IS590PR Final Projects – Fall 2019

Due Dates: Nov. 12ish=proposal in forum; Dec. 3rd&4th or 10th&11th =in-class presentations & feedback; Dec 15th=final files in GitHub

Overview and High-Level Requirements:

1. You may work individually or as a team of up to 3 students. Multi-member teams must clearly show collaboration from every member and are expected to perform at a proportionally-increased level of complexity, sophistication, depth, and/or scope to earn a high grade.
2. Select from one of these TYPES of analytics projects to implement in Python:
   I. An original Monte Carlo simulation
   II. An original analysis linking 2+ published data sets from distinct sources, investigating some topic like: societal or environmental changes possibly affected by changes in laws, industry, new inventions, or corporate practices; or analysis of complex biases within data; or testing scientific hypotheses; or uncovering evidence of corruption in any industry, company, or government; historical changes of health, economic, or other aggregate life quality factors between countries, cities, or similar.
   III. A formal critique of weaknesses, flaws, or limitations in a published data analysis and major code rewrite/enhancement of the program to improve the rigor of the analysis AND make the code more reliable & maintainable.
3. Do not select a Machine Learning-focused project. Enroll in Machine Learning Team Projects for that interest. Thus, libraries such as sklearn, PyTorch and TensorFlow should not be used here.
4. You must submit your original unique work, created specifically for 590PR. If the project is related to work you did earlier or are now doing for any other course or a job, you must get prior written approval from all the relevant instructors and the supervisor. Not doing so is subject to sanctions per the Student Code.
5. PROPOSAL Stage. Post the summary into Moodle's **"Final Project Topic Proposals"** forum. See expected information there.
6. Unlike all other assignments in 590PR, you are allowed and expected to openly publish your unique project work. You may consider it part of your student portfolio, link it from resume, etc. Typically this is done on Github.com since you'll be committing there as work proceeds.
7. PRESENTATIONS & Draft Stage: You will create and deliver a presentation that summarizes your project's purpose, hypotheses, design reasoning, and results so far. The program should be sufficiently operational for meaningful and beneficial code review, but does not have to be 100% final. Make sure your GitHub repository is up to date with all the work you've done so far (code, documentation, example outputs). It's okay if there are some final scenario explorations or even minor flaws left to resolve in your project at this stage, but you want constructive feedback from others. All students will also be submitting formal evaluations about other teams' projects, details will be given in class.
8. Final submission expectations:
   - PROPERLY CITE ALL YOUR SOURCES! Any citation style is fine, but make sure you do it. Students who have used code or materials without clearly indicating its true source AND

delineating which parts are not original have received failing grades as sanction and been reported to iSchool & UI through FAIR.

- ☐ Edit your README.md to create a good introduction and overview of the project, written with new visitors to your repository in mind. Summarize the conclusions you came up with, including how results are either supportive of or refute your hypotheses. You can embed images into the README file, if that is relevant.
- ☐ Use the "Factors in Code Quality and Code Reviews" like a checklist. Apply as many of the skills we've discussed this semester as applicable, to create the best quality program you can. Example expectations:
    i. Doctests or other unit tests. 1-person team minimum 30% actual test coverage; 2-person minimum 70% actual test coverage; 3-person minimum 80% actual coverage AND use Travis CI during development.
    ii. All functions (methods included) need complete Docstrings. 1-person=minimum 4 functions; 2-person=7+ functions; 3-person= 10+ functions
    iii. 3-person teams also should incorporate one of these efficiency techniques that will be discussed in class (Numba or Cython compilation or parallel processing).
- ☐ Consider each hypothesis or alternative situation you proposed to investigate (you may have added more after feedback). Your program code should be able to run the simulation for all of the hypotheses just by changing parameter values and/or through using different input data files -- do not hard-code such configuration aspects into the functions themselves.

(Type I projects) Specifics for Monte Carlo Simulations:

- ☐ Design your own scenario. Make certain your simulation is ORIGINAL in some way(s).
- ☐ You can simulate an engineering or manufacturing problem, business/management situation, (certain types of) human behaviors, physical phenomena, or a game. To encourage original thinking, **AVOID scenarios that have been done many times and/or discussed in class:** a "random walk" of stock prices, stock options, or similarly naïve financial "predictions"; a traffic simulation with only one intersection or only one road; parking lots or parking meters; customer seating/dining at a restaurant or serving them at a counter; the games *Tic-tac-toe*, *four*-in-a-row*, *Go-moku*, *chutes and ladders*, *Monopoly*, *Rock-Paper-Scissors, Blackjack, Poker*. If you want to model a sports game or tournament, ask about it – the obvious things to do are not original.
- ☐ You must have several well-chosen random variables in the model, to explore a variety of possible outcomes and derive the non-obvious probabilities of the overall model. Make sure you think carefully to choose appropriate ranges of values and a sensible distribution type for every randomized aspect! For example, if you simulate "number of swimmers in the pool" at some point in time as a uniform distribution, it's wrong. If you simulate the individual finish times of all runners in a marathon as uniform, triangular, or even normal, it's wrong. We'll discuss this in class.
    i. 1-person teams must have at least 2 different kinds of randomized variables plus the deterministic aspects to the model. Most interesting simulations require more.
    ii. 2-person teams must have at least 4 different kinds…
    iii. 3-person teams must have at least 6 different kinds …

- If there is any relevant public data available, try to incorporate real data as part of your simulation model. For example, any sports simulation must use real performance statistics about players and/or teams so the randomized variables get sampled from realistic ranges & distributions. If data you seek is not in downloadable form, you can still research the scenario in books to avoid making flawed designs.

(Type II Projects) Specifics for an Original Data Analysis [Non-simulation]:

An original analysis linking 2+ published data sets from distinct sources, investigating some topic like: societal or environmental changes over time (possibly) affected by changes in laws, industry, new inventions, or corporate practices; or analysis of complex biases within data; or testing scientific hypotheses; or showing evidence of corruption in any industry, company, or government; historical changes of health, economic, or other aggregate life quality factors between countries, cities, or similar.

There are *thousands* of public data sets that could be of interest to you, from many US and foreign government agencies, scientific organizations, universities, companies, and more.

Some of the earlier assignments this semester were similar to this type of project, but simpler. This project should be a bit larger in scope or sophistication than any individual assignment you did earlier in 590PR, especially for multi-student teams.

(Type III Projects) Specifics for a Formal Critique of Weaknesses in a Published Data Analysis:

This type of project is very flexible, but might require more effort during the Proposal stage, to make certain that the previously-published analysis you choose has sufficient weaknesses in the code, data, or conclusions to justify reworking it. The original publication you critique and rework does *not* have to be a scholarly peer-reviewed analysis (you'd have a hard time identifying a good candidate in the short time we have). So it could be a less formal analysis that was published on an open website (e.g. a blog, Kaggle.com, or millions of other places).

I don't want to see a project where you essentially just convert an already-good analysis from another language into Python. The original might even be in Python already, but it needs to be somehow flawed, fragile, or incomplete in its code, statistical analysis, and/or conclusions. That could have happened because they used biased or incomplete data (that you'd need to improve, augment, or work around), made logical mistakes during analysis, or based conclusions on incorrect (buggy) software. Another more subtle possibility is that they improperly ignored uncertainties within the raw or processed data.