Problem Set 2 (68 Points)

MLHC 2023

March 2, 2023

Submission Instructions

This problem set is due 3-16-23 at 11:59pm EST. Please submit your write-up and code to Gradescope. When you write up your report, put all writing into one file and name it `mit_email_username_pset1.pdf` (e.g. `lehmer16_pset1.pdf`), furthermore make sure your report follows the same structure as the problem set. You must write up their problem sets individually. You should not share your code or solutions (i.e., the write up) with anyone inside or outside of the class, nor should it be posted publicly to GitHub or any other website. You are asked on problem sets to identify your collaborators. If you did not discuss the problem set with anyone, you should write "Collaborators: none." If in writing up your solution you make use of any external reference (e.g. a paper, Wikipedia, a website), both acknowledge your source and write up the solution in your own words. It is a violation of this policy to submit a problem solution that you cannot orally explain to a member of the course staff. Plagiarism and other dishonest behavior cannot be tolerated in any academic environment that prides itself on individual accomplishment. If you have any questions about the collaboration policy, or if you feel that you may have violated the policy, please talk to one of the course staff.

Late Day policy:

- (2 "slack" days) We understand that sometimes things outside one's control prevent submitting by the deadline. As such, each student is given 2 "slack" days that they can use throughout the semester (e.g. you could submit two psets one day late each or you could submit one pset two days late) without a late penalty. The days do not subdivide into sub-day units: 2 hours late would spend one of the slack days without 22 hours of "rollover". In your pdf writeup, specify how many slack days you are using (they cannot be used retroactively).
- (10% off per unexcused late day.) If you submit a pset 3 days late and use 1 slack day, then this is 2 unexcused late days, which translates to 20% off your homework.
- (max 4 days late.) Homework will not be accepted after 4 days late, regardless of how many slack days are used, absent communication from S3 or OGE.
- (write on homework) In order to use a slack day, students must include it in writing on their submission pdf. Otherwise, TAs will assume no slack days used and deduct 10% for each late day.

Scenarios:

- Sam uses 2 slack days on HW3. This is the first time Sam has used any slack days. Sam now has 0 remaining slack days and receives her homework score with no penalty.
- Jamie uses 1 slack day on HW3 but submits 52 hours after the deadline. Therefore Jamie is 3 days late (rounded up) and receives 20% off the graded homework. This is the first time Jamie has used any slack days, so Jamie now has 1 slack day remaining.
- Cory has never used any slack days, but submits HW5 100 hours late with no note / communication from S3 or OGE. This is more than 4 days after the deadline, so the work is not accepted, and Cory receives a zero.

1 Predicting Sepsis (38 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 (8 points)

The starter notebook can be found here: https://colab.research.google.com/drive/1IC9HVgEszmNdCB0eDl1p4WnBAdkX39 *sharing*

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. (1 point) Plot a histogram of the length of patients data.

4. (1 point) Plot a histogram of the first hour in which patients develop sepsis.

5. (3 point) 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 $\frac{2+5}{40+39}$. What 5 features have the most missingness?

1.2 Learning a model to predict sepsis (30 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. If there are features with more than 50% missingness, feel free to drop them.

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 (10 points): Instead of constructing summary features, you can feed the data hour by hour into a recurrent neural network, you can get 10 bonus points if you implement an LSTM model (or similar models) on this data and evaluate it (meaning you can get 10 points for 2.2, 10 points for 2.3 and 10 bonus boints additionally)

3. (10 points) Feed the data into a neural network with PyTorch in a similar fashion to the notebook in recitation #3. Obtain the AUC on the test set and report it here. Please show all your code.

   For half the points (maximum 5/10): use an XGBOOST classifier instead of a neural network on PyTorch

   Bonus (5 points): In addition to implementing the neural network, implement an XGBOOST classifier.

2 Length of Stay Prediction Using Notes (30 Points)

In the last assignment, you ran a logistic regression to perform length of stay prediction using lab data. In this homework, you will do length of stay prediction using **clinical notes**. Please turn in your colab notebook. Make sure you are running using a GPU. **Ensure that you follow good machine learning principles**. The starter notebook can be found here: https://colab.research.google.com/drive/129mfmlHIFeFZ-0bkgjKY6s-4b-058suN?usp=sharing

2.1 Training the model (10 Points)

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

1. Sample 20% of the dataset for training and testing (40% total).
2. For each patient, take the first and the last note from their record. Use the `HOURS` field to determine this. This field was calculated by taking the time of the note with their ICU admission time. Some of these values are negative, which means that it is a note from the ED. It is okay if you take either the note from the ED or from the ICU (i.e., take the smallest absolute value).
3. Run the notes through the `distilbert` tokenizer, truncating any notes that are longer in length than 512.
4. Treat these as a independent training instances.
5. Train a `distilbert` model on this task. 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 best validation accuracy. This will take roughly 20 minutes.

2.2 Test Set Eval Questions (15 Points)

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

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

3. (2 Point) Graph a histogram of the test predictions. 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. (2 Point) What is the **accuracy and AUC** of the model on the test set, ignoring the fact that every patient has two notes?

5. (1 Point) Now, take into account that every patient has two notes. What is the **accuracy** on the test set if you only use the first note?

6. (1 Point) Take into account that every patient has two notes. What is the **accuracy** if you only use the last note?

7. (4 Points) One issue with this method is that we are treating the first and last note as independent training samples. This is known as **multiple instance learning**. Try two simple methods to combine the predictions from the first and last note. Describe your methods below, and report the accuracy of both on the test set.

8. (4 Points) Name two methods that you could use to jointly model **all** of the patient's notes, taking into account that the model context length is only 512 tokens.

2.3 Adding Lab Data (4 Points)

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

3 Prompt Engineering (8 Points)

Congrats on your new position! You've been hired as a prompt engineer at OpenAI. Your boss, Maggie, has asked you to explore different prompts for MedNLI. She suggests that you first learn about the task. Then, she asks that you do the following

1. Experiment with 7 very different prompts.
2. Select the best one, and play with the wording 2 different times (i.e., change small words).
3. Select the best wording, and try adding 1, 2, and 3 "demonstrations".
4. You should have run on 12 total prompts. Select the best performing of the 12 and report the performance on the test set.

Maggie has put together an initial version of the code: https://colab.research.google.com/drive/1e6QUhbNzOpGUGivpCY
brsAZczfgrJdU?usp=sharing. **Ensure that you follow good machine learning principles**. Your grade will be determined by both methodology and performance (70+% accuracy).

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.