Final Report

Salim Damerdji, Mandi Zhao, Christine Giang

May 23, 2019

Abstract

Hospitals provide more care and better care when they correctly predict a patient's mortality risk and how long a patient will stay at the hospital. Trained on the MIMIC-III dataset, our neural network provides first-day predictions for patient length-of-stay, with accuracy of 90.438% with RMSE of 19.762 hours. Our model is the first to learn directly from notes written by doctors and nurses.

1 Motivation

Better length-of-stay predictions help hospitals provide more care to patients. Suppose a hospital knew exactly when patients will leave. Then, they could schedule more appointments for when a new bed opens up [4]. They could also optimize the use of costly equipment [1].

Better length-of-stay estimates also enable better care. Long hospital stays lead to "hospital-acquired infections, adverse drug events, poor nutritional levels, and other complications" [1]. To prevent long stays, hospitals can identify patients with longer predicted stays and provide tailored care [4].

Mortality predictions are also crucial to a hospital's success because they enable hospitals to direct more resources towards high-risk patients [9]. Moreover, hospitals are in the business of optimizing both metrics simultaneously: they want to lower a patient's length of stay without increasing mortality risks.

2 Literature Review

There exists a large literature base for length-of-stay and mortality predictions. A majority of length-of-stay articles build models for specific types of patients. Carter and Potts [2] develop a model for predicting the length of stay of patients who receive primary total knee replacements. Tsai et al. [5] study the length of stay of patients who receive one of three cardiovascular diagnoses. Walczak et al. [6] study trauma patients and patients with acute pancreatitis. These models inherently have limited scope of application.

Other authors developed more general models. For example, Gentimis et al. achieved 80% accuracy with a binary length-of-stay classifier that predicted,

for any type of patient, whether they stayed less or more than five days [8]. Harutyunyan et al. built a general length-of-stay regression model with 110 hour mean absolute difference, as well as a mortality predictor with an accuracy of 91% [3]. Both papers use neural nets.

In the past, the literature base relied on coarse classification models. Tsai at al. suggested that, for the sake of practicality, researchers categorize patients into risk groups instead of predicting their actual length-of-stay [5]. This is no longer necessary with the release of large datasets like MIMIC-III, which provide enough data for regression models to be practical. This is great news since regression models offer more information than coarse classifications.

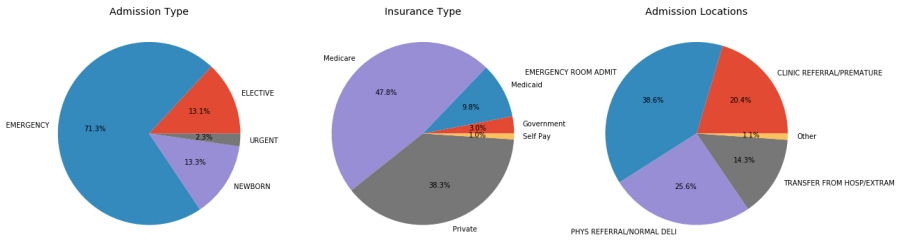
Our aim is to contribute to the emerging literature that uses neural nets to predict length-of-stay and in-hospital mortality. Our model applies generally, no matter the diagnosis. Moreover, we treat length-of-stay as a regression problem, which leads to more informative results. Finally, our model is the first to incorporate Natural Language Processing by learning from the notes written by doctors and nurses.

3 Data

We acquired our data from MIMIC-III data base [12] [13].

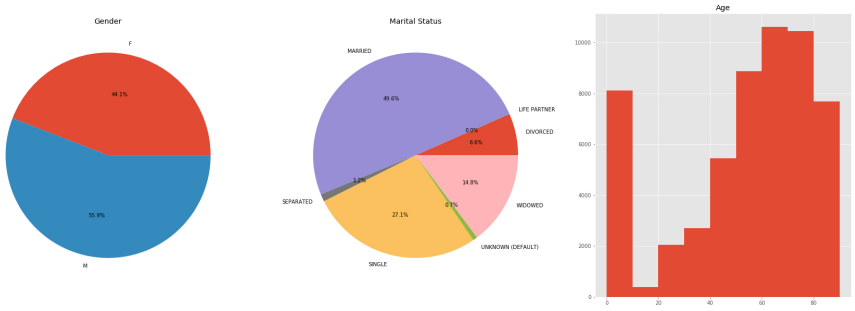
MIMIC-III's data was collected over the span of a decade from critical care units at the Beth Israel Deaconess Medical Center in Boston, MA. It describes 53,423 adult admissions from 2001 through 2012, as well as the admissions of 7870 newborns from 2001 through 2008 (Pollard and Johnson 2016).

MIMIC-III has data on patient demographics, doctor's notes, imaging notes, mortality data, diagnoses, and medications taken. Plus, for each admission, there are, on average, 4579 data points on vital signs and 380 data points from lab reports (Pollard and Johnson 2016). This provides ample data for a neural network to learn from.

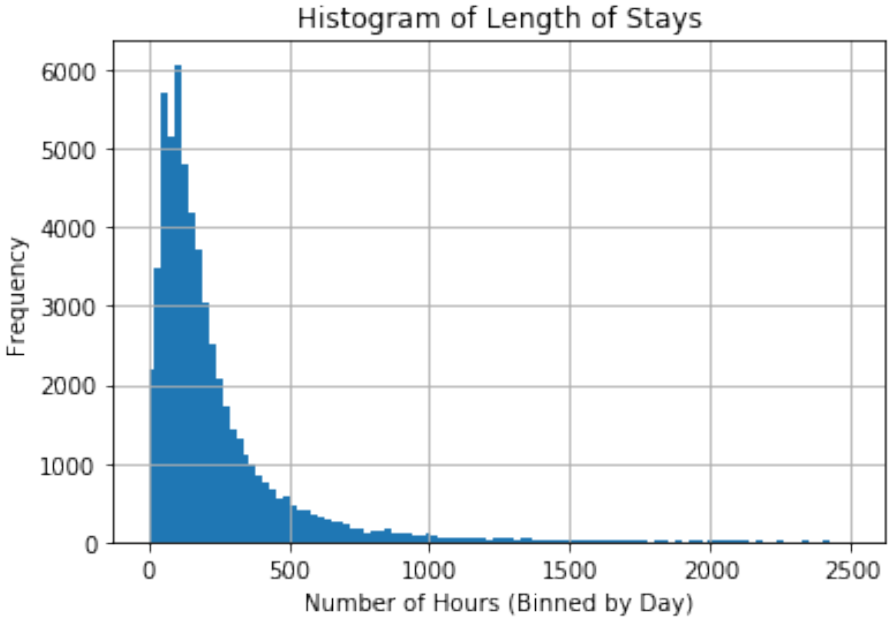


Plotted above and below is a selected subset of basic information available immediately after a patient's admission to hospital. As shown, MIMIC-III patients are comprised of roughly equal portions of male and female, and nearly half are married. Age 60 to 80 is found to be the most frequent age group that visit the

hospital; note that since the total admission table is counted by individual visits and not patients' head counts, the Age histogram demonstrate the relative visit counts of different age groups. Also noteworthy is that, in the same histogram, age 0 to 10, which also includes newborn babies, takes up a much larger portion of visits than age 10 to 20.



All length-of-stays are plotted in the histogram below. There is a clear right-skew to the graph. This implies there are some exceptional patients with unique circumstances that lead to long stays. In fact, in order to visualize the bulk of the distribution, the histogram does not include the 0.2% largest values. The longest stay is almost 10 months long.



4 Data Preprocessing

To process the data, we first removed all entries where a person was reported to have a negative length of stay, which is, of course, logically impossible. (In fact, these entries are intended to indicate that the person died prior to arriving to the hospital.)

We then addressed the categorical variables in our dataset. For encoding variables like diagnoses, we simply created dummy variables. Other categorical variables were harder to encode because they gave semantic meaning to numerical variables stored in other columns. For example, the chart events table details a patient's vital signs by having one column indicate the type of measurement taken - a categorical variable - and another column report the measurement recorded - a numerical variable. To process this table and others like it, we added one column per categorical variable and let the cells under that column be the measurement recorded for that type of measurement. If no such type of measurement was recorded for the admission at the time, we set the cell to zero.

All data was aggregated on a by-day basis so as to reduce the space required for storage. Though our regression model is trained only on data from the first day, the data we processed - save the BioBERT word embeddings - could inform a time-series analysis as well.

Space complexity was a real challenge. The chart events table alone took up 30 gigabytes. To make the task more manageable, we used parallelization and pkl files.

5 Feature Engineering

5.1 BioBERT Embedding

The table NOTEEVENT.csv contains an extensive collection of notes written by doctors and nurses. For 52,976 hospital visits, there are a total of 2,083,180 pieces of texts: each patient can have multiple visits, each visit could span multiple days, and each day might have multiple notes. We dropped variables with no plausible relationship with mortality or length-of-stay. Then we deleted duplicated or erroneous rows. Then we removed all notes categorized as "discharge summary" because a mortality or length-of-stay prediction becomes pointless after the discharge summary is written.

This yields a smaller table with columns identifying the patient ID, admission ID, a timestamp, and the text written at that time. The table has 96,955 rows and is 2GB large. Furthermore, we found from the original notes that they can be very long (more than 20,000 characters), formatted in a messy manner, and contain numeric values. Thus, to both extract continuous information and keep the data reasonably large, we group the notes by visits, sorted by dates, and use the below pre-processing methods to keep the "most informative" note segment of a given length (512 characters in practice):

1. Remove all numbers, which mostly record patient's examination results and measurements, and are contained throughout other tables in the dataset.

2. Remove all irrelevant special characters: e.g. throughout the notes, the patients' and doctors' names are replaced with "[***last name***]"; another example is long consecutive "_ _ _ _"'s used to section the notes.

3. Because the notes are often randomly split with new lines, and often "\n" appears in the middle of a sentence, instead of finding individual lines, we first merge all lines, then use str.split(': ') to find sentences after ":". Heuristically, we found these lines had the most valuable information not captured by data in other tables.

4. Other housekeeping routines such as merging repeated blank spaces.

As a result, we have a collection of much shorter, information-condensed notes. Here is an example of the effectiveness of this processing:

```
-------Original Note-------
[**2108-4-10**] 6:14 AM
 CHEST (PORTABLE AP)                                          Clip # [**Clip Number (Radiology) 33133**]
 Reason: Please assess for infiltrates.
 _____________________________________________________________________________
 [**Hospital 3**] MEDICAL CONDITION:
    48 year old woman with hx of asthma multiple myeloma, pulmonary embolism
  comes  in s/p URI and now worsening sob. Bibasilar crackles.
 REASON FOR THIS EXAMINATION:
  Please assess for infiltrates.
 _____________________________________________________________________________
                                 FINAL REPORT
 CLINICAL INDICATION:  48 year old female with history of asthma, now with
 worsening dyspnea.

 TECHNIQUE:  Portable AP chest.

 COMPARISON:  [**2108-4-6**].

 FINDINGS:  The cardiac and mediastinal contours appear stable.  There is no
 pulmonary vascular engorgement.  The lungs appear clear, with no confluent
 areas of opacification.  There are no pleural effusions.  The visualized
 osseous structures and soft tissues are unremarkable.

 IMPRESSION:  No acute cardiopulmonary abnormalities.

-------Our Preprocessed Note-------
The cardiac and mediastinal contours appear stable. There is no pulmonary vascular engorgement. The lungs appear
clear, with no confluent areas of opacification. There are no pleural effusions. The visualized osseous structures
and soft tissues are unremarkable. FINAL REPORT CLINICAL INDICATION Please assess for infiltrates. Hospital MEDICAL
CONDITIONAM CHEST (PORTABLE AP)ClipClip Number (Radiology)Reason No acute cardiopulmonary abnormalities.
```

Then, we use the pre-trained BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) model to obtain vector representations of the processed notes. Biomedical text mining is becoming extracting valuable information from biomedical literature BioBert bases itself off BERT (Bidirectional Encoder Representations from Transformers) [11], the recent contextualized language representation model. BioBert specifically uses a large number of unannotated biomedical text corpora for pre-training, effectively transferring the knowledge from a large amount of biomedical texts to a variety of biomedical text mining models with minimal task-specific architecture modifications.

It took an unexpected amount of time to configure the pre-trained models from BioBERT since, as we later discovered after many attempts, the scripts rely on a very specific set of dependencies including CUDA 9.0, cuDNN 7.0, tensorflow-gpu 1.11. This required careful setup on the EC2 instance.

Generating a word embedding for each note takes more than ten seconds, and our EC2 instances were not reliable enough, breaking off too often from

connection, to finish all 96,955 notes. Even with a perfect connection to the EC2 instance, processing all these notes would take 11 days. As a result, we reduced the scope of our task by only generating word embeddings for the first note of the first day for each individual stay. (This is why we settled for a regression model that only made first-day predictions in lieu of a time-series model.)

5.2 Admit Time

We applied a trig transformation to admission time by taking the cosine and sine of the time of the day a patient was admitted. On our training set, both the cosine and sine transformations were found to be statistically significant predictors of mortality at a .05 confidence level.

5.3 Age

In order to protect privacy, MIMIC-III added noise to date of birth and admission time. To extract a person's age, we simply subtracted the admission time from their date of birth.

For patients older than 89, the dataset sets their date of birth to be 300 years prior to their first admission time. Because the true median age for these patients is 92, we set these ages to 92 (Pollard and Johnson 2016).

Age is a strong predictor of mortality, with a Pearson coefficient of 0.30. Age is also a statistically significant predictor of length-of-stay and mortality, far below any reasonable the cut-off for any reasonable alpha level.

5.4 Diagnosis

MIMIC-III lists the diagnoses given to patients. There are 15,000 diagnosis codes, and many turn out to be quite related. For example, dozens of codes all denote various types of pneumonia. Converting each to a distinct dummy variable would squander this information. Thus, we coarsened the 15,000 diagnosis codes into around 400 families of related diagnoses, using a categorization scheme developed by the Healthcare Cost and Utilization Project.

5.5 Microbiology Events

The table MICROBIOLOGYEVENTS.csv contains microbiology information, including tests performed and sensitivities. The two main categorical variables are specimens, which is tested for bacterial growth, and antibiotics, tested against a given organism for sensitivity. 72 different types of specimen are recorded on all 631,726 rows, including blood culture, urine, and tissue. Thirty types of antibiotics are tested, but on 275,834 rows; this means about two-thirds of the specimen tested on not get antibiotic check. Therefore we choose to do one-hot encoding on the specimen types, but future projects should consider including the antibiotics variable into account for completeness.

We also noticed that each visit can require multiple testings of the same specimen, and even one day during a several-day visit may require multiple testings (which explains why size of the table outnumbers the total visit count). Hence we group all testing information by day, and use pandas.sum() to stack the one-hot encoding rows and reduce the total row numbers to 172,989.

Similar to other tables, to extract first-day information, we again select the row in each HADM-ID group that corresponds to the first day, and as a result further reduce down the data to 48,740 rows. This is smaller in size than 58,976 total visits because some patients might not require a specimen check at all, and for those visits we fill in corresponding rows with zeros.

5.6 Services

The SERVICES.csv table describes the service that a patient was admitted for. Intuitively, the type of service a patient is receiving should be correlated with their length-of-stay: for example, patients admitted under a surgical service should expect to stay longer than patients getting a service for their ears.

The original table contains 73,343 rows, which is bigger than total visit count because a patient might get transferred to another service during their stay, and the information of their previous service is recorded in a separate column. Similar to the microbiology table, we do one-hot encoding on the 19 different types of services. There could be a corner case where a patient gets transferred between services within one day, but in general this does not happen. Therefore we dropped the "PREV-SERVICE" column and specify service at a day-to-day basis.

Again, we further reduce the table for first-day information, which fortunately results in exactly 52,976 rows since every visit gets categorized into one service.

5.7 Prescription

PRESCRIPTIONS.csv contains all information about the medication orders for each patient, and each of the 4,156,450 rows corresponds to a single drug prescription. (Most patients are prescribed multiple drugs.) After dropping columns that provide additional drug representations and cleaning up repeated rows, we cut down the table to 3,440,200 rows. At first we wanted to do one-hot encoding on drug names or types, but 3,267 different drugs were counted, and there isn't an appropriate categorizing available like the services table above.

A closer look at the table suggests an alternative approach: the number of different types of prescribed medication tends to be larger at the beginning of a visit, or for patients with more severe conditions. For example, during the number "100001" visit, the patient takes 18 different types of medications on day one, and 18 types on day two, but then 10 types on day three, 8 on day four, 5 on day five, only 1 on day six, and 2 types on the last day. It also

makes intuitive sense that the number of medication types is correlated with the remaining length of stay.

Therefore, we decided to generate a new table that records drug-counts, instead of drug names. Because the original table also specifies the exact start date and end date for each drug, we are able to construct a table that records not just the number of drugs prescribed, but also the number of drugs currently taken by the patient at each day. But again, for first day information, the prior should suffice, and as a result we have a much smaller table with 52,976 rows.

6 Models

6.1 Feature Selection

For both our mortality classifier and length-of-stay regression models, we used the same feature vector, which we have described above.

We should also note a deliberate omission in our feature vector: demographic information. While this information is quite useful for predictions, there is a practical reason for not including these variables into our feature vector. Suppose people of different races have different health outcomes. These outcomes may differ because the groups are treated differently in the hospital. If we include this demographic information into our model, we may end up reinforcing this bias. Admittedly, omitting these features does hamstring our model slightly. (Previously, our MLP model attained an RMSE of 19 hours using this information.)

When constructing our feature vector, we tried to include nearly all variables that could plausibly explain length-of-stay or mortality risks. As a result, our feature vector is quite large and probably contains dozens of unnecessary features. To address this undue complexity and likely source of collinearity, we use LASSO as a form of automatic feature selection prior to training any model. We use cross-validation to tune the hyper-parameter for LASSO.

We did not, however, use LASSO to eliminate any variable associated with the BioBERT word embedding because we suspected that interactions will be crucial to extract insight from these word embeddings.

Ultimately, this process reduced the number of non-BioBERT features from 437 to 120 when predicting length-of-stay, and from 437 to 285 when predicting mortality.

6.2 Metrics

We evaluated our regression models using RMSE and an accuracy measure with a set threshold. This accuracy was determined by the difference of the prediction and labels. If the difference was within the threshold 15% of the actual label, this prediction was classified as accurate. We thought it made more sense to have the margin-of-error scale up as the LOS increases than to have one similar to the

two-day margin of Carter and Potts [2]. We include this accuracy metric so that our results are roughly comparable with the metrics used by other authors in the literature base. We include the RMSE metric because it is a highly intuitive metric for evaluating a regression model.

6.3 Linear Model for Length-of-Stay

To create a baseline model for length-of-stay predictions, we ran a simple linear regression. After normalizing the data, we found an R-squared coefficient of .45 on the training data and a RMSE of 219.7 hours. The performance on the test set was lousy, with an RMSE of 110,779 hours. Using our threshold accuracy metric, the model had 17.0% and 15.7% accuracy on the training and test sets respectively. By examining the predictions made, we discovered the model was predicting extremely large negative values for hundreds of observations. We could avoid this with a log-transformation of the response, but in order to have an apples-to-apples comparison, this would force us to do the same with other models; we did not want to make our results less interpretable by doing this.

This large test RMSE resulted from overfitting, so we added regularization by penalizing the L2 norm, using cross-validation to find the optimal penalty weight. This slightly lowered the R-squared coefficient to .44 on the training set and increased the RMSE to 222.4 hours. Using our threshold accuracy metric, the model had decreased slightly to 16.4% and 15.9% accuracy on the training and test sets respectively. But it drastically improved performance on the test set to 231.4 hours. This is still not great performance, but it is close to the training RMSE, suggesting that the overfitting problem has been addressed.

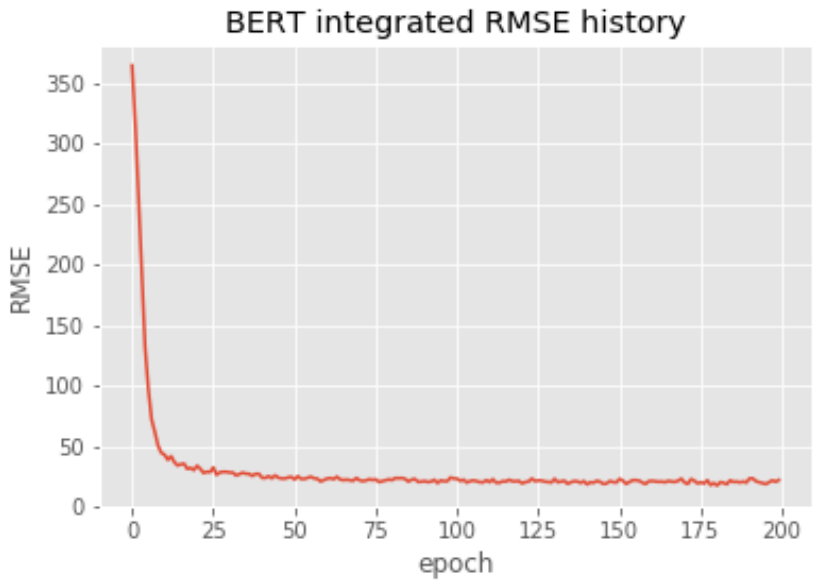
6.4 Logistic Regression Model for Mortality

Our baseline model for mortality predictions is a logistic regression model. After using cross-validation to tune the hyperparameter, our model yielded a training accuracy of 74.6% and a test accuracy of 74.3%.

Given that 74% of people survive, this baseline model performs about as badly as a model that only ever predicts survival. This is not inspiring performance.

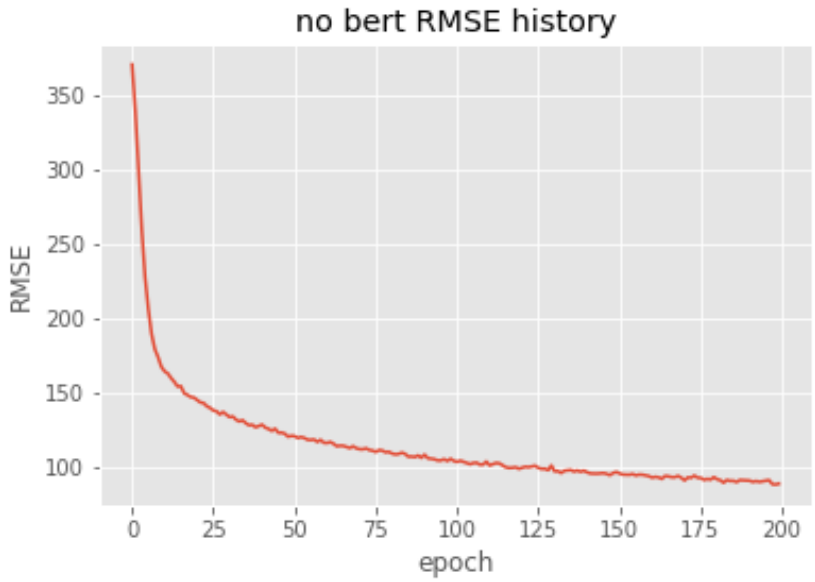
6.5 Multilayer Perceptron (MLP)

After tuning parameters with cross-validation, our final multi-layer perceptron has two fully connected hidden layers with 256 and 128 neurons, respectively. The input dimensions are 47102 observations with 890 variables. The model uses the Adam optimizer with learning rate 0.00001, a batch size of 256 and 200 epochs. Additonally, batch normalization has been added between each of the hidden layers. All the variables selected from Lasso were merged with the BioBERT word embeddings and 20% of the observations were set aside for testing.



We reached a test accuracy of 90.438% and a train accuracy 90.631% with a RMSE 19.762 hours.

Below we plot the RMSE results from our model trained without the BioBERT word embedding.



This model performs remarkably worse than the model where the BioBERT

word embeddings are included into the feature vector. In fact, this new model scores only a 16.355% test accuracy and 36.686% training accuracy. The disparity between test and training performance is much larger than that of the better model with doctor's notes. Also the RMSE is 89.534 hours, which is about 4.5 times greater than the RMSE from the model with the BioBERT word embeddings. This confirms our hypothesis that notes from doctors and nurses contain information that significantly help predict a patient's length of stay.

7 Conclusion

We found a remarkably large difference in performance between our baseline models and MLP models.

For predicting length-of-stay, our baseline linear model with the L2 penalty had a 231.4 hour RMSE and 15.9% threshold accuracy on the test set. In contrast, our MLP model had an RMSE of 19.76 hours and a 90.438% threshold accuracy. We also found that our BioBERT word embedding was crucial to this success, and improved our threshold accuracy by 79% and reduced our RMSE by 70 hours.

For predicting mortality, our baseline logistic regression model had a 74.3% accuracy, which is barely greater than the proportion of survivals. In contrast, our MLP classifier had an accuracy of 78.7%. Considering that 74% of people do not die, these results are not impressive.

However, our length-of-stay results are among the best results obtained in the literature base. For models restricted to specific subgroups of patients, models are typically accurate 60% to 80% of the time, when given a margin of error of two days. Among models that study all diagnoses, our length-of-stay predictions fare even better. Careful feature engineering, in our view, made all the difference.

Further research still remains to be done. For research teams with access to vast amounts of compute, we recommend that they use BioBERT to generate word embeddings from every note, not just those from the first day. This would enable a time-series analysis that could yield real-time predictions during a patient's stay. After all, if the dataset includes information that a surgery ended with a complication, then it would be a good opportunity to update the length-of-stay prediction to be longer. Researchers could then train a Recurrent Neural Net on this time-series data, and supply hospitals with better predictions. Furthermore, we would advise future researchers to use survival analysis when constructing such a time-series model.

Furthermore, we think future researchers could explore the benefits of treating the tasks of length-of-stay prediction and mortality prediction as a combined task. They could approach this as a classification problem. Suppose you bucketed all length-of-stays into n buckets. For example, one bucket may denote a stay of 1 to 2 days. Another may denote a stay of more than two weeks. Then for each bucket, there would be two classes: death or survival. In other words, this classification problem would predict both when a patient will leave the hospital

and why, simultaneously. We suspect this is a useful approach to the problem because many people with a high probability of dying will, realistically, either die quickly or have a long hospital stay to help them recover. A single predicted value from a regression model loses this complexity. Moreover, it's beneficial to predict both mortality and length-of-stay simultaneously because the interaction between the two is meaningful for practitioners. If, for some types of patients and diagnoses, long stays have severe health consequences, that would be important to pick up on. All this said, we expect more data would be needed to identify these kinds of patterns.

References

[1] Barnes, Sean, et al. "Real-Time Prediction of Inpatient Length of Stay for Discharge Prioritization." Journal of the American Medical Informatics Association, vol. 23, no. e1, 2015, doi:10.1093/jamia/ocv106.

[2] Carter, Evelene M and Henry W Potts. "Predicting length of stay from an electronic patient record system: a primary total knee replacement example" *BMC medical informatics and decision making* vol. 14 26. 4 Apr. 2014, doi:10.1186/1472-6947-14-26

[3] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask Learning and Benchmarking with Clinical Time Series Data. arXiv:1703.07771 Dec. 2018, pp. 1–19.

[4] Kulinskaya, Elena, et al. "Length of Stay as a Performance Indicator: Robust Statistical Methodology." IMA Journal of Management Mathematics, vol. 16, no. 4, 2005, pp. 369–381., doi:10.1093/imaman/dpi015

[5] Pei-Fang (Jennifer) Tsai, Po-Chia Chen, Yen-You Chen, et al., "Length of Hospital Stay Prediction at the Admission Stage for Cardiology Patients Using Artificial Neural Network," *Journal of Healthcare Engineering*, vol. 2016, Article ID 7035463, 11 pages, 2016. https://doi.org/10.1155/2016/7035463.

[6] Walczak, Steven, et al. "A Decision Support Tool for Allocating Hospital Bed Resources and Determining Required Acuity of Care." *Decision Support Systems*, vol. 34, no. 4, Mar. 2003, pp. 445–456., doi:10.1016/s0167-9236(02)00071-4.

[7] Resar, Roger et al. "Using Real-Time Demand Capacity Management to Improve Hospitalwide Patient Flow." *The Joint Commission Journal on Quality and Patient Safety*, vol. 37, no. 5, 2011, pp. 217–227., doi:10.1016/s1553-7250(11)37029-8.

[8] Gentimis, Alnaser et al. "Predicting Hospital Length of Stay Using Neural Networks on MIMIC III Data." 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber

Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Orlando, FL, 2017, pp. 1194-1201. doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.191

[9] Silva, Ikaro et al. "Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge 2012." Computing in cardiology vol. 39 (2012): 245-248.

[10] Jinhyuk Lee, et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." Jan, 2019. arXiv:1901.08746

[11] Devlin, Jacob et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Oct. 2019, arXiv:1810.04805

[12] 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: http://www.nature.com/articles/sdata201635

[13] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220, June, 2000. [Circulation Electronic Pages; http://circ.ahajournals.org/content/101/23/e215.full]