CSE 519 -- Data Science (Fall 2023)
Prof. Steven Skiena
Homework 2: Exploratory Data Analysis in iPython
Due: Thursday, September 28, 2023 (9:45 AM)

This homework will investigate doing exploratory data analysis in iPython. The goal is to get you fluent in working with the standard tools and techniques of exploratory data analysis, by working with a data set where you have some basic sense of familiarity.

This homework is based on the CommonLit Challenge on Kaggle, revolving around measuring the quality of written summaries (both for content and wording) produced by grade school children. More than just data exploration, you must also join the challenge and submit your model, to get a score from Kaggle. **I expect your Kaggle scores to be terrible on this** because we are not using modern NLP technologies, but instead exploring the data to uncover interesting observations about student summaries. You will need to submit your code files in three different formats (.ipynb, .pdf and .py). Make sure to have your code documented with proper comments and the exact sequence of operations you used to produce the resulting tables and figures. The submission steps are discussed below.

Data downloading

First of all, you need to join the challenge and download the data here. The description of the data can also be found at this page.

Python Installation

Instead of installing python and other tools manually, we suggest installing **Anaconda**, which is a Python distribution with a package and environment manager. It simplifies a lot of common problems when installing tools for data science. More introduction can be found here. Installation instructions can be found here.

Another option can be using **Google Colaboratory.** This is another option for those who want to run their Jupyter notebook remotely instead of installing the required packages locally. Colab allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

If you are an expert of Python and data science, what you need to do is install some packages relevant to data science. Packages that I believe you will definitely use for this homework include:

- pandas
- scikit-learn

- numpy
- Matplotlib
- seaborn (maybe)

The Google colab notebook (must access using @stonybrook.edu, not @cs.stonybrook.edu) contains boilerplate code to download the data to your google drive and a dictionary containing the features along with its data type. **Make a copy of the notebook before you start your HW.**

Tasks (100 pts)

1. Load the summaries_train.csv and prompts_train.csv files, joined to replace the prompt_IDs with the relevant text fields into a dataframe. (5 points)
2. Construct a table of five features (really 7) from the text for each instance: (10 points)
    1. Number of words in student response (text) and prompt (prompt_text)
    2. Number of distinct words in student response (text) and prompt (prompt_text)
    3. Number of words common to student response (text) and prompt (prompt_text)
    4. Number of words common to student response (text) and prompt_question
    5. Number of words common to student response (text) and prompt_title
3. Now fortify this list with at least five other numerical features. Consider readability indices, counts of words from particular classes (e.g character length, part of speech, popularity). Use your imagination as to what might be helpful for identifying well written summaries of texts. (15 points)
4. Look at the distributions of scores for content and wording, as histograms and scatterplots? What is the range of values here? How well correlated are they? Do the shapes of these distributions differ for the different prompts? (10 points)
5. Which words are over-represented in good essays (as per content and wording) while being under-represented in bad ones? Conversely, which words appear disproportionately in the bad essays? What is an appropriate statistic to use here? (10 points)
6. Create **three** plots of your own using the dataset that you think reveal something very interesting. Explain what it is, and anything else you learned from your exploration. (15 points)
7. Now build a baseline model for this task. We will call this *Model 0.* You will train linear regression models for both content and wording on 80% of the training data and test it on the remaining 20% *chosen at random*. Use only the original five features described above. Report the mean squared error of each model. What do you make of the error rate? (10 points)
   [Note: These baseline models should be **terrible** at this particular task. They are usually used for checking out some preliminary ideas and their performance.]
8. The basic features as defined above are not really suited for the task. Features can be preprocessed (or cleaned) to improve them before feeding into the model (e.g. normalize them, do a special treatment of missing values, etc). This can significantly improve the

performance of your model. Do preprocessing for all the features (the original five plus the extra you add). Explain what you did. (10 points)

9. For each of the two tasks (content and wording) create two models:
    - *Model 1* should use the cleaned features and linear regression for training. You can do some (potentially non-linear) scaling to keep the scores in range.
    - *Model 2* should use the cleaned features and an algorithm other than logistic regression (e.g. Random Forest, Nearest Neighbor, etc) for training.
      [Note: *scikit-learn* is a user-friendly library which is used to perform data loading, pre-processing, transformations, algorithms and metrics needed for Data Science and Machine learning]

Compare their performance and explain your reasoning for the differences in their performances. (10 points)

10. Submit your scores on the test set to the Kaggle competition for the best model you develop. Report the private and public score for your best submission along with the number of submissions. Include a snapshot of your best score on the website as confirmation. Be sure to provide a link to your Kaggle profile. (5 points) Further update: because students are having a hard time with the uploads and cannot get a score before the deadline, part 10 is now optional: every student will get 5 points whether they submit or not.

Be honest. This is your first modeling experience, and I am hoping to see you learned something, not where you are ranked on the leaderboard.

Rules of the Game

1. This assignment must be done **individually by each student**. It is not a group activity.
2. If you do not have much experience with Python and the associated tools, this homework will be a substantial amount of work. Get started on it as early as possible!
3. All of your written responses should be put in the appropriate place in your notebook template. Get the template notebook form from here. *You must access the notebook via your @stonybrook.edu email address, not @cs.stonybrook.edu!*
   You are allowed to add more cells, but definitely fill out the cells we give.
4. I will give a brief introduction to topics like linear regression in detail before the HW is due, with detailed treatment to come. Muddle along for now, and we will understand the issues better when we discuss them in the course. Feel free to read ahead in the book.
5. To ensure that you are who you are when submitting your models, have your Kaggle profile show your face as well as a Stony Brook affiliation.
6. There are some public discussions and demos relevant to this problem on Kaggle. It is okay for students to read these discussions, but they must write the code and analyze the data by themselves.

7. KISS is an important philosophy in data science: *keep it simple, stupid*.. For this homework that means that you should start by making a pass through the assignment doing simple things, instead of over-optimizing each part. *Then* go back and improve things where it counts once you know how simple works.
8. This is not a course on NLP, and the regression-based models I propose will perform **terribly** compared to modern LLMs. **You will not impress me if you use higher-powered technology as opposed to the problem-specific features we discuss, so do not worry about this.** But you are free to play around with them for Part 9 if you like.
9. You may use ChatGPT if you want, provided that you cite it through the class policy in the syllabus. But you will be doing yourself a terrible disservice if you use this to fake Python programming rather than learn it – now is your chance!
10. You will submit your code so we can run it through MOSS to detect copying and plagiarism. Do your own work!!
11. Our class Piazza account is an excellent place to discuss the assignment. Check it out at piazza.com/stonybrook/fall2023/cse519.

Submission

Submit everything through Google classroom. As mentioned above, you will need to upload:

1. The Jupyter notebook all your work is in (.ipynb file), derived from the provided template
2. Python file (export the notebook as .py)
3. PDF (export the notebook as a pdf file)

These files should be named with the following format, where the italicized parts should be replaced with the corresponding values:

1. cse519_hw2_*lastname_firstname_sbuid*.ipynb
2. cse519_hw2_*lastname_firstname_sbuid*.py
3. cse519_hw2_*lastname_firstname_sbuid*.pdf