Predicting Hospital Admission Length of Stay

Andrew Burton, RN, BSN, BS

Background

As healthcare becomes more complex alongside an aging US population, hospitals have been put under the increasing strain of high censuses and insufficient staffing. To assist hospital personnel in managing limited resources and improve patient care, being able to accurately predict patients' length of stay (LoS) can be useful. Having a reliable projected LoS can help in planning staffing, anticipating unit censuses, and guiding care. Because of this, I wanted to train a model that could reliably predict a patient's LoS given data gleaned from their chart.

Data

The data used for model training was obtained from the MIMIC database (specifically the MIMIC-III database). This database contains over 20 dataframes with information pulled from the charts of more than 50,000 hospital ICU admissions from 2001 to 2012. For this analysis, I utilized 4 of the dataframes: admission information, patient information, patient diagnoses, and diagnostis-related groups (DRGs are codes used for billing purposes and carry information about diagnoses and interventions).

All visits were identified with a unique Hospital Admission ID. It is important to note that this is different than the Patient ID – a patient could have more than one admission on record so multiple Hospital Admission IDs could map to the same Patient ID.

To maintain patient confidentiality, admission, discharge, and birth dates had all been randomly shifted while maintaining the same relative timeframe. The exception to this being patients older than 90, who had their date of birth shifted such that their age would be over 300 at the time of admission to further protect anonymity.

Target Variable

Length of Stay (days): Obtained by subtracting admission time from discharge time. There were a few cases in which admission time occurred after discharge time. In these situations, the patient's recorded time of death coincided with their discharge time. It seems these are patients that presented dead on arrival to the emergency department – such instances were set with a LoS of 0 days.

Features

Admission Location (8 levels, reduced to 3): All the distinct types of transfers were relabeled as 'transfer' and all referral types were relabeled as 'referral,' resulting in 3 levels, with the third being 'ED.'

Admission Type (4 levels, reduced to 3): Three labels, 'newborn,' 'emergency,' and 'elective' were left as is. Instances labelled 'urgent' were relabeled as 'emergency,' resulting in 3 levels.

Language (75 levels, reduced to 2): This attribute was mapped to reflect the presence of a language barrier (if the patient was English-speaking or not).

Ethnicity (41 levels, reduced, to 7): These labels were generalized to reflect patients' race: 'Asian,' 'Black,' 'Latinx,' 'Multiracial,' 'Native' (Hawaiian or Alaskan), 'White,' and 'Unknown.'

Admitting Diagnosis (>15,000 levels, reduced to 76): This column was derived from free text input, so entries contain typos and inconsistent labeling. I used regular expressions to pull 76 various keywords from the text to be used as labels.

Gender (2 levels): Left as is.

Date of Birth: Because all patients over 90 years of age had a recorded age of over 300, I changed all their ages to 90.

Diagnoses (> 6,000 levels, reduced to 320): This contains the ICD9 code for each diagnosis recorded in a patient's chart. I mapped these values to 320 broader diagnostic categories. The value for each category was weighted as the number of diagnoses that fell under said category. For example, all diagnoses under Acute Cardiac Ischemia (ICD9 codes 410 and 411) were grouped together. A patient with an Anterior MI (ICD9 code 410.1) and an Inferior MI (ICD9 code 410.4) would be recorded as 2 under the label 'Acute Cardiac Ischemia.'

DRG Description (> 1,300, reduced to 91): There are different DRG code systems that have changed over time, so using the codes themselves would be cumbersome. Instead, I performed a direct mapping of the text description to create attributes corresponding to the general organ system involved (Neuro, Cardiac, etc.) in addition to various interventions and events (if the patient was ventilated, underwent a procedure, presented as comatose, if they experienced any kind of organ failure, etc.). Multiple tags could be pulled from a single entry.

Missing Values

Several columns contained missing values. For ethnicity, there were over 7,000 instances recorded as 'unknown', which I chose to keep as its own level – I did not think there was an effective way to impute these missing values.

In over 24,000 instances, no language was recorded. Based off my own experience, hospitals often only record a patient's primary language only if it is other than English. Because of this, I felt it appropriate to assume all missing language values were English-speaking.

There were 285 entries missing their admission location. To impute these values, I started by looking at the corresponding admission diagnosis. Nearly all of these had the diagnosis of 'newborn.' Looking at all admissions shared the 'newborn' diagnosis showed over 7,000 had 'referral' as their admission location and only 3 had 'transfer'. Accordingly, I filled these missing admission locations with 'referral.' The remaining missing admission locations were mostly trauma cases and one admission in respiratory failure – all clearly emergency admissions. These missing values were updated as such.

Finally, 11 entries were missing their admitting diagnosis. These values were replaced based off the primary diagnosis.

Data Prep

I created dummy variables for all categorical features, after which I applied a 70/30 train/test split on the data. Numerical features, (including those obtained from the ICD9 codes) were scaled using a min max scaler. Finally, I visually inspected the mean LoS across the levels for all features, removing any features that had an equal mean LoS across all feature levels. In total this eliminated 22 columns, with the final dataframe having 58,929 rows and 406 columns. I created a second train/test set after applying PCA. I chose the number of components such that 0.99 of the variance was retained, which reduced the 406 columns down to 315 principal components.

Data Exploration

Bias

Healthcare has a long history of bias and discrimination towards minority populations and women. This data is no different – males make up about 56% of the admissions and 70% of admissions were labeled as white (figure 1). Additionally, gender is record as either 'M' of 'F,' with no allowance for any non-binary labels. Because of this, models may underperform for these underrepresented populations.

Length of Stay

The average length of stay was about 9.79 days and had a notably right-skewed distribution, with most visits being in the range of 0 to 50 days and a few longer than 100 days (and a maximum of 294 days). I chose not to

remove any of these outliers – a LoS of 100 days or more is not unusual and being able to predict such long stays would be useful to hospital staff (figure 2).

Modeling

For evaluating models, I chose to use RMSE with a basic linear regression scoring 84.64. I trained several additional models – a random forest, neural network, lasso, and ridge regressions. This was repeated on the reduced-dimension data as well, though PCA did not improve the results. Overall, the random forest regression performed best – I chose to proceed with refining this model (figures 3,4).

		Linear Regression	Ridge Regression	Lasso Regression	Random Forest	Neural Network
RMSE	All Features	84.64	84.68	160.97	77.76, 74.23*	88.65
	PCA	86.15	86.15	160.97	89.42	90.96
R^2	All Features	0.47	0.47	-3.98	0.52, 0.54*	0.45
	PCA	0.46	0.46	-3.97	0.44	0.43

* After Optimizing Hyperparameters

Feature Selection

I wanted to assess if the random forest performed better after eliminating any superfluous features. This was implemented by sorting the features by the magnitude of their linear regression coefficients and including them in modeling following a foreword stepwise process. I was unable to pare down the number of features – the model that performed best utilized all 406.

Hyperparameter Tuning

The random forest showed an appreciable drop in performance from training to test sets, having worse R^2 and RMSE on the new data. To address this, I performed a grid search with a focus on tuning hyperparameters to limit overfitting.

	Pool					Optimal
Max Depth	25	50	75	None		50
Max Number of Features	50	100	150	200	None	100
Min Sample per Leaf	1	5	10	15	20	1
Number of Estimators	100	150	200			200

Results

Applying these parameters improved the test-set RMSE to 74.23. RMSE on the training set was 10.25, so overfitting is still an issue. Overfitting is also evident in the R^2 score for train and test sets; 0.94 and 0.54 respectively. The model seems prone to underestimating, especially with longer stays (figures 5,6). About 60% of projections were within 4 days, and 80% were within 6 days. Looking at the ratio of the observed versus predicted, about 69% of predictions were within 0.7 to 1.3 times the observed (figures 3,4).

Discussion

There are several means by which models could be improved. For this analysis, I only used the MIMIC-III database. More data could be added by merging this data with that from the MIMIC-IV database. Additionally, there are several other dataframes within the MIMIC-III database. These include information on patient lab values, procedures, medications, and more. Incorporating this data into model training may boost performance. It would also be worth looking into instances the model struggled.

Additional improvement could possibly be found by addressing the skewedness of LOS – applying an oversampling algorithm or cost function penalizing poor estimates of long stays. Finally, there might be other regression algorithms that perform better.

Data Source

Johnson, A. E. W. *et al.* MIMIC-III, a freely accessible critical care database. *Sci. Data* 3:160035 doi: 10.1038/sdata.2016.35 (2016).

Accessed From

https://physionet.org/content/mimiciii/1.4/

Figures

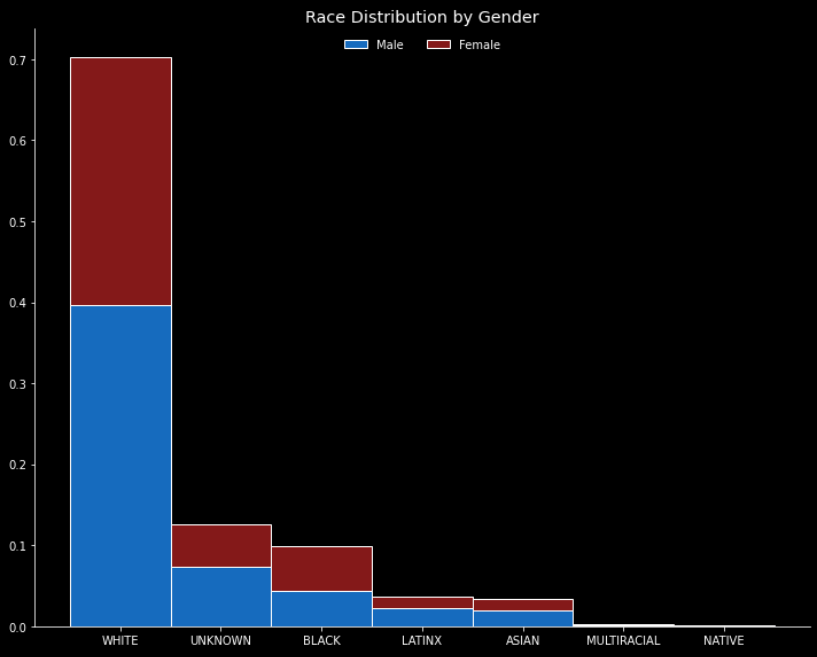

Figure 1

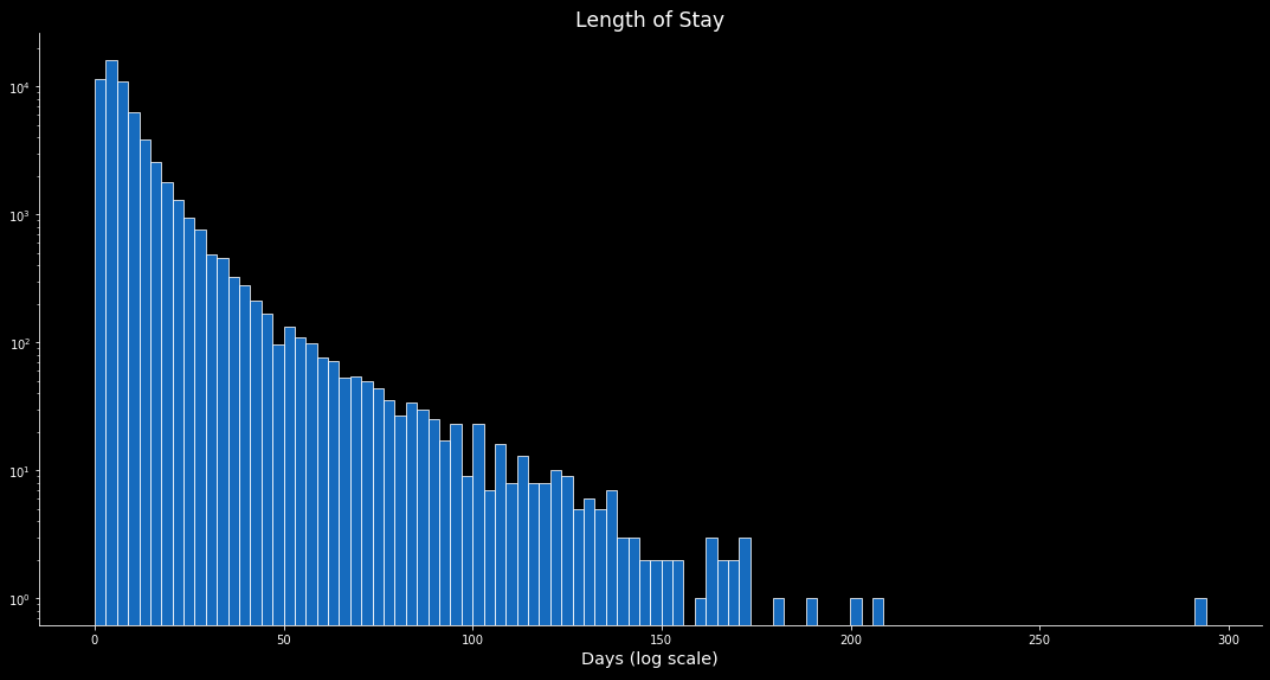

Figure 2

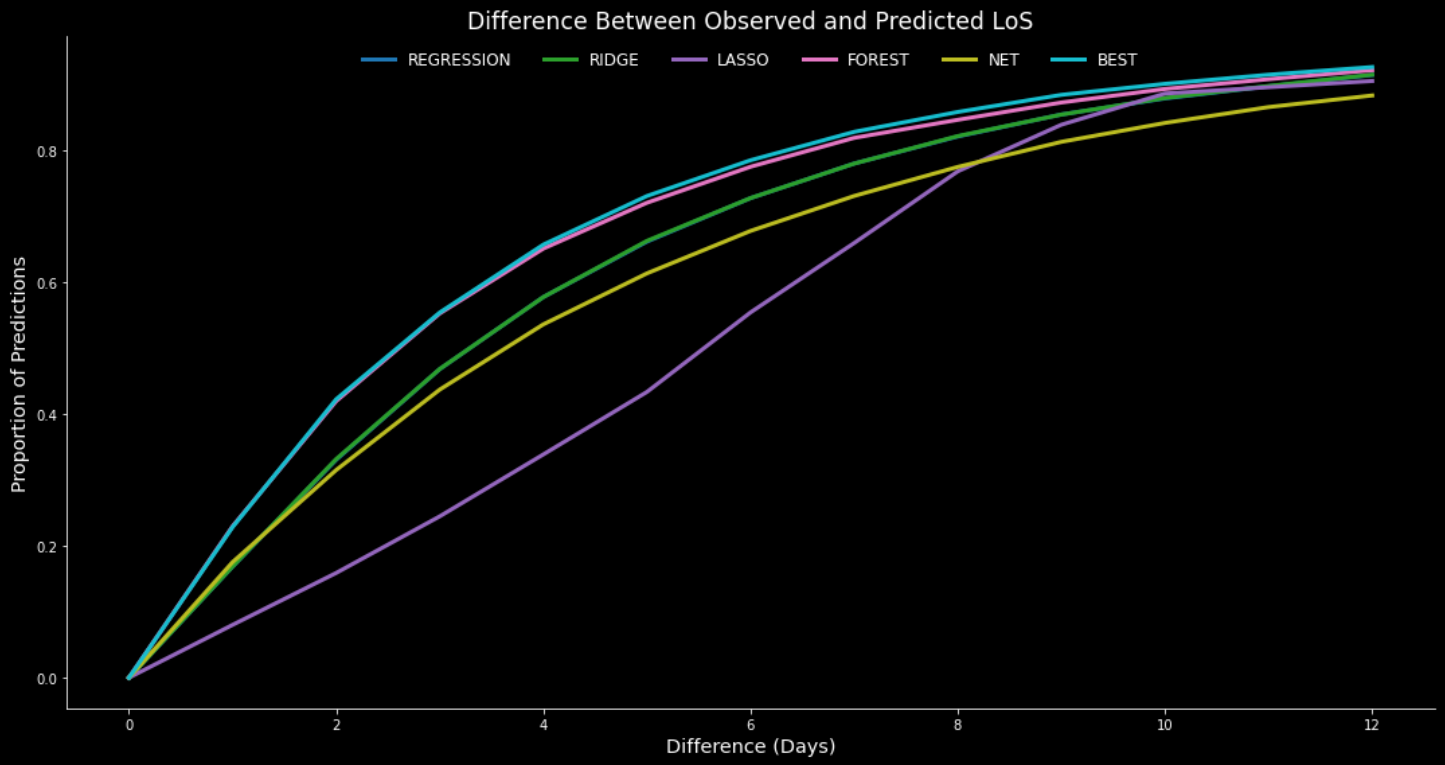

Figure 3

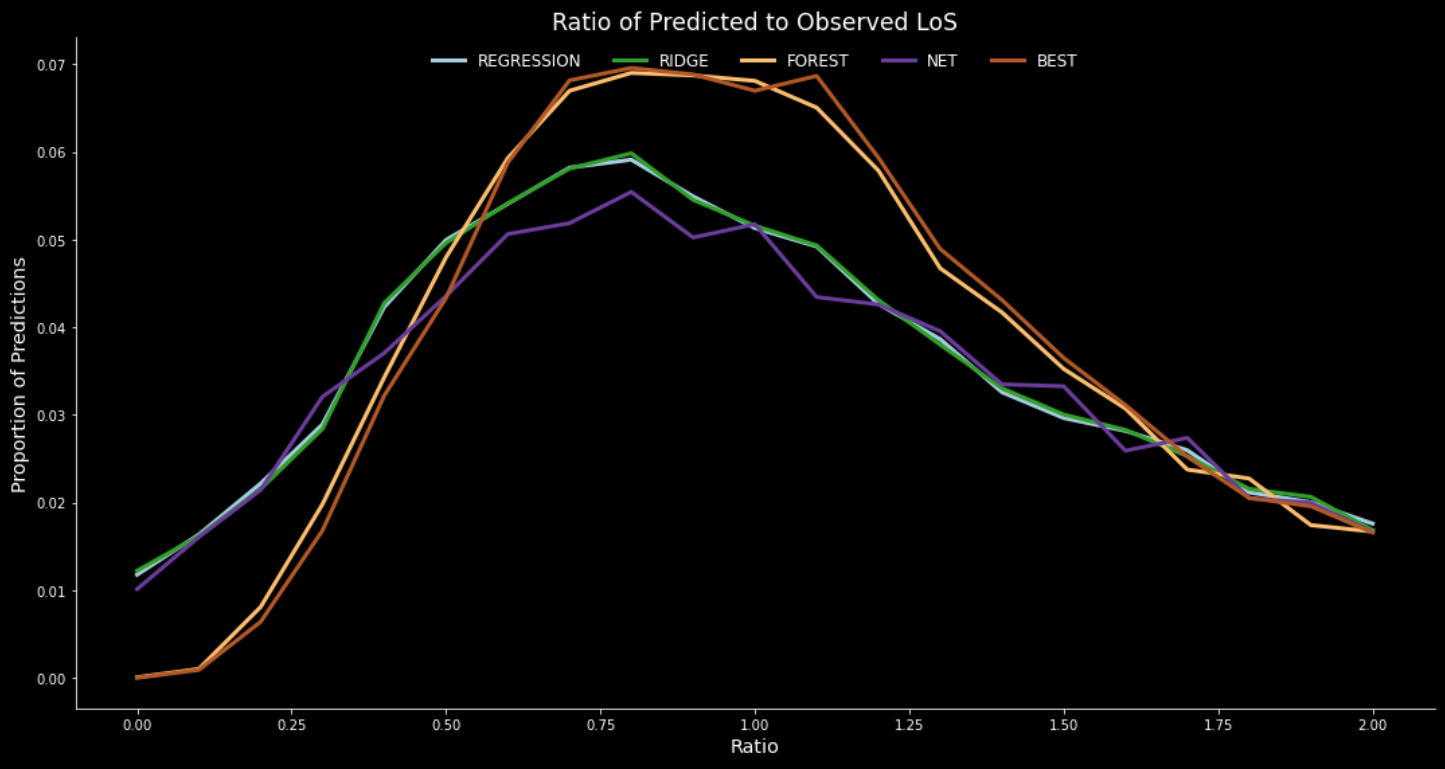

Figure 4

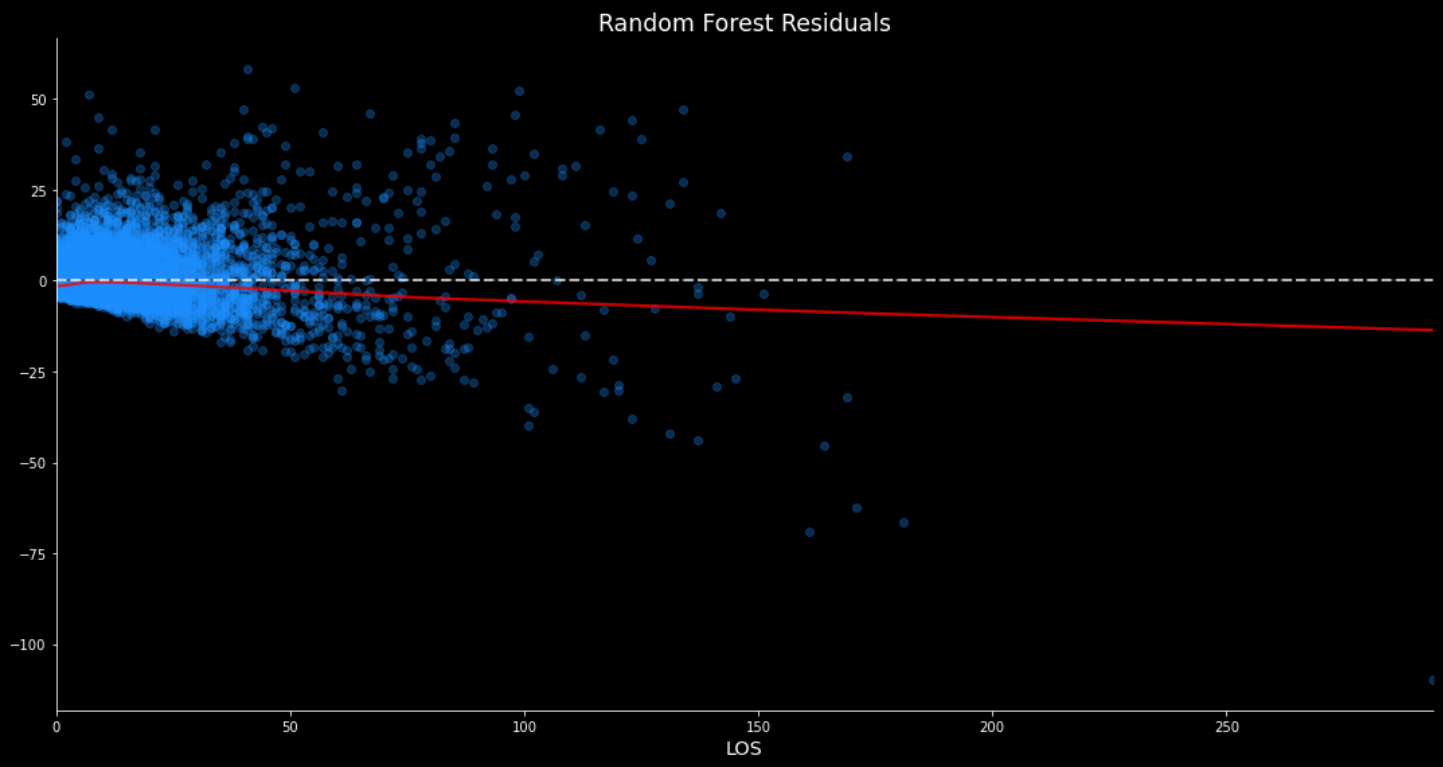

Figure 5

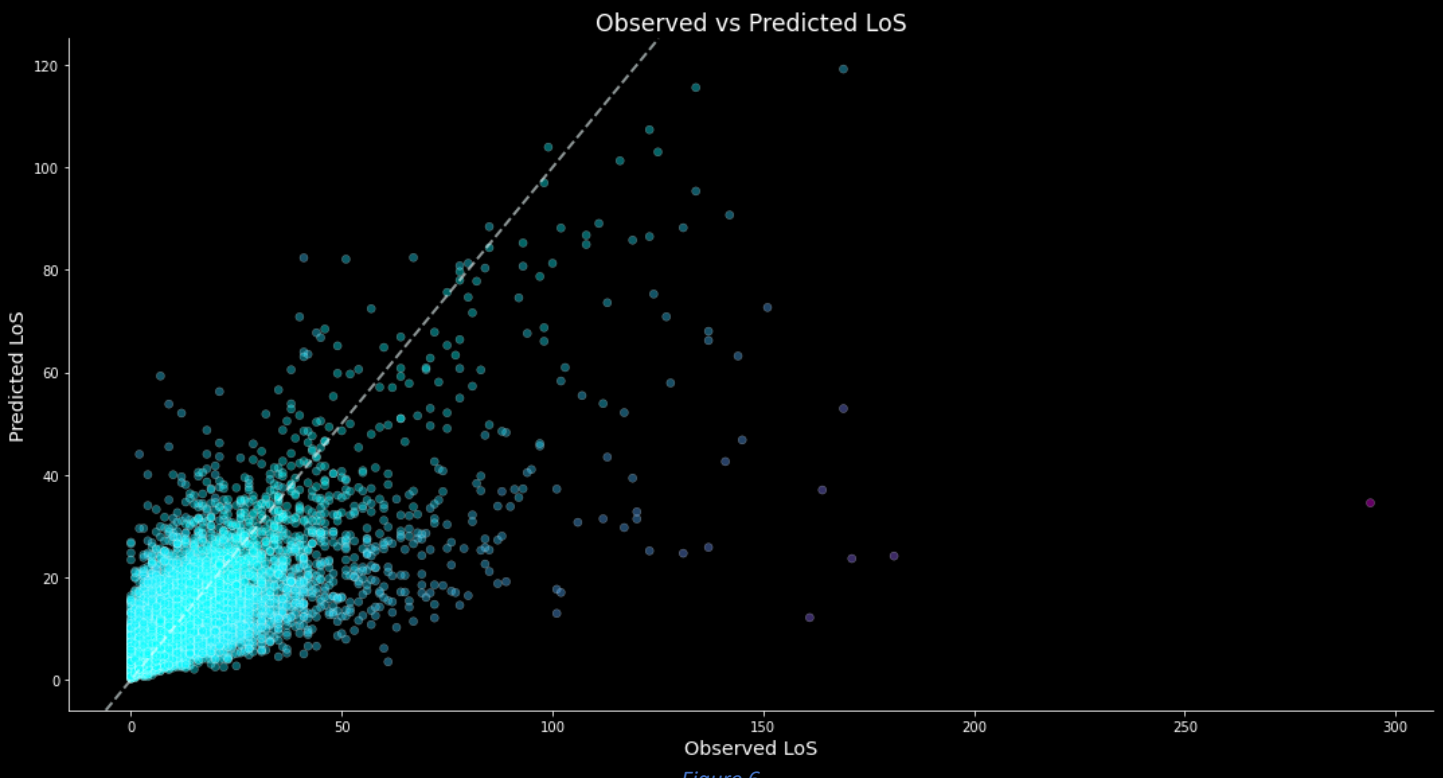

Figure 6