Submission of Work

All work for the course is expected to be completed by the due date and time and must be submitted using the appropriate platform (**not via email**). **Late submissions past the due date are subject to a 10% reduction in grade for each day late, with specific rules stated for each graded assessment item. Late submissions past two days are NOT accepted**.

All deadlines in this course are in Eastern (Boston) Time.

Once an assignment has been graded, students will be able to view the grade and feedback. Assignments are to be done in a professional manner -- points will be deducted for any work that is not at an appropriate level of quality. All code must properly run from start to end and all libraries, packages, files must be installed and/or loaded as part of the submitted code. Ensure that your work is properly documented and all graphical illustrations should be labeled appropriately.

Assessments

Please follow the 30-minute rule: if you have been stuck on a problem more than 30 minutes and have made no progress please stop and get help. Message the professor, attend office hours, or consult a TA. Do not become fixated on one problem or bug in your code. Asking a question can get you past the bug quickly and teach you how to fix it in on your own next time.

The course grade will be based on the following (which are subject to change):

1. **Mini-labs (10%)**: provide students with an opportunity to apply their understanding of the week's core concepts by solving problems and activities. Students will be expected to upload a Jupyter Notebook with their answers to the questions posed in the exercise. This is a very easy way of guaranteeing 10% of your grade!

2. **Homework assignments (30%)**: provide students with an opportunity to solidify their knowledge through the application of what they have learned to new problems. As such,

students are expected to work on the assignments individually. **Unauthorized collaboration is immediately reported to OSCCR**.

3. **Topic Quiz (20%)**: There will be multiple online quizzes throughout the course. The quizzes will be posted online in Canvas and will be open-book. You may use any notes, lecture slides, and readings to answer the questions in the quiz; however, you should NOT collaborate with each other.

4. **Final Project (40%)**: The final project is designed to give you hands-on experience applying the skills from this class to a life-like DS project. You are required to identify a problem and dataset; clean, process, and prepare the dataset for analysis; analyze and visualize the data using Python libraries; and report your findings and insights for decision making.
   Students are required to periodically submit progress reports to ensure that they are on-track. The dates and expectations for each progress report will be available in Canvas.
   You will be working in teams of 3-4 people and submit the deliverables of the project as a team. Therefore, you will submit peer evaluations of your team members' contribution to the project throughout the semester. Given the scope of the project, peer evaluations will have a substantial impact on your grade for the final project. It is important that all team members fully participate.

5. **Bonus Credits**: This is an opportunity for each student to receive extra credits to boost their overall score. These bonus credits will take the form of: 1) short quizzes that are periodically available in select modules, and 2) reflections -- which is your opportunity to reflect on your personal journey and experiences in this course.
   The bonus points are applied at the end of the semester and the maximum that can be applied is 5%.

Textbook & Grading Scheme

- **Textbook**: Python Data Science Handbook, by Jake VanderPlas. The entire book is available online on GitHub at **the following link (https://jakevdp.github.io/PythonDataScienceHandbook/)** .

Grading

The course grade system aligns with the university grading system:
95% and above A

90% - 94.9%	A-
87% - 89.9%	B+
84% - 86.9%	B
80% - 83.9%	B-
77% - 79.9%	C+
73% - 76.9%	C
70% - 72.9%	C-
Less than 70%	F

Academic Integrity Policy

The University views academic dishonesty as one of the most serious offenses that a student can commit while in college and imposes appropriate punitive sanctions on violators. Here are some examples of academic dishonesty. While this is not an all-inclusive list, we hope this will help you to understand some of the things instructors look for. The following is an excerpt from the University's policy on academic integrity; the complete policy is available in the Student Handbook.

- *Cheating* – intentionally using or attempting to use unauthorized materials, information or study aids in an academic exercise; this includes submitting work of another student or work prepared for another course
- *Fabrication* – intentional and unauthorized falsification, misrepresentation, or invention of any data, or citation in an academic exercise
- *Plagiarism* – intentionally representing the words, ideas, or data of another as one's own in any academic exercise without providing proper citation, including code fragments from websites such as stackoverflow
- *Unauthorized collaboration* – instances when students submit individual academic works that are substantially similar to one another; while several students may have the same source material, the analysis, interpretation, and reporting of the data must be each individual's independent work.
- *Participation in academically dishonest activities* – any action taken by a student with the intent of gaining an unfair advantage
- *Facilitating academic dishonesty* – intentionally or knowingly helping or attempting to violate any provision of this policy
- *Impersonation* – working on behalf of another students or allowing someone else to represent a student online, in discussion groups, for presentation, in any communication with the instructor, or in exams
- *Multiple Submissions* – submitting the same or substantially the same work in two courses

Any incident of academic misconduct will result in a 0 for the graded item, a report to OSCCR,

and a full two-letter reduction in the final course grade, except in cases of impersonation or fabrication. In those two cases the student will receive a report to OSCCR with a recommendation for dismissal from the University and an automatic failing grade of F.