HST.953 *Clinical Data Learning, Visualization, and Deployments* Friday, Oct 8
Massachusetts Institute of Technology, Fall 2021
Marzyeh Ghassemi and Leo Celi revised Wednesday 13th October, 2021, 05:24

Problem Set 2

Due Date: 11:59pm on **Friday, Oct 24, 2021**

Submission: You will need to make two submissions:

- Please submit all code that you wrote and the output of your scripts (including diagrams) as a zip file (code.zip) under the "Problem Set 2 Code" assignment on Canvas. If you use Jupyter notebooks, submit both the ipynb file and generate an html file using nbconvert.
- Additionally, please submit a written report (report.pdf) under the "Problem Set 2" assignment on Canvas. The submission will be uploaded through Gradescope, where you will be able to select the individual pages in your report corresponding to each problem.

Late Submission: Late assignments will not be accepted without a valid medical certificate or other documentation of an emergency.

Collaboration: What you turn in must be your own work. You may not work with anyone else on any of the problems in this assignment. If you need assistance, ask on the Piazza board for this course, or contact the instructor or TA.

Grading: Make sure that your solutions to the problems are written clearly and concisely. You will be graded both for answering the questions correctly and for writing up your answers in a readable manner.

GitHub Repo: Unless otherwise noted, all scripts needed for this problem set can be found at: https://github.com/criticaldata/hst953-2021

Data: All the data needed for this homework can be found at: https://drive.google.com/drive/folders/1antnaOopVCldAMatFLWCV9Zy7rEhOuGe?usp=sharing

Problem 1. Charting Patient Data [2.5 points]

For the first part of this problem set, we will take a look at visualizing the MIMIC-III data itself. Use the provided notebook and add your own code to it to answer the questions below.

(a) Write an SQL query to load from the chartevents table all the Respiratory Rate measurements (Item ID 618), and the times of those measurements, of patient 7363 during admission (HADM ID) #121492 into a pandas DataFrame.

(b) Use matplotlib to plot the Respiratory Rate data as a line graph. Include the plot in your report. From the graph, what was the Respiratory Rate of the patient on 2175-04-06 at time 06:00:00? (NOTE: it might be helpful to re-plot only the last few measurements to answer the last question.)

(c) Is the Respiratory Rate value you found in part (b) accurate? Would you be able to estimate that value in real time in a clinical setting? Why or why not?

(d) For each of the following patients and their corresponding admission IDs, plot as a bar chart the mean value of their SpO2 (Item ID 646) over the entire length of that admission. Based on your plot, which patients seem healthy? (Normal pulse oximeter readings are usually in the 95-100% range. Values below 90% are considered low and might indicate a need for supplemental oxygen).

Patient ID (SUBJECT_ID)	Admission ID (HADM_ID)
17440	169094
22389	100154
29123	185864
14212	189431
18867	120552

(e) Plot all of the SpO2 measurements you loaded in the previous part for the same patients as a violin chart. Based on your plot, which patients seem healthy? Does your answer change from part (d)? Why do you think that is the case?

Problem 2. Revisiting Mortality Prediction in the ICU [2.5 points]

In this problem, we will look at different ways of assessing the performance of our model. Recall that in problem 3 of problem set 1, we trained a logistic regression model to predict in-ICU mortality from different features.

Download the data for this problem (**adult_icu.gz**) from the link at the beginning of this problem set, and use the provided notebook to train a logistic regression model on that data. Then add your own code to the notebook to answer the questions below.

In what follows, you are expected to implement the performance metrics yourself rather than use an existing implementation of those metrics from libraries such as scikit-learn.

(a) For each of the training and testing splits of the dataset:

- What is the model's accuracy overall?
- What is the model's accuracy for each ethnic group (black, white, Asian, Hispanic, other)?
- Use matplotlib to plot the above results as a bar graph, and include the plot in your report.
- How well does the model perform overall? How well does it perform for each group of patients?

(b) We would like to investigate how well the model performs when assessed with different performance metrics.

- For each of the training and testing splits, compute each of the following performance metrics both for the model overall and for each ethnic group independently:
  - Precision
  - Recall
  - AUC Score
- Use matplotlib to plot the above metrics jointly as a single bar graph, and include the plot in your report.
- How well does the model perform overall? How well does it perform for each group of patients?
- State one discrepancy you notice in the performance of the model. What could be a potential reason for that discrepancy?

(c) Is the model fit to be deployed in a clinical setting? State one undesirable outcome that may arise from the use of this model in such a setting.

Problem 3. Visualizing ClinicalBERT Embeddings [5 points]

In this problem, we will explore different ways of visualizing the embeddings learned while training machine learning models. Rather than retrain a model from scratch, we will use the ClinicalBERT embeddings, which we have seen in problem set 1.

Download the data for this problem (**cohort.h5** and **embeddings.npy**) from the link at the beginning of this problem set. Then add your own code to the provided notebook to answer the following questions. A brief description of the data files can be found below:

- **cohort.h5:** Contains for each patient, and admission ID, a set of features corresponding to that patient obtained from the MIMIC-III dataset. This is a subsampled version of the file that was generated by the data preprocessing scripts in problem set 1.
- **embeddings.npy:** A num_admissions $\times$ embedding_size numpy array containing for each admission, the ClinicalBERT embedding of the last clinical note taken during that admission.

(a) The embeddings learned by ClinicalBERT are too high-dimensional. Use PCA (from scikit-learn) to reduce the dimensionality of the embeddings to 275 and store the resulting embeddings in a numpy array. (NOTE: we choose the PCA output embedding size to be 275 because an embedding of that size is sufficient to explain 95% of the variance in the data).

(b) Use t-SNE to visualize the embeddings you obtained in part (a). For each perplexity value in $\{5, 10, 20, 30, 40, 50\}$, plot the resulting visualization, and include the plot in your report. (NOTE: For this part, make sure you use the t-SNE implementation imported from the MulticoreTSNE package, as the scikit-learn implementation can be significantly slower.)

- How do the clusters in the plots change as the perplexity value is varied?
- For which perplexity value are the clusters the most well-separated from each other?

(c) Using the best perplexity value you found in part (b), plot the ground truth labels for COPD, Hypertension, and mort_icu (each on a separate figure) on top of the t-SNE visualization. Include the figures in your report.

- How well do the visualized clusters correspond to the ground truth labels?
- Provide one reason that could explain the above outcome.

(d) Use UMAP to visualize the same embeddings you obtained in part (a). Vary the values of the **n_neighbors** and **min_dist** hyperparameters and plot the corresponding visualizations. Include some of the resulting figures in your report.

- How do the clusters in the plots change as the hyperparameter values are varied?
- For which hyperparameter values are the clusters the most well-separated from each other?

(e) Using the best hyperparameter values you found in part (d), plot the ground truth labels for COPD, Hypertension, and mort_icu (each on a separate figure) on top of the UMAP visualization. Include the figures in your report.

- Which of the two visualization methods (t-SNE or UMAP) yields a visualization closer to the ground truth?
- Do you expect one visualization method to always be more accurate than the other? Why or why not?