Problem Set 2 (96 points)

MLHC 2025

March 9, 2025

Submission Guidelines

Due 3/13/2025, 23:59 on Gradescope

Please make a submission only if you are registered as a regular student. Submit your write-up as `[mit_email]_pset2.pdf` (*e.g.,* `sophiejg_pset2.pdf`), with all written work in a single PDF file following the problem set structure. Please append your code at the end of the report. For those who prefer to typeset in LaTeX, the source code of this file is here .

Submit your own work and note any collaborators - if none, state "Collaborators: none." If using external sources, cite them properly. You should be able to explain your solutions verbally. Please do not share your code or report with anyone inside or outside of the class, nor post them publicly online. Course staff welcomes any questions about these policies. Please see the "Late Policy" on the course website (`https://mlhcmit.github.io/`).

1 Predicting Sepsis (46 points)

Background: Sepsis is a serious and wide-spread health issue, contributing to 6 million deaths per year (4). Sepsis occurs when an infection causes a systemic inflammatory response, disrupting normal physiologic functioning. This can lead to septic shock, a situation in which the body cannot maintain proper blood pressure and so organs do not get the perfusion they need. Early deployment of broad-spectrum antibiotics and fluid resuscitation can save lives by the hour (1). This has inspired a goal of sepsis prediction using machine learning (2). In this problem set question We will use the dataset from the 2019 PhysioNet Computing in Cardiology Challenge in this problem (3).

1.1 Exploratory Data Analysis (10 points)

The starter notebook can be found here: in this link

1. (1 point) What is the fraction of patients that eventually develop sepsis?

2. (1 point) What is the Gender distribution of the patients? Additionally, what is the average and median age of patients?.

3. (2 points) Plot a histogram of the length of patients data.

4. (2 points) Plot a histogram of the first hour in which patients develop sepsis.

5. (4 points) Plot a histogram of the how often each feature is missing across all patients. For example, if patient1 has feature1 missing 2/40 (across the 40 hours of their stay), and patient2 has feature1 missing 5/30, the missingness for feature 1 is $\frac{2+5}{40+30}$. What 5 features have the most missingness?

1.2 Learning a model to predict sepsis (36 points)

We are going to build a machine learning predictor to predict whether a patient will develop sepsis using data until 3 hours before they develop sepsis . The goal is to build our predictor and evaluate it on a test set. We will try to guide you on this journey.

The general steps will be:

- Split the data appropriately into a train, validation and test set.
- Impute missing features for each patient in a valid way that is informative. A valid way to impute missing features with 0, however, this not a very informative way to do things. (feel free to look up ways online).
- Appropriately trim each data frame depending on the time of sepsis (if sepsis occurs).

- Do some feature engineering for each data frame.
- Train a neural network on the data.
- Evaluate the performance of our neural network on the test set.

More specifically:

1. (10 points) Impute missing values for each feature in each dataframe appropriately. Explain your methods taken to impute these values.

2. (10 points) For each patient we will construct a single feature set X that summarize all their data. For patients who develop sepsis, only retain data until 3 hours before sepsis develops. For patients who don't develop sepsis, only keep the first 20 hours of their data. Then construct a single feature set X that summarizes the variablity across the hours of their stay. Specifically, for each time-varying feature, create features that summarize the 1) average, 2) minimum, 3) maximum. Describe but do not implement, **TWO** other summary features that you may want to add. Each patient will now have instead of a dataframe of features rows for each hour, will now have a single row with summary features.

   Bonus (6 points max): Instead of constructing summary features, you can feed the data hour by hour into a recurrent neural network (3 points) or if you implement an LSTM model (3 points) on this data and evaluate it, or both (6 points).

3. (16 points) Feed the data into a neural network with PyTorch in a similar fashion to the notebook in recitation #4. Obtain the AUC on the test set and report it here. Please show all your code.

   For half the points (maximum 8/16): use an XGBOOST classifier instead of a neural network on PyTorch

   Bonus (4 points): In addition to implementing the neural network, implement an XGBOOST classifier.

2 Length of Stay Prediction Using Notes (50 points)

In the last assignment, you ran a logistic regression to perform length of stay (LOS) prediction using lab data. In this homework, you will predict whether a patient's LOS is lesser or greater than 7 days using **clinical notes**.

To prevent running into Google Colab's GPU rate limits, make sure you're connected to CPU when developing your code and **ONLY SWITCH TO GPU WHEN TRAINING. Ensure that you follow good machine learning principles**. The starter notebook can be found here: in this link. Please append your code at the end of this pset.

2.1 Evaluate data distribution and summarize notes (10 points)

Here, we have clinical notes taken from the first 48 hours of a patient's ICU admission. You can observe this using the `HOURS` field. This field was calculated by taking the time of the note with respect to their ICU admission time. Some of these values are negative, which means that it is a note from the ED, which is fine and we will still include them in our dataset.

1. (1 point) You may notice that the patients have multiple notes for each `HADM_ID`. One method to take into account all these notes is to summarize them with the help of an LLM, such as Gemini. Start off by concatenating all the notes from each `HADM_ID` together. Make sure you concatenate them sequentially in the order they were recorded and add a specifier, such as "New note:" in between. You should now have a dataset with one row per unique `HADM_ID`. Submit a screenshot of your processed dataset.

2. (2 points) Plot a bar plot of the sample distribution of LOS greater than vs less than 7 days.

3. (2 points) You will notice that the dataset is imbalanced between positive and negative samples. Although most machine learning algorithms handle this imbalance fine, to reduce the running time of ML for the purposes of the problem set, we'll create a smaller dataset with with just 7000 samples of each label for a total dataset of 14,000 samples. Plot a bar plot of the now balanced distribution.

4. (5 points) Now we can use the Gemini Vertex AI API to feed in a batch of our notes and generate summaries. Give an example of a sample note summary and visually inspect it. Comment on the quality of the summary as a function of its raw clinical note counterparts. Does the LLM do a good job synthesizing the patient story from across multiple notes?

2.2 Bag-of-Words Model (12 points)

1. (5 points) First implement a classic Bag-of-Words model with Logistic Regression for the classification task on the summarized notes using a 70/30 training vs validation split. Use `scikit-learn`'s packages, along with their `roc_auc_score`, `classification_report`, and `confusion_matrix` to evaluate the model. Report these metrics.

2. (5 points) Add `L1` regularization to your model. Experiment with a range of 5 different regularization constants. Plot validation AUC with respect to your regularization constants. Which constant provides the best AUC?

3. (2 points) Report the top 10 features with the most positive weights and the top 10 features with the most negative weights from the model that obtained the highest held-out AUC in the previous question. Explain what it means to be a positively weighted feature vs a negatively weighted feature.

2.3 Training the DistilBERT model (15 points)

We describe the requirements, as well as the steps you should take below:

1. Sample 70% of the dataset for training and 30% for validation. Use the same data splits from the BoW model for an effective comparison.

2. Run the summarized notes through the `distilbert` tokenizer, truncating any notes that are longer in length than 512. However, notice we set `maxTokenOutput = 512` in the Gemini API outputs earlier so most notes should be within 512 tokens.

3. Train a `distilbert` model on this task (SWITCH TO GPU). Train for at most 5 epochs, with an effective batch size of 64, and a learning rate of 2e-5. Evaluate at every epoch, and save the model that has the lowest validation loss. Generate a plot of the training vs validation loss, validation accuracy, and validation AUC curves across epochs. This will take roughly 30 minutes.

2.4 Test Set Eval Questions (8 points)

1. (1 points) What is the epoch with the best validation accuracy?

2. (1 points) What is the epoch with the best validation loss?

3. (2 points) Graph a histogram of the test prediction logits. Use 10 bins. Explain in 2-3 sentences what you are seeing (i.e., is the distribution normal/multivariate Gaussian, where do most of the values go in, etc).

4. (4 points) What is the **accuracy and AUC** of the best model on the validation set? How does this compare to the accuracy and AUC from the Bag-of-Words model?

2.5 Adding Lab Data (5 points)

Describe, BUT DO NOT IMPLEMENT, two different ways that you could add the lab data into the predictions.

References

[1] Ricard Ferrer, Ignacio Martin-Loeches, Gary Phillips, Tiffany M Osborn, Sean Townsend, R Phillip Dellinger, Antonio Artigas, Christa Schorr, and Mitchell M Levy. Empiric antibiotic treatment reduces mortality in severe sepsis and septic shock from the first hour: results from a guideline-based performance improvement program. *Critical care medicine*, 42(8):1749–1755, 2014.

[2] Joseph Futoma, Sanjay Hariharan, Mark Sendak, Nathan Brajer, Meredith Clement, Armando Bedoya, Cara O'Brien, and Katherine Heller. An improved multi-output gaussian process rnn with real-time validation for early sepsis detection. *arXiv preprint arXiv:1708.05894*, 2017.

[3] Matthew A Reyna, Christopher S Josef, Russell Jeter, Supreeth P Shashikumar, M Brandon Westover, Shamim Nemati, Gari D Clifford, and Ashish Sharma. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. *Critical Care Medicine*, 2019.

[4] WHO. Sepsis, 2020.