Prevention of Death:
Predicting ICU Mortality Using Natural Language Processing

Cara Shrivastava, November 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]

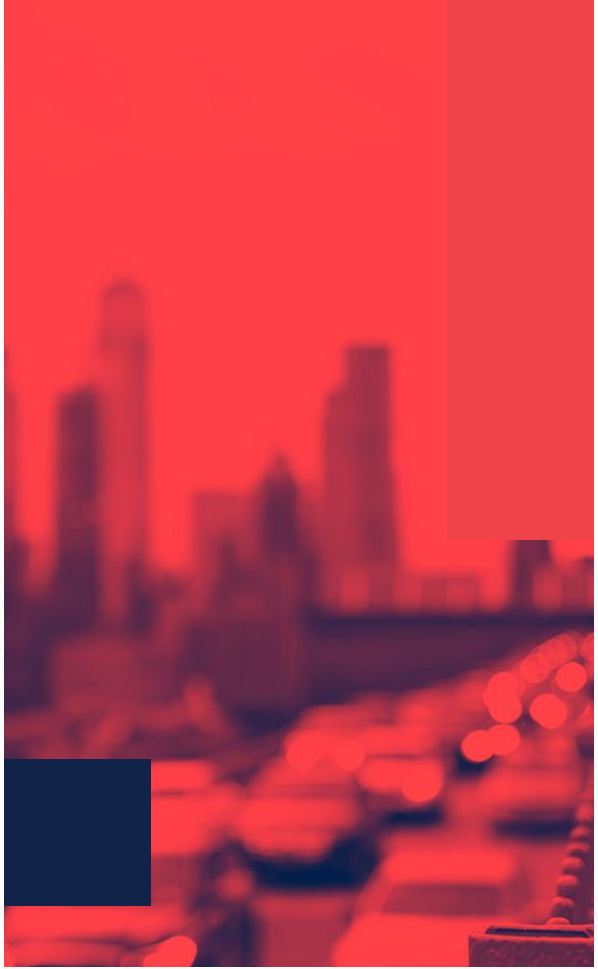
IMPACT OF HIGH MORTALITY RATES

- 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 ICU mortality rate.



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.

THE SOLUTION

Patients tagged as “high risk” will be considered for palliative care or discharged to their home or another facility for hospice care.

- Alternatively, in-hospital, palliative care saves, on average, $3237 per patient.*[4]
- Hospice care results in an average per-patient savings of $27,480.*[5]

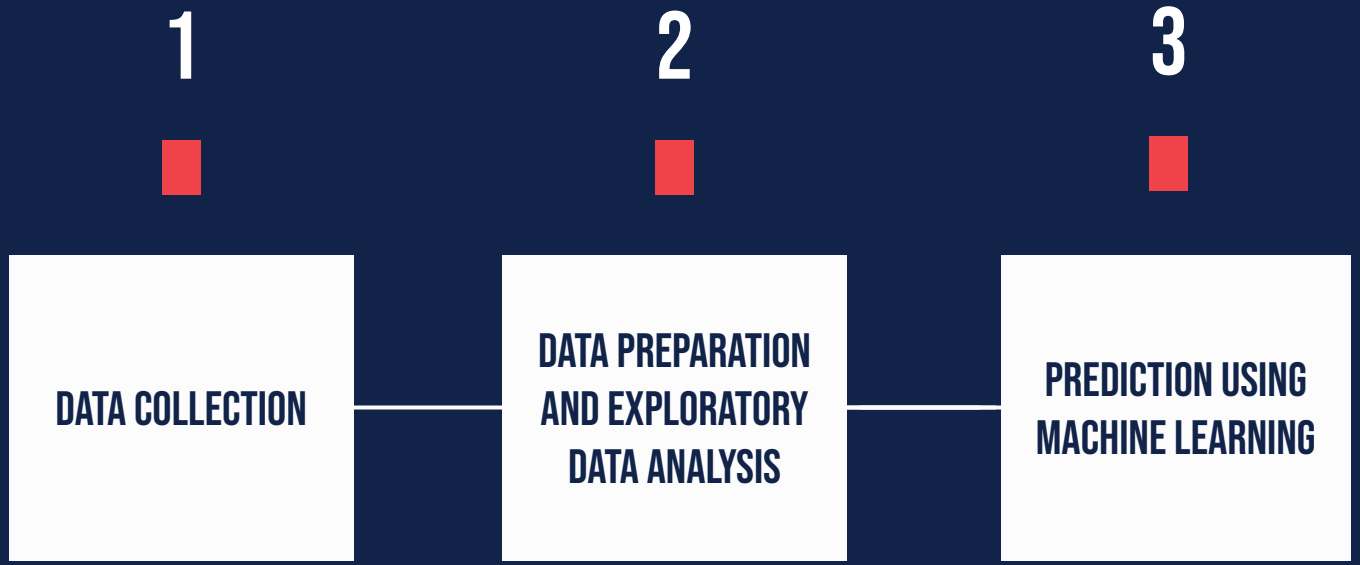
*Compared with ICU stay

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 - 0=negative death, 1=positive death)
- 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

Balancing the Training Data

- The dataset was imbalanced, since only about 11% of the patients were in the positive mortality group.
- After splitting the data into training, validation, and test sets, the negative group was subsampled, so that there were an even number of patients in positive and negative groups.

DATA PREPARATION

Text Preprocessing with Bag of Words (BoW)

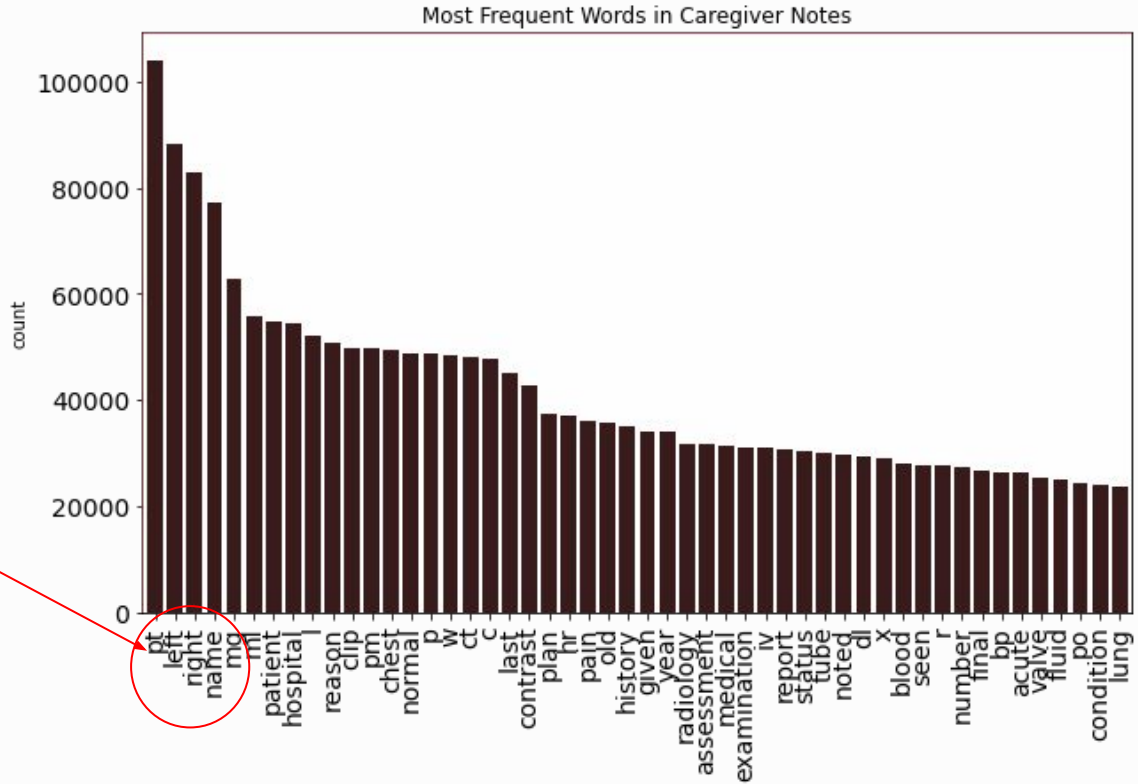
- Caregiver text from the training set was preprocessed using the NLP Bag-of-Words (BoW) model.
- BoW is a simple way to represent text with numbers.
- In the end, each note is represented as a bag of words vector (i.e., a string of numbers).

DATA PREPARATION

Bag of Words - Steps

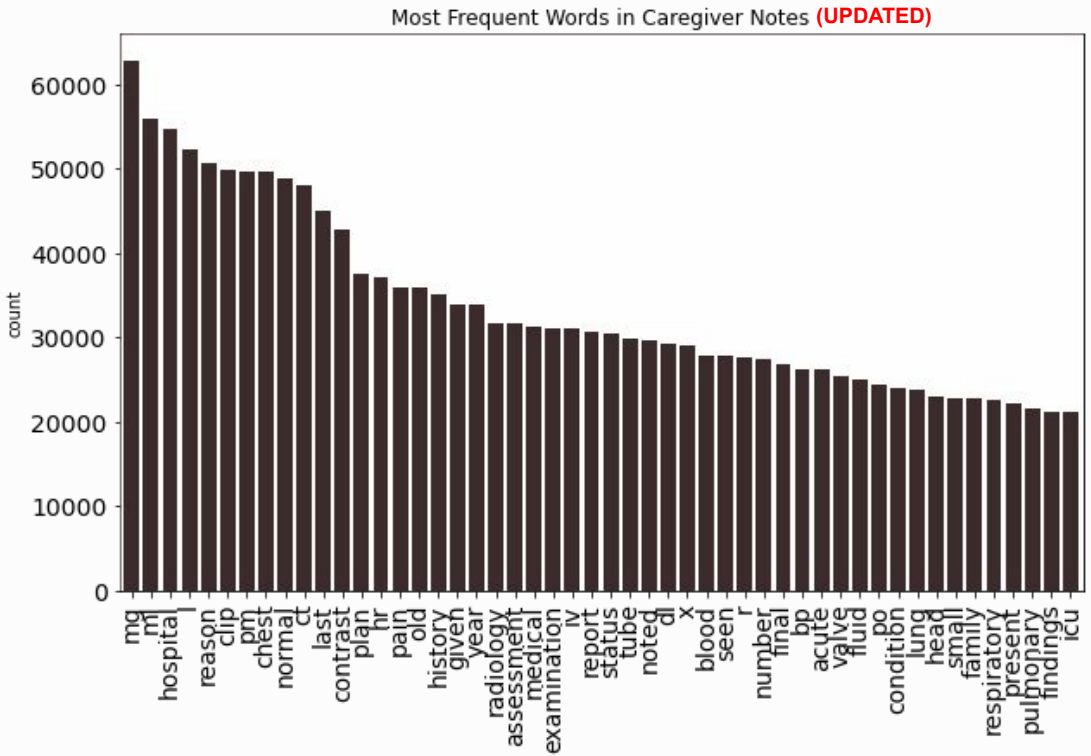
1. Numbers, punctuation, and irrelevant characters were removed from text.
2. A pre-made list of stop words was used to remove irrelevant words (such as “a”, “the” “and”).
3. Most frequent words were noted. Additional stop words were added to the list (see next 2 figures).
4. A vocabulary of ‘tokens’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

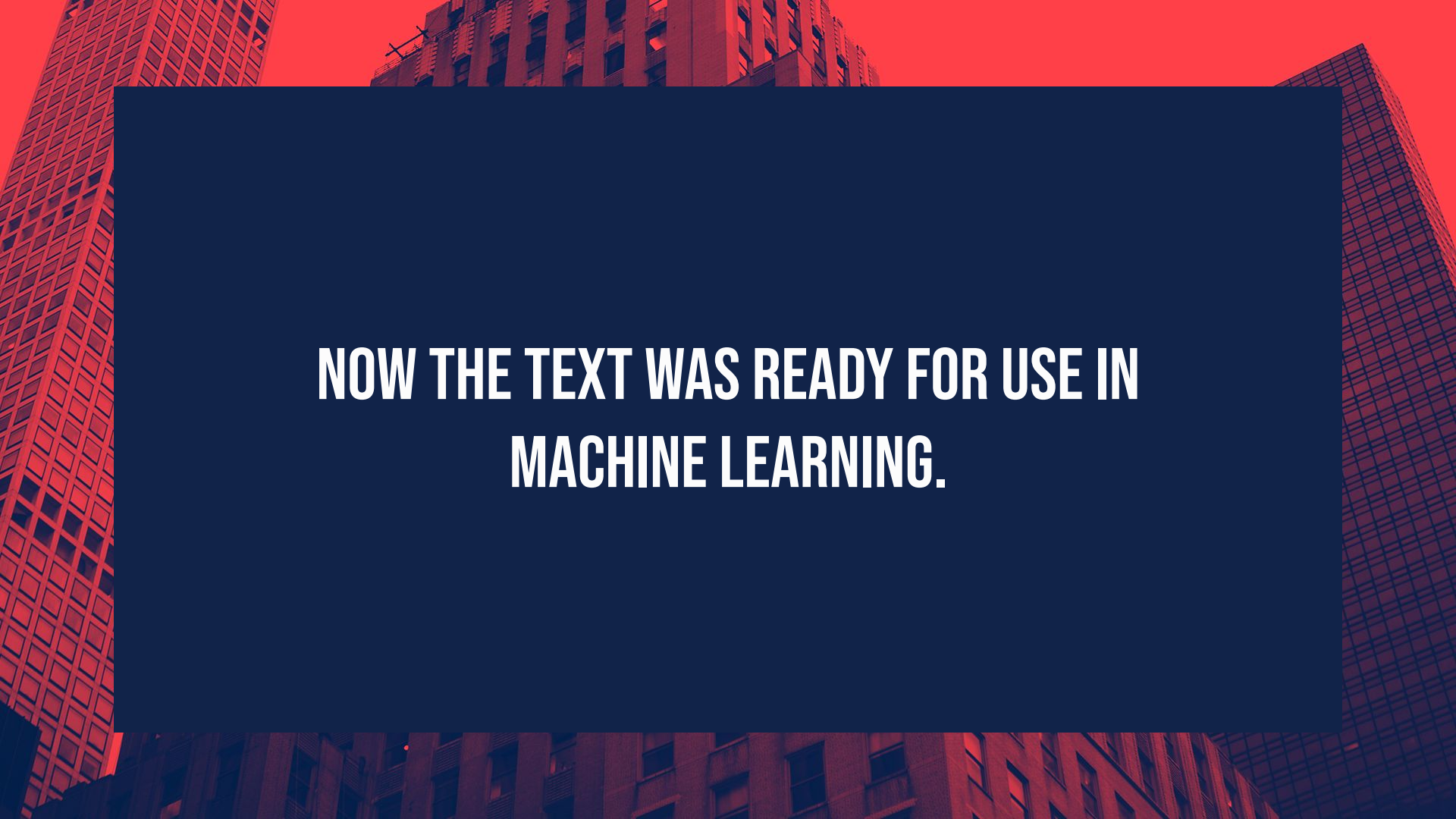
EXPLORATORY DATA ANALYSIS



Some of the top words like ‘pt’ (abbreviation for ‘patient’), ‘left’, ‘right’ and ‘name’ could be considered irrelevant.

EXPLORATORY DATA ANALYSIS




NOW THE TEXT WAS READY FOR USE IN
MACHINE LEARNING.

03.

PREDICTION USING MACHINE LEARNING

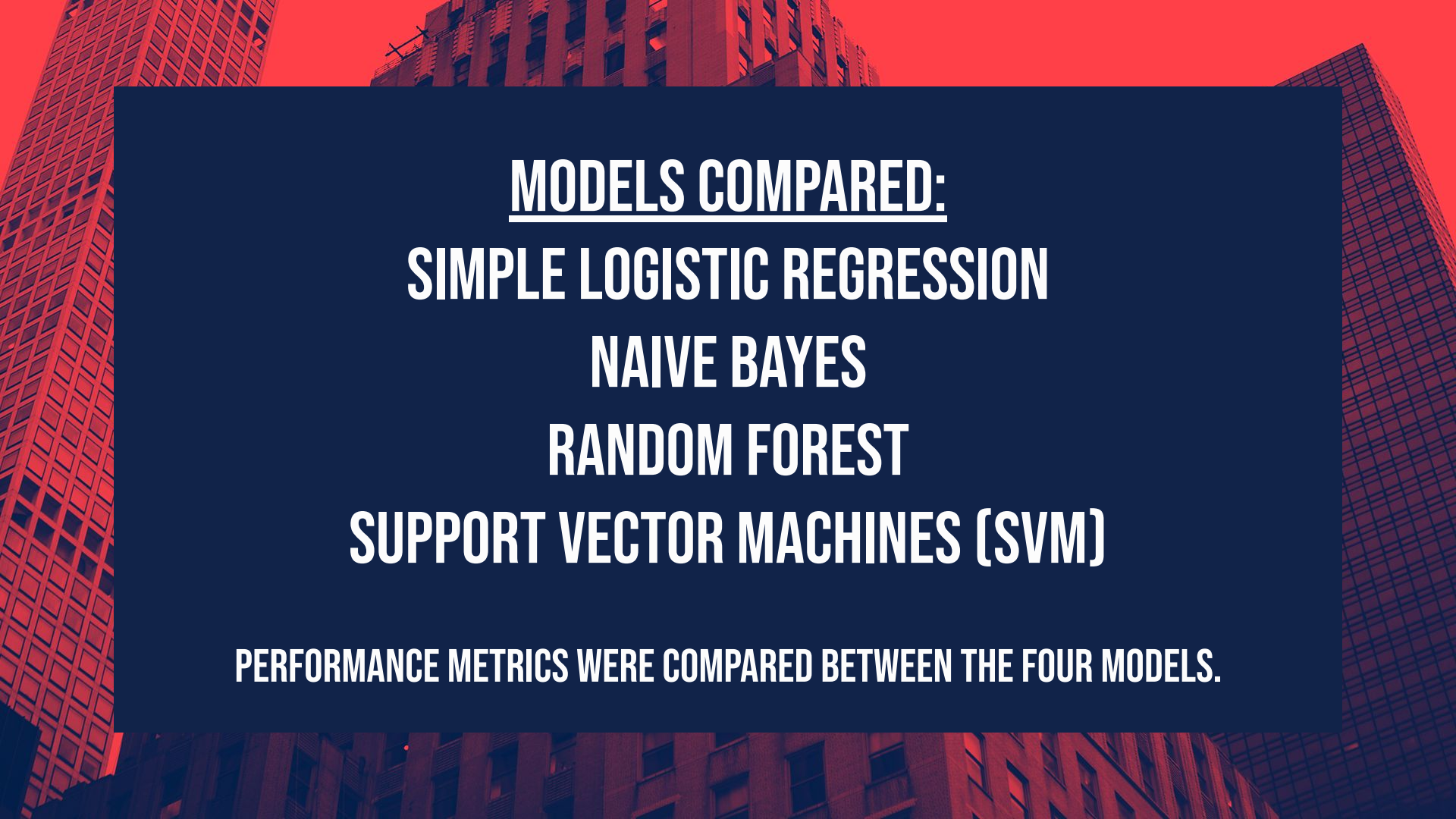
MACHINE LEARNING

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.



MODELS COMPARED:

SIMPLE LOGISTIC REGRESSION

NAIVE BAYES

RANDOM FOREST

SUPPORT VECTOR MACHINES (SVM)

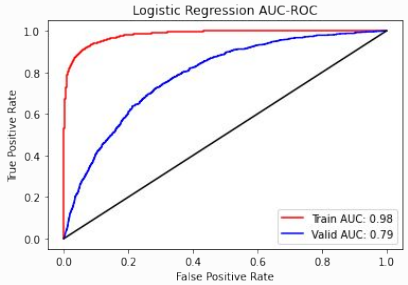
PERFORMANCE METRICS WERE COMPARED BETWEEN THE FOUR MODELS.

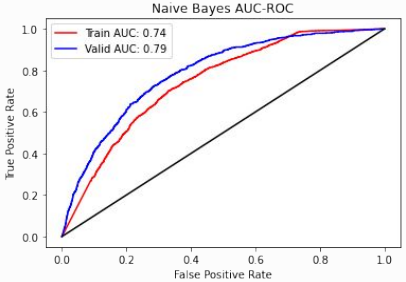
MODEL COMPARISON

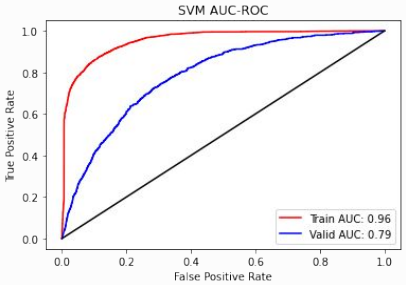
PERFORMANCE METRICS				
	LOGISTIC REGRESSION	NAIVE BAYES	RANDOM FOREST	SVM
AUC	TRAIN: 0.978 VALID: 0.781	TRAIN: 0.743 VALID: 0.699	TRAIN: 1.000 VALID: 0.814	TRAIN:0.955 VALID: 0.814
ACCURACY	TRAIN: 0.917 VALID: 0.717	TRAIN: 0.594 VALID: 0.830	TRAIN: 1.000 VALID: 0.661	TRAIN: 0.876 VALID: 0.671
RECALL	TRAIN: 0.917 VALID: 0.718	TRAIN: 0.283 VALID: 0.248	TRAIN: 1.00 VALID: 0.804	TRAIN: 0.859 VALID: 0.788
PRECISION	TRAIN: 0.917 VALID: 0.187	TRAIN: 0.749 VALID: 0.161	TRAIN: 1.000 VALID: 0.180	TRAIN: 0.895 VALID: 0.177
SPECIFICITY	TRAIN: 0.917 VALID: 0.717	TRAIN: 0.905 VALID: 0.883	TRAIN: 1.000 VALID: 0.648	TRAIN: 0.892 VALID: 0.660

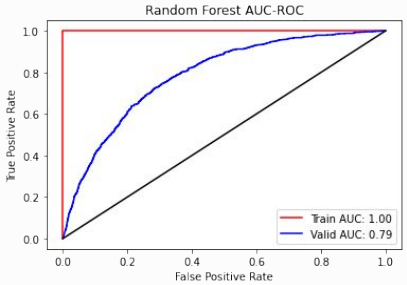
- Logistic Regression(LR), Random Forest(RF), and SVM better than Naive Bayes.
- RF scored highest in nearly all metrics for the training data, but this was due to over-fitting. This model was eliminated from the running.
- LR was the second best, with AUC-ROC of 0.978 (training) and 0.781 (validation) (see next slide) .
- LR also scored best in Specificity, which is important here; it would be desirable to find a model that is unlikely to predict false positives (i.e., incorrectly predict death).

MODEL COMPARISON



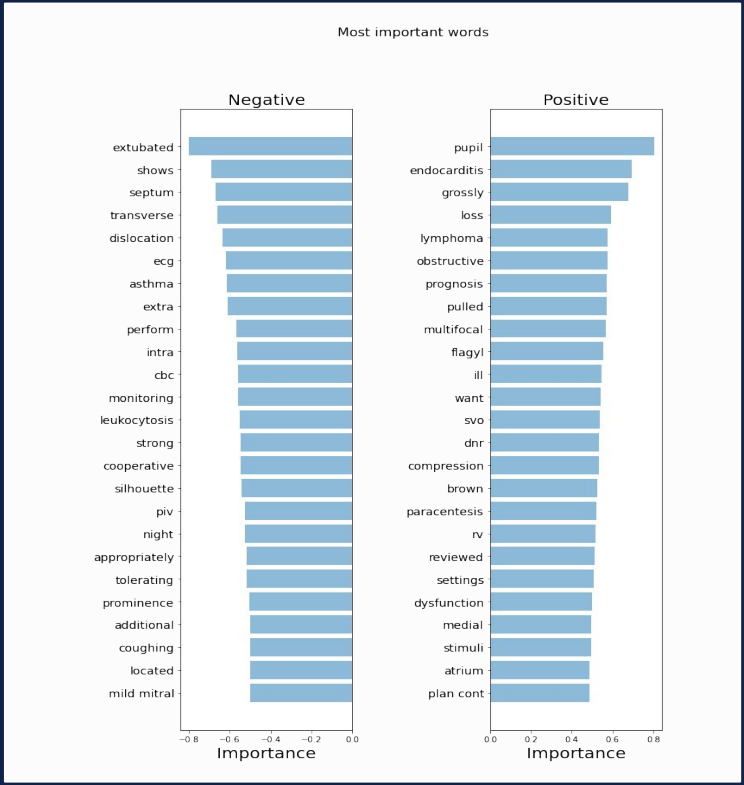






Logistic Regression was chosen as the best model, with AUC-ROC of 0.982 (training) and 0.787 (validation), and less overfitting compared with Random Forest.

IMPROVING THE MODEL: FEATURE IMPORTANCE



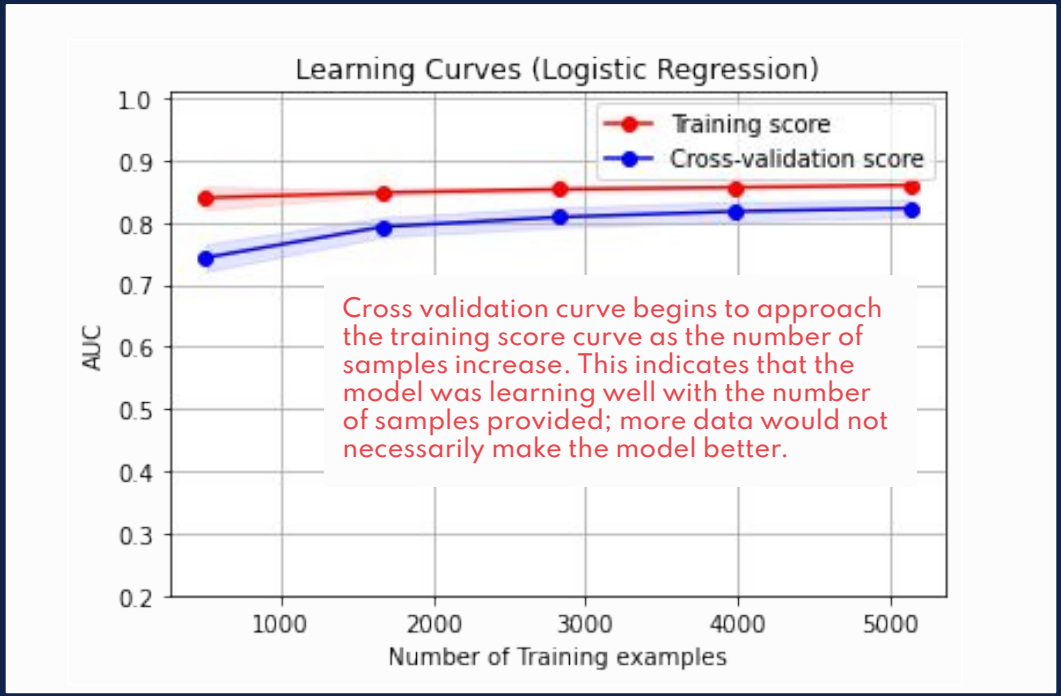
Features with the highest coefficients predict death and the lowest coefficients predict survival.

Most important words make sense:

- ‘Positive’ flagged death words include those describing severe infections, chronic severe medical conditions, and sepsis (endocarditis, flagyl (treatment for sepsis), lymphoma), and possile signs of imminent death ('pupil', 'prognosis', 'dnr' (do not resuscitate').

- The most important ‘negative’ flagged word (i.e. surviving patients) was 'extubated'. Patients who who have been extubated (taken off vent) are doing well!

IMPROVING THE MODEL: SAMPLE SIZE



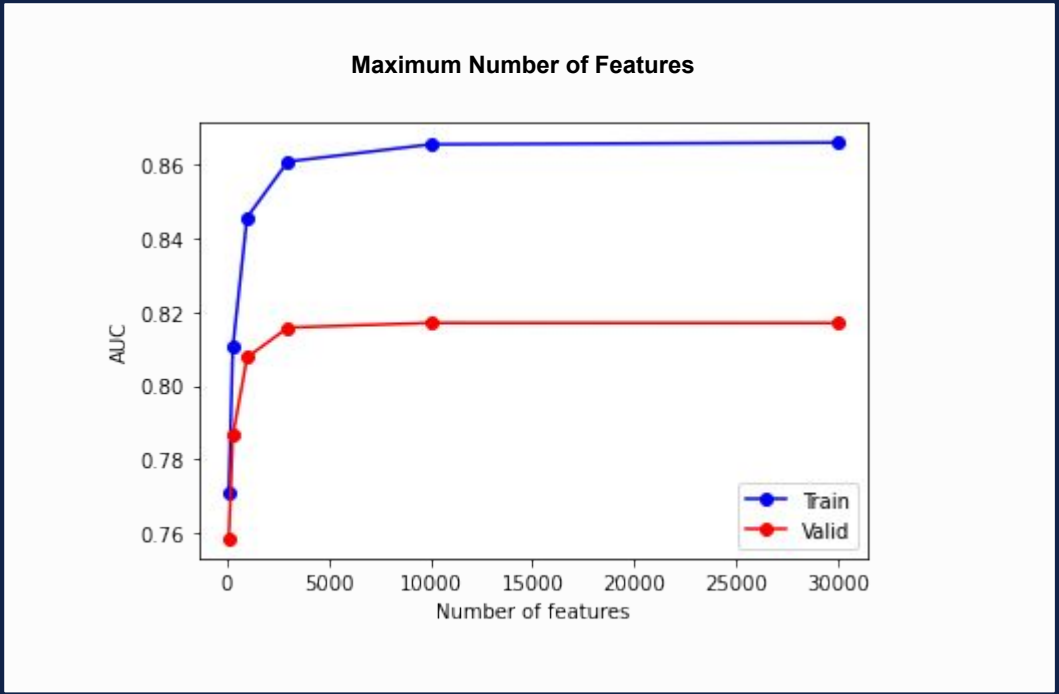
A learning curve was plotted to help assure enough samples are available for the model to make good predictions.

The AUC of the training set was about 0.85, and the Cross-validation score was similar, averaging about 0.8.

Some overfitting is indicated since there is a gap between the training and cross-validation curves.

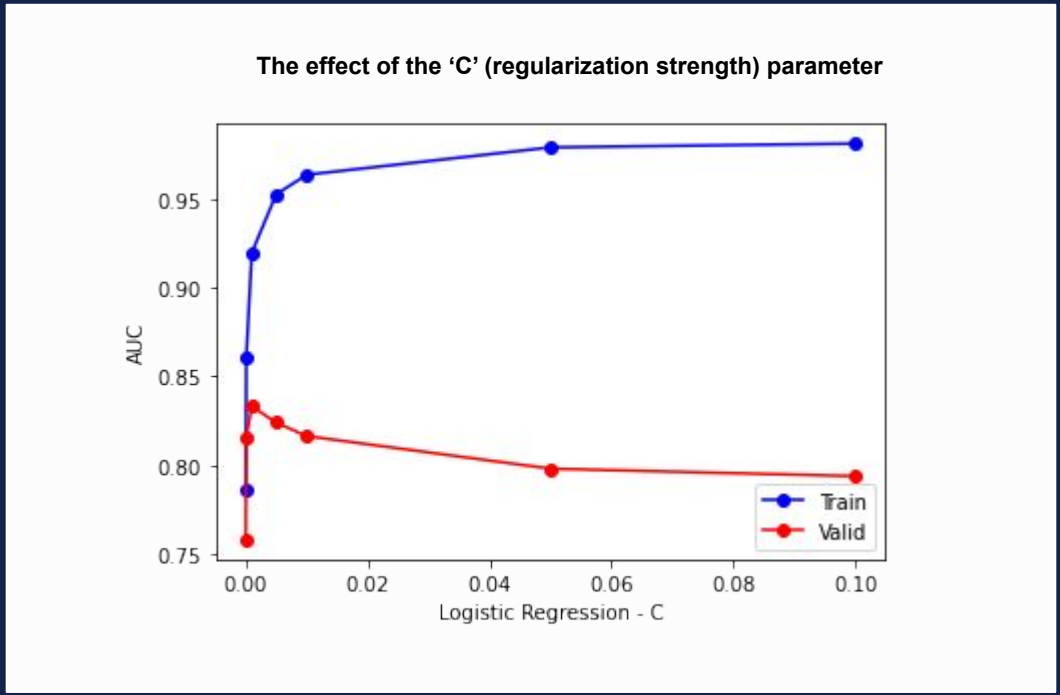
IMPROVING THE MODEL: MAXIMUM FEATURES

The maximum number of features input in CountVectorizer were examined to determine the effect on model's results. As the maximum number of features increase, overfitting occurred. The current max_features value of 3000 looked appropriate.



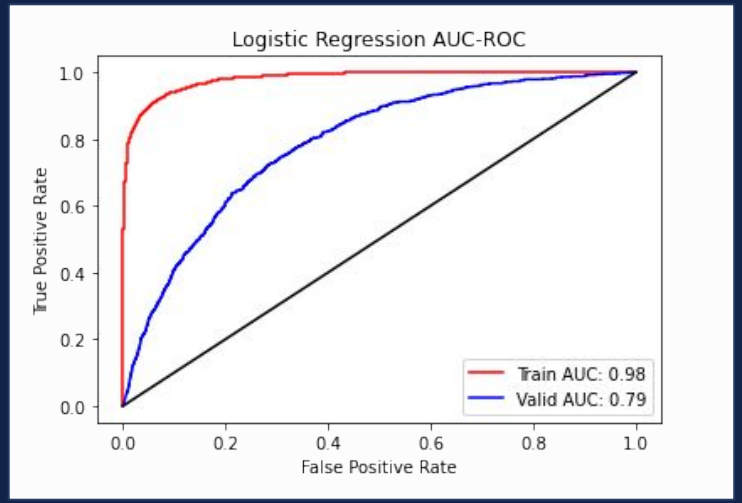
IMPROVING THE MODEL: HYPERPARAMETER OPTIMIZATION

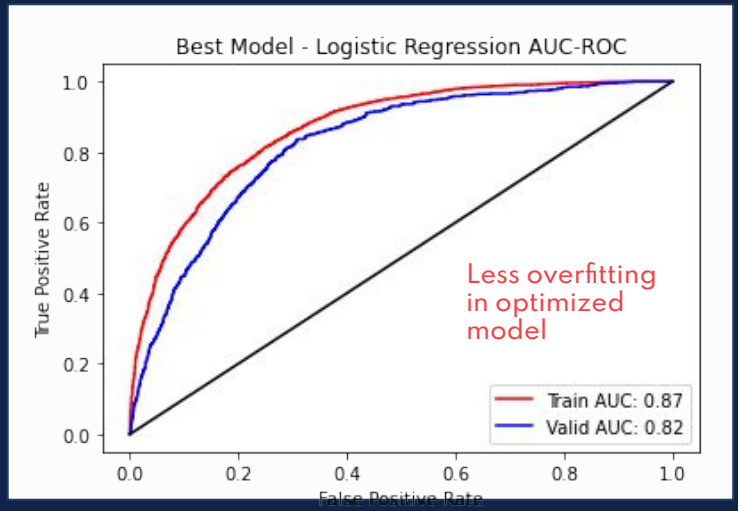
The effect of C on model performance was explored. C is a parameter that determines the strength of regularization; higher values of C correspond to less regularization.



IMPROVING THE MODEL: HYPERPARAMETER OPTIMIZATION

Finally, a GridSearch was performed to help determine best parameters.
GridSearch determined the best performance parameters to be:
'C': 0.0001, 'max_iter': 100, 'penalty': 'l2', 'solver': 'sag'





IMPROVING THE MODEL: HYPERPARAMETER OPTIMIZATION

In nearly all cases, the optimized model had less overfitting when compared with the original logistic regression model.

PERFORMANCE METRICS - LOGISTIC REGRESSION		
	BASIC MODEL	BEST MODEL
AUC	TRAIN:0.978 VALID:0.781	TRAIN:0.868 VALID:0.822
ACCURACY	TRAIN:0.917 VALID:0.717	TRAIN:0.781 VALID:0.703
RECALL	TRAIN:0.917 VALID:0.718	TRAIN:0.829 VALID:0.837
PRECISION	TRAIN:0.917 VALID:0.187	TRAIN:0.757 VALID:0.197
SPECIFICITY	TRAIN:0.917 VALID:0.717	TRAIN:0.734 VALID:0.691

RESULTS - BEST MODEL

This is a visualization of the model's performance, It describes the quantity of patients who were true positives (actual deaths) and true negatives (actual survivors) with those who were predicted positives and predicted negatives.

Performance metrics are based on these values.

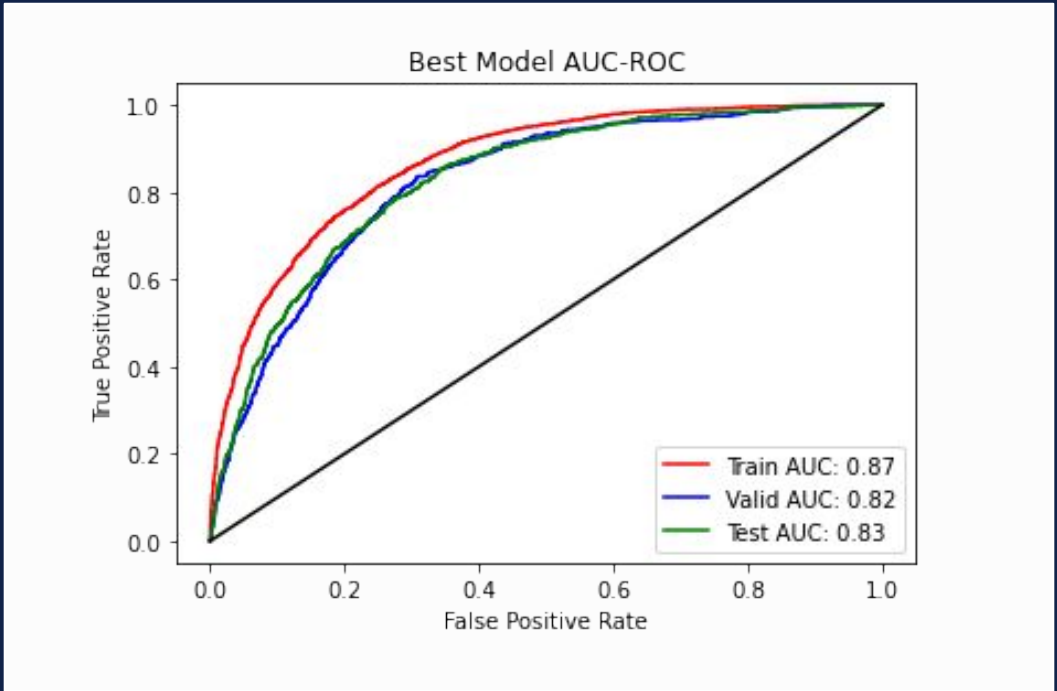
CONFUSION MATRIX - BEST MODEL TEST DATA			
	PREDICTED +	PREDICTED -	TOTAL
+ DEATH	587	148	735
- DEATH	2910	4451	7361
TOTAL	3497	4599	8096

TEST RESULTS - BEST MODEL

PERFORMANCE METRICS: TRAIN, VALIDATION, AND TEST DATA			
	TRAIN	VALIDATION	TEST
AUC	0.868	0.822	0.828
ACCURACY	0.781	0.703	0.711
RECALL	0.829	0.837	0.799
PRECISION	0.757	0.197	0.211
SPECIFICITY	0.734	0.691	0.702

- *Accuracy*: Aggregation of model's performance. Out of all patients in the database, 71% of patients' +/- mortality was predicted correctly.
- *Recall/Sensitivity*: 80% of actual deaths were predicted correctly.
- *Specificity*: 70% of patients who survived were correctly predicted to survive.
- *Precision*: 21.1% of all predicted deaths were correctly predicted to be death. This means that although the model captured 80% of the patients who were truly dying, many of those predicted as deaths actually survived.

RESULTS - BEST MODEL



There remains a small amount of over-fitting, but the model is much improved in that respect compared with the baseline logistic regression model.

INTERPRETATION OF RESULTS: A CASE STUDY

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

Based on data from the American Hospital Association[6], the average hospital in the US has approximately 6,589 admissions per year.

ICU stays accounted for 27% of these admissions.[7]

Therefore, it can be predicted 1,779 hospital stays would include at least one ICU stay.

INTERPRETATION OF RESULTS: A CASE STUDY

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

The confusion matrix shows counts of actual and predicted ICU deaths and survivals applied to the average hospital.

CONFUSION MATRIX - BEST MODEL AVERAGE US HOSPITAL			
	PREDICTED +	PREDICTED -	TOTAL
+ DEATH	129	33	162
- DEATH	639	978	1618
TOTAL	768	1011	1779

INTERPRETATION OF RESULTS: A CASE STUDY

For the average hospital in the US, total costs per year from ICU patients who die during hospitalization amount to an average of over $5.3 Million.[6,7,8,9]

Assumptions:

- Average cost of ICU hospital stay: $32,879[8,9]
- Cost of hospice care in place of hospitalization: $11,820[5]
- If option 2 used, 75% of those truly at end of life would be discharged to hospice 24 hours after admission.

INTERPRETATION OF RESULTS: A CASE STUDY

The following cost savings analysis is based on two possible actions a health system could take:

Option 1: No intervention

- Costs from in-hospital mortality remain the same.

Option 2: Employ machine learning model.

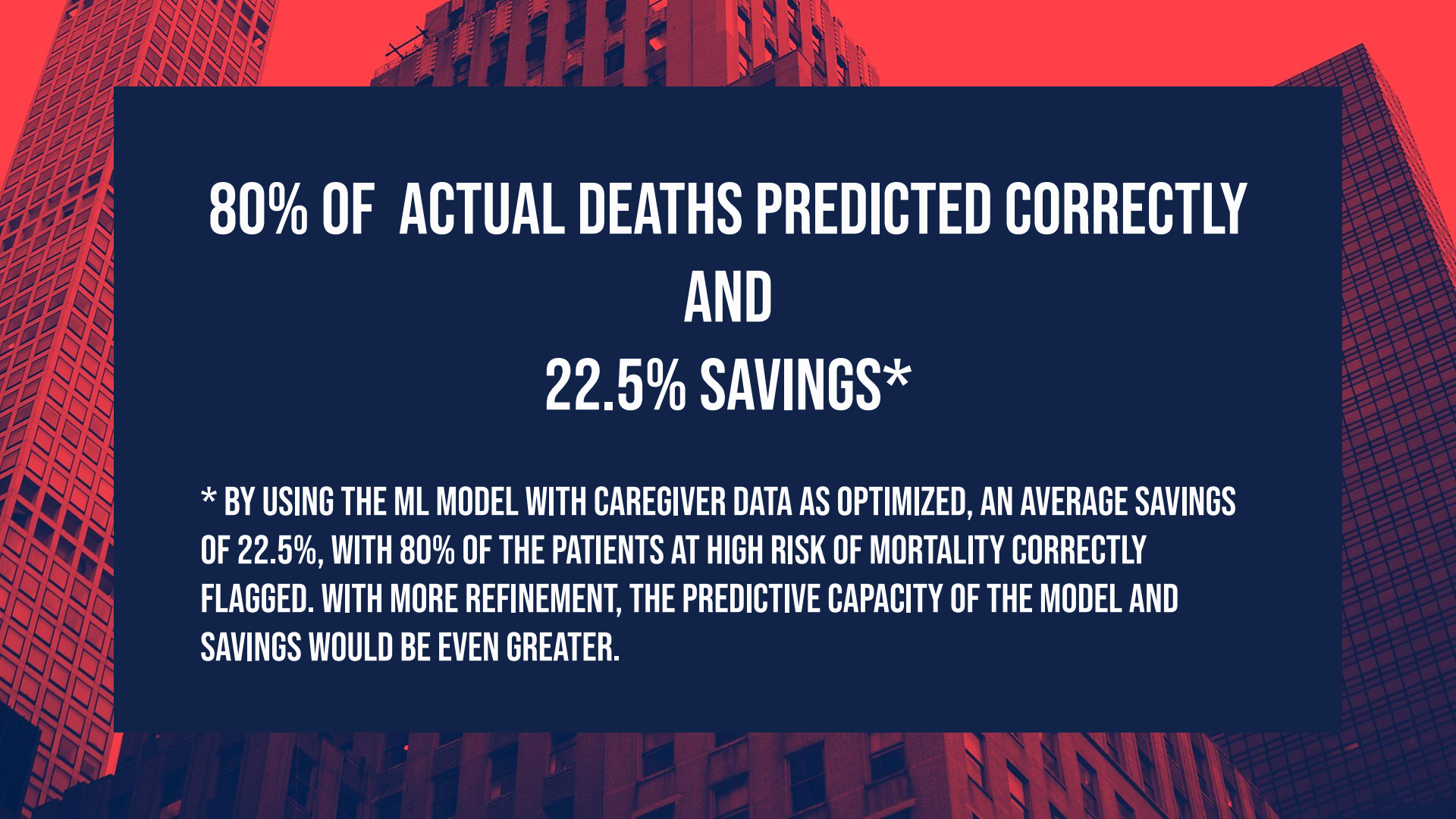
- Patients who are flagged as ‘high risk’ of death are given extra attention within 24 hours of admission.
- Per physician’s discretion, consideration is given for discharge to home, hospice or palliative care.

RESULTS INTERPRETATION - A CASE STUDY

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

If the machine learning model were used, an average savings of approximately **$1.19 million** could be achieved.

SAVINGS ANALYSIS		
	OPTION 1	OPTION 2
COST OF HOSPITALIZATIONS	$5,310,550	$5,310,550
SAVINGS	$0	$2,336,273
HOSPICE CARE COSTS	$0	$1,143,534
TOTAL SAVINGS	$0	$1,192,739
PERCENT SAVINGS	0.00%	22.46%



80% OF ACTUAL DEATHS PREDICTED CORRECTLY AND 22.5% SAVINGS*

* BY USING THE ML MODEL WITH CAREGIVER DATA AS OPTIMIZED, AN AVERAGE SAVINGS OF 22.5%, WITH 80% OF THE PATIENTS AT HIGH RISK OF MORTALITY CORRECTLY FLAGGED. WITH MORE REFINEMENT, THE PREDICTIVE CAPACITY OF THE MODEL AND SAVINGS WOULD BE EVEN GREATER.

LIMITATIONS AND OPPORTUNITIES (1 OF 2)

- Adjust the data being used:
  - Use only Admission note (rather than all notes from first 24 hours)
  - If sufficient data were available, create separate algorithm for specific medical conditions (e.g., liver disease or cancer)
- More feature engineering, optimization, and choice of classification model:
  - NLTK's regular expressions,
  - TF-IDF in place of countvectorizer,
  - Normalizing and grouping tokens with stemming and lemmatization
  - During model comparison, logistic regression and SVM were quite close. An optimized SVM model could be compared with the optimized linear regression model.

LIMITATIONS AND OPPORTUNITIES (2 OF 2)

- Deep learning framework:
  - As an alternative to the traditional machine learning models of logistic regression and SVM, the preprocessed data could be fed into a deep learning framework which could take the sequence of words into account, creating a more sophisticated model that may be more accurate in its predictions.
  - Possible Python libraries to use: Tensorflow, HuggingFace Transformer, Keras, and spaCy
- Create a meta-model:
  - A stacked ensemble model could be created using the best NLP model along with a model that predicts mortality based on other features included in the MIMIC III dataset, such as laboratory data, vitals, and other patient data. The combination of the two models into a could likely reduce the number of false negatives and improve accuracy.

CONCLUSION

Using caregiver notes from the first 24 hours of hospital admission, a better-than-random model can be built to predict mortality in ICU patients.

- This could be used to support physicians in a high-stress environment who are often pressed for time.
- Such a model would be beneficial to both patients and healthcare organizations, helping to reduce unnecessary and intrusive procedures and substantially reducing financial costs.
- It is important to emphasize that the model's predictions are meant to be used to flag patients who are at high risk of mortality based on caregiver notes, so that a timely discharge is considered. Many surviving patients were, in fact, predicted to die with the model. It is up to the physician and care team to determine the patient's risk of death.
- A feasible next step would be to create a stacked ensemble model, which would allow for a stronger algorithm to predict mortality in ICU patients.

THANKS!

Do you have any questions?

carashri@hotmail.com

Credits

Special thanks to all those who helped:

- Tommy Blanchard (my Springboard mentor)
- Springboard

REFERENCE LINKS

1. ICU Outcomes - UCSF
2. Outcomes From Intensive Care in COVID-19 Patients
3. Long Term Care Market Size, Share & Trends Analysis Report by Service (Home Healthcare, Hospice, Nursing Care, Assisted Living Facilities), by Region (North America, Europe, APAC, Latin America, MEA), and Segment Forecasts, 2020 - 2027
4. Patterns of Cost for Patients Dying in the Intensive Care Unit and Implications for Cost Savings of Palliative Care Interventions
5. Better care of sickest patients can save hospitals money, says largest study of its kind
6. Fast Facts on U.S. Hospitals, 2020
7. ICU Occupancy and mechanical ventilator use in the United States
8. Average hospital expenses per inpatient day across 50 states
9. Critical Care Statistics - ICU costs per stay
10. This blog post, by Andrew Long, was a helpful reference for the completion of this project: Introduction to Clinical Natural Language Processing: Predicting Hospital Readmission with Discharge Summaries.