Andre Yin
OURSIP Final Research Report
2 August 2019

Predicting Hospital Length of Stay Using Multilayer Perceptron Neural Networks

Abstract

Accurately predicting hospital length of stay (LOS) can ease patients' emotional strain, help optimize hospital operations, and reduce costs. Previous research has used multilayer perceptrons (MLPs) to predict LOS using a binary threshold, but research is lacking in predicting multiclass LOS (i.e. specific day ranges). In addition, current literature lacks systematic testing to determine which combinations of comprehensive patient attributes most strongly correlate with LOS. Using MLPs, this experiment optimizes parameters to predict binary and multiclass LOS, then assesses the relative importance of patient attributes in determining LOS. The first phase of this project consisted of pre-processing data from the Medical Information Mart for Intensive Care (MIMIC-III) clinical database to select and format a group of patient attributes to train the MLP. The second phase entailed training and testing the MLP while varying neural network parameters to achieve a baseline predictive accuracy. The final phase focused on removing attributes from the training inputs to determine their relative importance based on a decrease in accuracy. The MLP increases hospital length of stay predictive accuracy by up to 100%, with ICU length of stay as the most important attribute.

Introduction

An extra day at the hospital costs $10,400 on average, while hospital stays each year cost the US health system $377.5 billion (HealthCatalyst, n.d.). For patients and their families, knowing a substantiated estimate of their length of stay may improve their mental state and ease their anxiety, potentially leading to better health outcomes. As more emphasis is placed on value-based care, nurses and doctors will be better able to provide such care in light of how long a patient will likely stay at the hospital. Predicting hospital length (LOS) is of substantial value to optimizing hospital operations, including resource planning and allocation. For example, administrative staff can more efficiently allocate wards and bedspace if they can better predict the flow of incoming and outgoing patients. Thus, they may have a better idea for how many patients they can accommodate and when. At the same time, an accurate prediction may relieve financial burden off payers who will have a better estimate in advance of the healthcare costs.

A crucial first step is to focus on LOS prediction accuracy without much consideration for what patient attributes are used to build the model. However, it is important to realize that data collection comes with a cost. For instance, certain laboratory tests that may be good indicators for LOS may be financially infeasible for the patients or the hospital. As a result, it is paramount to assess the relative importance of LOS factors, thereby reducing the number of factors needed without compromising high predictive accuracy. Based on current literature, determining which patient attributes contribute most to predicting LOS, without targeting a specific demographic or condition, has not been methodically analyzed.

Especially in the past decade, the shift from paper-based to electronic health records has facilitated research studies on aggregate patient data. Leveraging the

electronic nature of this data in conjunction with powerful digital data analysis tools, researchers are able to analyze trends and make predictions infeasible before. The MIT Lab for Computational Physiology developed the Medical Information Mart for Intensive Care (MIMIC-III), which contains data on more than 20,000 critical care patients admitted to Beth Israel Deaconess Medical Center in Boston (Johnson et al., 2016). Using patient attributes in the MIMIC-III database, this paper trains and tests a neural network model to predict LOS and assess which attributes are most important in determining LOS.

Literature Review

While there exist various models that have worked with MIMIC-III data for LOS prediction, currently there are no systematic models for *general* populations that 1. determine the relative importance of input factors contributing to hospital length of stay and 2. predict multiclass ranges for LOS. In this context, *general* refers to a population not filtered for a specific ethnicity or disease.

MIMIC-III is a single-center database consisting of records and attributes of more than 20,000 patients admitted to critical care units. Johnson et al. introduced MIMIC-III to the public in 2016, surveying its various relational tables including vital signs, lab measurements, observations, codes, survival data, LOS, and more (Johnson et al., 2016).

Due to its comprehensiveness and accessibility, many people have worked with MIMIC-III data to analyze trends and other information that can be extracted. For example, Huang et al. explored how a pre-extended version (MIMIC-II) can be used to analyze the results of laboratory tests, detailing steps to access the database and how to effectively work with its relational structure (Huang, Badrick, & Hu, 2017).

With MIMIC-III's release, researchers focused on two broad predictive fields: 1. disease-related outcomes such as complications and 2. time-based outcomes such as LOS.

In the former category of disease-related outcomes, Mohan analyzed the MIMIC-III database, focusing specifically on patients designated by icd9_code 996, indicating that they experienced complications following certain procedures. Mohan's neural network model, which outputs a binary variable for the existence of such complications, achieved a predictive accuracy of over 80% (Mohan, 2018). Meanwhile, Mao et al. used a machine learning-based sepsis-prediction algorithm known as InSight using just six basic vital signs. They were able to achieve an AUROC curve of 0.92 for the detection of sepsis (Mao et al., 2018).

In the latter category of time-based outcomes, Kelly et al. determined that age, co-morbidity levels, and marital status are associated with LOS (Kelly, Sharp, Dwane, Kelleher, & Comber, 2012). Meanwhile, targeting specific conditions is evident in the work of Wang et al. (acute exacerbation of chronic obstructive pulmonary disease) and that of Hachesu et al. (cardiac problems) (Hachesu, Ahmadi, Alizadeh, & Sadoughi, 2013; Y., K., F.A., S., & T., 2014).

Meanwhile, Gentimis et al. trained a neural network on a broad range of patient data, from admission information and ethnicity to age and primary diagnosis, to predict hospital LOS with 80% accuracy. This experiment differs from previous ones in that it did not select for a particular ethnicity or disease, thus enabling LOS prediction for a broader demographic (Gentimis, Alnaser, Durante, Cook, & Steele, 2018). While this paper achieved remarkable accuracy given its generalized input factors, it did not produce multiclass classification models, and it provided no insight into how important the factors were relative to each other.

The current literature illustrates that a combination of complex factors influence LOS, suggesting the importance to assess the relative importance among these factors in a comprehensive manner.

Methodology

MIMIC-III Database

The first major component of this experiment was to analyze MIMIC-III, which contains hospital records of more than 20,000 patients who had been admitted to critical care units between 2001 and 2012. In accordance with HIPAA regulations, the data has been de-identified and anonymized. MIMIC-III is composed of multiple datasets stored as csv files. Below are representative datasets and brief descriptions:

- **Admissions**: patient registration, healthcare, background, and other personal information
- **Caregivers**: caregiver type, e.g. research nurse (RN), medical doctor (MD), or pharmacist (PharmD)
- **Diagnoses**: information and codes on diagnoses
- **Discharges**: details of patients leaving hospital
- **Prescriptions**: medication information from the hospital computerized hospital entry system
- **Procedures**: details on procedures (and their corresponding codes) carried out on patients during stay
- **Patients**: additional demographic information not in 'Admissions'
- **Patient Notes**: notes and observations recorded by nurses or doctors

A comprehensive overview of attributes in the database is available at Physionet, MIMIC-III's host website ("MIMIC-III Critical Care Database," 2016).

Data Pre-processing Using Excel

MIMIC is a relational database, meaning that patient information is spread across data tables on different files. IDs present in these tables enable cross-linkage and comparison. For example, merging data regarding a patient's admission and gender requires linkage using 'subject id'.

Excluding dataset linkers such as subject id or hospital admission id, the finalized list of input attributes used, with specifications if applicable, is as follows:

1. ***Admission type***
2. ***Admission location***
3. ***ED wait time (admit time – registration time)***
4. ***Insurance***
5. ***Religion***
6. ***Marital status***
7. ***Ethnicity***
8. ***ICU LOS***
9. ***Admission season (divided year into quarters)***
10. ***Admission time of day (divided day into quarters)***
11. ***Gender***
12. ***Age (registration time – date of birth)***
13. ***First care unit***
14. ***Last care unit***
15. ***First ward***
16. ***Last ward***

The output used as validation is **Hospital LOS**, calculated as (discharge time – admit time). For binary classification, outputs < 6.3 days (median) were assigned 0, and outputs >=6.3 days were assigned 1. For X-class classification, outputs were split into X classes based on iso-percentile boundaries, with an equal number of patients in each output class.

The following are rationales for attribute selection: 'hadm_id' (hospital admission ID) and 'subject_id' (uniquely identifies patient) are used as identifiers to appropriately merge data across the relational database. Admit time and discharge time are used to determine season and time of day, which could influence LOS due to potential staffing shortages or peak hour visits. Insurance type may reflect a patient's financial capability, while religion and marital status may reflect a patient's lifestyle. Both the type of care units during ICU stay and ICU stay LOS could reflect severity of a patient's condition, as could admission type or location. Factoring in

gender can account for genetic differences, while DOB is used to determine age.

After the aforementioned attributes were selected, data pre-processing in Excel began. The first step was to merge all relevant attributes into one central excel file through 'subject_id' or 'hadm_id'). Since not all data tables provide complete or accurate information for the same patients, a standard copy/paste or filter would not be sufficient. Rather, Excel's VLOOKUP function was used since it is designed to merge datasets based on a common identifier. Because a single VLOOKUP function was inefficient, a double VLOOKUP was implemented, leveraging binary search without compromising precision. Storing hundreds of thousands of functions that would never be changed was unnecessary and inefficient, so all formulas were converted to raw values with the merging of each attribute. After all input attributes were finalized, the data was cleaned to remove any N/A, VALUE! errors, blanks, incorrect data (e.g. negative LOS values), and other inconsistencies.

A multilayer perceptron only takes quantitative values as inputs. Thus, the last step of pre-processing required converting all categorical values to quantitative values. Here is a representative example:

MARITAL_STATUS:
- Married/life partner → 0
- Widowed → 1
- Single → 2
- Divorced/separated → 3
- Unknown → 4

The mapping shown above was compared to a standard normalized mapping to contrast baseline accuracy.

Multilayer Perceptron Neural Networks

(Artificial) neural networks (NN) are modeled on the neural system of the human body. Just like how the human nervous system is composed of individual neurons highly specialized in what information they process, NN comprise layers of artificial neurons. Inputs in the NN are combined through a system of weighted synapses to produce an output.

Multilayer perceptrons (MLP) are a class of neural networks. In addition to an input layer and output layer of neurons, they may contain hidden layers that help determine the optimal model for specific data. Initially, randomly normalized weights are assigned to each neuron in each layer. Each input neuron intakes one attribute and associates it with the random weight before passing on the value to the next layer. Each neuron of the next layer then calculates a weighted average of all attributes and passes on this value to the next layer. The process repeats until the MLP outputs a prediction. **Fig. 1** shows a representative skeleton of a binary classification MLP.

The MLP learns through iterative two-step back-propagation. The process by which inputs are assigned weights, taken a weighted average of, and passed onward to the next layer is known as feeding forward (1st step). The ultimate goal for an MLP is to minimize an error function, often highly complex and non-linear. Thus, the output produced for one set of attributes is compared to the theoretical output and the total error is calculated. Through a loss function, that error is propagated backward to each layer of the MLP to recalculate the weights (2nd step).

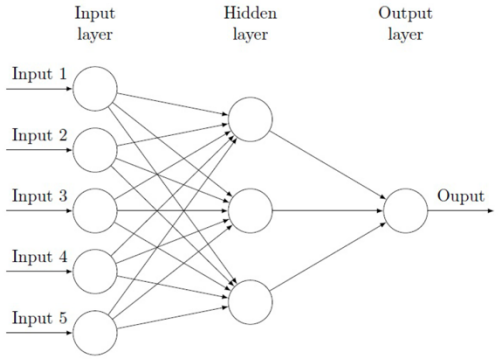

Fig. 1: A typical skeleton of a binary classification multilayer perceptron (MLP) neural network. This experiment uses 16 attributes as baseline input and X neurons as output, determined by X-class classification. Relative importance assessment was conducted by stripping away combinations of input factors.

MLP Training and Testing

The pre-processed spreadsheet was loaded into a Jupyter Notebook file. Initial exploration of the MLP borrowed python code from Medium.com to facilitate understanding underlying structure at a fundamental level (Spencer-Harper, 2015). Two key takeaway points from this initial model are the usage of the log loss function and the need for batch randomization. The steep gradient of the log loss function for values (errors) close to 0 makes it better at differentiating between similar inputs. Meanwhile, batch randomization lessens the likelihood of overfitting to the data.

Eventually, the basic skeleton was replaced with a model via the Python Keras machine learning package. Keras provides a flexible user interface that enables convenient manipulation of parameters in building and testing the model.

A k-fold cross-validation procedure was used to train and test predictive accuracy. This procedure is commonly preferred in machine learning applications due to less bias than other methods, such as the simple train/test split. Following is the k-fold cross-validation procedure:

1. Shuffle the dataset randomly.
2. Split into k groups.
3. For each group—
    a. Use as the validation set.
    b. Train NN on remaining groups.
    c. Fit a model on training set.
    d. Evaluate model on this group.
    e. Retain performance and discard model.
4. Summarize the overall performance of model using the k different scores.

Findings

Nomenclature

The following nomenclature is necessary to understand the underlying parameters of an MLP:

- **Neuron:** a node in a neural network that combines weighted input(s) into an output
- **Layer:** a group of neurons all at the same level of an MLP hierarchy
- **Kernel initializer:** initial random distribution of weights
- **Activation function:** function that takes in weighted inputs and combines them to form an output
- **Loss function:** function used to determine error and how it is back-propagated to redefine weights
- **Optimizer:** a set of conditions that characterizes how the MLP learns, including learning rate
- **Epoch:** one pass of the entire training set through the MLP
- **Batch size:** how many data points used to train the MLP at one time (batch size * # batches = 1 epoch)
- **Splits:** how many times the entire dataset is split for high-efficiency training and cross-validation

Because the LOS data is skewed to the right (**Fig. 2**), the median of 6.3 days is used as the binary threshold instead of the mean. Meanwhile, for X-Class classification, thresholds were established based on iso-percentile partitions into X equal-size classes. For a specified set of parameters (defined above), the two main metrics used to determine performance are the following:

- **Mean accuracy:** average frequency of correct predictions, where ‘correct’ means MLP predicts the same class as a patient’s actual class
- **Standard deviation:** average std. dev. from the mean accuracy

Establishing Binary Baseline Accuracy

The MLP achieved the highest consistent binary accuracy of 71% using the following parameters:

- **Attributes**: Admit time of day, Admit season, ED wait time, Admission type,

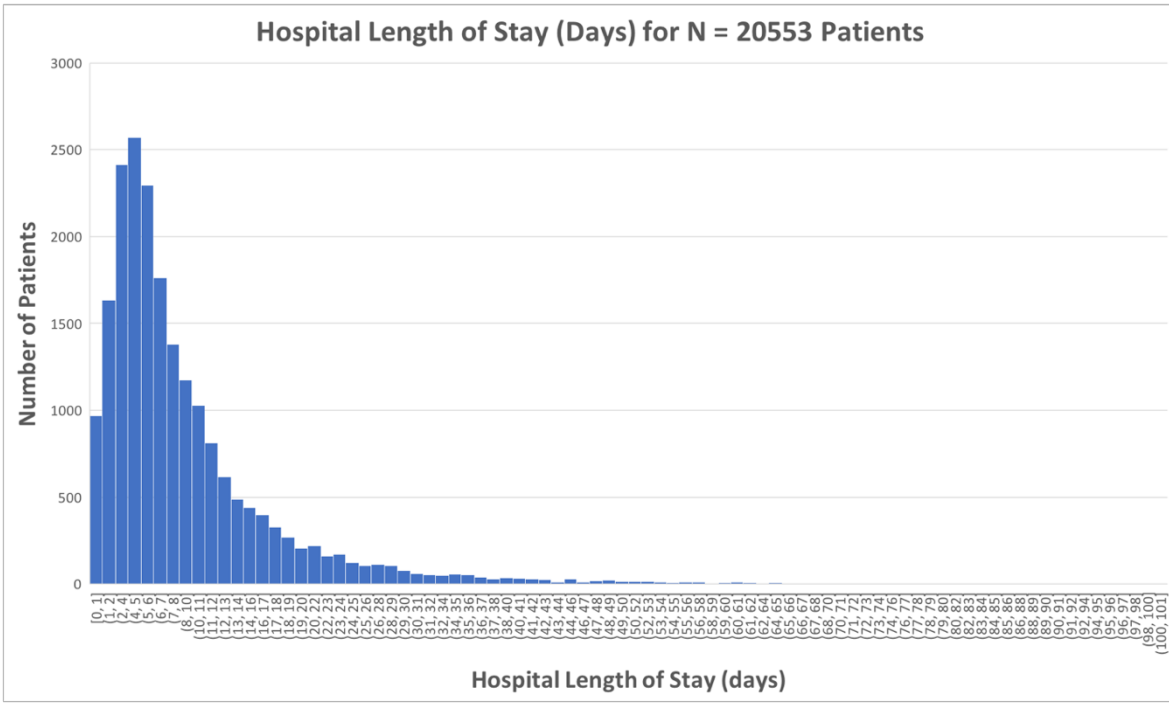

Fig. 2: Hospital length of stay (LOS) is heavily skewed to the right. Thus median (6.3 days) is used instead of mean (9.3 days) as threshold for binary classification. Using the median ensures that the neural network's predictive accuracy can be fairly compared to the 50% guessing baseline since there are an equal number of patients below and above the threshold.

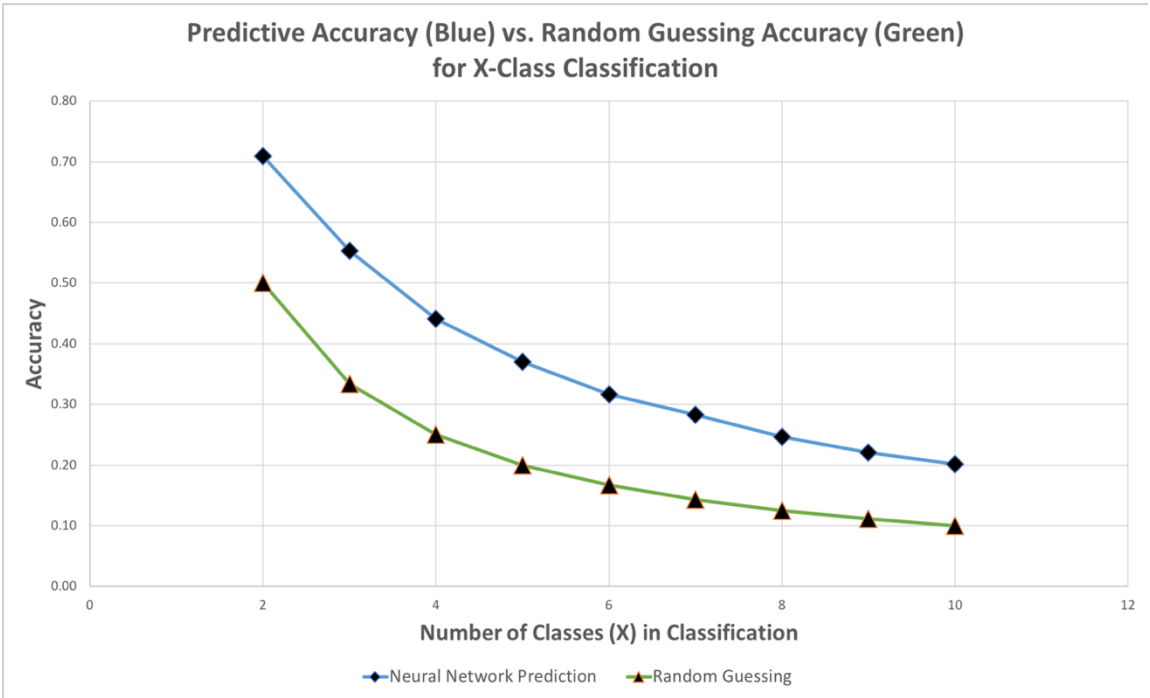

Fig. 3: MLP model prediction rate is 40% to 100% higher than random guessing baseline. ~0.004 std. dev for each class. As the number of classes for classification increases, so does the improvement from random guessing to model prediction. Random guessing accuracies are equal to 1/X (X = number of classes) since thresholds were determined by iso-percentile boundaries.

Admission Location, Insurance, Religion, Marital status, Ethnicity, Gender, Age, First care unit, Last care unit, First ward ID, Last ward ID, ICU LOS

- **Validation**: Hospital LOS, where a 0 means actual LOS<6.3 days (median), and a 1 means actual LOS>=6.3 days
- **Model**: Sample size = 20553, 16 input neurons, 0 hidden layers, normal kernel initializers, relu activation for input layer, sigmoid activation for output layer, binary cross entropy (log loss) function, adam optimizer, epochs = 100, batch size = 1250, splits = 5

Establishing X-Class Baseline Accuracy

Optimal X-class accuracy was established using the following parameters: 16 input factors using 'relu' loss, X output nodes using 'softmax' loss, 0 hidden layers, nsplits=3, epochs=100, batch_size=1250). The MLP multiclass models increased LOS predictive accuracy by 40% to 100%, compared to random guessing (**Fig. 3**).

Testing Different Optimizers

Although the 'adam' optimizer was used to determine baseline accuracy, there was no significant difference in performance when any of these optimizers was used instead: 'rmsprop', 'adagrad', 'adadelta', 'adamax', or 'nadam'.

Optimizing Sample Size

Immediately after any excel data is imported into python, the data is completely randomized and re-indexed to minimize any biases due to ordering of the original dataset. A sample size of N was selected from the first N rows of the randomized data table. Sample sizes range from 100 to 20553 (entire table). For consistency, batch size is 1/10 of sample size and the number of epochs remains constant at 100. Four configurations of MLPs were tested for each sample size:

1. Baseline (non-standardized, but all inputs are still between 0 and 1)
2. Normalized (all input attributes are scaled to have a distribution of mean = 0 and std dev = 1)
3. Half Input (half as many (8) neurons are used as input attributes (16) in the input layer)
4. Extra Hidden (include a hidden layer with half as many neurons as attributes)

The following results hold for each of the four configurations: Above the threshold N=500, average accuracy does not increase. This result bodes well for increasing efficiency due to a relatively small sample size. Average binary accuracy remains stable in the 70-71% range. Above the threshold N=1250, average standard deviation stabilizes in the 1-2% range. For the sake of accuracy, efficiency, and minimal variation, N=1250 appears to be ideal.

Optimizing Batch Size

In machine learning, data is often split into batches to train to improve efficiency. For consistency, N=20000 and the normalized configuration were used for all batch sizes ranging from 156 to 20000 by powers of 2. Batch sizes ranging from 312 to 5000 delivered mean accuracies in the 71% range. Meanwhile, mean accuracy drops considerably as batch size increases to 20000 (65%). Average standard deviation remains stable in the 1% range across all batch sizes. Based on these results, batch size = 1250 appears to be ideal, producing the highest accuracy without compromising efficiency.

Optimizing Number of Epochs

Using epochs ≥30 resulted in no significant different in predictive accuracy. In light of this threshold, and in consideration of standard practice, 100 epochs was used to establish baseline predictive accuracy.

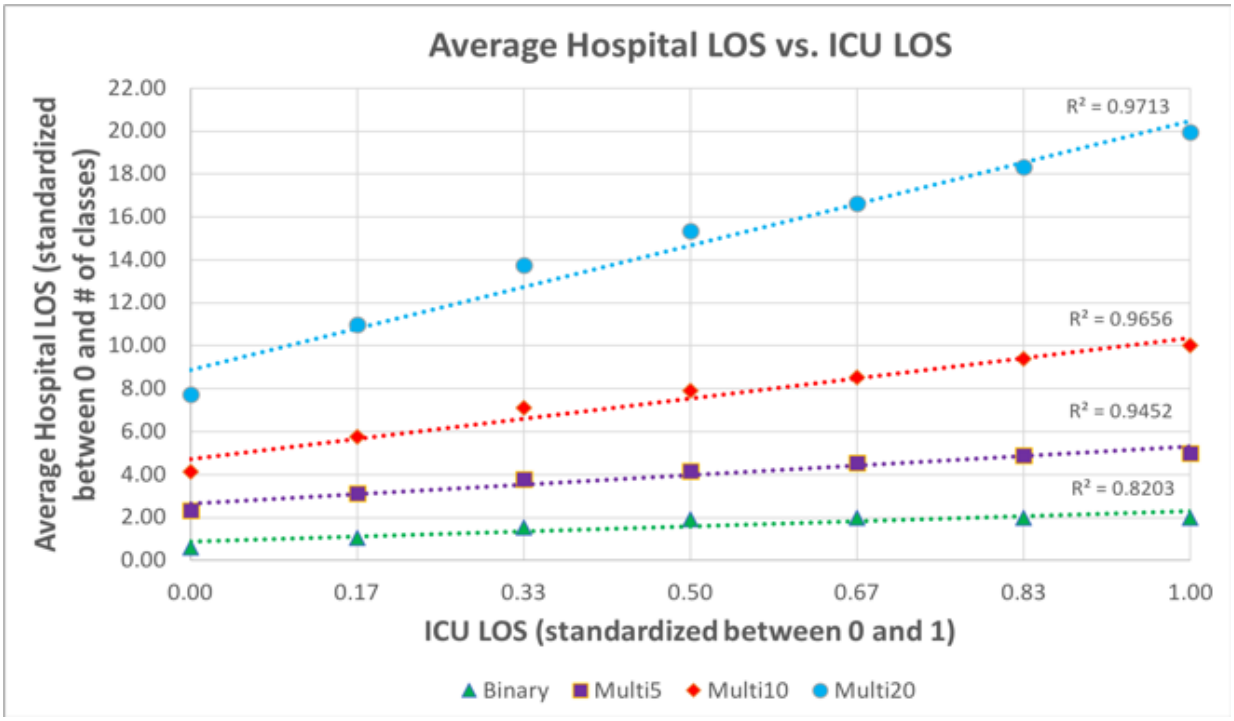

Fig. 4: Strong positive correlation between ICU LOS and hospital LOS. The highest classification of ICU LOS (1.00) corresponds with the highest average hospital LOS for each class (binary, 5, 10, and 20). These trends indicate that ICU LOS per se is a good indicator of hospital LOS. A rationale for the strong correlation is that ICU LOS reflects the severity of a patient's condition, which in turn affects the duration of their treatment in the hospital.

Optimizing Number of Neurons

A single input layer and single output layer with no hidden layers configuration is used to test the effect of # input neurons on predictive accuracy. Parameters held constant include N=20553, a normalized configuration, N/10 batch size, and 100 epochs. Using 3 or more input neurons, the MLP predicted stably at ~71% accuracy with ~1% standard deviation. This result bodes well for increasing efficiency, especially since the number of input neurons needed (3) is significantly less than the number of input attributes (16).

Next, a hidden layer was added to test the performance of MLP with various combinations of # of input and hidden layer neurons. A total of 25 combinations were tested (1,2,4,8,16 for each layer). Using more than a (2 input, 4 hidden) neuron combination (~71% average accuracy) did not significantly improve average accuracy. Standard deviation, as in previous cases, stayed in the 1% range. It is important to note that both the (2,4) and (8,1) combinations produced ~71% average accuracy, which suggests that a hidden layer is not needed.

Relative Importance of Attributes

An initial round of relative importance testing was conducted by removing each of the 16 attributes one at a time to see which removal resulted in the largest decrease in accuracy. Removing ICU length of stay as an input variable resulted in the sharpest decrease in accuracy of 39%, compared to a ~0-3% decrease upon removing any other factor. These results strongly indicate that ICU LOS is by far the most important attribute for predicting length of stay. Indeed, the graph of hospital LOS vs. ICU LOS exhibits strong positive correlation (**Fig. 4**).

Upon first glance at patient attributes, it is intuitive that patient diagnoses—

especially the primary diagnosis—would have a significant effect on LOS. However, adding in 'diagnosis' as the 17th attribute did not improve baseline accuracy. In fact, accuracy even decreased to ~68%. This decrease in accuracy is counterintuitive since disease is expected to heavily correlate with LOS, but a plausible explanation for lowered accuracy is the significantly smaller sample size (1205 with diagnosis info vs 20553 total).

Fig. 5 provides a visualization of the relative importance of patient attributes in predicting hospital LOS.

Fig. 5: Relative importance of patient attributes weighed by size and color.

Conclusion

Key Take-Away Points

X-class classification models increase predictive accuracy by up to 100% compared to random guessing. The MLP predicts binary classification with ~70% accuracy. The most important patient attribute in determining hospital LOS is ICU LOS.

Future Work

Since this project could not possibly test all combinations of input factors, a natural extension of this project is to continue forming various plausible combinations of input attributes, removing them from the training set, and analyzing the change in predictive accuracy. In this manner, relative importance among more variables can be flushed out, aiding the effort to predict LOS while increasing efficiency and saving costs due to data collection.

One interesting question concerns how well an NN will perform when given individual variables vs. conjoined variables (i.e. if the individual variables are operated upon linearly, quadratically, etc.). For for the linear case, theoretically there should be no difference in performance since MLPs are designed to work with linear combinations of inputs. However, what if the combinations are formed non-linearly? And what implications do conjoined variables have on efficiency?

Ultimately, more work needs to be done on analyzing the reasoning behind the relative importance of various attributes. When researchers or hospitals need to explain why they feed certain attributes but not others into a neural network or any machine learning tool, they need to explain their rationale in human terms. This research may need to rely on the power of not only intuition and logic but also other fields such as biology, psychology, anthropology, and sociology.

Acknowledgements

Foremost, thank you to David Ramirez and Alex Dytso for being my close mentors on this project. Alex, I am honored to have taken your ORF 245 class on statistics, helping me solidify my logical thinking and analytical abilities. David, thank you for your tailored guidance on machine learning and for helping me answer questions from the fundamental to the nitty gritty. Professor Vincent Poor, thank you for inviting me into your lab and allowing me to extend your lab group's research interests and directions. And thank you to the Office of Undergraduate Research for giving me the opportunity to do summer research with a stipend. A final thank goes to God, my parents, and all the buildings and secret places where I found the motivation to do productive research work.

References

Gentimis, T., Alnaser, A. J., Durante, A., Cook, K., & Steele, R. (2018). Predicting hospital length of stay using neural networks on MIMIC III data. *Proceedings - 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 2017 IEEE 15th International Conference on Pervasive Intelligence and Computing, 2017 IEEE 3rd International Conference on Big Data Intelligence and Compu*, *2018-Janua*, 1194–1201. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.191

Hachesu, P. R., Ahmadi, M., Alizadeh, S., & Sadoughi, F. (2013). Use of data mining techniques to determine and predict length of stay of cardiac patients. *Healthcare Informatics Research*, *19*(2), 121–129. https://doi.org/10.4258/hir.2013.19.2.121

HealthCatalyst. (n.d.). Patient-Centered LOS Reduction Initiative Improves Outcomes, Saves Costs. https://doi.org/10.1161/CIRCHEARTFAILURE.112.000265

Huang, Y.-L., Badrick, T., & Hu, Z.-D. (2017). Using freely accessible databases for laboratory medicine research: experience with MIMIC database. *Journal of Laboratory and Precision Medicine*, *2*, 31–31. https://doi.org/10.21037/jlpm.2017.06.06

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L.-W. H., Feng, M., Ghassemi, M., … Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*, 160035. https://doi.org/10.1038/sdata.2016.35

Kelly, M., Sharp, L., Dwane, F., Kelleher, T., & Comber, H. (2012). Factors predicting hospital length-of-stay and readmission after colorectal resection: A population-based study of elective and emergency admissions. *BMC Health Services Research*, *12*(1). https://doi.org/10.1186/1472-6963-12-77

Mao, Q., Jay, M., Hoffman, J. L., Calvert, J., Barton, C., Shimabukuro, D., … Das, R. (2018). Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and ICU. *BMJ Open*, *8*(1), 1–11. https://doi.org/10.1136/bmjopen-2017-017833

Milo Spencer-Harper. (2015). How to Build a Multi-layered Neural Network in Python. Retrieved July 4, 2019, from https://medium.com/technology-invention-and-more/how-to-build-a-multi-layered-neural-network-in-python-53ec3d1d326a

MIMIC-III Critical Care Database. (2016). Retrieved July 4, 2019, from https://mimic.physionet.org/about/mimic/

Mohan, N. (2018). *Predicting Post-Procedural Complications Using Neural Networks on MIMIC-III Data*.

Y., W., K., S., F.A., D., S., H., & T., H. (2014). Factors associated with a prolonged length of stay after acute exacerbation of chronic obstructive pulmonary disease (AECOPD). *International Journal of COPD*, *9*, 99–105. https://doi.org/10.2147/COPD.S51467