Syllabus
Projects in Programming & Data Science,
INFO-UB.0024.01, Fall 2020

Course information

- When: Tuesdays, 4.55pm - 7.30PM
- Where: Online / Zoom
- Slack: https://ppds-f20.slack.com/

Professor information

- Prof. Panos Ipeirotis
- Email: panos@stern.nyu.edu
- Website: http://www.ipeirotis.com
- Blog: http://www.behind-the-enemy-lines.com
- Twitter: @ipeirotis
- Office: KMC 8-84
- Office Hours: Through Zoom / by appointment

Teaching assistant

- Joe Barrows

Registration information for non-Stern students

Course Description

This is a **project-based course**, for students that are **already familiar** with programming in Python and SQL, and want to apply this knowledge (and beyond) to a topic of their interest. The students should have taken INFO.UB.0023 (Introduction to Programming for Data Science), or have equivalent experience.

This is not an introductory course. Students who are interested to learn about programming and SQL need to take the Introduction to Programming for Data Science (INFO.UB.0023) first.

The class will be split into two parts. Early in the semester, we will have lectures about a variety of topics, building on top of the material already covered in INFO.UB.0023. In the second part, the students will work on their projects, with the guidance of the professor, but without a lecture component.

In INFO.UB.23 (the natural prerequisite for the class) the goal of the final project was to create a replicable, executable report (in the form of an iPython notebook). In this class, the goal of the project for this class is to build a **data product.** We will discuss more what this means, but at

the minimum we expect the final deliverable to give the ability to users to interact with the project (through a chatbot, or through a web front end), and allow the users to get something meaningful from the underlying data.

Student teams are encouraged to talk to their classmates and the instructor about the evolution of their own project and get feedback on what techniques, technologies, and datasets would be helpful to move the project forward.

Conceptually the class will have the following modules:

- **Module 1a, "Backend": Web API's**
    - *Description*: We will learn about Web APIs, as a powerful way to interact with various data sources on the web, and as a method for processing data. We will examine how to call web APIs, learn about parameters, headers, status codes, and we will learn the parse the responses that come back. We also will learn about various authentication mechanisms and practice using APIs that require authentication (e.g., Twitter, IBM Watson).
    - Topics
        - Web services/APIs for retrieving and processing data
        - Scheduling processes to run in the background and at specified intervals
    - **Assignment**: Identify an API that returns information that changes over time. Write a script that fetches the data over time, and stores the data in a database. The assignment requires the design of a database to store the time-invariant and time-varying information. *The script can be scheduled to run in the background periodically.*
- **Module 1b, "Backend": Regular Expressions**
    - Description: We will first learn how to use regular expressions, which will allow us to extract data from text using patterns.Will see that regular expressions are also useful for general data processing tasks, and are often used for data cleaning.
    - Topics:
        - Regular expressions from extracting data using patterns
        - Regular expressions for data preparation and data cleaning
- **Module 1c, "Backend": Web crawling/scraping**
    - Description: Not all the web sources provide an API to access their data. For that reason, it is often necessary to "crawl" web pages and "scrape" the content directly from the web pages of a website. We will learn to use XPath to parse HTML pages and learn to write patterns that extract specific parts of a web page.
    - Topics:
        - Parsing XML documents using XPath
        - Identifying relevant content within HTML pages
    - Assignment 2: Crawl a website to extract the information that appears in its web pages. Store the retrieved results in a database.
- **Module 2, "Analytics": Descriptive Data Analysis and Visualization**

    - Description: Python Pandas is a powerful library, which is widely used for data processing tasks. We will learn how we can use Pandas for managing and transforming data, and see how we can integrate Pandas functionality with SQL databases. We will also examine how to visualize data. We will learn the basic concepts of mapping variables to visual channels, and go through examples of visualizations that allow complex datasets to be represented visually.
    - Topics:
        - Reading data from a variety of sources
        - Data manipulation
        - "Tidy data" discussion
        - Multidimensional data analysis
        - Time series: Trends, periodicity, autocorrelation
        - Spatial data: Spatial distributions, data operations for spatial data
        - Data visualization
    - **Assignment**: Analyze a dataset using Pandas and generate plots to explain aspects of the data.
- ***Module 2b, "Analytics": Text Mining. Introduction to Classification***
    - *Description: When we go beyond structured data, text is one of the most common forms of data that we need to analyze. We will spend some time understanding the basics of text data (tokenization, Zipf's law), and discuss some of the common tasks that are associated with text. We will examine the use of semantic resources (e.g., Wordnet and/or Freebase) for facilitating our text analysis. We will also examine concepts such as part-of-speech tagging, entity extraction, chunking, dependency parsing, and apply these techniques to a variety of textual data sets.*
    - *Topics:*
        - *Tokenization, stemming, Zipf's law*
        - *Part-of-speech tagging, chunking*
        - *Entity extraction, disambiguation*
        - *Dependency parsing, sentiment analysis*
        - *WordNet and embeddings*
        - *Introduction to text classification (featurization, training/testing, evaluation)*
    - ***Assignment**: Given a set of reviews, extract the most commonly discussed noun and noun chunks and their frequencies among the reviews. Using a dependency parser, extract the adjectives that are most frequently associated with these entities, and their corresponding frequencies.*
- **Module 3: "Frontend": Building a basic data-driven website (Flask)**
    - **Description:** While this is not a web-development course, we will also examine briefly the concept of creating a data-driven website. We will examine how we can expose our data and visualizations through a website, and see how we can query the "backend" data from a web front-end. We will learn about web forms, server-side scripts, and see how we can present the data that we have been collecting and analyzing to a general audience on the web.

    - Topics:
        - Running web server using Flask
- **Semester Project: Building a Data Product**
    - Description: During the semester, we will learn how to fetch, organize, and present data. In this part of the course, we will discuss how we can create a **data product**, i.e., a place where other users can connect to interact with our data, analysis, and so on.
    - This is a substantial part of the course. In this part, you will work in teams to create your own data product. You are encouraged to use a large fraction of the material that we covered during the semester.

Attendance

I expect students to attend all classes, or at least watch the recordings of the class. If you plan on not attending more than 2 sessions during the semester, consider taking the class at some other time.

Prerequisites

Students are expected to have taken the class “Introduction to Programming for Data Science” (check the syllabus), or have the equivalent knowledge (mainly, know Python and SQL). If in doubt, please contact the instructor.

Grading

- 75% semester project
- 20% assignments (pass/fail)
    - Assignments are pass/fail. Students that get a “fail” should try to fix their submission and resubmit to get a “Pass” and a 50% credit for the assignment. Students that do not fix their assignment will get a 0% credit.
- 5% participation

Late Assignment Submission Policy

You are free to submit your assignment late, but there is a 3% grade penalty on your assignment for every additional day after the deadline. You can be at most 7 days late. (That will be a reduction of 7x3% = 21%.) Given the generous late submission policy, penalties are strictly enforced, and no extensions are granted. Family events, recruiting, holidays, are not valid reasons for waiving the late submission penalty. **Please plan accordingly, and do not leave submission for the last minute.**

PS: I will consider exceptions for health reasons, but I expect a notice well in advance of the deadline. In other words, catching a cold the day of the deadline will not be a valid reason for an extension.

FAQ about grading

- Q: How do you assign grades?
  A: Expect roughly 50% A/A-'s and the rest being B+/B's and, in some cases, Cs'.
- Q: How is the class graded?
  A: I expect most students to do well in the assignments. Assignments are mainly for practice, and to get you to understand better the material. As expected by the name of the class, the majority of your grade depends on the project. Projects are roughly rank-ordered and the top projects get an A, and the worst projects get B's.
- Q: How do you grade the projects?
  A: I **stack-rank** the projects, based on a combination of factors, including ambitiousness, technical competency, interestingness, and coverage of class material. There is also a peer-evaluation component. Interestingly enough, the two are often strongly correlated; the correlation tends to break for some projects with "unpopular" topics but overall, I would say that peer-evaluations are correlated with the professor grade around 80% of the time.
- Q: Does this mean that I may spend a *lot* of time on my project and still get a B?
  A: Yes. The stack-ranking approach is very intentional, and not accidental: Setting a minimum set of goals for the project is artificial; it either leads to the class becoming too hard or too easy. I fully expect student teams to compete with each other to deliver the best possible project at the end.

Course Policies

Unless otherwise noted, we follow the default Stern Policies.

Tentative Timeline

Session	Date	Topic	Readings
1	Sep 8	Web APIs I	4/A (GeoIP, OpenweatherMap) **Assignment**: Setup Jupyter/MySQL, Form teams and signup for presentations 4/B2 (IBM Watson, text analysis), **Assignment**: NewsAPI + IBM Watson
2	Sep 15	Web APIs II	4/B1, 4/B3 (Google Vision API, Google Maps) session1/A5 session1/A6 (Store Citibike data into database) **Assignment**: Store data from API into database If we have time: Spotify API, Yelp API (examples with authentication)
3	Sep 22	Regular Expressions	5/A, 5/B **Assignment**: Process a dataset to identify anomalous entries
4	Sep 29	Web Crawling	6/E, 6/F **Assignment**: Crawl a website and save the results in a database
5	Oct 6	Descriptive Data Analysis & Visualization I	3/A (restaurant dataset)
6	Oct 13	Descriptive Data Analysis & Visualization II	3/A & 3/A2 (accidents dataset)
7	Oct 20	Web Frontend Development	9-Flask
8	Oct 27	Command line and Scheduling Tasks	Command-line basics (12/A) Python scripts vs iPython notebooks Running tasks in the background (12/D) Using cron for scheduling (12/D)
9	*Nov 3*	*Optional lecture: Temporal Data*	*3/D, 3/F*
10	*Nov 10*	*Optional lecture: Geospatial Data*	*8/F, 8/D*
11	*Nov 17*	*Optional lecture: Text Mining*	*7/A, 7/B, 7/C*
12	*Nov 24*	*No lecture - working on projects*	
13	*Dec 1*	*No lecture - working on projects*	
14	Dec 8	**Class Presentations**	