1-Year Mortality Prediction of Compassus Hospice Patients

Master of Analytics - Final Practicum

https://github.gatech.edu/dlevesque3/GaTech_Practicum

Derek Levesque
Georgia Institute of Technology
dlevesque3@gatech.edu

1 Introduction

The home health care industry has steadily grown over time as the prevalence of communicable diseases has declined and the rise of lifestyle-based chronic conditions has increased. As demographic trends continue and the baby boom generation retire, the average age of the US population is the highest its ever been. The COVID-19 pandemic which we are just exiting has contributed even more to the growth of this industry. The market size in terms of revenue of the Home Care Providers industry is $114.9bn in 2022 and is one of the fastest growing segments of the healthcare industry today. [1]

The home health industry is generally split into home health care and hospice with hospice care being home health that focuses exclusively on end of life care where patients have less than 6 months to live. This patient segment has different and more specialized needs than the more general home health care population and thus there is much focus on developing technological and analytical tools to help determine the point in time when patients should be considered for hospice. In the easiest of cases it can be quite obvious such as a diagnosis of terminal cancer with less than 6 months to live. However for the majority of cases where patients succumb to old age or 'natural causes' it is much less clear.

The US government is actively involved in this area in terms of setting regulations, although to a much lesser degree than seen in Canada or Europe. Unfortunately there is much work to do in terms of modernizing the industry. There is little coordination or standardization of individual health records meaning any analytical task is faced with data quality issues. Thankfully The Office National Coordinator for Health Care Policy (ONC) entire mission is working to standardize Electronic Health Records (EHR) but there is still a long road ahead.

2 Problem Definition

Compassus, a leading home health and hospice provider has asked us to develop a model and analytical approach to predict mortality within 1 year. As a corporate provider of both general home health care and hospice care, a core focus of their business is determining when to move patients into hospice.

Accurate predictions offer quality of life benefits to patients and economic benefits to providers as resources can be allocated more effectively and the proper attention and care can be prioritized to those patients with the most urgent needs. Accordingly, there exists strong incentives for research into new and improved prediction approaches. [2]

I implemented a data pipeline and model to classify patients and determine the likelihood a given patient will perish within one calendar year so that they may be transitioned to hospice care at the optimal time.

3 Data

I was provided an excel data file with 28,000 records of patient-level health data manually compiled from various internal Compassus systems. There were 5,930 patients in total and 250 different independent features including the usual demographic information along with features representing individual patient outcomes. From the highest level view the dataset was comprised of 3 distinct patient populations, identified in the **DispositionId** column: hospice patients who had perished, home health care patients who perished before being moved to hospice, home health care patients who were still alive.

As is common in the health-care industry the data was manually compiled from multiple different internal systems and thus the dataset had numerous problems that prevented a quick and straightforward analysis. The main issues discovered were:

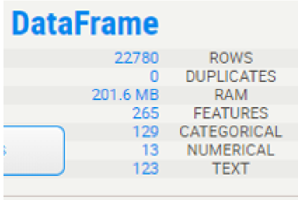
1. **Patients with two different patient ids (pa_id).** It seems that patients who eventually transitioned to hospice were given a new patient id which would skew the results as the events would not be attributed to the same individual. By matching on the Gender and Year_born columns, along with the ceo_id (visit id) we were able to identify these patients and sync their patient numbers.

2. **Patients who had a discharge date, AFTER their start of care date.** This is obviously not possible so we either needed to eliminate them completely or manually identify what happened in each individual case.

3. **Extraneous rows that were added due to the inclusion of HIPPS/HHRG codes.** These rows were essentially duplicates except for these columns only, which meant the dataset has way more rows than it needed to. Using python/pandas we found rows that were identical except for these columns and then consolidated the info into a single row.

4 Data Processing

This dataset was a perfect example of the crucial importance of data-wrangling and data-cleaning as garbage into a model will skew the results and in such important area as end-of-life health care, it is



DataFrame

22780	ROWS
0	DUPLICATES
201.6 MB	RAM
265	FEATURES
129	CATEGORICAL
13	NUMERICAL
123	TEXT

Figure 1: The Dataset Provided by Compassus

all the more critical. If the model can't attribute lab test results and diagnoses to the correct patient it will not learn correctly. With the data integrity issues in mind, I felt the most valuable approach was to build a new data framework so that modeling might be easier to replicate in the future, even if the dataset changes form or data becomes available from a different source. This framework had me conceptualize the data into four main clusters and programatically split them out into separate tables/dataframes.

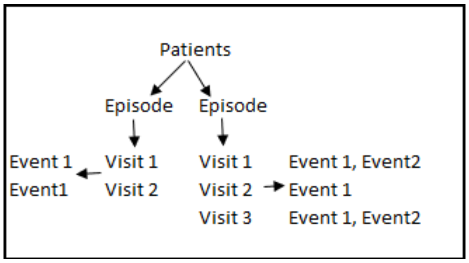


Figure 2: Standardized Mortality Prediction Layout

Patients - Each distinct patient and their basic demographic information and mortality status

Episodes - episodes of care are the set of services provided to treat a clinical condition or procedure over a 60 day period. This is an industry standard and is how insurance companies including Medicare reimburse and monitor care.

Visits - are events where medical provider care is actually delivered along with physically observing the patient and documenting findings according to OASIS standards. The first visit is known as the Start of Care (SOC) date with subsequent visits occur periodically thereafter. Each patient may have one or several visits in any particular episode.

Events - Lab results, diagnosis codes, patient questionnaires and observations. In this data-set the events are standardized OASIS results. The Outcome and Assessment Information Set (OASIS) is a group of standard data elements developed and tested by the Centers for Medicare Services (CMS). OASIS standards were designed to enable systematic comparative measurement of home health care patient outcomes.

Any mortality prediction exercise can be fundamentally broken down into these four main components. This standardized approach allows for an organized process that can be reproduced as more data becomes available even if structured differently. [3]

4.1 Class Imbalance

A common problem in mortality prediction (and binary classification in general) is that of a class imbalance which occurs when one of the two classes is significantly under-represented in the data.[4] In our case, we have approximately 2900 patients who have perished in less than 1 year (majority class), and only 200 patients who we can definitively determine have lived longer than 1 year (minority class). The sampling distribution of the training features overlap significantly which exacerbates the problem of learning from imbalanced data.[5] We had to throw out a substantial number of patients data because they have not died, and we don't have more than 365 days of data to definitively determine they will live more than 1 year which is the same as classifying them as a member of the minority class. In other words, they may live longer than 1-year and they may not, we don't have enough data to know.

To initially determine any effect the class imbalance may have, I artificially created a balanced data-set by reducing the data used for validating the model. I utilized 100% of the underrepresented class and then selected an identical number of patients from the majority class by filtering the patients based on the most typical cases. This obviously will throw away a multitude of patients but by choosing patients with the most unique characteristics this approach actually did not substantially change the results.

5 Analytical Approach

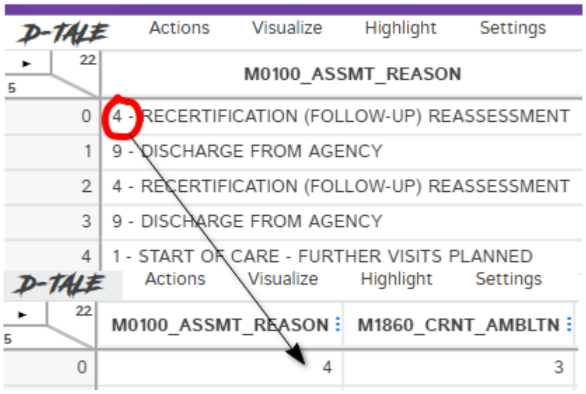
This project primarily used the Python programming language for both the data processing and the modeling. Jupyter notebooks were used for the front-end due to its ease of interactivity but python files outside of Jupyter were also created to implement the model due to some compatibility issues with Jupyter. I used the Windows Subsystem for Linux as the OS due to its advantage in leveraging the GPU for faster modeling. Linux tends to have fewer problems than Windows

Some helpful python packages heavily used were:
Pandas was the main workhorse for its unbeatable and seemingly infinite number of useful data-wrangling methods
CUDF leverages the GPU with pandas dataframes for resource intensive tasks
DTALE for an excel-style interactive data exploration and manipulation of the dataframe including sorting and filtering right inside the browser window.
SweetViz tool for data cleanup and initial feature selection and general overview of the data. It offers a web-based GUI and provides information on every feature/column.

Visit the following link see my work or use for your own analysis: https://github.gatech.edu/dlevesque3/GaTech_Practicum

In addition to the previously mentioned high-level data integrity concerns, the effectiveness of a mortality prediction model/algorithm is directly contingent upon the quality of features representing the data. Given that we initially were provided up to 250 features to work with it was imperative to perform some feature selection to not only eliminate any redundancies but also ensure the information contained would not skew the model output.

Since most of the features are structured and standardized in the OASIS format it allowed us to programmatically extract the useful information. A typical format is: *"1 - Yes, the patient is experiencing x", "0 - No, the patient is not y"*. This structure allows us extract the integer away from the entire string so the model will receive a simple integer input for each feature. For the categorical features that are missing the integer and have Yes/No we can simply convert them to integers and in the case of string only information, it is almost always limited to 6 or fewer possible responses so therefore also possible to convert to integers.



	M0100_ASSMT_REASON
0	4 - RECERTIFICATION (FOLLOW-UP) REASSESSMENT
1	9 - DISCHARGE FROM AGENCY
2	4 - RECERTIFICATION (FOLLOW-UP) REASSESSMENT
3	9 - DISCHARGE FROM AGENCY
4	1 - START OF CARE - FURTHER VISITS PLANNED

	M0100_ASSMT_REASON	M1860_CRNT_AMBLTN
0	4	3

Figure 3: Standardizing Feature Value Strings into Integers

Since all of the features are dependent on actual human health care providers to collect, there are certain items that seem to be more prevalent across the patients. For example, almost every patient has their height and weight calculated, but some of the more obscure questions are largely blank. In this light I looked at which features most patients in both classes had in common and prioritized those. We need to be careful in that it is certainly possible for patients who died quickly to have features that patients who lived longer might not have, so its not simply eliminating features with a high proportion of N/A rows. More thought and analysis in selecting which features to keep is warranted.

5.1 Parallel Processing

With a dataset of this size I wanted to fully utilize my high core CPU and GPU to leverage parallel processing. I decided on the Joblib python package because it was quickest to implement. It significantly cut down processing time because each of the 12 CPU cores could simultaneously process the data. For some reason I tried to use the GPU and it was actually slower so not sure if there was a configuration issue.

6 Long Short-Term Memory (LSTM) Model

The mortality prediction task can be approached in many different ways according to the objectives and the type of data provided. I wanted to try to challenge myself a bit so I chose to implement a deep learning model and try to see if I could get my GPU involved on my local machine. Given that the data-set provided has several time/date components of critical importance each patient can be viewed as a distinct entity that is subject to a sequence of events. These events are episodes which are 60 day time periods used to manage patient care and observe. There are additional sequences of Health Care provider visits in which observations are collected.

Since the sequence of the events are inherently meaningful we need to utilize a model that can derive meaning not only from the events themselves but from their sequential relationship. For example, imagine a patient, who develops cancer, and then 2 months later, his appendix burst. This is inherently different than someone who is perfectly healthy and has has their appendix burst. An event that occurs while a patient's immune system is suppressed may have a different outcome.

So the question we are trying to model is the following: **Did a given sequence of medical events that occurred within a 365 day time period result in patient mortality?**

A recurrent network has the ability to keep track of past events inside its hidden states by 'remembering' the previous sequence of events and passing them to the next layer in the network. [6]

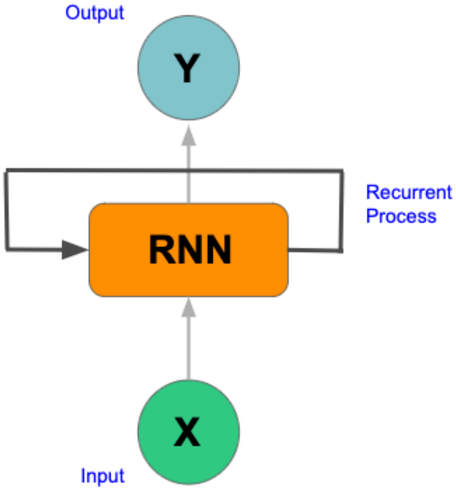


Figure 4: An RNN allows previous outputs to be used as inputs

I chose to implement a long short-term memory (LSTM) model as it was a specifically developed recurrent neural net designed to address the challenges of time-series data. By feeding the model chronological episodic visit data, the order of the events that occur to any patient become significant predictors. LSTM models are uniquely compatible for mortality prediction because they 'remember' the sequential occurrence of events and find patterns over time. Contrast this with more simplistic models which will simply look at ALL the event that have happened to a specific patient in a specific time period without considering the chronology. LSTM are an improvement upon traditional RNN because they have a longer-term memory and are not affected by only the most recent events. [7]

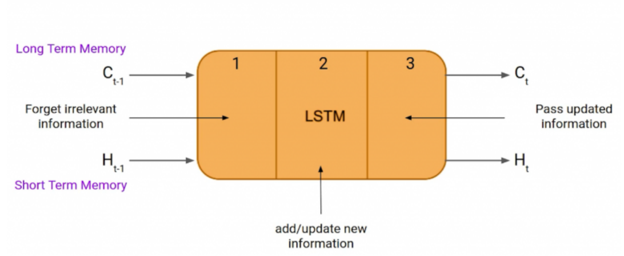


Figure 5: 3 Parts to LSTM: 1. remember/forget previous 2. New Info Learned 3. Pass updated info

6.1 Hyperparamater tuning

After splitting the data into training, validation and testing tranches it was time to tune the model by experimenting with the hyperparameters of the LSTM model to try and prevent over-fitting to the training data.

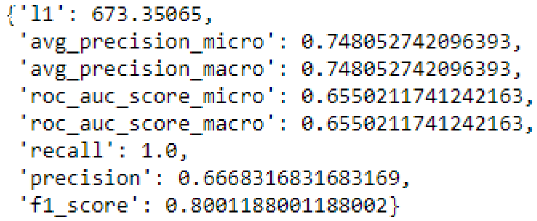
The number of epochs is a hyper-parameter that controls the number of complete loops through the training data-set. A larger number means the model runs for longer period of time and spends more time learning, however this can lead to over-fitting. I settled on 50 Epochs which had similar results as 200 epochs in much less time. I also experimented with batch size which controls the number of samples processed before the model updates. I settled a batch size of 32. Finally I tried different learning ratios which control how much the weights of the model change in response to the estimated errors during learning. At very low values my model simply ran forever and I didn't have the patience. I was probably getting stuck in a local minimum. At the larger end the results were sub-optimal. The final setting was .0001. I settled on 32 for the hidden layer size (single layer). I also ran a bidirectional LSTM with the same parameter values.[8]

Most of this experimentation had more of an effect on speed of the model and resource consumption on my machine than they did the classification results.

7 Results

Before checking the actual classification results. I wanted to make sure the model itself was running correctly. I compared the training loss vs validation loss and validation loss was just slightly higher which indicates that the model trained for enough time and is not under-fitting/over-fitting the data to a significant degree.

All iterations of analysis used a logistic simgoid function for the binary classification. For evaluation metrics I used the area under the receiver operating characteristics (ROC-AUC score) and Precision-Recall Curve (or Average Precision) to report the model's performance.



```
{'l1': 673.35065,
 'avg_precision_micro': 0.748052742096393,
 'avg_precision_macro': 0.748052742096393,
 'roc_auc_score_micro': 0.6550211741242163,
 'roc_auc_score_macro': 0.6550211741242163,
 'recall': 1.0,
 'precision': 0.6668316831683169,
 'f1_score': 0.8001188001188002}
```

Figure 6: LSTM Mortality Prediction Model Results

Since we did have a class imbalance as previously mentioned I used both micro and macro measurements to try and gain an understanding of how this affected the results. A macro-average treats both classes equally by computing the metric independently for each class and then taking the average, while a micro-average aggregates the contributions from both classes. [9]

The precision metric demonstrates the models effectiveness in predicting true positives. In our context it is showing how well the model correctly predicts mortality within 1-year. Our precision of approximately .75 means that the model is definitely better than a random guess. It does have some false positives, meaning it would suggest some patients transferred to hospice too early.

An issue that deserves further scrutiny is the recall score. It seems that the model may only be predicting 1 class, the majority class. This is to be expected in such a highly unbalanced data-set, but even when correcting for this imbalance by creating an artificially reduced data-set with an equal number of patients from each class I am only seeing the majority class being predicted by the model. Once again this may be related to the data itself which is not of greatest quality.
The recall metric indicates the proportion of actual positives identified correctly by the model.

$$\text{Recall} = \frac{TP}{TP + FN}$$

A recall of 1.0 means there were no false negatives, or in our case, the model did not suggest that someone should be moved to hospice when they really shouldn't. On its surface this is a desired outcome. In many medical outcomes we want to be conservative and err on the side of caution. We don't want to give someone a death sentence when they are not deserving. However, a perfect recall score suggests a deeper look is needed.

The F1 score combines both precision and recall and indicates model quality in terms of false positive and false negatives. Our F1 of .80 is pretty good and that means are identifying the right people to transfer to hospice while simultaneously not transferring patients who don't meet the criteria.

$$F1 = 2 \times \frac{\text{precision} \ \times \ \text{recall}}{\text{precision} \ + \ \text{recall}}$$

What proportion of actual positives was identified correctly?

$$\text{Precision } P = \frac{TP}{TP + PP}$$

The Roc_AUC_Score also indicates the model is providing some value vs a random guess. It is demonstrating an ability of the model to distinguish between classes and thus has detected signal in our data that allows it to make accurate predictions a majority of the time.

The precision-recall curve helps us to dive deeper into the class imbalance and its possible effect on results. The PR curve is graphical representation of a classifier's performance across many thresholds. For our purposes by default we used standard Logistic sigmoid threshold of 50%

PR curves can be advantageous to ROC-AUC curves for highly unbalanced data-sets because ROC curves are insensitive to class imbalance and thus might not clearly indicate results. [10]

We can see that by

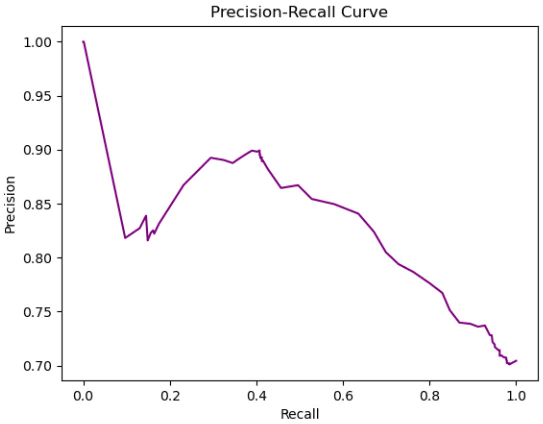


Figure 7: Precision vs Recall at various Thresholds

8 Conclusion

An LTSM recurrent neural net model fed standardized medical event data from the Outcome and Assessment Information Set (OASIS) does accurately predict the one-year mortality of Compassus home health care patients. It is therefore a valuable approach and is more accurate than simple random guessing. I would recommend to implement this approach at the company but perhaps more importantly would be doing everything possible to ensure a more robust data gathering pipeline as a first-priority.

As we learned in ISYE 6501, all models are wrong but some are useful. Our results do indicate decent model quality with a few potential issues. The degree of false positives and false negatives is low yet the recall is a perfect 1. Upon closer inspection it seems that only 1 class is being predicted. All models have trade-offs in these areas and more introspection and tuning may be necessary.

Ideally I would have liked to see even higher quality results and I believe by experimenting with different RNN's and having a cleaner data-set it is certainly possible to improve. I have seen other papers where precision in the upper 80% percentile so I believe data quality/integrity issues may be skewing the results rather than a sub-optimal analytical approach.

9 Possible Improvements

The dataset we were provided had alot to be desired. There were data integrity issues and the historic length was limited. To truly assess and leverage the power of an LSTM model we need data for a longer historic time period with more events per patient. We should incorporate EHR from before they entered care under Compassus supervision. Ideally we would have real data that doesn't have such a pronounced class imbalance which can skew results.

9.1 Feature Selection

I experimented with several different feature selection techniques to see what the effect on the results would be. I manually reduced the original features from 250 to 115 using SweetViz and through a manual search for the degree of null values across patients in both classes. I kept the features with the greatest percentage of populated values. I realized that this approach may have been flawed in that a null value for one class might actually be ok, because patients who died quickly may have certain characteristics that patients who lived long do not have. Null values are not necessarily "missing" values or indicative of data quality issues.

References

[1] Steven Landers, Elizabeth Madigan, Bruce Leff, Robert J. Rosati, Barbara A. McCann, Rodney Hornbake, Richard MacMillan, Kate Jones, Kathryn Bowles, Dawn Dowding, Teresa Lee, Tracey Moorhead, Sally Rodriguez, and Erica Breese. The future of home health care: A strategic framework for optimizing value. *Home Health Care Management & Practice*, 28(4):262–278, 2016. PMID: 27746670.

[2] Cari Levy, Monica Morris, and Andrew Kramer. Improving end-of-life outcomes in nursing homes by targeting residents at high risk of mortality for palliative care: Program description and evaluation. *Journal of Palliative Medicine*, 11(2):217–225, 2008. PMID: 18333736.

[3] Yue Zhao, Zhi Qiao, Cao Xiao, Lucas Glass, and Jimeng Sun. Pyhealth: A python library for health predictive models. *arXiv preprint arXiv:2101.04209*, 2021.

[4] Rui Zhang. Healthcare data analytics. chandan k. reddy and charu c. aggarwal. boca raton, fl: Chapman hall/crc press (2015) 724 pp. *Journal of Biomedical Informatics*, 2015.

[5] Misha Denil and Thomas Trappenberg. Overlap versus imbalance. pages 220–231, 05 2010.

[6] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. Published online 2014; based on TR arXiv:1404.7828 [cs.NE].

[7] Joon Lee, David M Maslove, and Joel A Dubin. Personalized mortality prediction driven by electronic medical data and a patient similarity metric. *PloS one*, 10(5):e0127428, 2015.

[8] Joon Lee, Joel A. Dubin, and David M. Maslove. *Mortality Prediction in the ICU*, pages 315–324. Springer International Publishing, Cham, 2016.

[9] Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 65–74, New York, NY, USA, 2017. Association for Computing Machinery.

[10] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):1–21, 03 2015.