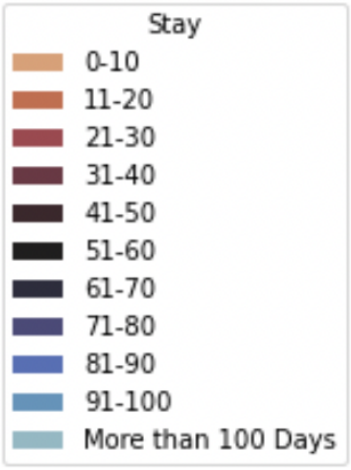
Hope Frost
Capstone project

Hospital Stay Prediction Report

Hospital Management

Management in hospitals, holds a lot of responsibility and they need to have answers questions such as: How long will a patient be in the hospital? Will the hospital have enough resources to meet that patient's needs? Every patient and their family will be intensely concerned with getting home and healthy. It is, however, imperative that Hospitals can plan for how long a patient will be admitted.

HealthMan, a non-profit focused on the management and functioning of Hospitals, has posted a dataset on Analytics Vidhya. Hospitals have to constantly shuffle people and resources to meet every patient's needs. With the recent pandemic, many people died simply because hospitals didn't have enough resources. If the hospital knows how long a patient will need care they can plan their use of resources to save stress on patients and staff; while allowing them to optimize the care and treatments given.

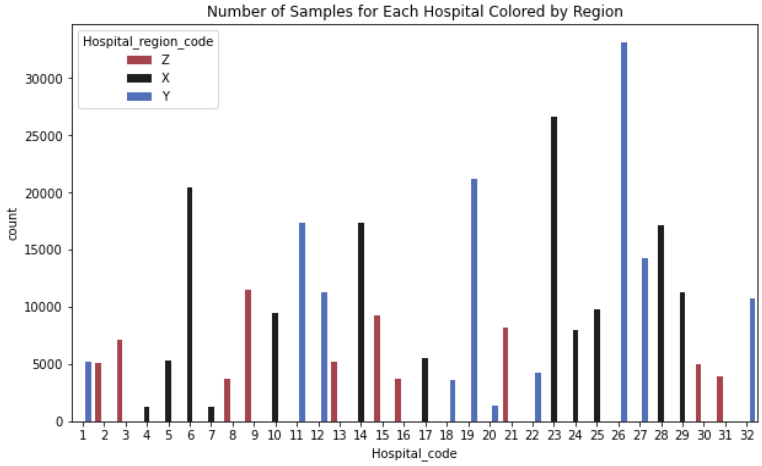


Data Wrangling

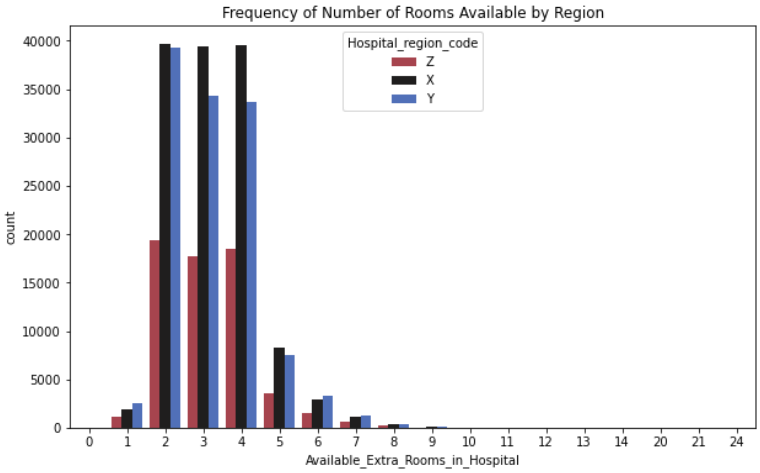
The train set contains 16 features we can pass to a model plus the target feature for the length of stay containing eleven 10-day ranges for classification. See classification ranges in the Stay chart. The data consist of 318,438 samples of admittance data to create and evaluate models. The final test set predictions can be submitted to the hackathon for final accuracy scores on both a public and a private dataset.

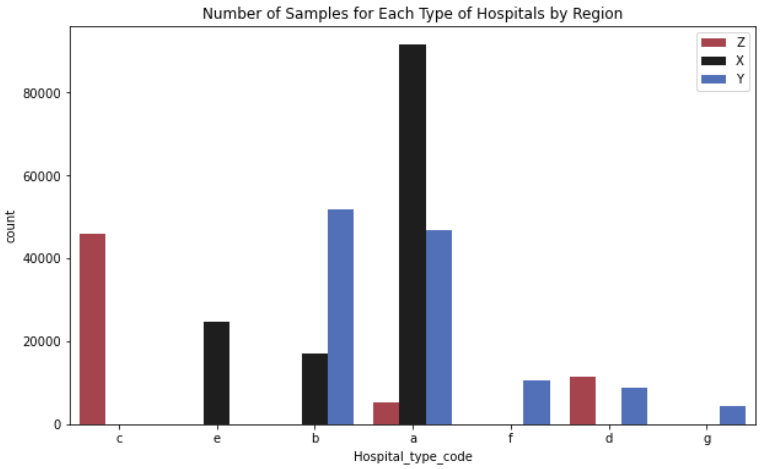
Only two columns had missing values to impute. Bed Grade has been filled with the mode from the train set. For City_Code_Patient we assume that the patient lives in the city the hospital is in and fills the missing value with City_Code_Hospital.

Inconsistency in column names doesn't affect the data but to make it easier to work with the spaces have been replaced with underscores. For privacy, the dataset has been processed by replacing names and type descriptions for the hospitals with letters and numbers. This makes it hard to



determine the meaning behind some of the data.



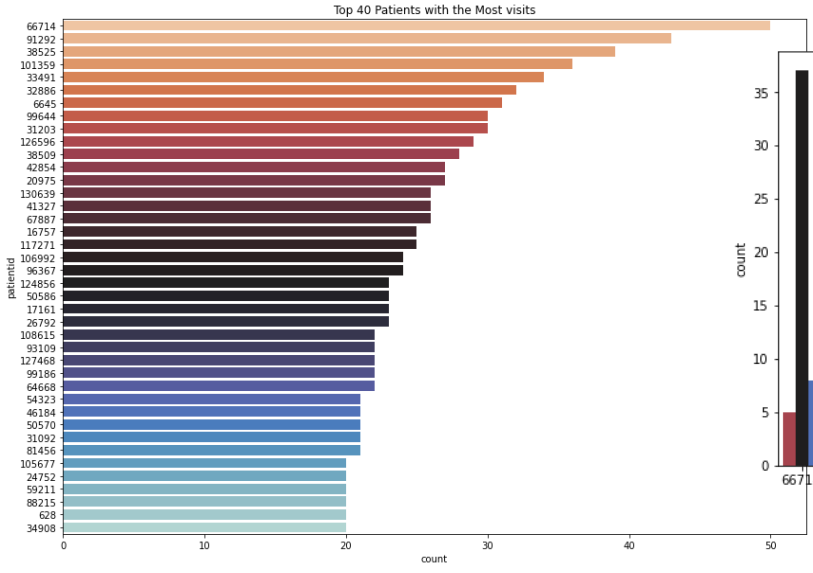


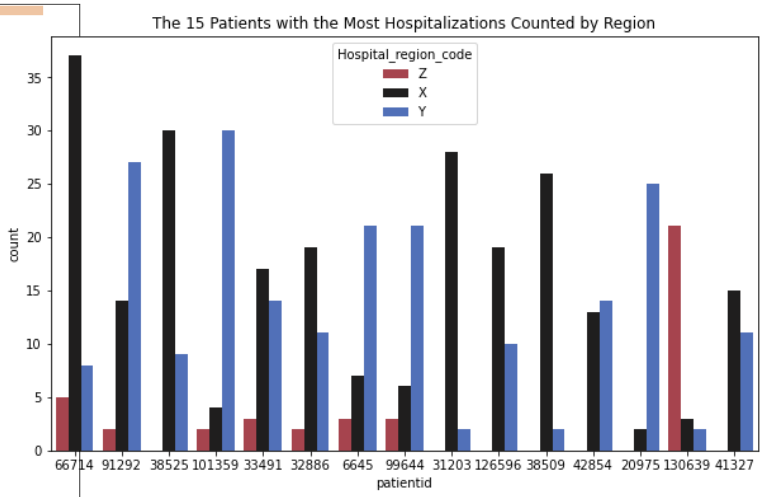
EDA

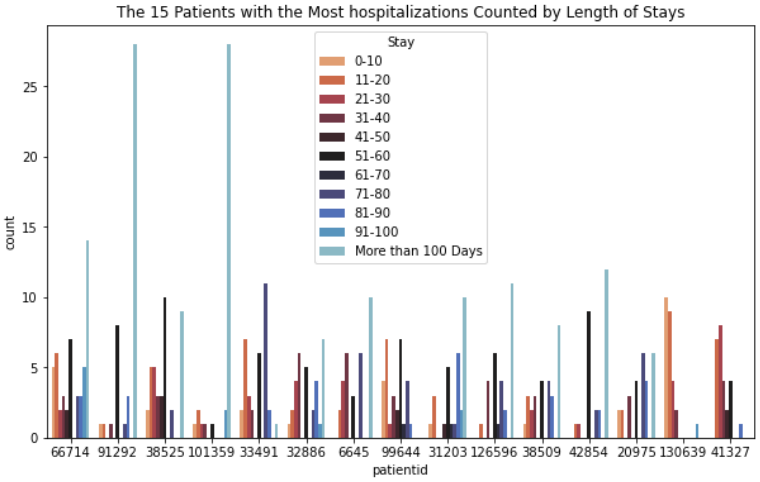
The dataset has 32 hospitals from 3 different regions with several descriptive features for the hospitals and a couple for the patients' admission assessment. The Regions are X with 12 hospitals, Y and Z both with 10 hospitals. As you can see in the "Number of Samples for Each Hospital Colored by Region" plot and "Frequency of Number of Rooms Available by Region", Region Y has more hospitalizations and the capacity to handle them than Z. If we had more information on the regions such as population or area we could deduce more about the size of the hospitals in region X. By far hospital 26 in region Y is the busiest hospital.

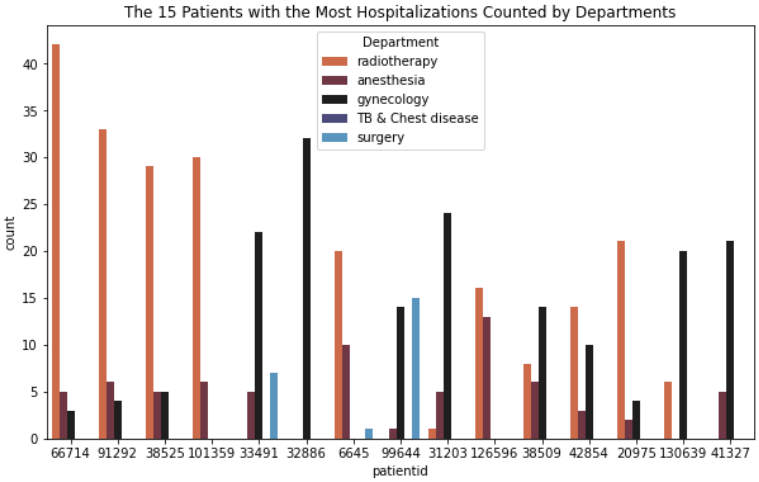
When examining the type codes in "Number of Samples for Each Type of Hospitals by Region" only type code 'a' is in all 3 regions. Type 'g' has the fewest hospitals and is only in region 'Y'. Types 'c', 'e', and 'f' are also only found in one region. This may mean patients have to travel to these hospitals for specialized care. Even though the type code suggests different kinds of hospitals for the most part they seem to all have the same departments except for maybe surgery. Many plots show the majority of visits were to the gynecology department.

Every visit is a unique sample, but the dataset has no dates to know how many years of records were collected. In examining the patient information many patients have had multiple hospitalizations. The plot



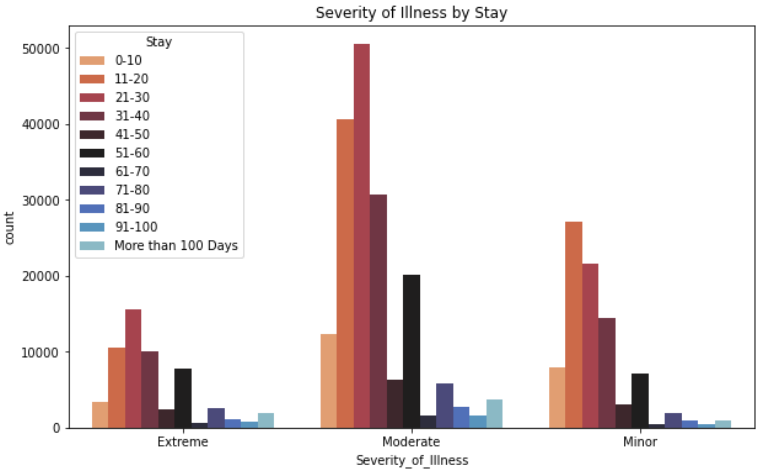






"Top 40 patients with the Most visits" shows that some patients have a lot of hospitalizations. For a single patient to have 20-50 visits to the hospital could be a decade of really sick years or a whole lifetime of patient records. "The 15 Patients with the Most Hospitalizations Counted by Region," is a closer look at the top 15 patients from "Top 40 Patients with the Most Visits." Two patients have almost 30 stays in the hospital that were more the 100 days long. Assuming none of these can overlap the data must cover at least 8 years of records. There is only one 10-year age range for each patient. This could indicate the system is updating age to the patient's current age range or that the data only contains one decade worth of records. It's hard to read this dataset because it has been manipulated so much to provide privacy. Continuing the investigation looking at "The 15 Patients with the Most Hospitalizations Counted by Length of Stays," patients with lots of hospitalizations have been to hospitals from multiple regions. This may indicate travel to see a specialist or the patient has moved to another region during the time the data was recorded. Looking at "The 15 Patients with the Most Hospitalizations Counted by Departments," Radiotherapy visits would support that the patient was very sick and needed to see a specialist.

Throughout the investigation of this dataset, the target feature Stays repeatedly shows a long tail distribution. The "Severity of Illness by Stay" plot shows this distribution and particularly highlights the drop in samples for the 41-50 day, and 61-70 day range, with a pickup between them. Any model used is likely to struggle with correctly predicting these classes. With this in mind as well as the processing the dataset has gone through stripping the



data adding useful features may be important.

Feature Engineering

Before adding any new features, the train set has been split into a test set and a train set, to ensure no leakage as new features are imputed. Each of the sets of new features were defined in a function with a function that calls each, so that all of the features are added to the dataset with one line. This setup allows new data to be processed quickly. The features added to the dataset fall into one of 3 types of features. Count features that impute a count of how many samples have that value for the feature from the train set. This function is passed a list of 9 new features to count. Feature by feature counts occurrences of a feature by another feature. This feature counts rooms available, bed grade, or patient visitors by a list of seven other features, adding 21 new features. The last function adds just one new feature, Patient Traveled. If the patient traveled to get to the hospital it may increase the time needed to recover for discharge. This is a binary feature with one representing that the patient lives in a different region than the hospital. In total 31 new features are added to the dataset.

Preprocessing and Modeling

Random Forest requires the least preprocessing, using bootstrapping to avoid overfitting decision trees to the train set. Independent features are used for decision boundaries so scaling and standardizing are not required. If other models are needed the best option for this dataset would be to pipeline the standard scaler for the quantitative features. Since the original dataset only had 3 quantitative features, and the imputed features were mostly counted features, Random Forest should handle the data well without scaling. The categorical features are one-hot encoded. The Y labels are encoded for the model using Sklearn Label Encoder.

The Analytics Vidhya hackathon score is based on 100% accuracy. At the time the contest closed, the highest accuracy achieved was 43.908% with 19638 submissions

	precision	recall	f1-score	support
0-10	0.45	0.11	0.18	5912
11-20	0.41	0.50	0.45	19349
21-30	0.42	0.66	0.52	22041
31-40	0.42	0.24	0.30	13734
41-50	0.00	0.00	0.00	2933
51-60	0.41	0.50	0.45	8752
61-70	0.00	0.00	0.00	721
71-80	0.43	0.01	0.03	2526
81-90	0.41	0.22	0.28	1220
91-100	0.36	0.01	0.01	719
More than 100 Days	0.54	0.45	0.49	1703
accuracy			0.42	79610
macro avg	0.35	0.25	0.25	79610
weighted avg	0.40	0.42	0.38	79610

entered during the time of the contest. With the default settings, Random Forest predicts 38% accuracy on the test. Tuning the Hyperparameters with random search brings this accuracy rate to 42.103% on the test set and 52.92% on the train set. This tells us the model is not learning enough from the train set. The Random Forest model predicts 0 test samples in the two classification ranges we expected it to struggle with. Overall it received the lowest scores for the ranges of 41-50, 61-70, and 91-100, with the highest precision scores for 0-10, 71-80, and More than 100 Days. The predictions this model gives on the final test set received a public accuracy score of 42.001% and a private accuracy score of 41.705%. If the contest was still going it would be submission rank 266. This ranks the model in the top 1.5% of competitors.

The point of the project is to assist Hospitals in predicting how long a patient will be admitted so they can plan efficient care. Most of the miss classifications are one 10-day range off. While a 42% accuracy may seem low, almost 70% of predictions are within a 20-day range. The model is producing a ballpark for hospitals to plan off. While the model accuracy on 10-day ranges has room for improvement, a 20-day error is a decent starting place. Hospitals can plan for an 11-30 day range vs a 71-90 day window.

Further Steps

It is possible to create more features, but features imputed on repeat patients will not generalize well to new data with new patients. A lot of information has been stripped from this dataset for privacy making it hard to impute methodical features. To increase our accuracy additional information is needed. Date of admission, or at least age at the time of admission would be very helpful. Information on regions such as population, area, and size of hospitals would also be beneficial.

While standardizing and scaling with a pipeline to try other models is an option, the train scores and the hackathons leaderboard indicate that models are not learning enough from this dataset. While a combination of models could catch what one model is missing, more data is key to developing a better prediction.