EAS 560 – Data Science Challenge

(Due: Friday, 7 May 2021, 5:00pm)

The goal of this data science challenge is to **practice the development, implementation, and application of a real-world data mining and modeling code**. Your challenge is to take one of the following COVID-19 modeling codes, which are all used in the actual modeling of the current disease spread and impact predictions, and continue its development on your own. You can also choose to start this development from scratch, in which case you are free to reuse parts of these codes as you see fit.
https://medium.com/data-for-science/epidemic-modeling-101-or-why-your-covid19-exponential-fits-are-wrong-97aa50c55f8

https://www.kaggle.com/lisphilar/covid-19-data-with-sir-model

https://towardsdatascience.com/building-your-own-covid-19-epidemic-simple-model-using-python-e39788fbda55

https://towardsdatascience.com/visualise-covid-19-case-data-using-python-dash-and-plotly-e58feb34f70f

Your tasks in detail are to:

1) understand the modeling code you have chosen;
2) add new features and functionality to the code or the underlying mathematical models, thus increasing their scope and level of detail, as well as improving the results/insights they yield;
3) improve and expand the provided code as needed consistent with best practices for producing a deliverable software product. That means, your code should have adequate documentation and commenting; if you implement plotting functionality, the resulting images should be *meaningful*, *attractive*, and "*publication quality*"; if you produce text outputs, they should be nicely formatted; you may want to employ an integrated development environment (e.g., Eclipse, Spyder, PyCharm) and version control (Git and your own Github repository);
4) apply your code and the new models you have created to test different hypotheses and scenarios that may be of interest and attempt to validate these using available data (e.g., from Johns Hopkins University or the WHO); make sure that your reports clearly state your hypotheses and findings;
5) plan and document all your development activities and modeling "experiments" using the Redmine project management software (I will make a dedicated Redmine instance accessible to you shortly); **I expect you to interact with Redmine daily**, if only to log time you spent on a particular task (I will assume that you have not worked on your project on days you have not logged your work); it is good practice to have a browser tab with Redmine open at all time and log your efforts as you are conducting them;
6) every Friday, 5pm, (starting on 02/12) you have to submit a 1-page summary of your past week's activities and your plans for the next week to me; if warranted, you may submit longer weekly reports; please only report new results from week to week, not repeat all old results over and over (except potentially for comparison with newer results); you should base your reports on your Redmine activities and the two should be consistent; include a screenshot of your Redmine Gantt chart in your weekly reports;
7) on the last day of the challenge (05/07), you will submit a summary report of your project (which you may base on your weekly reports) of at least 10 pages as well as your code, documentation, and a screencast demo/tutorial of your work; the report has to include a Redmine Gantt chart of all your project activities; the documentation has to clearly highlight and enumerate the changes you made and the new features/functionality you implemented.

You have 15 weeks to work on this open-ended, free-form assignment, and **you are expected to deliver a quantity and quality of work that is in line with the full-time effort of a senior data science graduate student over such a period of time.**

Rules for the reports:

- pdf file format
- 1-inch margins
- 11pt Times New Roman font (9pt for captions, 12pt for section headers)
- single line spacing; 1 blank line between sections/subsections; 0.5 blank line between paragraphs
- justified orientation of text
- images and bibliography do not count towards the page requirement
- within these constraints, the report has to be formatted in clean, consistent, visually appealing fashion (e.g., the scaling and positioning of images, added page numbers, informative headers/footers if warranted, sensible emphasizing *via* bold, underlined, or italics fonts; consistent indentations of bullet points; well aligned table contents; meaningful numbers of digits in reported numbers; etc)
- required sections:
    - 1. Introduction
    - 2. Background [of the project field and employed data science techniques; broad overview that contextualizes the performed work]
    - 3. Methods [technical details of the employed data science approaches, e.g., codes, algorithms, (hyper-)parameters, setup, etc, needed to reproduce the presented work]
    - 4. Results and Discussion
    - 5. Conclusions and Outlook
    - 6. Bibliography/References [do not count towards page requirement]
    - if you wish to omit required sections, this needs to be pre-cleared; additional sections or subsections may be added if warranted
    - sections and subsections have to be sequentially numbered and reflect a logically concise scheme that makes it easy to navigate your report [e.g., 1, 1.1, 1.2, 1.3, 2, 3, 3.1, 3.2, 4, …]
- use the formal language that is typically found in technical reports or scientific papers, not informal language (i.e., use first- or third-person active voice; no short forms, no colloquialisms; use full sentences in general, not only bullet points [although these may be used where warranted])
- make sure that the text uses correct English grammar (including syntax and punctuation) and spelling
- the write-up has to be specific, substantive, compelling, and comprehensible; precision is a virtue, i.e., fluffed out, waffling, and/or filler text will be penalized
- make sure to cite, credit, and attribute the work of others
- include a Redmine Gantt chart of all your project activities
- **please make sure you follow all instructions exactly. Non-compliant reports will receive poor marks!**

We expect the quality at the level of a professional technical report written for public dissemination!
Your report will be evaluated for both the content and the communication skills it exhibits!

Please submit your weekly reports, as well as your final report and zipped software package (or a link to a repos) back to me *via* email. Please use the following email subject line "EAS 560 DSC submission by <your name>" so that I can keep track of everyone's work. Please use your **@buffalo.edu account** when you communicate *via* email.

Your submission will be graded based on:

- the quantity and utility of new features/functionality, the quality of your implementation (including efficiency, documentation, readability, clarity, design, extensibility), and whether your program works correctly or not; as part of the evaluation, we will try to run your code, so please provide adequate instructions for how to do so;

- the quality of your hypotheses, modeling “experiments” you have conducted, as well as their results, derived insights, and discussion you provide;
- the quality (and compliance) of your reports as well as the screencast demo/tutorial;
- the way you utilized tools such as Redmine or Github.

Failure to follow all assignment instructions or work that is obviously nonsensical may result in penalties. Particularly original solutions may be rewarded. Submissions that are disorganized, illegible, or otherwise unprofessionally presented may receive penalties.

This assignment comes with an additional perk: At the end of this challenge, you can make your code publically available (e.g., on GitHub, Sourceforge, BitBucket) and use it as a work sample or part of your portfolio. This will come in handy should you ever decide to apply for a job that requires coding skills and proof thereof.

Tip 1: If you have trouble getting your code to run, try to google the error message you get. They will likely lead you to a useful stackoverflow page that will help you solve the problem.
Note: **You may collaborate, exchange ideas, and interact on technical questions with your classmates, but every student has to submit an individual solution that has been independently written up. No two submissions can be alike.** Collaborators should be listed and contributions should be credited. Failure to do so constitutes academic dishonesty and carries penalties as discussed below.

Breaches of academic integrity (e.g., plagiarism, cheating) are unacceptable and will result in a failing grade for the entire course. It’s not right and it’s not worth it! Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university and of themselves while facilitating the university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas. It is expected that you behave in an honorable and respectful way as you learn and share ideas. To summarize UB’s policy on dishonesty: A student will not present, as his or her own, the work of another, or any work that has not been honestly performed; will not take any examination by improper means, and will not aid and abet another in any dishonesty. Please consult UB’s Academic Integrity Policies at:
https://catalog.buffalo.edu/policies/integrity.html
https://grad.buffalo.edu/succeed/current-students/policy-library.html#academic-integrity