CSC2541 Problem Set 1 - Machine Learning using Clinical Data

Due: Wednesday Feb 6 at 11:59pm (Markus)

Instructor: Marzyeh Ghassemi (marzyeh@cs.toronto.edu)
TA: Bai Li (bai@cs.toronto.edu)

Introduction

In this assignment, you will explore some medical datasets and apply several machine learning models on them. You must clearly list all tools that you use to perform the analyses. If you use any tools / algorithms / ideas not developed by yourself, you must reference it properly.

Please submit all code that you wrote and output of your scripts (including diagrams) as a zip file **(code.zip**) and a written report (**report.pdf**) to Markus by **Wednesday, Feb 6 at 11:59pm**. If you use Jupyter notebooks, submit both the ipynb file and generate an html file using nbconvert. Also, please submit a physical copy of your written report at the beginning of class on **Thursday, Feb 7 at 10am**.

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 in the Piazza board for this course, or contact the instructor or TA. Late assignments will not be accepted without a valid medical certificate or other documentation of an emergency.

Preliminaries

In this assignment, you will be working with the MIMIC dataset (https://mimic.physionet.org/) which consists of about 40,000 patients admitted to ICUs in the Beth Israel Deaconess Medical Center in Boston between 2001 and 2012.

1. Before you can download the dataset, you need to complete an online ethics course, “Data or Specimens Only Research” and register an account on PhysioNet. It will take several days to approve your request. Detailed instructions are given here: https://mimic.physionet.org/gettingstarted/access/

2. Next, download the dataset and import it into a PostgreSQL database. This takes several hours; make sure you have at least 100GB of free disk space. The data should be access restricted. I.e. a personal computer with a password, or in a folder on a server that only you can access. Detailed instructions: https://mimic.physionet.org/tutorials/install-mimic-locally-ubuntu/

Part 1: Getting started with MIMIC (4 marks)

To get warmed up with MIMIC, explore the hospital experience of patient ID 40080. For each question, give the SQL query you used.

1. Give a brief summary of the patient's demographics (race, age, marital status, etc).
2. What was the patient's primary diagnosis (seq_num = 1) and the ICD-9 code?
3. How long did the patient stay in the ICU? According to the discharge report, what was her condition when she was discharged?
4. What was the patient's highest and lowest heart rates during the stay?

Part 2: Predicting mortality in ICUs (12 marks)

In this part, you will train models to predict mortality in ICUs using data from the first 48 hours of the ICU stay and clinical notes. First, run the script to preprocess data from MIMIC (https://cs2541-ml4h2019.github.io/problem_set/mort_icu_script.py) which generates the following files:

- `adult_icu.gz`: tabular data from adult ICUs
- `adult_notes.gz`: clinical notes from adult ICUs

The binary target variable to predict is `mort_icu`, which is 1 if the patient died in the ICU. Use the last column `train=1` for training data, and `train=0` for testing. The rest of the data contains variables such as the heart rate, blood pressure, respiratory rate, and other variables, measured in the first 48 hours.

Part 2a: Train a logistic regression model (using scikit-learn) to predict in-ICU mortality from all of the variables. Use L2 regularization.

Plot the ROC graph and compute the AUC score. Comment on the model performance.

Look at the top 5 risk factors of mortality and the lowest 5 and explain what they mean.

(Hint: in order to compare the coefficients of the features included here regardless of their scale, it is useful to standardize the non-binary variables).

Part 2b: Use the clinical notes text data to train a bag-of-words model to predict in-ICU mortality.

Use NLTK to tokenize and remove stopwords and punctuation.

Use TF-IDF to transform bag-of-word counts into numerical features, then train a L1 regularized logistic regression to predict mortality.

Report the top 5 words associated with a high risk of mortality and the lowest 5.

Plot the ROC graph and compute the AUC score.

Part 2c: Combine the models in parts 2a and 2b to predict mortality using both clinical notes and tabular data.

Plot the ROC graph and compute the AUC score.

Comment on how the two sources of data affect the model.

Note about preprocessing: The datasets that you worked with were preprocessed such that missing values were imputed. Specifically, we stratified the data by gender and age groups and replaced missing values with their group means. The topic of missing data is important and will be discussed further in class.

Part 3: Predicting hypertension using LSTM (10 marks)

In this part, you will train models to predict hypertension using sequential chart data. First, run the script to preprocess data from MIMIC (https://cs2541-ml4h2019.github.io/problem_set/make_hypertension_data.py) which generates the following files:

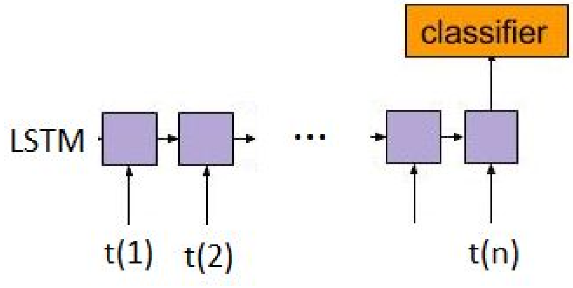
- `hypertension_patients.gz`: contains for each subject_id and hadm_id, a column `hypertension` which is 1 if the patient has hypertension. As with before, use the last column `train=1` for training data, and `train=0` for testing.
- `hypertension_charts.gz`: contains measurements of various instruments. Each row is one measurement. The itemid determines which type of measurement it is, and can be one of the following types:
    - Heart rate in bpm (itemid = 220045)
    - Respiratory rate in breaths / minute (itemid = 220210)
    - O2 saturation in % (itemid = 220277)
    - Blood pressure in mmHg (itemid = 220181)

Part 3a: As a baseline, train a logistic regression to predict hypertension, using the min, max, and mean heart rate as features. Remove patients with fewer than two heart rate measurements.

Report the AUC and F1 scores on the test set.

Next, do the same for respiratory rate, O2 saturation, and blood pressure.

Part 3b: Train an LSTM (using Keras) to predict hypertension, using the heart rate input sequence as input, and a dense layer at the last step to produce a binary classification (see diagram). Remove patients with fewer than two heart rate measurements.



Report the AUC and F1 scores on the test set.

Next, do the same for respiratory rate, O2 saturation, and blood pressure.

Part 3c: Comment on the performance of your models. Which measurements are the most useful for predicting hypertension? Do the results make sense?

Part 4: Topic Modelling on Clinical Notes (6 marks)

In this part, you will run Latent Dirichlet Allocation (LDA) on clinical notes. LDA is an unsupervised algorithm to extract hidden topics (sets of related keywords) from large volumes of text.

For our case, we use `adult_notes.gz` generated in Part 2, ignoring all columns other than chartext which contains the notes. Each row is the combined notes for one ICU stay, which we will treat as a document.

Part 4a: Use gensim to run LDA on the clinical notes, using 20, 50, and 100 topics.

Use NLTK to tokenize and remove stopwords and punctuation.

For each of the three runs, calculate the coherence score. How many topics gives the optimal coherence score?

Part 4b: Use the best model according to part 4a. For each of the keywords: *respiratory, vomiting, urine, pulse*, examine the topic(s) that contain the keyword.

What other words occur in these topics? Do they make sense? If possible, give a name to each topic.