DSCI 510 Final Project

Fall, 2020

This is a description of your project “milestones”. Please read this entire document carefully.

This project is to be done **individually.**

In general, the focus of the project is to show you can acquire, model, store and process multiple sources of data, and build reliable pipelines to do so. For this course, actual “analysis” of the data is secondary; you’ll be expected to say *something* about the data, but your conclusions are not the focus of the project.

Your project will be scored on a number of factors, including (but not limited to!) the complexity and size of your datasets, the quality of your pipeline and modeling code, and the writeup of your research statement and conclusions. It’s a sizable amount of work, but it’s your chance to actually do something substantial with data, so I hope you have fun with it!

The project contains three “milestones”, outlined below. Each one will be turned in separately, with the final submission being your final project.

Briefly, milestones are:

1). Data set and problem selection

2). Data acquisition and modeling infrastructure

3). Research conclusions and writeup

You are welcome – and encouraged! – to move faster than the milestone schedule. Especially in the beginning, it makes sense to write access code (i.e. scrapers and API crawlers) for your data as soon as possible!

**Homework 4/Project Milestone 1 (50 points)**

**Due Wednesday, November 11th at 11:59pm**

Find three (3) data sets on the web from a topic area that’s interesting to you. (If there isn’t any data that’s interesting to you, that’s a problem!):

1. One must be “scrape-able” (i.e. not available via an external API)
   1. “Data set” is not just a simple web page
   2. Needs to be something that requires automation to obtain
   3. If you could just cut-and-paste the data, it’s not a data set
   4. Data should be some what structured (maybe). If what you want to scrape are large text blurbs or images, that’s fine, as long as there is an algorithmic way of obtaining a large number of these items, while attaching meta-data to them.
   5. Again, if you could obtain the data manually vs. writing a script, it doesn’t count!
2. One must be available via external public API
   1. You have to be able to access it without a ton of trouble
   2. You can use OAuth if needed, but if you can’t get at the API, it doesn’t count
   3. If you decide to use an API that requires OAuth or some other authentication mechanism, ***you must test that you can access this before hand***
   4. The API must allow a “reasonable” number of free accesses per day/hour/lifetime, etc. Enough for you to be able to test and deploy assignments and projects based on it
   5. **It is up to you to maintain your API keys and stay within rate limits! Make sure you have a large enough call “quota” to be able to complete your project (i.e. don’t get banned!)**
   6. There is a list (by no means comprehensive!) of public APIs here: <https://github.com/toddmotto/public-apis/blob/master/README.md>
      1. Note which require authentication, and which do not!
3. The third can be either (API or scraped)

If you’re a Ph.D. student, something related to your research would be great! If you’re a MS student, here’s your chance to work on data of your choosing! I don’t mind “double-dipping”, and using this data for another course project, etc.

The deliverable for Milestone 1 is a text file (not .MD, or .docx or .pdf) named LASTNAME\_FIRSTNAME\_proj1.txt (where LASTNAME and FIRSTNAME are your own names) submitted to the course website with the following:

This document should contain the following:

\*\*\*

Q1: DATA SOURCE 1:

URL for website or API:

(If API, link to API docs):

Brief description of data/API (what it contains, what it represents, etc.):

Grading Scheme:

+5pts for providing data source

+5pts for description of data source

Q2: DATA SOURCE 2:

URL for website or API:

(If API, link to API docs):

Brief description of data/API (what it contains, what it represents, etc.):

Grading Scheme:

+5pts for providing data source

+5pts for description of data source

Q3: DATA SOURCE 3:

URL for website or API:

(If API, link to API docs):

Brief description of data/API (what it contains, what it represents, etc.):

Grading Scheme:

+5pts for providing data source

+5pts for description of data source

Q4: Briefly (4-6 sentence) describe how you might combine these datasets (i.e. how do they relate to each other? What are the commonalities between them? How might you connect them? How do they enrich each other?). For example, if you scraped census data that contains a person’s “home town”, google maps API data, and data with median income per zip code, you might discuss how you would use the google maps API to translate the hometown to a particular zip code, and then combine that with the income data.

Grading Scheme:

+10pts for sensible answer to above questions

G5: Briefly (4-6 sentence) describe what you might hope to find in the data overall. Basically, what are you trying to accomplish in this research project? What relationship are you trying to explore, or what pattern are you trying to discover, etc.

Grading Scheme:

+10pts for sensible answer to above questions

\*\*\*

Be sure to pick datasets that provide a meaningful amount of data, that you can actually access (in terms of rate limits, etc.) and that you’re interested in working with for the duration of the project!

**Project Milestone 2 (50 points)**

**Due Monday, November 30th at 11:59pm**

In this assignment, you’ll take the data sets you described in the first milestone, and build software to access, model and store the data.

Taking a closer look at each step:

1). Accessing the data: You’ll need to build scrapers and/or API crawlers for **each** data set you described above. These will need to be robust against failure, and will need to respect API rate limits, authentication, etc.

Grading scheme:

+10pts for writing code that sensibly access data sources

2). Modeling the data: Build a data model, using whatever method you prefer. This can be SQL relationships, a Python class/object hierarchy, setting up Pandas dataframes, SQLalchemy, etc. There are other options too! You have the freedom to interface with your data however you’d like, but keep in mind that regardless of how simple you *think* the data is, your solution will be graded on how useful, extensible, modular and robust your solution is. That means that if you turn in a solution that is great for your data as it is, but fails if anything about your data changes, that is not a great solution!

Grading Scheme:

+5pts for providing sensible model of data sources, including the relationships that exists between sources

3). Store the data: This should be relatively straightforward based on your modeling decision. SQL databases have a built-in capacity to store its data on disk. For other methods of modeling (i.e. Pandas) you’ll need to explore *serialization*. For Pandas, this may be CSV, Feather, Arrow, etc. For Python class objects, you could explore Pickle. Basically, you have to be able to save your data to disk and reload it back into your data model (see the discussion of “--source local” below) ☺

Grading scheme:

+10pts for writing code that stores the data in a manner that allows it to be saved and reloaded, and is consistent with data model mention in part 2

Your code should be modular in that it allows you to obtain the data from the scraper/API (“remote”) but also obtain it from local storage (“local”). How you implemented this (text files, CSV, cached webpages, SQL files, Feather serialized dataframes, etc.) is up to you. Your main script should have a

--source remote or --source local

command line parameter, that chooses where to obtain the data from

When invoked, your Python script should grab the data (either locally or remotely) and stick it into your data model.

**ADDENDUM:** You should also include a --grade flag. When this flag is invoked, your code should grab a maximum of **3** of each data source (i.e. 3 calls to an API, or scrape 3 web pages). This will allow us to grade your projects much more easily (which means better grades for you!) Note, you should also include API credentials for the project in either your Python code or documentation so we can test your code!

The deliverable for Milestone 2 is a Python **script** (or collection of Python scripts; note this is NOT a notebook!) submitted to the course website in the following format:

Name your archive of files: LASTNAME\_FIRSTNAME\_proj2.zip

The main file to run should be called LASTNAME\_FIRSTNAME\_proj2.py. Following these naming conventions is part of the grading process.

In addition, your zip file should turn in a *plain text* file named LASTNAME\_FIRSTNAME\_proj2.txt that answers the following questions:

1. What are the strengths of your data modeling format?
2. What are the weaknesses? (Does your data model support? Sorting the information? Re-ordering it? Only obtaining a certain subset of the information?)
3. How do you store your data on disk?
4. Let’s say you find another data source that relates to all 3 of your data sources (i.e. a data source that relates to your existing data). How would you extend your model to include this new data source? How would that change the interface?
5. How would you add a new attribute to your data (i.e. imagine you had a lat/long column in a database. You might use that to access an API to get a city name. How would you add city name to your data?)

Grading Scheme:

+5pts for reasonable answers to any of the above questions

**The Final Project (100 points)**

**Due the day of our final, Tuesday December 8th, at 2pm**

[Hint: The most important things in the project are that you had a clear research plan, and you turned in something that actually works. It's better to have working code that comes to a simple conclusion that an amazing research idea that throws an exception. Incrementally debug your code!]

What conclusions you come up with is up to you, but basically you want to answer whatever question originally prompted you to do the project in the first place. This could be a statistical analysis, a visualization, a surprising feature of the data or something else entirely!

For this assignment, you’re to submit **a zip file** named LASTNAME\_FIRSTNAME\_proj3.zip. Your zip file should *only* contain data and a notebook that processes the data (i.e. no need to turn in your scrapers, etc.)

Your zip file should have the following format:

* final\_project.ipynb (explained below)
* /data: Store the data you downloaded and wrote to disk in Milestone 2 here
* /src: If you have any extra python scripts that are invoked via the notebook, store them here
* environment or requirements.txt (optional). If you need packages that are not a standard part of the Ananconda distribution, please create an environment file. Instructions for that are here: <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html> and here: <https://blog.usejournal.com/why-and-how-to-make-a-requirements-txt-f329c685181e?gi=c77c6a4f2994>. You should also the steps to recreate your environment in this step #3 below.

The contents of final\_project.ipynb contain answers to the following questions. Put each question in a notebook cell (see the examples on the course website for the format):

1. Your name
2. Any major “gotchas” to the code (i.e. things that don’t work, go slowly, could be improved, etc.)
3. Any libraries that need to be installed to run your code (see above)
4. Anything else you feel is relevant to the grading of your project your project.

Also, answer some questions about the project itself:

1. What did you set out to study? (i.e. what was the point of your project? This should be close to your Milestone 1 assignment, but if you switched gears or changed things, note it here.)
2. What did you Discover/what were your conclusions (i.e. what were your findings? Were your original assumptions confirmed, etc.?)
3. What difficulties did you have in completing the project?
4. What skills did you wish you had while you were doing the project?
5. What would you do “next” to expand or augment the project?

Overall, the grading will be based on what your code looks like (is it well written, robust against errors, well documented, well commented), the detail of your research plan (did you have realistic goals? Did you describe your goals clearly? Was your topic properly scoped, I.e. not too big, not too small, etc.) and you ability to follow directions.

One final hint: **Debug your code incrementally. If your code does not run, it is unlikely you’ll get a decent grade on the project, or the class in general.** Always be working from a stable source, and add features incrementally (re-watch the week 1 lecture for hints on how to do this)

Please contact the TA or I with any questions. And START EARLY ☺