Predicting Mortality Using Diagnoses

Gaurab Banerjee

Background

Electronic medical records (EMRs) are the main form of storing patient information in clinical settings. Previous work has used EMRs to establish links between hospital stay duration and mortality[1]. In this project, I predict the mortality of a patient using both coded diagnoses and free text diagnosis. I use logistic regression, a neural network, and Google's BERT embedder.

Data Collection

The data is from the MIMIC-III[2] (Medical Information Mart for Intensive Care III) dataset: aggregated EMRs from Beth Israel Deaconess Medical Center's ICU. I merged on SUBJECT_ID as the common pivot column.

I one-hot encoded the categorical variable, ICD9_CODE which contains coded patient diagnosis. The DIAGNOSIS column is a free text column which is capped at 192 characters. The HOSPITAL_EXPIRE_FLAG column is a binary column.

Features

SUBJECT_ID	ETHNICITY	ICD9_CODE	LABEL	212	HEF
1	Cuban	280	heart rate	80	0

Fig. 1 Sample example of inputs and the output, ICD9_CODE.

Feature	Description
ROW_ID	Unique Data Linker
SUBJECT_ID	Admission/Patient ID
HADM_ID	Admission ID
ADMITTIME	Admission Time
DISCHTIME	Discharge Time
DEATHTIME	Time of Death
ADMISSION_TYPE	Elective, Urgent, Newborn, Emergency
ADMISSION_LOCATION	Pre-admit location: Categorical Var.
DISCHARGE_LOCATION	Categorical Var.
INSURANCE	Categorical Var.
LANGUAGE	Categorical Var.
RELIGION	Categorical Var.
MARITAL_STATUS	Categorical Var.
ETHNICITY	Categorical Var.
EDREGTIME	ED Entry
EDOUTTIME	ED Out
DIAGNOSIS	Free text notes
HOSPITAL_EXPIRE_FLAG	Did patient survive to discharge?
HAS_CHARTEVENTS_DATA	Chart populated?
SEQ_NUM	Priority order of ICD9
ICD9_CODE	Patient diagnosis

Models

This was modeled as a reflex-based problem. The input was either the coded diagnoses or the free-text diagnoses the medical providers had written. The output was the HOSPITAL_EXPIRE_FLAG value. Specifically, I used three techniques:

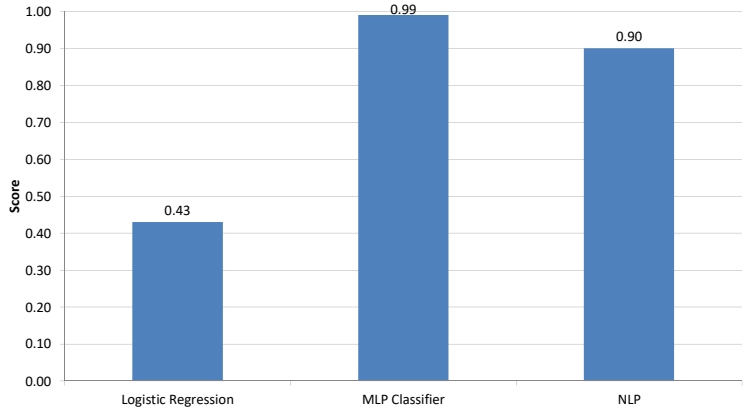
Logistic Regression
This is a classification model which uses the function identified below. Y=1 when the probability is greater than 0.5.

$$log\frac{p(x)}{1-p(x)} = \beta_0 + x \cdot \beta$$

Multilayer Perceptron Classifier
This is composed of hidden and input layers feeding into a binary classification layer. I used hidden layer sizes (5,2)

NLP Approach
I used BERT to generate my word embedding and harnessed the power of transfer learning. Then there was a output layer predicting mortality.

Results and Discussion



Key Takeaways

- Logistic Regression is not optimal- all patients were predicted to survive so there was no useful classification
- Due to the high scores of the NN, there is likely overfitting occurring
- Why Overfitting? One theory is that while there are nearly 1.1 million patients in this data, the feature vectors for many of the ICD9_CODES are extremely sparse while others are overrepresented
- The NLP approach performs more poorly than expected: the medical text may be too niche or being embedded properly

Future Research

- Use NLP on the free text NOTES column and see if extraneous notes change outcome prediction
- Use NLP to auto-bucket ICD9_CODES

References

1. Pirracchio, R., Petersen, M. L., Carone, M., Rigon, M. R., Chevret, S., \& van der Laan, M. J. (2015). Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): a population-based study. The Lancet Respiratory Medicine, 3(1), 42-52.
2. MIMIC-III, a freely accessible critical care database. Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available from: www.nature.com/articles/sdata201635
3. Previous work done by me