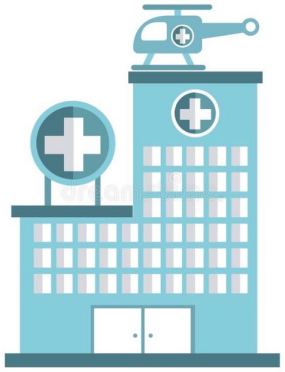


Predicting Hospital Length of Stay

Introduction

Overview of Dataset

The dataset was obtained from the UCI Machine Learning Repository[1] after it was donated on behalf of the Center for Clinical and Translational Research, Virginia Commonwealth University. The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes. Information was extracted from the database for encounters that satisfied the following criteria.

1. It is an inpatient encounter (a hospital admission).
2. It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.
3. The length of stay was at least 1 day and at most 14 days.
4. Laboratory tests were performed during the encounter.
5. Medications were administered during the encounter.

Though the encounters were filtered to include only those where a diabetes diagnosis was entered in the system, diabetes is often not the most responsible diagnosis/reason for admission to the hospital. The dataset does appear to have been created to facilitate the study of diabetic patients, it contains many other useful features standard in a hospital admission dataset which I believed made it suitable for predicting length of stay (LOS).

An excerpt of the Table 1 is shown below. The complete table can be seen in the Appendix.

Feature name	Type	Description and values	% missing
Encounter ID	Numeric	Unique identifier of an encounter	0%
Patient number	Numeric	Unique identifier of a patient	0%
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other	2%
Gender	Nominal	Values: male, female, and unknown/invalid	0%
Age	Nominal	Grouped in 10-year intervals: 0, 10), 10, 20), ..., 90, 100)	0%
Admission source	Nominal	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital	0%

[1] https://archive.ics.uci.edu

Time in hospital	Numeric	Integer number of days between admission and discharge	0%
Number of lab procedures	Numeric	Number of lab tests performed during the encounter	0%
Diagnosis 1, 2, 3	Nominal	The primary, secondary & tertiary diagnoses (coded as first three digits of ICD9); 861 distinct values	0%
24 features for medications	Nominal	Metformin, repaglinide, nateglinide, etc. The feature indicates whether the drug was prescribed or there was a change in the dosage. V	0%

The dataset contained most of the fields one might expect from a standard hospital dataset. Features like age, gender, race, diagnosis, admission source and unique identifiers, both for the patient and the specific admission encounter.

Objective:

The initial goal was to predict the nominal value of LOS, between 1 and 15 days. However, after exploring the data it began to appear that this would not be possible given the homogeneity of patients in the 1-6 day cohort. As such, the decision was made to build a binary classifier to predict if a patient would be admitted between 1-6 days vs. 7-15 days.

Methodology:

Data Preprocessing

In this section I will discuss the steps taken to clean the data and prepare it for modelling. It is important to note that, while these activities are described separately from the Data Exploration section, it was an iterative process between the two. That is, cleaning the data and exploring it were done almost simultaneously. For example, in order to prepare descriptive plots of one variable to be used in the data exploration phase, the variable had to be cleaned first.

Data cleaning was completed in two main phases - as seen in the two .R files; “Data Cleaning Phase 1” and “Data Cleaning Phase 2”. The first phase of data cleaning was done in conjunction with Data Exploration. This used the original dataset, and was designed to facilitate data exploration as well as preparing the data for modelling.

The second phase is commenced after the completion of data exploration in order to make the final changes needed to prepare the data for modelling. This primarily consisted of deleting variables. Variables were deleted for one of two reasons:

- They are extremely sparse. For example: acetohexamide had a single occurrence out of the 50K records in the training data set
- They are redundant. For example, The “medical_specialty” variable had been grouped and the spread into a set of binary variables.

The very first step in Phase 1 was to separate the data into training, testing and validation data sets. The plan from the outset was to attempt an ensemble model approach, which requires not only train and test sets, but a validation set as well. It is also important to note that the validation and test data sets were not included in any of the data exploration output. This was to prevent “information leak” and to simulate a real life situation, where the model is fully developed and deployed and only then is finally tested on as of yet unseen data.

A split of 70-15-15 (train-test-validation) was decided upon. This decision was more due to convention (train-test splits are most often 70-30 or 80-20) and to retain at least 10K records in the training and validation sets. Due to the sparsity of many of the variables, I was wary of having fewer records in training/validation sets as these sparse variables would then be unlikely to be present.

Data Preprocessing activities fell into one of four main categories.

1) Grouping categorical variables

The dataset had a large number of categorical variables where the grouping of different levels was not conducive to predicting LOS. For example, the Medical Specialty variable contained 70 levels, many of which represented less than 0.1% of was in creating useful grouping of the existing variables.

A similar issue occured with the Admission Source, Admission Type, and Diagnosis variables. In all of these cases, assumptions were made to group the different factor levels to a more manageable size. Many of these choices were obvious (grouping together “Radiology” and “Radiologist”) but other requires more in depth knowledge. For example, grouping together of surgical specialties requires

special knowledge of hospitals, particularly in regards to LOS. It is widely accepted that surgical units have shorter LOS than their medical counterparts. It is for this step in particular that a subject matter expert would have been useful, though my own expertise in this area is not insubstantial (having worked on LOS-related projects at a major tertiary hospital for several years).

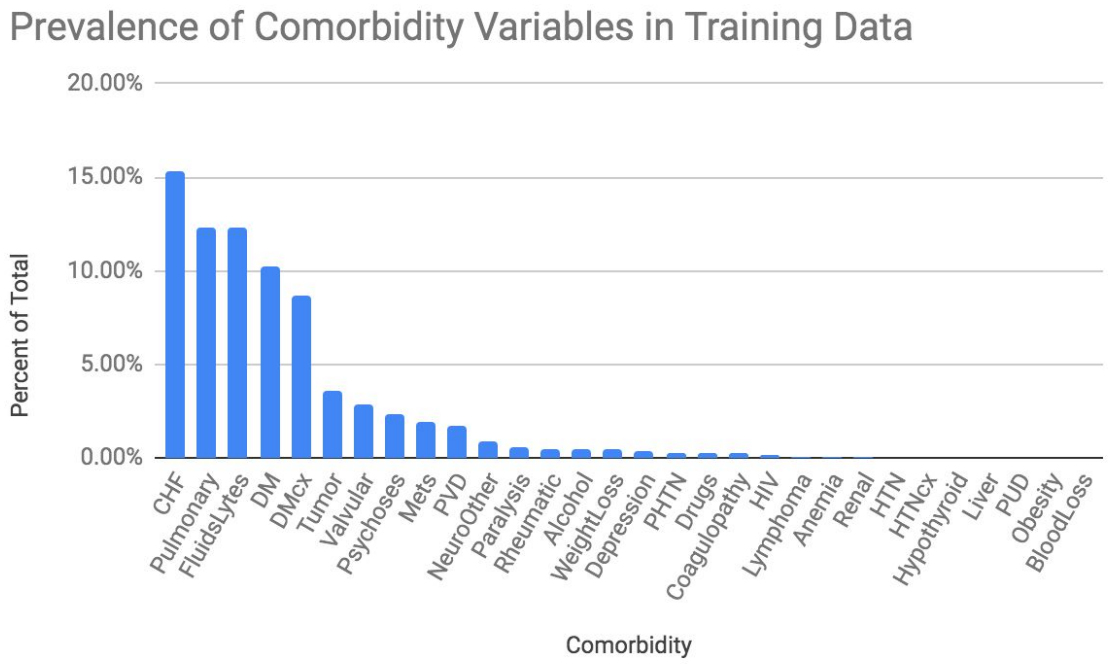
Grouping of the factor levels of the diagnosis variables was a similar problem. Given that this was likely one of the more predictive variables, I dedicated a substantial amount of time in researching how best to proceed. Eventually, I decided to use the ICD-9 groupings on Wikipedia (https://en.wikipedia.org/wiki/List_of_ICD-9_codes) which I felt provided a sufficiently concise representation of the data while still maintaining a distinction between high LOS and short LOS diagnoses.

2) Formatting variables

Both Admission Type and Admission Source are presented in the original dataset as a numeric code . A table is available to cross-reference these codes so I imported and merged these tables to make the variables appear as a text description instead. Additional tasks included formatting variables to be consistent with the style of the rest of the code (changing "Medical-Specialty" to "medical_specialty") and formatting a categorical variable into a series of binary variables (one for each level).

3) Feature creation

In an effort to extract the most value from the diagnoses variable, the ICD package was used to extract comorbidities from each diagnosis. The ICD package is designed to "calculate comorbidities, medical risk scores, and work very quickly and precisely with ICD-9 and ICD-10 codes[2]. There are several different methodologies available for how to extract comorbidities, but lacking knowledge in this area I choose what appeared to be the most popular.



[2] https://cran.r-project.org/web/packages/icd/index.html

4) Removing variables

After completing the data exploration, the decision was made to streamline the modelling process by removing variables with limited predictive power. These were either categorical variables where, for example, one level had 99% of the observations. While there were some concerns about removing variables that might be helpful in identifying certain cases, I was also conscientious of including too many variables in the model for fear of getting trapped in a local minima during the tuning phase. It was primarily for this reason that these variables were removed.

The other case were variables that were conditional on the outcome variables. For example, number of lab procedures, number of medications administered during the admission. In a real world setting, where the value of the model is in predicting a patient's LOS at the start of the admission, these details would not be available yet.

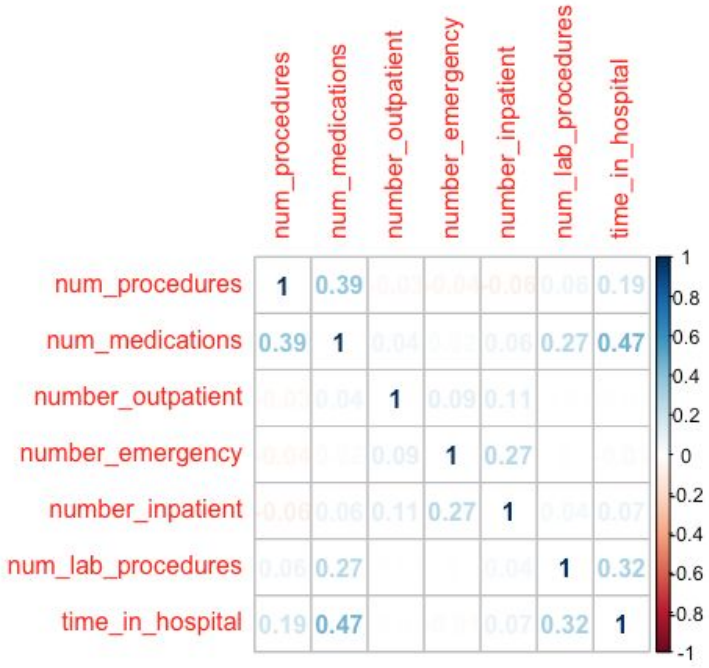
Lastly, the decision was made to randomly delete all but one encounter from patients that had multiple encounters. This was done in an effort to maintain independence between records, since patients with multiple encounters would likely have similar LOS for each encounter.

The entire cleaning process was then streamlined into a single function, so that it could be replicated with ease on the training and validation datasets.

Data Exploration

The first step was to examine each variable on its own. A description of what the variable represented was found on a publication done using the dataset.[3] For categorical variables, this involved examining the number and type of each level present. Numerical variables were examined using a variety of descriptive statistics and plots, including but not limited to: mean, median, standard deviation, density plots & histograms, sometimes using a log transform to make the plot more interpretable.
The next step was to explore the relationships between certain variables of interest. For example, a correlation plot was used to examine the linear relationship between all the available numeric variables.

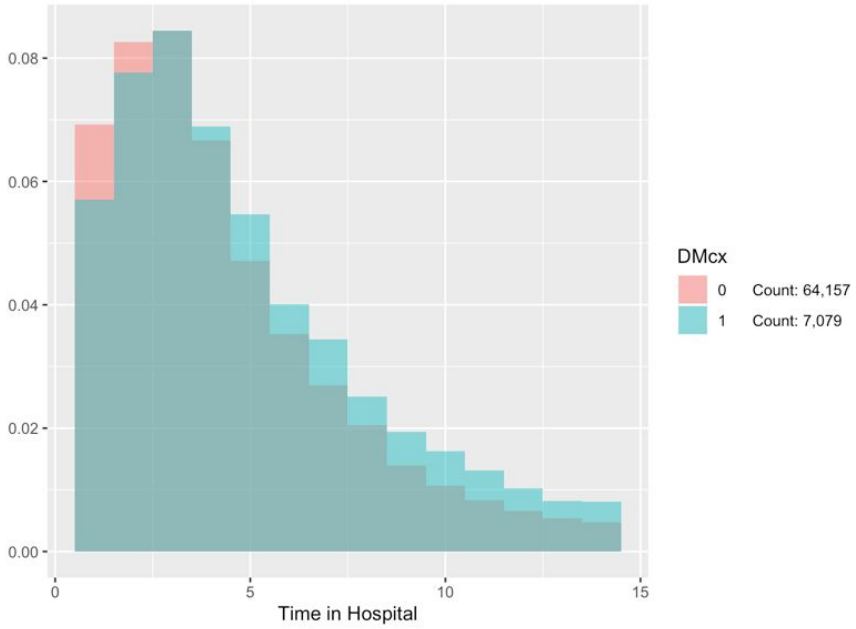
[3] https://www.hindawi.com/journals/bmri/2014/781670/



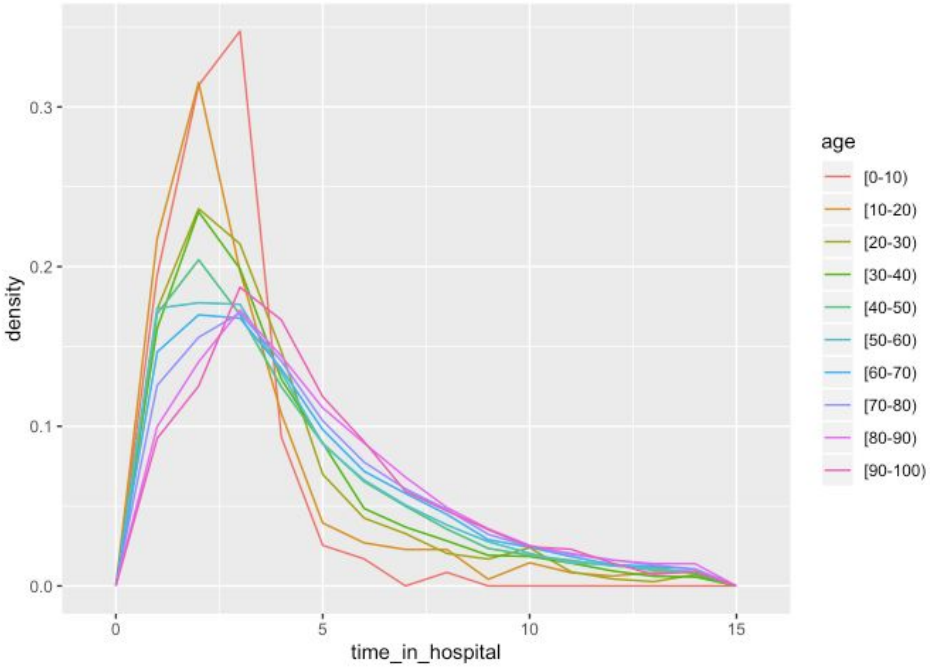
The most illuminating task was in examining each variables relationship to LOS (both as a binary "1-6 days" or "7+ days" variable, and as a continuous variable between 1-5 days). Again, a variety of tools were used to accomplish this task.

var_name	level	count	percent_of_total	mean	median	p25	p75	p90	plong_los
race	Caucasian	53318	74.8%	4.38	4.0	2.00	6.00	9.0	20.5%
race	AfricanAmerican	13419	18.8%	4.50	4.0	2.00	6.00	9.0	22%
race	Other	4499	6.32%	4.17	3.0	2.00	6.00	8.0	18.8%
gender	Female	38194	53.6%	4.48	4.0	2.00	6.00	9.0	21.2%
gender	Male	33042	46.4%	4.28	3.0	2.00	6.00	9.0	20.1%
max_glu_serum	None	67509	94.8%	4.38	4.0	2.00	6.00	9.0	20.7%
max_glu_serum	>300	893	1.25%	5.28	5.0	3.00	7.00	10.0	29.7%
max_glu_serum	Norm	1813	2.55%	3.95	3.0	2.00	5.00	8.0	16.2%
max_glu_serum	>200	1021	1.43%	4.73	4.0	2.00	6.00	9.0	23.3%

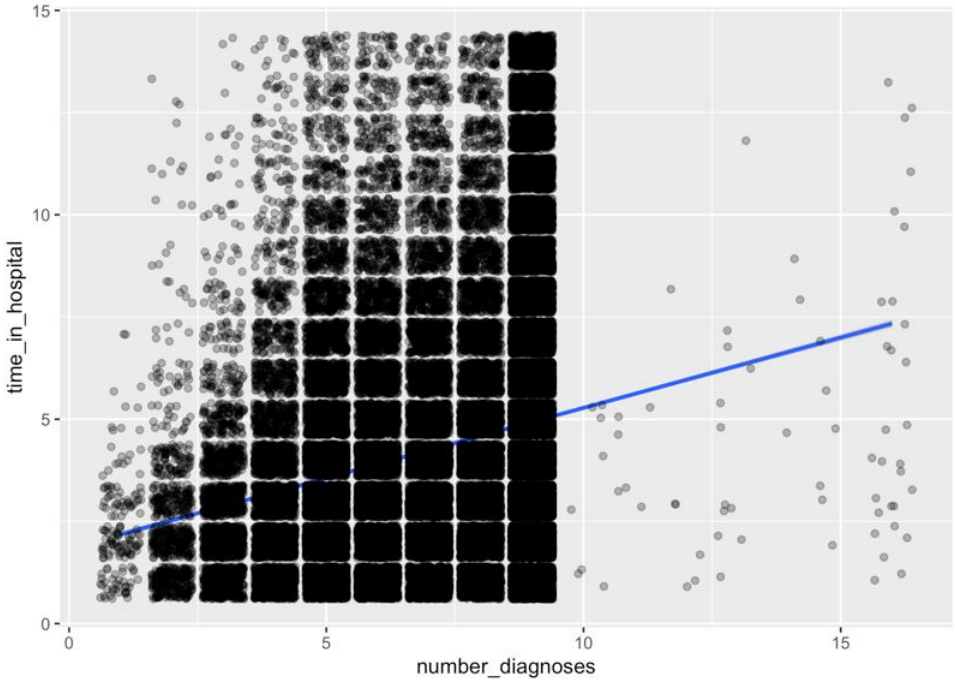
I examined each factor level of every categorical variable in the data set. The aim was to see if there were particular where there was a substantial difference in LOS between the factor levels. I looked at both the numeric LOS difference (1-15 days) and the binary difference. The mean value for the entire dataset for "long LOS" was 20%. So, for example, the max_glu_serum variable at level = ">300" has a higher than average occurrence of long LOS, although a fairly small count (1.25% of the total). This was a frustrating trend. Very few variables appeared to have significant differences between the factor levels in relation to the outcome variable.



To get a better sense of where the separation occurred, if at all, each categorical variable was plotted against LOS to examine if there was a difference that could be used for predictive modelling.



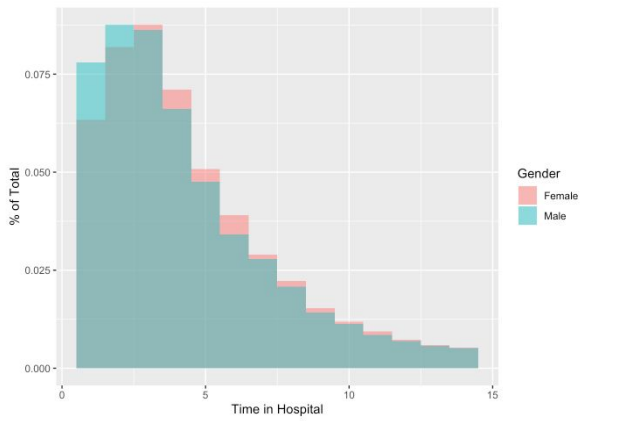
Where there were multiple factor levels, a stacked line chart was used to visualize all the factor levels at the same time.

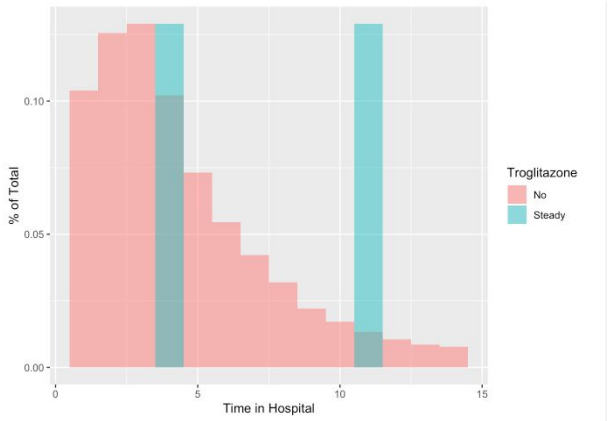


Numeric variables were examined using scatter plots with a linear regression line overlaid. The reaffirmed lessons learned from the correlation plots - that there likely aren't very strong linear relationships present in the data that would lead to strong predictive power.

Key Learnings from Data Exploration Phase

Throughout this entire phase the main question I sought to answer during this phase is how the analysis can be structured to improve the chances of success. The more I explored, the less optimistic I was that not only would a continuous prediction model be feasible (outputting a number between 1-15 days) but even a binary classifier would struggle to properly discriminate between long LOS and short LOS cases. This was due largely to the homogeneity of the variables across different values of LOS. For example,





The Male and Female have near identical LOS across the entire range of 1-15 days. Or, where separation did occur, as it the Troglitazone variable, it was usually only a handful of occurrences (2 out of 70K cases).

All of this exploration lead me to two conclusions:

1) There were very few linear relationships to be exploited in modelling. Hopefully there are some non-linear, or more complex relationships (interactions, for instance) that can be discovered through a creative choice of model. A General Additive Model was selected to leverage this possibility.
2) It seemed extremely unlikely that I would be successful in predicting on a 1-15 day scale. A binary classifier has a much higher chance of success. A split of 1-6 days vs 7+ days was selected as a balance between properly segmenting the outliers from the normal cases, while still maintaining at least 20% of the data in the "outlier" segment. Initially, I considered 1-7 days vs 8+ days but there were simply too few cases present in the 8+ days segment.

Model Selection

The goal from the outset of the project was to use an ensemble model. Ensemble modelling is a technique of training several individual models and then combining their results to, ideally, increase accuracy while reducing variation. The logic is that some models are good at certain tasks (e.g. identifying outliers), and combining them leverages the strengths of each individual model. Many machine learning algorithms already use this idea, for example XG Boost, or Ada Boosted decision trees, which iteratively improve on weak learners to create a much stronger model. Ensemble modelling takes this one step further, by allowing for more than one boosted model to be combined, leveraged not just the weak learners contained in the individual model, but the combined intelligence of the entire set of models.

Ultimately, I selected three models to be part of the ensemble. A Generative Additive Model (GAM), a penalized Logistic Regression Model, and an XG Boost model. My criteria for model selection was straightforward

- *Computationally inexpensive.* I was training these models on a single core Macbook Air with 4GB of RAM. Any algorithm I selected had to complete its training overnight, within an 8 hour block of time. This excluded models like Random Forest which would likely have required 24+ hours to run to completion.
- *Relatively simple to learn and implement.* I was wary of simply choosing a model and implementing it without any knowledge of its inner workings. I had some familiarity with XG Boost in the past and spent time researching penalized logistic regression so felt relatively comfortable with these two.
- *Complementary strengths and weaknesses*. To make use of the ensemble model, the composite models should have different predictions. Anything about 75% correlation on model predicitions implies that ensembling the models won't lead to much improvement. It was for this reason that a GAM was selected, as I was hoping it would be able to capture some non-linear relationships that were missed by the other two models.

- *Personal interest.* XG Boost and Penalized Logistic Regression are both widely used in industry and I had a personal interest in gaining some familiarity with them.

Model Development

The Caret package by Max Kuhn was utilized to automate cross validation and model tuning tasks. Repeated 10-fold cross validation (with three repetitions) with a range of tuning parameters was used for the logistic regression and XG boost models.

After a prolonged effort, I was unable to make the GAM model operational within the context of the Caret package. I trained this model on its own, and without cross validation or different tuning parameters. This was largely due to time constraints was certainly not an optimal approach.

The ensemble model was constructed out of the optimized Logistic Regression and XG Boost models. For reasons to be discussed later, the GAM model was not included in the ensemble.

Model Evaluation

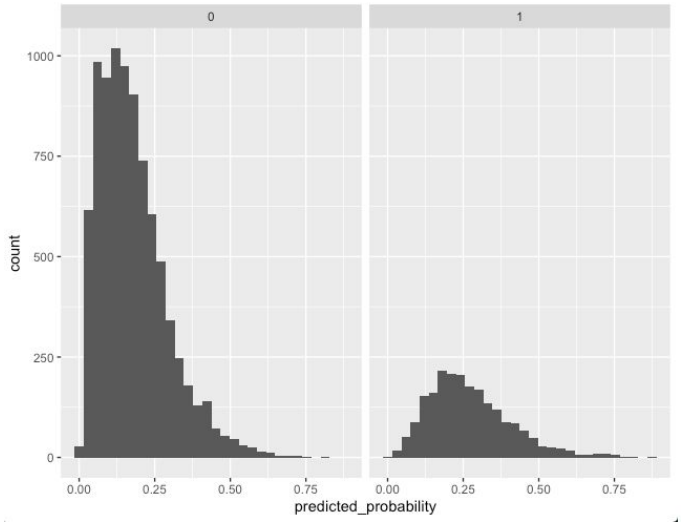
The model evaluation process was the same for all three models and was performed using the testing data set. Sensitivity and specificity were examined using a Confusion matrix and ROC curves. Variable Importance was also examined using standard functions available in the Caret package. Finally, the probability of predicting long LOS was visualized in a histogram to compare how probability varied over LOS for the two groups.

1) Logistic Regression Model

		Actual	
		Long LOS	**Short LOS**
Predicted	**Long LOS**	1.21%	1.28%
	Short LOS	18.56%	78.95%

Accuracy	0.8017
Sensitivity	0.06129
Specificity	0.98408

The high accuracy is comprised almost entirely of predictions for short LOS. The model has extremely low sensitivity, reflecting a serious inability to properly categorize Long LOS cases.



For each model, I examined how the predicted probabilities compared for the two levels of the outcome measure, as well as the shape and scale of the ROC curve.

2) XG Boost Model

		Actual	
		Long LOS	**Short LOS**
Predicted	**Long LOS**	1.49%	0.98%
	Short LOS	18.28%	79.25%

Accuracy	0.8074
Sensitivity	0.07544
Specificity	0.9878

The XG Boost model performed marginally better in terms of sensitivity, though struggled with the same issue of identifying the Long LOS cases.

3) Generalized Additive Model

Though the other diagnostic tools were also used to evaluate the GAM, after examining a correlation plot comparing predictions of the three models, the decision was made to remove the GAM from further consideration.

	XG Boost	*Logistic Regression*	*GAM*

XG Boost	---	0.8603	0.8567
Logistic Regression		---	0.9978
GAM			---

The GAM was nearly identical in its predictions compared with the penalized Logistic Regression model. I struggled with tuning this model and in the end had to use the default parameters and without cross validation. Still, it is interesting that it behaved almost identical to the fully tuned logistic regression model.

4) Ensemble Model

After removing the GAM from consideration, the ensemble model consisted of only two models. This is less than ideal, especially when considering the Logistic Regression and XG Boost models we 86% correlated.

The test dataset was loaded and predictions made using both XG Boost and Logistic Regression models. This resulted in two new variables to be fed into the ensemble model. These two variables, in theory, contain all the information contained in the other 73 variables. The ensemble model was created from training random forest was trained on the test dataset using only the predictions and the outcome variable.

Finally, the validation data set was used to test the ensemble model.

		Actual	
		Long LOS	**Short LOS**
Predicted	**Long LOS**	3.55%	6.08%
	Short LOS	16.73%	73.65%

Accuracy	77.28%
Sensitivity	17.52%
Specificity	81.5%

The ensemble model achieved an increase in sensitivity at the expense of specificity. Given that separately the models struggled to identify the long LOS cases, it is reassuring to see the ensemble model improve in this regard. Depending on the use case of the model, this outcome may be preferable.

Discussion

Overall, none of the models were especially useful in identifying the long LOS cohort of patients. The best of the four was the ensemble model, which achieved a sensitivity of 17.52%. I attribute this result to several factors.

First, the data set itself was probably not ideally suited to this task. The data was originally compiled to aid in the study of diabetes in hospital setting. While many useful variables were present, some key pieces of insight were missing. For instance - which each record was identified at both the patient and the encounter level there was nothing available to indicate in what order the encounters occurred. For instance, if a patient had several admissions over a period of time, it would have been useful to be able to create variables such as “time since last encounter” or “mean LOS of previous encounters”. Instead, these encounters had to be randomly removed in order to maintain independence of the data.

Second, while I have above average experience for a data analyst investigating this type of problem, it is definitely recommended to consult a subject matter expect to validate many of the assumption I have made. For example, the grouping of the Medical Specialty or Diagnosis variables, to name a few.

Further improvement to the predictive model could be made by:

a. Adjusting the way variables are grouped (Diagnosis, Medical Specialty, etc.)
b. Using a different methodology for comorbidity extraction from ICD-9 codes.
c. If the goal is to improve predictive power at the expense of interpretability, creating dozens or even hundreds of new features as combinations of the existing features (e.g. interactions, log or square transforms) at random has been shown to work in some Kaggle competitions. This depends on the goal of the analysis - if merely predicting LOS is the goal vs understanding the drivers behind LOS.
d. Adding additional models to ensemble, particularly those with lower correlation that the two used.
e. Test different methods of ensembling. A Random forest was used but it was a fairly naive decision to do so.

Appendix

Complete Table 1:

Feature name	Type	Description and values	% missing
Encounter ID	Numeric	Unique identifier of an encounter	0%
Patient number	Numeric	Unique identifier of a patient	0%
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other	2%
Gender	Nominal	Values: male, female, and unknown/invalid	0%
Age	Nominal	Grouped in 10-year intervals: 0, 10), 10, 20), ..., 90, 100)	0%
Weight	Numeric	Weight in pounds.	97%
Admission type	Nominal	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available	0%
Discharge disposition	Nominal	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available	0%
Admission source	Nominal	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital	0%
Time in hospital	Numeric	Integer number of days between admission and discharge	0%
Payer code	Nominal	Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay	52%
Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon	53%
Number of lab procedures	Numeric	Number of lab tests performed during the encounter	0%
Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter	0%
Number of medications	Numeric	Number of distinct generic names administered during the encounter	0%
Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter	0%
Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter	0%
Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter	0%
Diagnosis 1	Nominal	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values	0%

Diagnosis 2	Nominal	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values	0%
Diagnosis 3	Nominal	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values	1%
Number of diagnoses	Numeric	Number of diagnoses entered to the system	0%
Glucose serum test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured	0%
A1c test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured.	0%
Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”	0%
Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”	0%
24 features for medications	Nominal	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed	0%
Readmitted	Nominal	Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.	0%