Data200 Capstone Project

Due April 6th; Presentations on April 5th.

Show me what you know!

Do a passion project that demonstrates what you've learned in this course. It doesn't need to be an in-depth, complete analysis - processing some data (regex or feature engineering) and doing exploratory data analysis is enough.

Your project will have two components:

1. A Jupyter notebook with all of the code needed to run your analysis. There should be markdown cells to describe the goal of the code cells, but they do not need to be a self-contained report.
2. A 10-15 minute oral presentation that describes your analysis. The presentation complements the notebook; this is where you can explain the logic and the process behind your analysis.

Ideally, the notebook can be run step-by-step with each step justified in a markdown cell. The end of the notebook should contain a brief (possibly point-form) section with a plain language interpretation of the results. The presentation will expand on the interesting parts of the study and motivate people to read your notebook. You may want to focus on the data collection, processing, and cleaning, since these are difficult and important steps.

An incomplete project can still get full marks. If you are unable to collect data, you can simulate it according to what you would expect and still demonstrate knowledge of course content. In this case, you should give clear details about how the data would have been collected (keeping in mind the concepts from the first two lectures). Focus on demonstrating knowledge, not superficial polish.

Your project can be closely based on an assignment, but you must demonstrate further knowledge of other concepts.

Note that the notebook is due after the presentation. I recommend getting finalized results as fast as possible, making the presentation, then polishing the notebook.

Project Ideas

- Find the transcripts of a TV show or movie series that you like. Search for lines spoken by one character about another character, and find the sentiment of the sentence (from VADER). Write down some information about each character from your own knowledge or other sources (such as age, race, etc.) along with information from the script (whether certain words appear in the same sentence, who the speaker was, sentiment). Fit a basic linear model to see what factors influence sentiment about given characters.
- Combine and analyse event data from football matches (here's a link to the data on Kaggle). The data spans multiple `csv`s, which must be joined in the right way before fitting a model. There are columns that may have information that can be extracted with regex.
- Compare students' smartphone usage to their answers to one of various survey questions. A sample dataset can be found here (note the "URL" links in both resources - you don't need to log in to get any data).
- Predict something about a song based on it's qualities (how many words, energy, time signature, etc.). An example dataset can be found here, which includes a target variable describing whether a song was a hit or a flop.
  - Alternatively, this dataset includes a spectral decomposition of the audio files for various songs. The associated code does a multi-class classification. An alternative project might involve you personally rating how much you like each track, then doing a regression of your ratings to see which features of music are more impactful to you (this could be done with the "hits" dataset as well).

Most data sets have already been analysed elsewhere. I strongly encourage you to familiarize yourself with others' work, but this project should represent your own efforts. I encourage the use of ChatGPT to help automate tasks that are not demonstrative of your knowledge, but I have absolutely no tolerance for students who present ChatGPT output (or any other work) as their own.

Evaluation

- **Presentation**
  - Quality of the oral presentation *and* the readability/presentation of the notebook.
- **Accuracy and Completeness**
  - No errors in information.
  - All relevant information is provided *concisely*. The report should not be a teaching resource, but you should define any concepts.
    - Example: "In this code cell, I use regular expressions, which is *blah blah blah*, to extract the features." (If you do something interesting with the regex, feel free to explain the expression as well.)

- **Content**
  - Covers *many* concepts from the course in the same analysis.
    - A really good regression on simple data would be worth less than a basic regression including features engineered from a regex pattern and requiring data cleaning/joining, with interpretation of results/errors discussion of the data collection.
  - For examples of "concepts" that you should consider, see slide 10 of Lecture 01. This is not an exhaustive list - concepts such as standardizing and covariance are fair game.

Bonus Marks

There are **5 bonus marks** available for peer-to-peer evaluation. There is 1 mark available for doing a practice presentation (possibly virtual) for your peers, and 1/3rd of a mark per peer for providing feedback via a provided rubric. The rubric should be fully completed, including specific comments and feedback about the content, and given both to the presenter and to me (I will not be at the practice, but I will be looking carefully at the comments/feedback to verify that you were present at the practice). Another three bonus marks are available for evaluating the presentations (1 per peer).

There are **3 bonus marks** available for publishing your results to a public-facing portfolio, especially git-based services such as GitHub, GitLab, or BitBucket. You may also wish to publish to a personal blog or to the Kaggle competetion where the data came from. To get all 3 bonus marks, the post/repo must:

1. Be easy to navigate
   - An `ipynb` and your slides that describe different aspects is not enough, but a single, self-contained, well-written `ipynb` is enough.
   - The code must be easy to follow and be well documented.
2. Be an advertisement for your expertise
   - On GitHub, you might want a detailed Readme.md file that describes how great your analysis is and how it demonstrates your expertise. For a personal blog, add an overview explaining which skills this demonstrates.
3. Be reproducible
   - You may assume that the reader can access the Kaggle data (or any other publicly available data), but any supplementary data should be included (*e.g.*, if you use your own subjective ratings for songs).
   - The notebook should call all modules/functions in the first cell. Readers should not get halfway through your work and then discover they need to install another module.
   - You may want to use an auxilliary code file, and if so this must be easily available with your code.