Prediction of Death: Using Natural Language Processing to Predict Mortality in ICU Patients

Capstone 2 Milestone Report

Background

As the world copes with the Covid-19 pandemic, mortality rates have come to the forefront. In-hospital mortality has long been used as a measure of hospital quality. The intensive care unit (ICU) has the highest mortality rate of all the units in a hospital. Of the approximately 4 million ICU admissions per year in the United States, the average mortality rate ranges from 8-19%, or about 500,000 deaths annually[1].

High death rates lead to unnecessarily high financial costs and poor patient experiences at the end of life. The average cost of an end-of-life ICU stay is $39,300 (median $24,000)[2]. This amounts to $19.65 Billion per year in the US. Costs of prompt palliative care can reduce hospital costs by $3237 per patient according to a study from the Icahn School of Medicine at Mount Sinai and Trinity College Dublin[2]. If hospice care is chosen as the end of life care option (in place of in-hospital palliative care), the average cost per patient is even less, at $11,820[3]. If patients nearing the end of life could be identified soon after hospital admission, they could be given prompt palliative care or discharged to their home or other facility for hospice, thereby having a significantly better patient experience by removing unnecessary, burdensome, and unwanted procedures, and also reduce financial burden significantly.

One possible way to help patient care providers and hospitals predict patients who are at the end of life is to analyze the data present in caregiver notes. Natural language processing (NLP) is a field of artificial intelligence that allows the computer to understand the human language. It can process words and phrases from text, like *sepsis* and *status worsening,* into a form that can be included in predictive models. The objective of this project is to use NLP to predict in-hospital mortality.

Client

The client is a healthcare system looking to reduce in-hospital mortality rates in order to reduce aggressive care (or to discharge to outside care) for patients who are at the end of life and thereby improve patient experiences, outcomes and quality of care scores.

Data

This project was completed using the MIMIC-III database from MIT. MIMIC-III is a large, freely-available database comprising de-identified health-related data associated with over 50,000 patients who stayed in ICUs of the Beth Israel Deaconess Medical Center between 2001 and 2012. The database includes information such as caregiver notes, demographics, vital sign measurements made at the bedside, laboratory test results, procedures, medications, imaging reports, and mortality. The database is accessible via: https://mimic.physionet.org/.

Data was obtained from two datasets in MIMIC-III: ADMISSIONS and NOTEEVENTS. ADMISSIONS contains the target variable, HOSPITAL_EXPIRE_FLAG (i.e., whether or not the patient died during hospitalization) and unique identifiers HADM_ID (hospital admission ID) and SUBJECT_ID. NOTEEVENTS contains TEXT (caregiver notes) and the unique identifiers HADM_ID and SUBJECT_ID. The following is a full list of database variables included in the analysis:

ADMISSIONS:

- HOSPITAL_EXPIRE_FLAG
- SUBJECT_ID
- HADM_ID
- ADMITTIME
- DEATHTIME
- ADMISSION_TYPE

NOTEEVENTS:

- SUBJECT_ID
- HADM_ID
- TEXT
- CHARTDATE
- CATEGORY

Target Variable

The HOSPITAL_EXPIRE_FLAG column was used as the target variable (1=death, 0=no death).

Data Preparation/Wrangling

Data preparation began with data cleaning and exploration. Python pandas, numpy, matplotlib, and seaborn were used for data wrangling and exploration. Natural Language Toolkit (NLTK) was used for the Bag-of-Words model of converting text into numbers that can be processed for prediction

Reading in the Data

Each dataset was provided in a GZIP compressed CSV format. The datasets were imported directly into individual pandas dataframes.

Data Exploration

Initial data exploration of the ADMISSIONS and NOTES tables found 58976 rows/19 columns and 2083180 rows/ 11 columns, respectively.

Addressing Null Values

The column DEATHTIME contained many null values, which were retained since this reflected the patients who did not expire during hospitalization. No other null values were noted.

Merging of Datasets

The two dataframes were merged and irrelevant columns were dropped, after which they contained 1,851,959 rows and 9 columns.

Feature Engineering

Initially, all of the variables were of the object and int data types. Date and time variables were converted to datetime format.

Once the datasets were merged, the feature ENDTIME was created. This feature indicates the time to complete note collection (24 hours after admission, or upon patient death - whichever happened first). The dataset was then updated to include only those notes that were taken between ADMITTIME and ENDTIME (also taking care to remove those patients who expired before admission), resulting in 444528 rows and 10 columns.

Splitting the Data into Training, Validation and Test Sets

Once the data underwent initial processing, in order to apply the NLP model, and for eventual use in machine learning, the data was split into training, validation, and test sets.

Text Preprocessing

To allow the machine learning model to accept the data, caregiver text from the training set was preprocessed using the NLP Bag-of-Words (BoW) model. Bag-of-Words is a simple way to represent text with numbers. Basically, it breaks up a text (here, each caregiver note) into individual words (or 'tokens') and then counts how often each word occurs. Each note is represented as a bag of words vector (i.e., a string of numbers).

BoW was implemented as follows:

Step 1: To free up valuable processing time, numbers, punctuation, and irrelevant characters, such as \n, were removed from the text and a pre-made list of stop words from the NLTK library was used to remove irrelevant words (such as "a", "the" "and"). The most frequent words were noted, and a few stop words were added to the list (Figures 1 and 2).

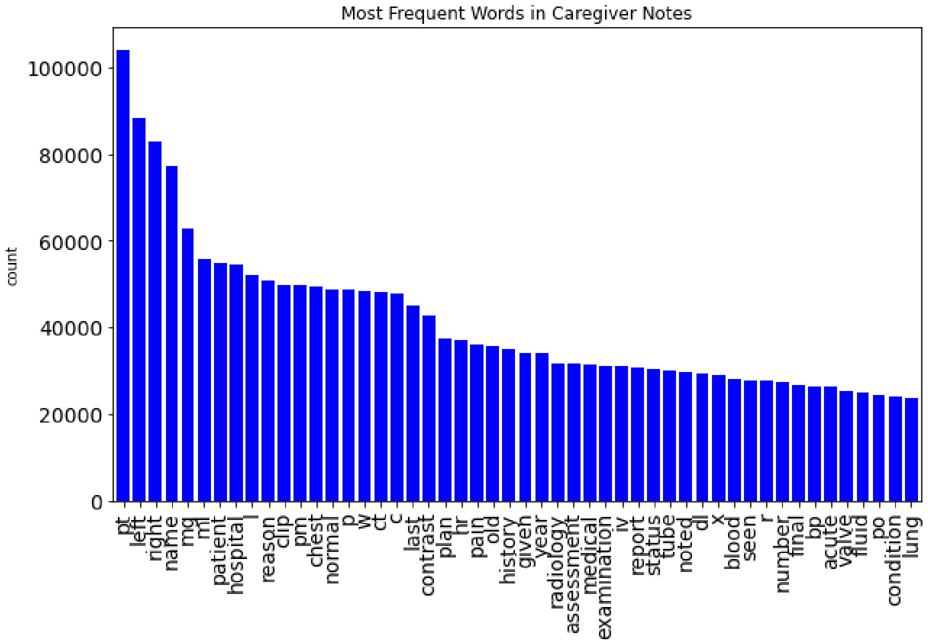


Figure 1. Most common words in caregiver notes.

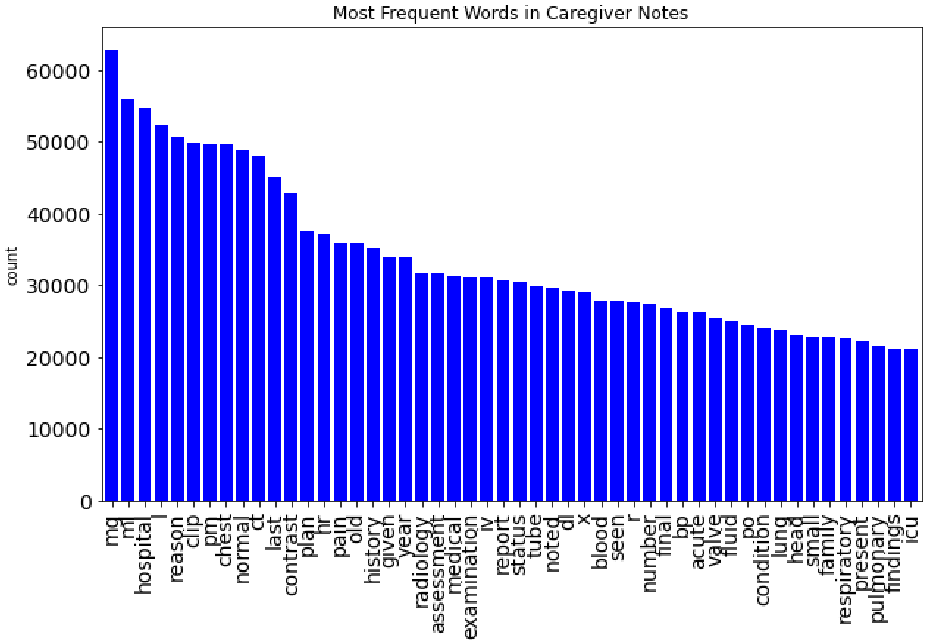


Figure 2. Most common words in caregiver notes (after additional clinical stop words filtered)

Step 2: A vocabulary of 'tokens' was built from all the unique words that were left in the caregiver notes.

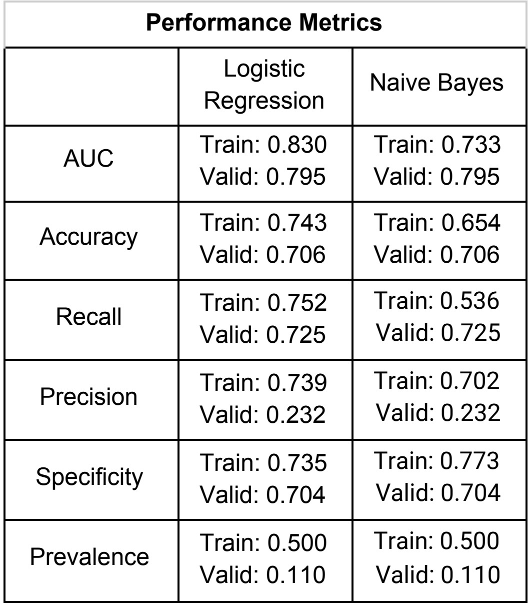
Step 3: Finally, each of the tokens was fed into a vectorizer, where it was converted into a feature (variable) that represented the different columns of the dataset. One column (vector), was created for each unique token in the dataset ('corpus'). For a given text sample, each time a word occurred, the value of 1 was added to its column, if not, the value added would be 0.

Now the text was ready for use in a machine learning model.

Prediction Using Machine Learning

Model Selection

Since the target variable, in-hospital mortality, is categorical and binary, a supervised learning model that would work well with a large, sparse matrix dataset was needed. Simple logistic regression and Naive Bayes models were applied to the data and performance metrics were compared between the two models on both the training and validation data (Figures 3 and 4).



Performance Metrics

	Logistic Regression	Naive Bayes
AUC	Train: 0.830 Valid: 0.795	Train: 0.733 Valid: 0.795
Accuracy	Train: 0.743 Valid: 0.706	Train: 0.654 Valid: 0.706
Recall	Train: 0.752 Valid: 0.725	Train: 0.536 Valid: 0.725
Precision	Train: 0.739 Valid: 0.232	Train: 0.702 Valid: 0.232
Specificity	Train: 0.735 Valid: 0.704	Train: 0.773 Valid: 0.704
Prevalence	Train: 0.500 Valid: 0.110	Train: 0.500 Valid: 0.110

Figure 3. Performance metrics - Logistic Regression and Naive Bayes.

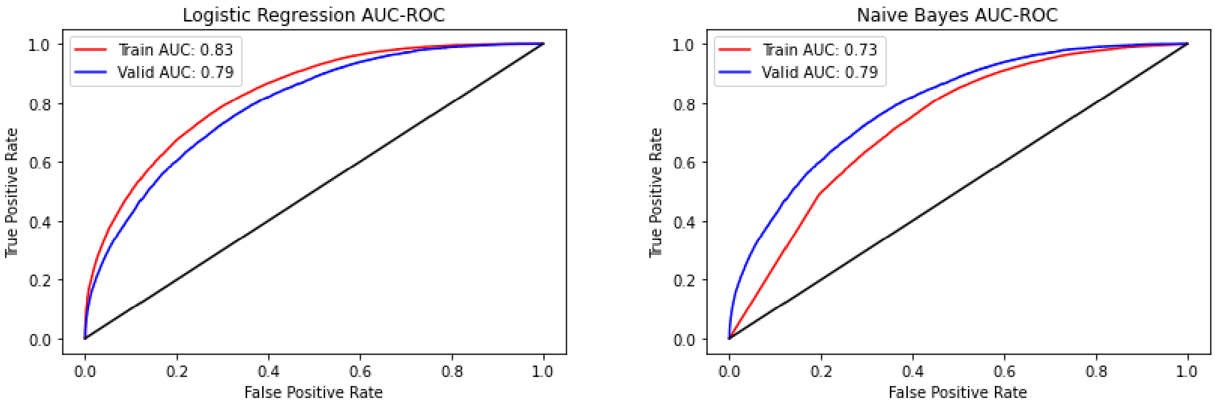


Figure 4. AUC-ROC for Logistic Regression and Naive Bayes.

Logistic regression had the greatest performance, with an AUC-ROC of 0.83 for the training set and 0.795 for the validation set, so it became the model of choice for further optimization and use on the test dataset.

Next to do:
Feature Engineering

Hyperparameter Optimization

Interpretation of Model Results and Implications

Conclusion

References

1) https://healthpolicy.ucsf.edu/icu-outcomes#:~:text=The%20modern%20intensive%20care%20unit,or%20about%20500%2C000%20deaths%20annually.
2) https://www.acc.org/latest-in-cardiology/journal-scans/2020/07/21/13/29/outcomes-from-intensive-care-in-patients#:~:text=This%20systematic%20review%20and%20meta%2Danalysis%20of%20ICU%20outcome%20in,ICU%20mortality%20fell%20over%20time.
3) https://tincture.io/the-hidden-costs-of-dying-in-america-2da0b81bbcd1
4) https://revcycleintelligence.com/news/palliative-care-reduces-hospital-costs-by-over-3k-per-patient