CMSC320 – Introduction to Data Science
Final Tutorial

Fall Semester 2021

John Dickerson
University of Maryland
Department of Computer Science

1 Summary

In lieu of a final exam, CMSC320 students will turn in a tutorial that will walk users through the entire data science pipeline: data curation, parsing, and management; exploratory data analysis; hypothesis testing and machine learning to provide analysis; and then the curation of a message or messages covering insights learned during the tutorial. Students may choose an application area and dataset(s) that are of interest to them; please feel free to be creative about this! For some ideas and possible data sources, see the slides from the first lecture; I've also listed a few sources below.

- Awesome Public Datasets, a crowdsourced-then-curated list of public datasets: https://github.com/awesomedata/awesome-public-datasets
- US government data: https://www.data.gov/
- Maryland state data (similar websites exist for other, worse states): https://opendata.maryland.gov/
- Microsoft, Amazon, Google, and other large tech firms host their own collections and provide search functionality (e.g., https://datasetsearch.research.google.com/)

The tutorial should be self-contained, a mix of Markdown prose and Python code, and delivered as a GitHub statically-hosted Page (described below).

As example tutorials, check out (some links may not work; not all results are completely correct, but these are examples of very good efforts that "fit the bill"):

- The golden age of rap: http://rstumbaugh.me/hiphop-analysis/
- Predicting a win in Rainbow Six: Siege: https://jiglesia3.github.io/
- What makes the best defensive footballers? https://bdaisey.github.io/
- Analysis of global suicide data: https://summerzzzy.github.io/
- Prediction of Alzheimer's and dementia: https://amygracecruz.github.io/
- Maryland and peer institutions' faculty/student counts: https://krixly.github.io/.
- Analysis of crime data in College Park: https://andresgogo.github.io/

- Analysis of Freddie Mac's Single Family Loan-Level data: https://amulyavelamakanni.github.io/data-science-pipeline-tutorial/

In general, the tutorial should contain at least 1500 words of prose and 150 lines of (non-padded, legitimate) Python code, along with appropriate documentation, visualization, and links to any external information that might help the reader. You are welcome to do this project individually or in a group of size at most three; we'll scale up the expectations accordingly as group size increases.

1.1 Github Pages

GitHub provides a service called Pages (https://pages.github.com/) that provides website hosting functionality backed by a GitHub-based git repository. We would like you to host your final project on a GitHub Pages project site. To do this, you will need to:

1. Create a GitHub account (or use the one you already have) with username `username`.
2. Create a git repository titled `username.github.io`; make sure `username` is the same as whatever you chose for your global GitHub account.
3. Create a project within this repository. This is where you'll dump your iPython Notebook file and an HTML export of that Notebook file.

These instructions are also given directly on the front page of https://pages.github.com/; following those instructions should be fine!

1.2 Deliverable

The deliverable to the CMSC320 staff will then be a single URL pointing to this publicly-hosted GitHub Pages-backed website. It is due by the CMSC320 university-wide pre-scheduled date of **4:00PM on Monday, December 20th**. We will not (*cannot*) accept late assignments.
Please make sure to include your name (and the names of all group members) at the top of your deliverable, after the title.

2 Grading

We will assign a numeric score between 1 and 10 for each of the following six dimensions:

1. **Motivation.** Does the tutorial make the reader believe the topic is relevant or important (i) in general and (ii) with respect to data science?
2. **Understanding.** After reading through the tutorial, does an uninformed reader feel informed about the topic? Would a reader who already knew about the topic feel like s/he learned more about it?
3. **Other resources.** Does the tutorial link out to other resources (on the web, in books, etc) that would give a lagging reader additional help on specific topics, or an advanced reader the ability to dive more deeply into a specific application area or technique?

4. **Prose.** Does the prose portion of the tutorial actually add to the content of the deliverable?

5. **Code.** Is the code well written, well documented, reproducible, and does it help the reader understand the tutorial? Does it give good examples of specific techniques?

6. **Subjective evaluation.** If somebody linked to this tutorial from, say, Hacker News, would people actually read through the entire thing?

Dimension	Points Received	Points Possible
Motivation		10
Understanding		10
Further Resources		10
Prose		10
Code		10
Subjective Evaluation		10
Total		60