HST.953 Clinical Data Learning, Visualization, and Deployments
Massachusetts Institute of Technology, Fall 2024
Marzyeh Ghassemi and Leo Celi

Problem Set 1

Due Date: Week 4 Class

Submission: Students were asked to make two submissions:

- Please submit all code that you wrote and the output of your scripts (including diagrams) as a zip file (your filename.zip) under the "Problem Set 1 Code" assignment on the internal class website.
- Additionally, please submit a written report (your filename.pdf) under the "Problem Set 1" assignment on the internal class website. 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 documentation of a valid emergency from S3/gradsupport for MIT students, or an equivalent student support hub for non-MIT students.

Collaboration: You may collaborate with your fellow classmates (and in fact, you are encouraged to!). However, what you turn in must be your own version of the work. If you do collaborate with some of your classmates, please make sure to list your collaborators on the first page of your submitted report.

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/hst953-2022

Problem set prepared with help from Adrien Carrel

Part 0. Preliminaries [0 points]

Throughout this course, you will be working with healthcare datasets that may contain varying levels of sensitive patient information. To access these datasets (for homework 2 and your course projects), you will need to complete an online ethics course, "Data or Specimens Only Research" and to register an account on PhysioNet.

This is a time-consuming process, and it will take several days to approve your request, so it is important that you start working on the course early to have access to the datasets you need by the time you need them.

CITI Certification

In addition to registering an account on PhysioNet, the students taking this course were instructed to complete an online ethics course called "Data or Specimens Only Research," which they could access through MIT or through the Collaborative Institutional Training Initiative (CITI Program) website:

Research, Ethics, Compliance, and Safety Training

Registering a PhysioNet Account

Please follow the instructions under the "PhysioNet Credentialing" section at the following link https://mimic.mit.edu/docs/gettingstarted/ to get access to PhysioNet. Please use your MIT email address username as your PhysioNetWorks email (otherwise, please email Abbas your PhysioNetWorks email so we can try to expedite PhysioNet access). One of the steps will require you to fill out a Data Use Agreement (DUA) where you will be asked for:

1. Reference category. Choose Supervisor
2. A reference name. Write Leo Celi.
3. If asked for an email, use lceli@mit.edu. If asked for a job title, use "Principal Research Scientist"
4. The general research area for which the data will be used: Write HST.953/6.S982.
5. Please ensure your form is complete, or it will not be approved.

Note: When you are asked to submit your CITI course results, you must submit the full CITI completion report (which lists all modules completed, with dates and scores) rather than just the CITI certificate, which shows only course completion.

Part 1. Mortality Prediction in the ICU [17 points]

In this exercise, we will look at a typical application of machine learning to tabular clinical data: predicting patient mortality based on information collected during their first 24h in the ICU. On a surface level, this looks like a straightforward task. However as we shall see in this part and the next, there are several common mistakes that can easily creep into our modeling and bias our results if we're not careful.

Follow the instructions in the jupyter notebook to download the GOSSIS dataset, then fill out the relevant code blocks in the notebook to answer the following questions.

(a) **[3 pts]** One of the mistakes that occurs most frequently when modeling based on clinical data is using information that would not be available to the model at the time it needs to make its prediction. Machine learning models operating in a clinical setting typically have to run in real time, and at the time they are run, not every feature available in the dataset might be available for the model to use. In fact, there are several tests that are timestamped at the time samples are collected in the ICU but that take a week or so for their results to be returned for analysis.

Another form of this mistake, commonly referred to as *data leakage* is to use features whose value is obtained as a result of the clinician's prediction of the outcome of the patient. Using these features leaks information about the true label that the model is trying to predict, even though in practice the model would be used to *help the clinician come up with their prediction in the first place*, and therefore it wouldn't have access to these data-leaking features.

The dataset that you are using contains around 200 features: (a) some that are useful for mortality prediction, (b) some that are/should be irrelevant, and (c) some that might be predictive but that inadvertently leak data to the model. Go over the list of features and their descriptions with your homework group members and indicate in the jupyter notebook which features fall into each of the three categories mentioned above.

(b) **[0.5 pts]** Now that we have filtered out the features, the next step is to standardize them to make it easier for our models to fit them. Follow the instructions in the notebook to standardize and impute the features. We will be using a SimpleImputer from the scikit-learn library to fill in any empty values with the mean/mode of that feature. We will also be using a OneHotEncoder for categorical features and a StandardScalar for numerical features to make the features more amicable for our models to learn from.

(c) **[1.5 pts]** Train a logistic regression model to predict mortality based on the standardized features. Compute and report the following performance metrics in a table on each of the train and test sets:

- Accuracy
- Precision
- Recall
- F1-score

- AUC score

(d) **[1.5 pts]** Repeat part (c) using a bagging random forest classifier (from scikit-learn) and using a gradient-boosted random forest classifier (from the xgboost library).
(Note: for simplicity, use the sklearn-compatible API from the xgboost library when training the gradient-boosted random forest)

(e) **[1 pts]** Plot on two histograms (one for the training set and one for the test set) the performance metrics you just obtained. Plot the names of the metrics along the x-axis, plot the values of the metrics along the y-axis, and color-code the bars based on their corresponding model.

(f) **[1 pts]** Which model has the best performance? Briefly explain your answer.

(g) **[1.5 pts]** To get a high-level overview of which features contribute the most to our models' predictions, use the shap library to compute and plot the Shapley values of the xgboost model's features on a beeswarm plot. Which features contribute the most to the model's predictions? Do they seem like reasonable features that the model can rely on, or is the model basing its predictions on spurious correlations?

(Note: it might be useful to refer to https://github.com/slundberg/shap for a quick overview of how to interpret shap plots.)

(h) **[1.5 pts]** At this point, we have a good idea of how well our models perform on the general population. However, just because a model performs well on the general population doesn't imply that the model will perform equally well on different cohorts within that population.

Write code in the notebook to split the test set into cohorts in three different ways:

- white and non-white patients
- male and female patients
- patients younger than 65 and patients 65 years old or older

(i) **[2.25 pts]** Compute the same performance metrics (accuracy, precision, recall, F1 score and AUC score) for the xgboost model on each test cohort separately. Similarly to part (e), plot these metrics on a single histogram for each split (metric names along the x-axis, metric values along the y-axis, and color-coded by cohort split).

(j) **[0.75 pts]** How well does the model perform on the two cohorts in each split?

(k) **[1.5 pts]** For each of the above three splits, select two data points from each cohort and visualize the Shapley values of each data point on a waterfall plot using the shap library. Include the plots in your report.

(Note: Similarly to part (g), it might be useful to refer to https://github.com/slundberg/shap for a quick overview of how to interpret shap plots.)

(l) [**1 pts**] Do you notice any discrepancies in the features used by the model to make predictions for the two cohorts in each split? If yes, briefly describe those discrepancies. If not, briefly explain why you think such discrepancies were not observed.

Part 2. Delving into Disparities [8 points]

One other issue common in machine learning on clinical data is that of data imbalance. Collected clinical data is typically biased towards the population of the hospital(s) the data is collected from, and depending on several factors (e.g. if the hospital population isn't representative of the general population), the trained model's performance might differ across sub-populations.

(a) [**2 pts**] Plot, each on a different pie chart, the distribution of the patients' genders, ages (bucketed into 5 year intervals), and races for each of the train and test splits (6 pie charts in total). Do you notice any imbalances in the data?

(b) [**2 pts**] We would like to check if varying the degree of imbalance in the GOSSIS dataset has an effect on the performance of a model trained on the imbalanced dataset. Follow the instructions in the notebook to reduce the number of training datapoints corresponding to female patients by 20%, 40%, 60%, and 80% respectively. For each reduction in the number of datapoints, train an xgboost model on the modified dataset and compute the same performance metrics introduced in part 1 (accuracy, precision, recall, F1 score, and AUC score) on both the modified training set and the **original** test set. Additionally, compute the same performance metrics on each of the male and female cohorts. Report your results in a table.

(c) [**1 pts**] How does reducing the number of female patient datapoints affect the performance of the model? Why do you think that is?

(d) [**1 pts**] Do you expect to see the same or different results when varying the degree of missingness of some populations in other datasets or cohort splits? Briefly explain why you think so.

(e) [**1 pts**] Another form of data imbalance is more implicit, and happens due to some patients undergoing fewer tests even if the number of patients is the same in each cohort. How do you expect increasing the degree of missingness of test results in some training datapoints would affect the performance of your model? Briefly justify your answer.

(f) [**1 pts**] Suggest one potential way of handling missingness of test results.

HST.953 Clinical Data Learning, Visualization, and Deployments
Fall 2024