Lecture 10

Outline

- Assignment 3
- Project Proposal
- How to plan a project: agile development
- A peek at Quarter II

Assignment 3 (Report)

- Puts together "full replication" (contains 1 and 2).

- Use section resources (e.g. work being replicated) as a model for how to write the report.

  - How much detail? Someone with background in the area, reading your report, should understand it *without* references to other resources. Experts should be able to assess your code from the content alone (*without* looking at your code).

  - Remove code cells using nbconvert:

    - `jupyter nbconvert <nb.ipynb> --no-input --no-prompt`

Assignment 3 (Code)

- Puts together *the entire* pipeline (data, processing, analysis).

- What should be library code? What should be in notebooks?

    - If a notebook takes a long time to run, *then it shouldn't be in a notebook!* (use library code + script => save data locally to load into notebook).

- You code should be *clean* and use the directory structure.

    - Notebooks may be cleared & executed! (No unnecessary ones).

- Code should contain your test data (see assignment 3) and will be run on the DSMLP server via the commands:

    - `launch-scipy-ml.sh [-i your_docker_hub/your_docker_image]`

    - `cd project && python run.py test-project`

Project Proposal

- Project Proposal Contains 3 parts:
    - Broad (what is the problem? why do we care?)
    - Specific (what approach are you trying?)
        - Task-oriented schedule (is the proposal reasonably scoped for the time allotted?).
- The broad portion *guides* you to related tasks if the first specific approach doesn't work out (it likely won't!).
- The specific portion grounds you, getting you started.

Proposal Comments

- The proposal is living document, it will change!
- Your domain expert may have you revise it (and that's ok!)
- Schedule is a *starting point* to be honest about your plan:
  - Look to the domain schedules this quarter as a guide.
- First addition: think of a *baseline* for your proposed project to measure your results against, and incorporate it into your proposal and schedule.

Creating a Schedule

- Problem: how to split up work on a project so that everyone can work at the same time, without blocking others' progress?

- Approach 1: work on independent (or parallel) tasks when possible.

- Approach 2: Use 'stubs' to begin on work later stages before earlier ones are finished.

- Requires defining small scope tasks and frequent communication.

Planning Examples

- Task: Classify conflict on Wikipedia contributions.
- Obs 1: Work on *scaling data ingestion* alongside calculating test-statistics/building models by using small test-data samples for later work.
- Obs 2: Once data ingestion is finished, create data-quality EDA alongside building models.
- Obs 3: Simultaneously prototype different possible statistics/models that capture the notion of 'conflict'.

Planning Examples

- It's important in each of these to have frequent communication between group members on progress:
    - Did format of data ingestion change?
    - Does the EDA suggest trying a different model?
    - Is the 'stub' you are depending on schedule? (or does that task need more help?)
- We will borrow from Agile Methodology in software development to help each team's project development.

Next Quarter: Sneak Peek

- Project consists of:
    - written report(s): a paper, a visually effective blog, art-piece, etc
    - code artifact: versioned and deployable project that follows best-practices for reproducibility and extension.
    - Oral presentations: presentations of varying length, appropriate to different audiences (general, specialized; 3-20 minutes)

Next Quarter: Sneak Peek

- Lecture will consist of the following content:
    - Observations for working in teams (Agile Methodology; proper GitHub conventions).
    - Communicating via writing and visual content (theory and tools).
    - Oral presentations (including practicing in class!)
    - Details on what project code must satisfy.

Next Quarter: Sneak Peak

- Before each Wednesday domain each team will update each other on their progress for the week (in-person, or on e.g. Slack).

- In each Wednesday domain, one (randomly chosen) person from each team will summarize:
    - The progress the team as a whole has made that week,
    - What problems were encountered in working the last week, and discussion about what's needed to solve those problems if they're impeding progress. We will spend most our time discussing these.
    - What the team will be attempting for the next week.

- These updates are also submitted via Google Forms.