Prevention of Death:
Predicting Hospital Mortality Using Natural Language Processing

Cara Shrivastava, October 2020

MORTALITY IN THE ICU

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]
- The ICU mortality rate of Covid-19 patients as of July 2020 was 41.6% across international studies.[2]

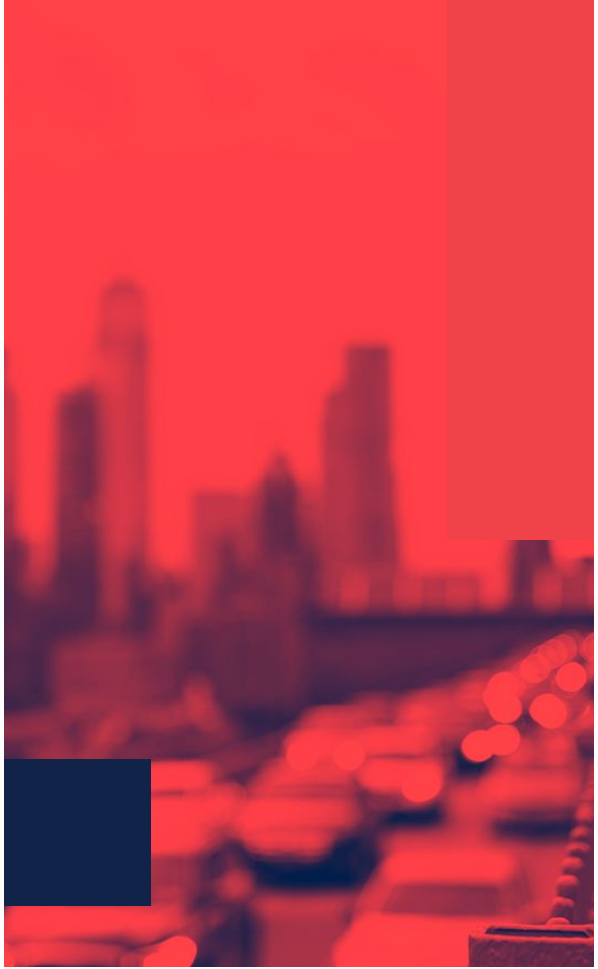
IMPACT OF HIGH MORTALITY RATES

- 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.[3]

- This costs the US **$19.65 Billion** per year.

THE CLIENT

The results from this project would be of particular interest to a healthcare system looking to reduce its in-hospital mortality rate.



THE SOLUTION

Reduce in-hospital mortality rates as much as possible.

THE SOLUTION

Using data they already have, hospitals can screen patients early in admission, selecting those at ‘high risk’ of being near the end of life.

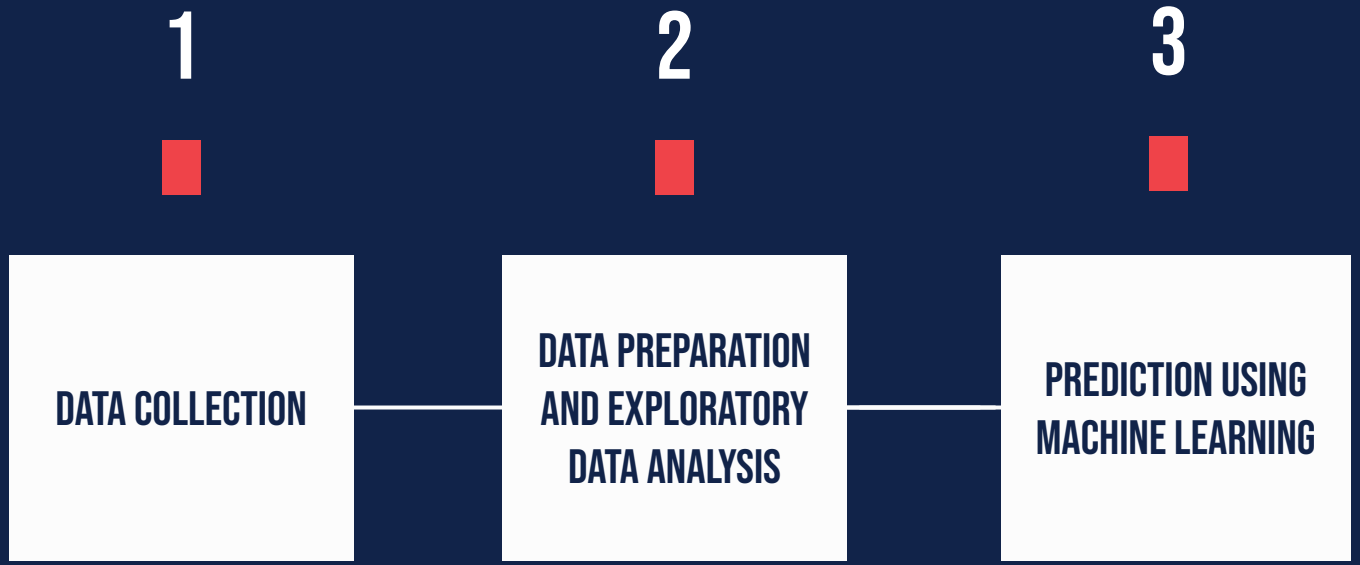
- Patients tagged as “high risk” will be considered for palliative care or discharged to their home or another facility for hospice care.
    - Hospice care results in an average per-patient savings of $11,820.[4]
    - In-hospital, palliative care saves, on average, $3237 per patient.[5]

THE SOLUTION

Caregiver Notes and Natural Language Processing (NLP)

- A simple 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.
- NLP helps the computer 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 PROCESS


1
2
3
DATA COLLECTION
DATA PREPARATION AND EXPLORATORY DATA ANALYSIS
PREDICTION USING MACHINE LEARNING

01.

DATA COLLECTION

DATA COLLECTION

Data was obtained from two datasets in MIMIC-III: ADMISSIONS AND NOTEEVENTS.

Columns used included:

- HOSPITAL_EXPIRE_FLAG (the target variable - indicates whether or not the patient died during hospitalization)
- TEXT (caregiver notes)
- HADM_ID (hospital admission ID - a unique identifier)
- SUBJECT_ID (another unique identifier)

02.

DATA PREPARATION AND EXPLORATORY DATA ANALYSIS (EDA)

DATA PREPARATION

Data was obtained from two datasets in MIMIC-III: ADMISSIONS AND NOTEEVENTS.

Columns used included:

- HOSPITAL_EXPIRE_FLAG (the target variable - indicates whether or not the patient died during hospitalization)
- TEXT (caregiver notes)
- HADM_ID (hospital admission ID - a unique identifier)
- SUBJECT_ID (another unique identifier)

READING IN DATA

Imported GZIP compressed CSV files into pandas dataframes.

INITIAL EXPLORATION

ADMISSIONS had 58976 rows & 19 columns; NOTES contained 2083180 rows & 11 columns.

NULLS AND OUTLIERS

DEATHTIME contained many null values, which were retained (patients who did not expire in the hospital).

FEATURE ENGINEERING

Date and time variables were converted to datetime format. A new feature, ENDTIME was created.

MERGING DATASETS

The two dataframes were merged and irrelevant columns dropped, resulting in 851,959 rows and 9 columns.

SPLITTING OF DATA

Data was split into training, validation, and test sets to prepare it for text preprocessing and for eventual use in predictive models.

DATA PREPARATION

Text Preprocessing with Bag of Words (BoW)

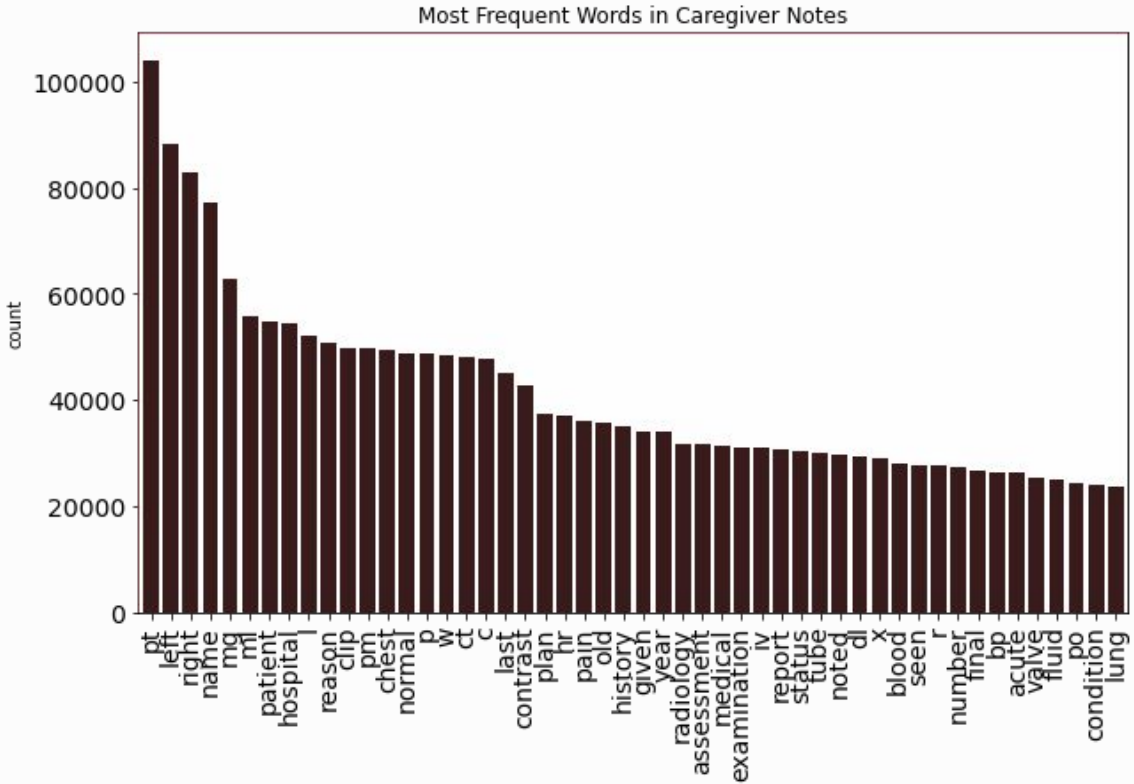
- Caregiver text from the training set was preprocessed using the NLP Bag-of-Words (BoW) model.
- BoW is a simple way to represent text with numbers.
- In the end, each note is represented as a bag of words vector (i.e., a string of numbers).

DATA PREPARATION AND EDA

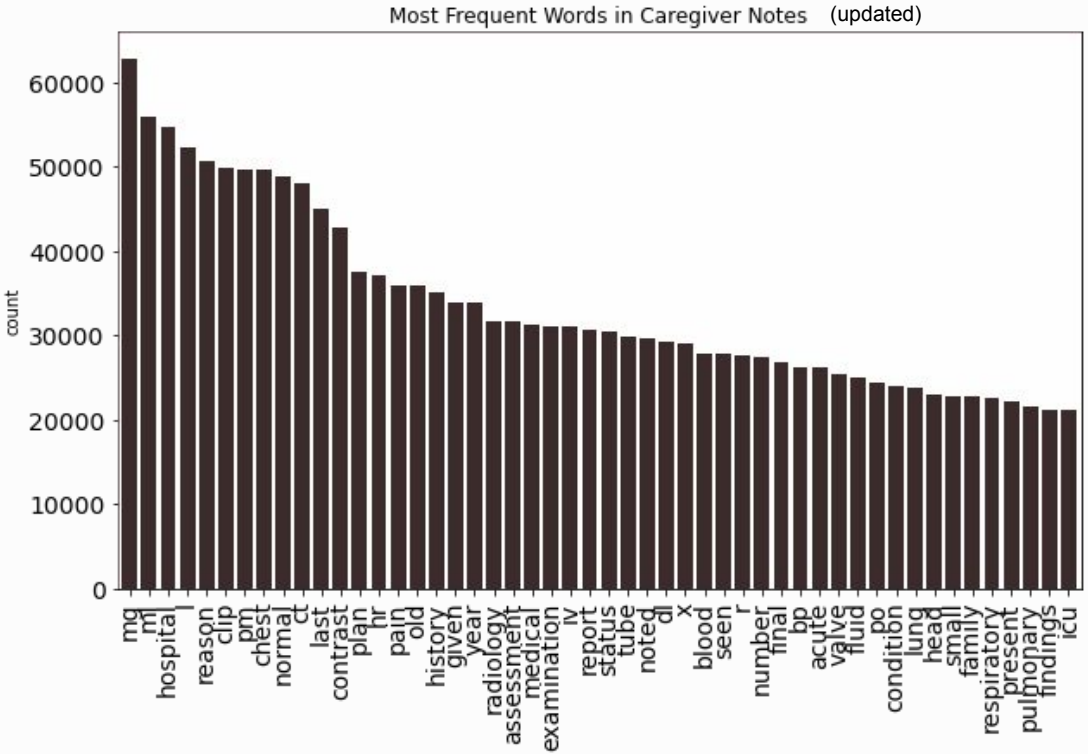
Bag of Words - Steps

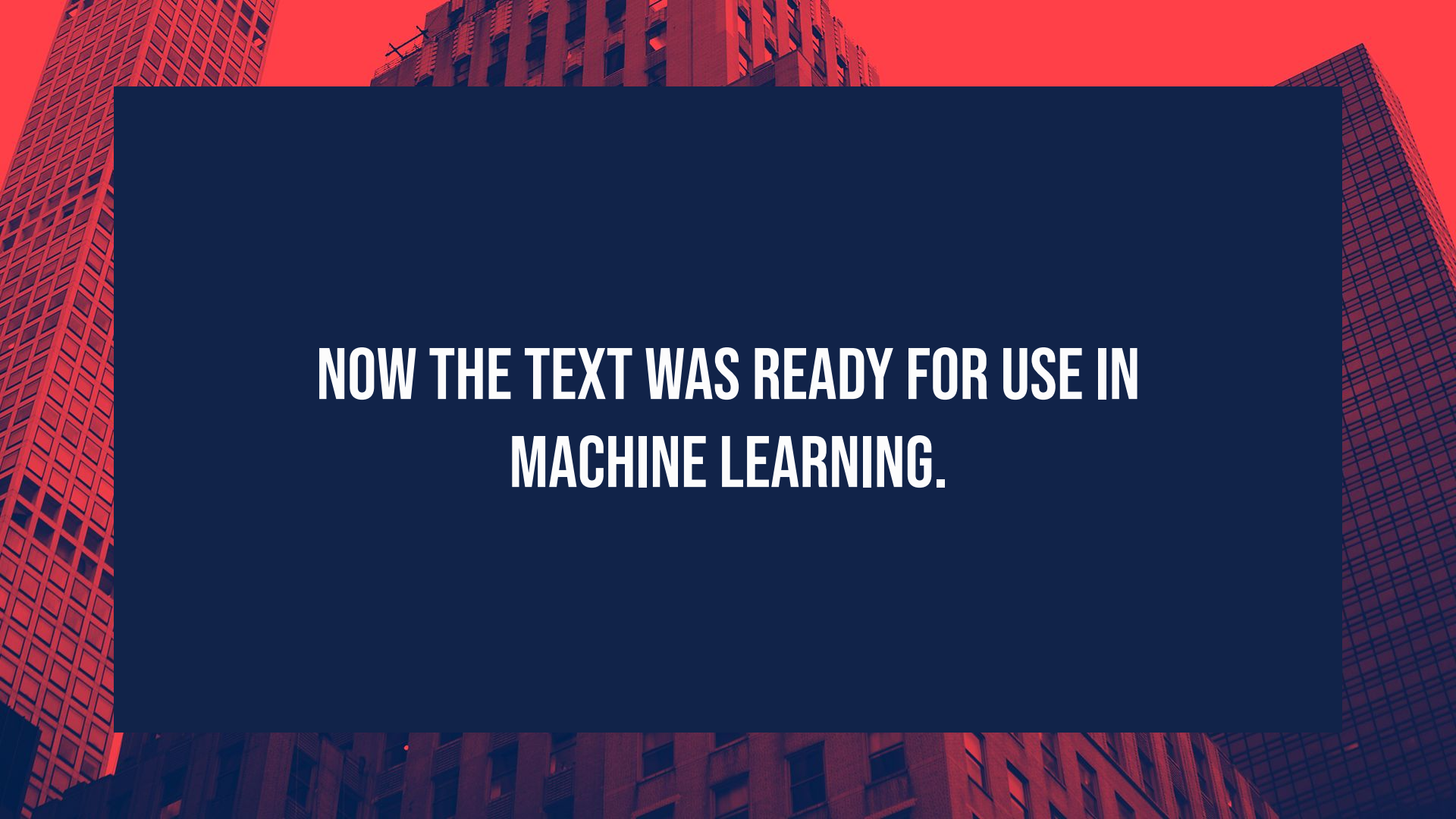
1. Numbers, punctuation, and irrelevant characters were removed from text.
2. A pre-made list of stop words was used to remove irrelevant words (such as “a”, “the” “and”).
3. Most frequent words were noted. Additional stop words were added to the list (see next 2 figures).
4. A vocabulary of ‘tokens’ was built from unique words left in the caregiver notes.
5. 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.

DATA PREPARATION AND EDA



DATA PREPARATION AND EDA




NOW THE TEXT WAS READY FOR USE IN
MACHINE LEARNING.

03.

PREDICTION USING MACHINE LEARNING

MACHINE LEARNING

Model Selection

This is a classification problem, since the target variable is categorical and binary.

Ideal model should be:

- A supervised learning model
- One that would work well with a large, sparse matrix dataset.

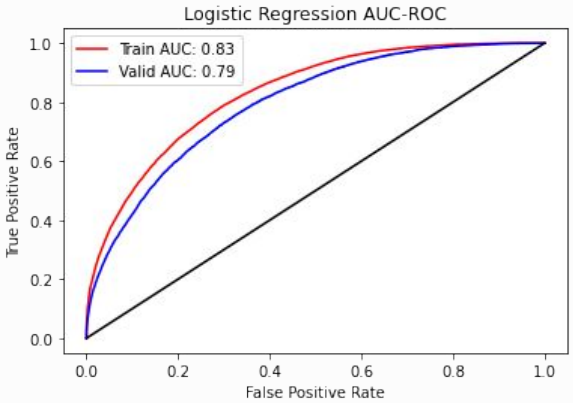
SIMPLE LOGISTIC REGRESSION AND NAIVE BAYES WERE SELECTED.

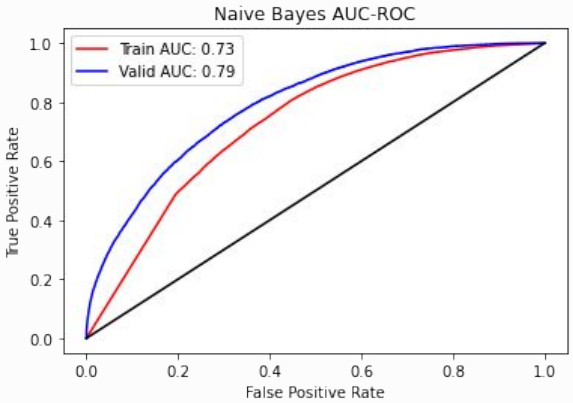
PERFORMANCE METRICS WERE COMPARED BETWEEN THE TWO MODELS.

LOGISTIC REGRESSION VS NAIVE BAYES

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

LOGISTIC REGRESSION VS 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.

IMPROVING THE MODEL

Feature engineering - (e.g., feature importance)

IMPROVING THE MODEL

Hyperparameter optimization

MODEL EVALUATION

Final model results and interpretation

RESULTS INTERPRETATION - A CASE STUDY

Given an average hospital, what kind of savings can be expected if this type of model were implemented?

CONCLUSION

Summary of what was learned

THANKS, ETC.

Contact info

Link to code

Credits