Predicting Delivery Hospital Stay

Abigael Bousquet
Brown University
GitHub[1]

1. Introduction

Preparing for a birth can be nerve-wracking for an expecting mother with all of its unknowns. As one's due date approaches, knowing how long to expect for the labor and delivery stay at the hospital can help an expecting family to (1) prepare childcare for other children during the hospital stay, (2) plan days off from work for a partner ahead of time, and (3) emotionally prepare for their stay at the hospital, bringing invaluable peace of mind to an overwhelming time.

To answer this question, I began with a dataset from the Statewide Planning and Research Cooperative System (SPARCS) under the New York State Department of Health (NYSDOH): Hospital Inpatient Discharges (SPARCS De-Identified): 2021[2]. Row-level data is gathered at discharge for each patient visit and the complete yearly dataset is published once annually, de-identified. At the time of this project's birth, the 2021 data was the most recent data available.

I extracted a particular subset of the NYSDOH dataset to answer the question of labor and delivery stays: discharges whose diagnosis description was "uncomplicated pregnancy, delivery or puerperium". I subsequently restricted my target audience to expecting mothers anticipating an on-time delivery without known pre-birth complications. Making this distinction allowed me to use the maximum number of features from the original dataset, including the labor diagnosis and metadata about the hospital where the birth took place as anticipated to make a meaningful prediction prior to the expecting mother's trip to the hospital.

2. Exploratory Data Analysis

The final dataset contains 18,804 i.i.d. rows (births) with 13 features and the target variable. Features include hospital metadata and limited patient metadata including age group, the first three digits of the patient's zip code, gender, race, ethnicity, and three forms of payment. It is important to note that metadata about the patient was extremely limited given the public nature of the dataset, and some rows had enhanced-de-identified columns which were intentionally blank.

Due in part to this enhanced-de-identification, there were missing values in this dataset. 54% of the 13 features contained missing values, some redacted by the NYSDOH and others like payment type data likely left blank by patients who didn't require three payment types. This amounted to 89% of the rows containing missing values.

I began my EDA exploring the target variable: length of stay. Stays ranged from 1-17 days and proved to be heavily unbalanced in favor of 1-3 day stays (Figure 1). Knowing this distribution helped to guide my splitting strategy and understand the range of reasonable predictions.

I also visualized the distribution of different features against the target variable (Figures 2-5). Figure 2 revealed some correlation between primary payment type (payment_type_1) and length of stay; although distributions of delivery stays were mostly shared across payment types, there were some interesting patterns in outliers. For example, a patient with self-pay as their primary payment type was most likely to have a 1 day stay (28.9%) whereas a patient with medicaid as their primary payment time was much less likely to have a 1 day stay (14.7%).

Some feature values that I had expected to correlate strongly with length of stay did not; particularly, I expected the distributions of length of stay to differ more significantly between age groups (Figures 3-4). However, since only age groups were provided rather than specific ages of patients for security, it is possible that such a correlation exists but is not obvious without the underlying distributions of ages within those groups.

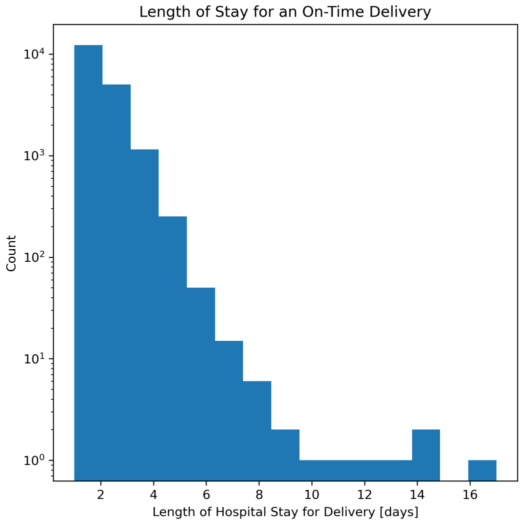

Figure 1: Distribution of Target Variable

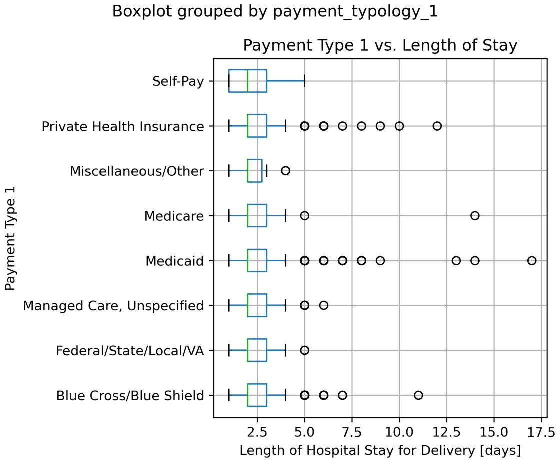

Figure 2: Primary Payment Type vs. Target Variable

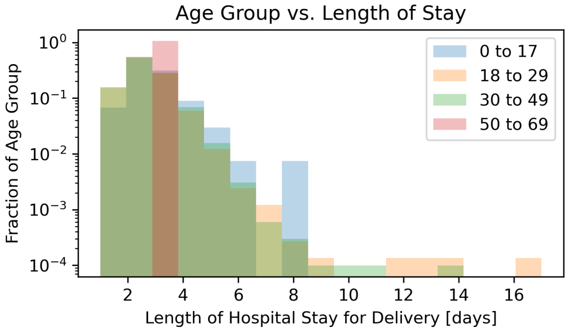

Figure 3: Age Group Category-Specific Histogram

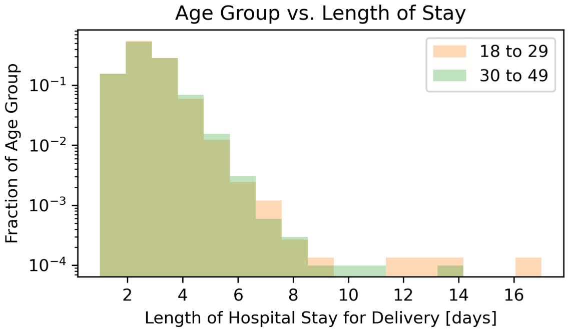

Figure 4: Subset of Age Group Category-Specific Histogram

The most variable distribution of the target variable across feature values was for permanent_facility_id—the specific hospitals of the birth (Figure 5). This was surprising to me, as I expected patient data to more obviously correlate to the target variable than hospital data.

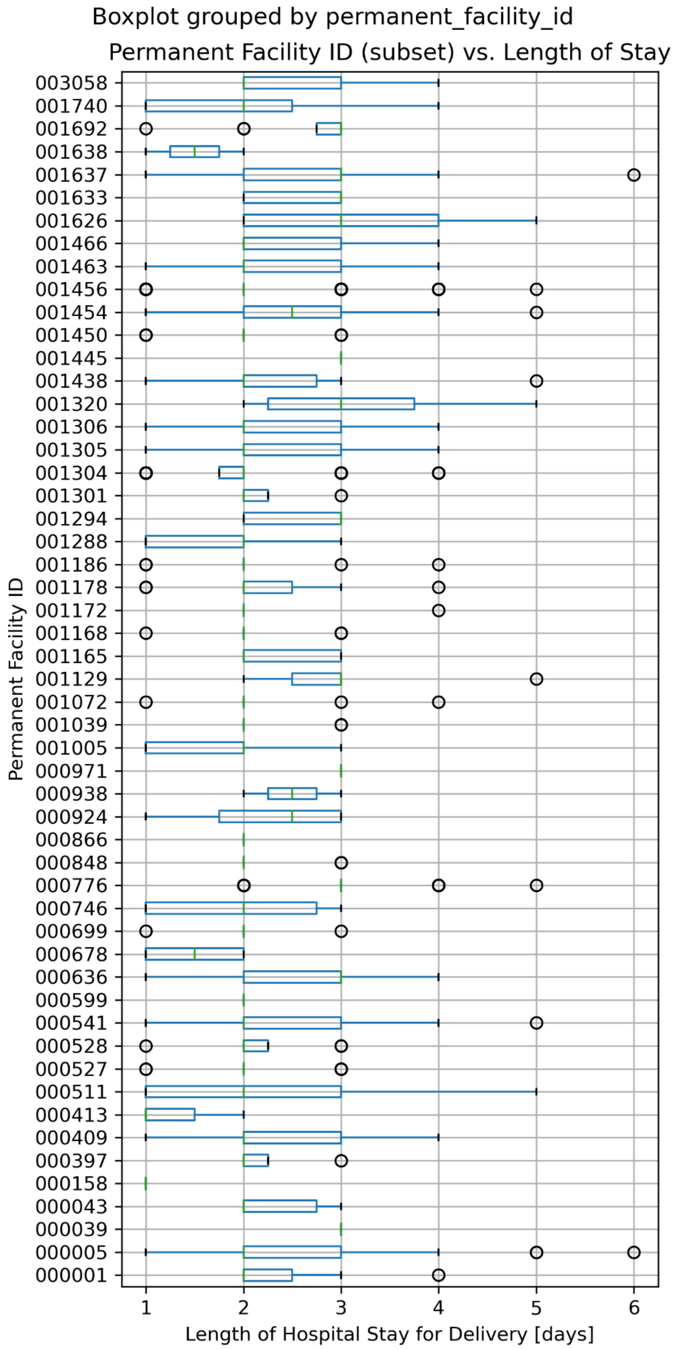

Figure 5: Subset of Facility IDs vs. Target Variable

3. Methods

3.1 Splitting

Given the imbalanced nature of the target variable, I employed a stratified splitting strategy to separate the dataset into training, validation, and test sets. In my stratified continuous split function, the range of the target variable is separated into many small bins—effectively creating categories out of the regression variable—which can then be stratified by train_test_split. I made use of the scsplit function from the Verstack package[3], which has the added functionality of combining bins with nearest neighbors when necessary to generate bins of the minimum size required by train_test_split.

Based on the size of my dataset, I split 60% train, 20% validation, and 20% test.

3.2 Preprocessing

For preprocessing, I first handled encoding of the categorical and ordinal features separately. The singular ordinal feature in the dataset—age_group—did not have any missing values, so sklearn's OrdinalEncoder formed the ordinal transformer pipeline. The remaining features were categorical and many contained missing values so their pipeline was more complex; first, missing values were imputed to the new category "not reported" and second, each feature was encoded with sklearn's OneHotEncoder. The combination of these two encoders formed my first preprocessor.

My second and final preprocessor then applied a StandardScaler to the encoded data, ensuring that each feature had a mean of 0 and standard deviation of 1 so that I would be able to accurately use linear model coefficients in determining feature importance.

At this stage, I also calculated the Pearson correlation coefficients between encoded features. For later interpretability of my models, it would be important to remove all but one of strongly correlated features. Based on this output, I determined that there were perfect 1.0 correlation coefficients between features permanent_facility_id, facility_name, and operating_certificate_number so I removed all but permanent_facility_id in my data accordingly.

3.3 Hyperparameter Tuning and Cross Validation

I chose to evaluate my models by root mean squared error (RMSE). I favored RMSE over MSE to maintain error in the same unit as my target variable for clearest error interpretation.

My cross-validation pipeline is shown in Figure 6. Of note is the aforementioned double preprocessing and the repetition of this pipeline 10 times with 10 different random states, allowing me to calculate uncertainty due to randomness in my models.

To perform hyperparameter tuning, I trained six different models on the training set with a range of values for each of their most influential parameters and ultimately chose the combination that minimized RMSE on the validation set (Table 1). When choosing hyperparameter ranges, I used linear ranges for bounded hyperparameters and log ranges for hyperparameters with no upper bound. To ensure that my ranges were wide enough I plotted parameter vs. RMSE graphs and confirmed the presence of regions of both underfitting and overfitting (e.g., Figure 7).

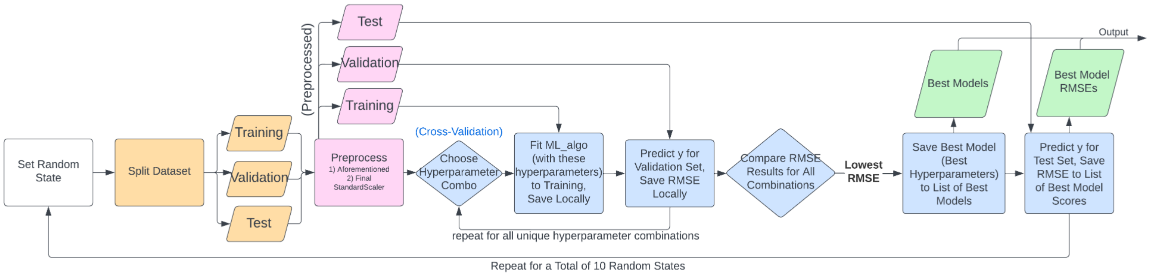
Set Random State
Split Dataset
Training
Validation
Test
(Preprocessed)
Test
Validation
Training
Preprocess
1) Aforementioned
2) Final StandardScaler
(Cross-Validation)
Choose Hyperparameter Combo
Fit ML_algo (with these hyperparameters) to Training, Save Locally
Predict y for Validation Set, Save RMSE Locally
Compare RMSE Results for All Combinations
Lowest RMSE
Save Best Model (Best Hyperparameters) to List of Best Models
Best Models
Predict y for Test Set, Save RMSE to List of Best Model Scores
Best Model RMSEs
Output
repeat for all unique hyperparameter combinations
Repeat for a Total of 10 Random States

Figure 6: Cross-Validation Pipeline

Lasso: - alpha (log scale)	Ridge: - alpha (log scale)
ElasticNet: - alpha (log scale) - l1_ratio (linear scale)	RandomForestRegressor: - max_features (linear scale) - max_depth (linear scale)
SVR: - gamma (log scale) - C (log scale)	XGBRegressor: - reg_alpha (log scale) - lambda (log scale) - max_depth (linear scale)

Table 1. Hyperparameters for ML Algorithms

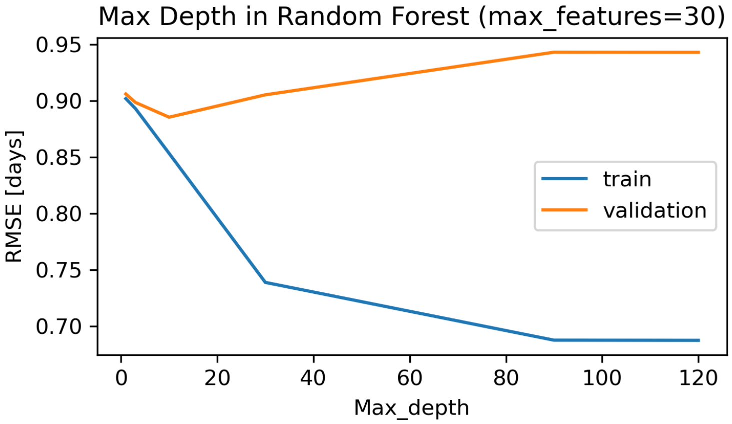

Figure 7: Hyperparameter vs. RMSE

4. Results

The global baseline RMSE 0.9 days when predicting the mean length of stay of 2.3 days for all patients. Across the 10 random states, the baseline RMSE of the test sets was 0.803 +/- 0.024 days. This already-low baseline is a result of the heavily skewed nature of the target variable; despite its range of 1-17 days only 1% of the dataset fell into the 4-17 day range. As a result, it was very difficult for any of the models I trained to learn the patterns for longer hospital stays and beat the baseline. The results of my models are summarized by Table 2 and Figure 8.

RMSE [days]	**Lasso**	**Ridge**	**Elastic Net**	**Random Forest Regressor**	**SVR**	**XGB Regressor**
Mean	0.897	0.864	0.895	0.931	0.946	0.862
Standard Deviation	0.015	0.012	0.011	0.013	0.015	0.014
Ranking by Mean RMSE:						
	4	2	3	5	6	1

Table 2: Results

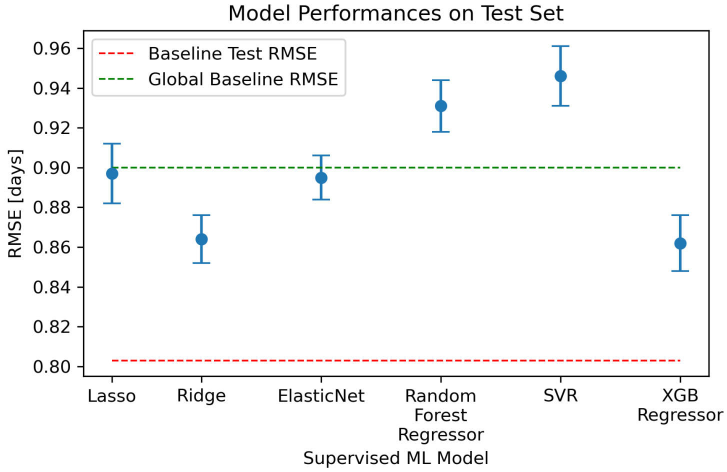

Figure 8: Results

My best model was XGBRegressor, which yielded a test set RMSE of 0.862 +/- 0.014 days. The standard deviation of this result demonstrates the uncertainty due to randomness. Compared to the baseline scores, XGBRegressor performed better than the global baseline but 2.5 standard deviations above the average test set baseline.

3.1 Interpreting Best Model

Since my best model performed worse than the 1:1 baseline, the model would never be deployed and any interpretations of it would be unreliable. However, for the sake of practice, I did work to interpret the XGBRegressor model.

To understand global feature importances, I considered mean SHAP values as well as weight, gain, and total gain from XGBoost (Figures 9-12).

Specific hospitals took many of the high importance ranks across the 4 metrics, which matched the variation I saw in permanent_facility_id in EDA. I was particularly surprised to see hospital metadata take higher importance than patient metadata, although my dataset did have very little patient metadata.

The one-hot encoding of race as white vs. not-white was a feature I was extremely surprised to see in these importances. However, in returning to my EDA I could see that for this dataset there were fewer long-stay outliers for patients who reported their race as white.

I was interested to see that payment types appeared in the metrics: medicaid as primary in SHAP, and unreported secondary and tertiary in gain. This was a correlation I suspected from prior research but wasn't confident would appear.

According to the SHAP values, some specific hospitals, unknown gender, and certain secondary payment types had the least importance in the final model (Figure 13).

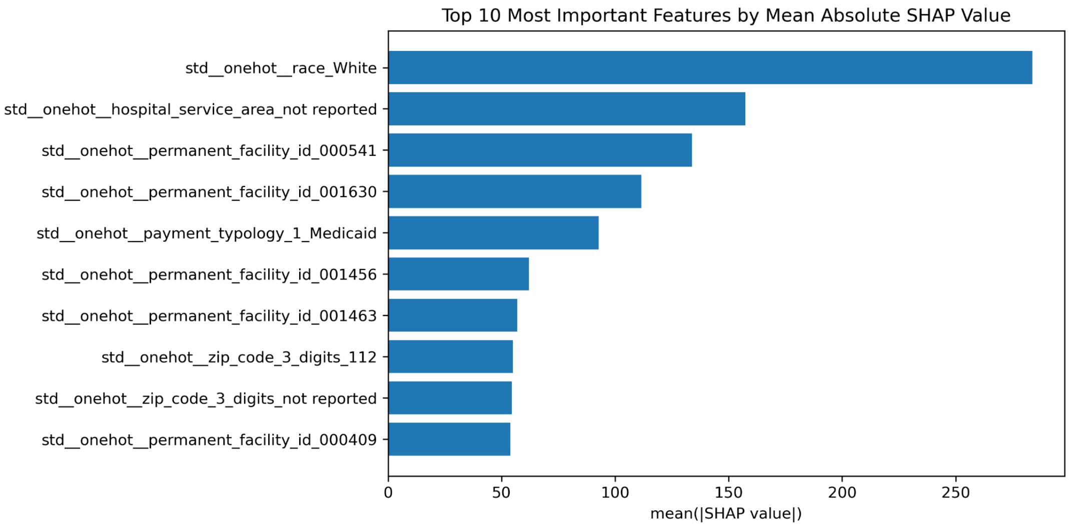

Figure 9: Global Importance, Mean SHAP Value

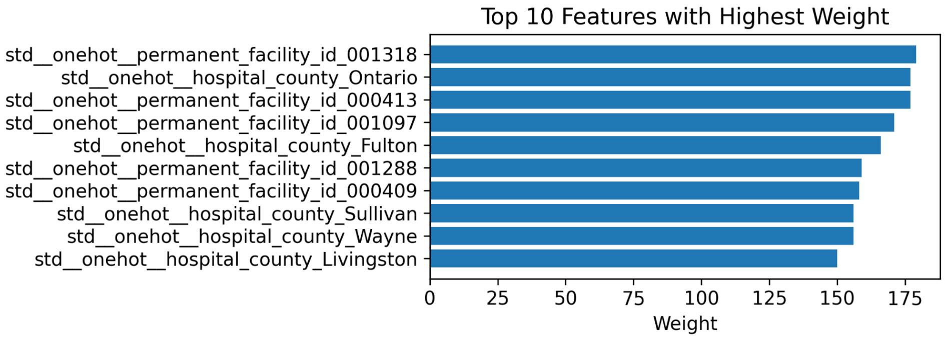

Figure 10: Global Importance, Weight

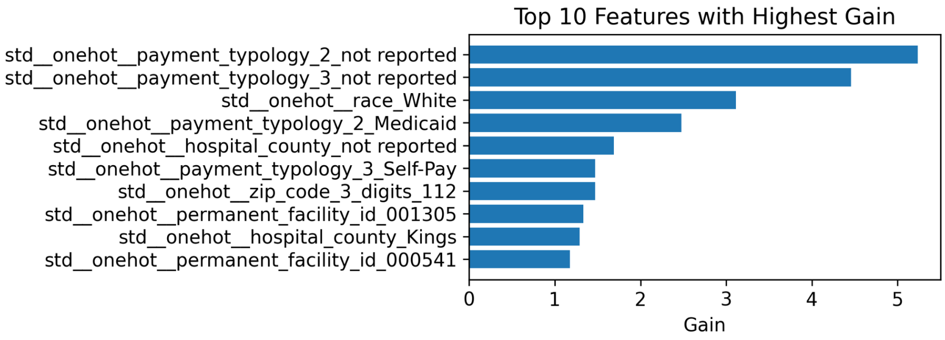

Figure 11: Global Importance, Gain

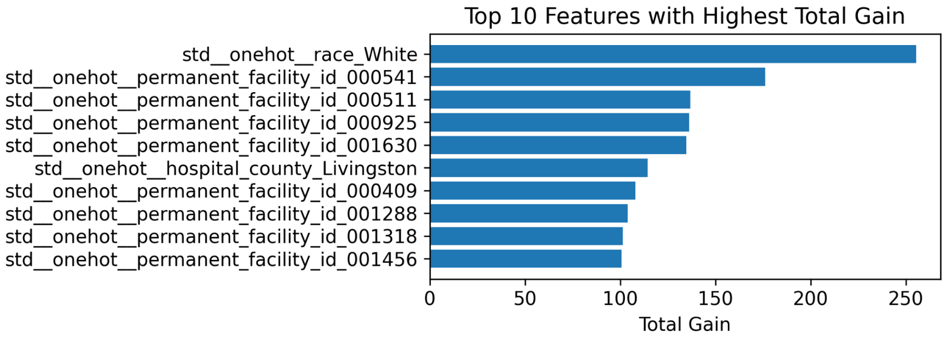

Figure 12: Global Importance, Total Gain

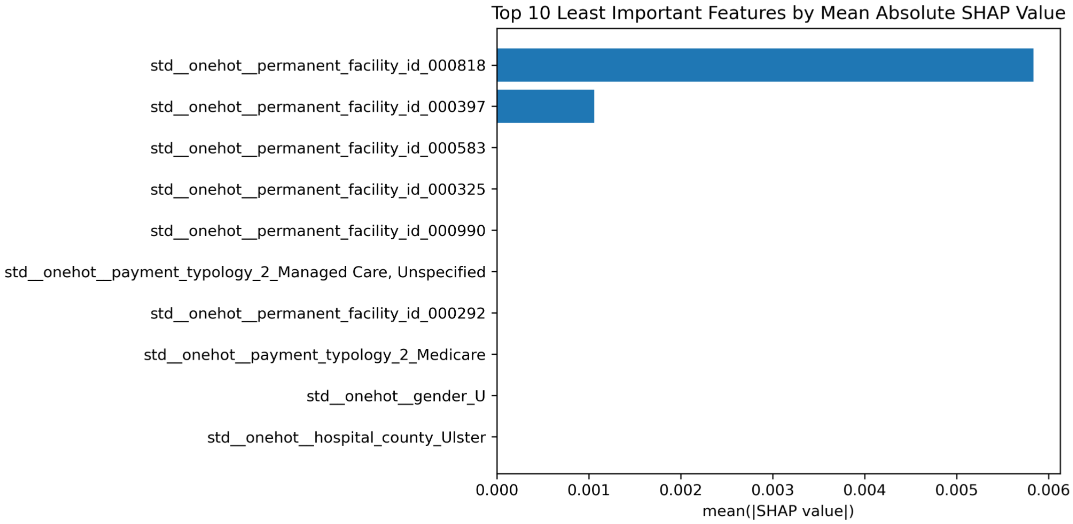

Figure 13: Least Global Importance

I used SHAP's TreeExplainer to determine local importances for example patients in my test set (Figure 14). The locally most important features matched those I saw globally.

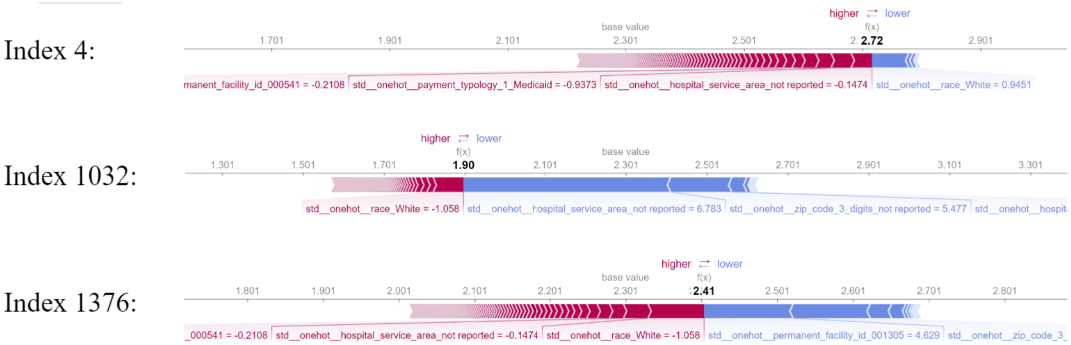

Figure 14: Local Importance

5. Outlook

I see two major paths to improving the power of my model: (1) improving my data with additional patient data or (2) feature engineering with existing features.

5.1 Additional Patient Data

According to experts, a patient's attributes have the greatest effect on the length of their labor and delivery. A woman's first labor is likely to be the longest, with subsequent labors often decreasing in time; knowing which—with historical data for the patient if applicable—could increase model performance. Additionally, the patient's birth plan of vaginal delivery vs. cesarean-section (C-section) could be a valuable indicator of length of stay since different recovery times are associated with each: usually around 2 days for vaginal births and about 4 days for C-sections[4].

Additionally, pre-existing health conditions for mother or baby are risk factors that could lengthen delivery stay. Having this data could help to improve my model's predictions specifically for longer stays. This data would typically be collected at health visits throughout the pregnancy, but were not included in my publically available dataset. This data would also allow me to expand application of the model past exclusively mothers expecting an uncomplicated birth.

5.2 Feature Engineering

Due to the sensitive nature of patient data, acquiring the aforementioned additional data would be difficult. A second route to model improvement without expanding the dataset would be feature engineering.

According to the Center for Disease Control (CDC), insurance coverage of hospital stays has historically affected length of hospital delivery stay[5]. By the Newborns' Act, insurers are only required to cover a certain number of days in the hospital (2 days for vaginal delivery, 4 for C-section)[6]; different insurance plans cover costs differently and different care providers may not take all plans[7]. Moreover, patients without health insurance are in a completely different situation.

As a result, it is likely that combinations of payment types or combinations of payment types with specific hospitals (by permanent_facility_id) may uncover stronger correlations to length of stay than individually. This is a hypothesis that I hope to explore in the future.

6. References

[1] Bousquet, Abigael. "Hospital Delivery Stays." *AbigaelBousquet/Hospital-Delivery-Stays*, GitHub, 8 Dec. 2023, github.com/abigaelbousquet/hospital-delivery-stays.

[2] New York State Department of Health. "Hospital Inpatient Discharges (SPARCS de-Identified): 2021: State of New York." *Hospital Inpatient Discharges (SPARCS De-Identified): 2021 | State of New York*, 1 Dec. 2022, health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/tg3i-cinn.

[3] Zherebtsov, Danil. "Verstack 3.8.12 Documentation." *Verstack 3.8.12 Documentation*, 2020, verstack.readthedocs.io/en/latest/#stratified-continuous-split.

[4] U.S. DOH. "Labor and Birth." *Office on Women's Health*, U.S. Department of Health & Human Services, 22 Feb. 2021, www.womenshealth.gov/pregnancy/childbirth-and-beyond/labor-and-birth#f.

[5] CDC. “Longer Hospital Stays for Childbirth.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 6 Nov. 2015, www.cdc.gov/nchs/data/hestat/hospbirth/hospbirth.htm.

[6] U.S. Department of Labor, Employee Benefits Security Administration. “FAQs about Newborns’ and Mothers’ Health Protection.” *Employee Benefits Security Administration*, U.S. Department of Labor, www.dol.gov/sites/dolgov/files/ebsa/about-ebsa/our-activities/resource-center/faqs/nmhpa.pdf. Accessed 8 Dec. 2023.

[7] “Health Insurance and Childbirth: How Much Is Really Covered When Having a Baby?” *eHealth*, eHealth, www.ehealthinsurance.com/resources/guide/everything-you-need-to-know-about-health-insurance-and-pregnancy. Accessed 8 Dec. 2023.

7. GitHub Repository

https://github.com/abigaelbousquet/hospital-delivery-stays