Patient Length of Stay Predictor

Team: Shaili Vemuri and Ahrian Vemuri
Dataset:
https://www.kaggle.com/datasets/bhautikmangukiya12/hospital-inpatient-discharges-dataset

Project Goals and Scope:

1. Create a model that predicts the length of stay of a patient in a hospital based on various factors from the Kaggle dataset.
2. Create 3-5 high quality visualizations that represent key relationships in the data
3. Learn the complete process for a project from finding the dataset to creating the model

About the Dataset:

The dataset was found in Kaggle and is titled Hospital Inpatient Discharges Dataset. It was created by Bhautik A. Mangukiya, a data scientist that works at Tech Up Labs. The most recent update was in 2024. The author, Mangukiya, writes that “This dataset offers a comprehensive collection of hospital inpatient records aimed at predicting the length of patient stays. With features covering demographic details, medical history, admission types, and more, it provides a rich resource for healthcare analytics and data-driven decision-making”.

He also states that “this data file contains basic record level detail for the discharge. The de-identified data file does not contain data that is protected health information (PHI) under HIPAA. The health information is not individually identifiable; all data elements considered identifiable have been redacted. For example, the direct identifiers regarding a date have the day and month portion of the date removed.”

In total, there are 33 columns and over 2,101,598 rows of data. All of the data is from hospitals based in the state of New York.

Choosing Features:

For predicting the length of stay (LOS), we selected 11 key features from the original 33 columns in the dataset, along with the LOS column as our target variable.

Target Column:

- **Length of Stay** (target): The number of days a patient stays in the hospital, which we aim to predict.

11 Chosen Features:

1. **Age Group**: We hypothesized that age would influence the length of stay, with older patients generally staying longer.
2. **Gender**: Gender-based trends were considered, as there may be differences in hospital stay length.
3. **Type of Admission**: The type of admission (e.g., emergency, elective) directly impacts how long patients stay.
4. **CCSR Diagnosis Description**: The type of diagnosis (e.g., flu, heart attack) strongly correlates with LOS.
5. **CCSR Procedure Description**: Different procedures often require varying recovery times, affecting LOS.
6. **APR DRG Description**: This provides information on the patient's primary condition and resource needs, which influence the length of stay.
7. **APR Severity of Illness Description**: The severity of the patient's illness impacts both care required and LOS.
8. **APR Risk of Mortality**: A higher risk of mortality often results in extended hospital stays.
9. **APR Medical Surgical Description**: Whether the patient underwent surgery is critical, as surgery typically leads to longer recovery times.
10. **Emergency Department Indicator**: Patients admitted via emergency services often stay longer due to more complex care requirements and the lack of preparation beforehand
11. **APR MDC Description**: This indicates the body system affected, and certain

2nd Most Important Features:

- **Race & Ethnicity**: Though race and ethnicity can influence disease severity, these factors were less critical compared to medical descriptions.
- **Total Charges & Costs**: These were excluded since they are decided post-discharge and don't directly affect the prediction.
- **Hospital Service Area & County**: Given the data only came from New York hospitals, geographic location didn't significantly impact our model.

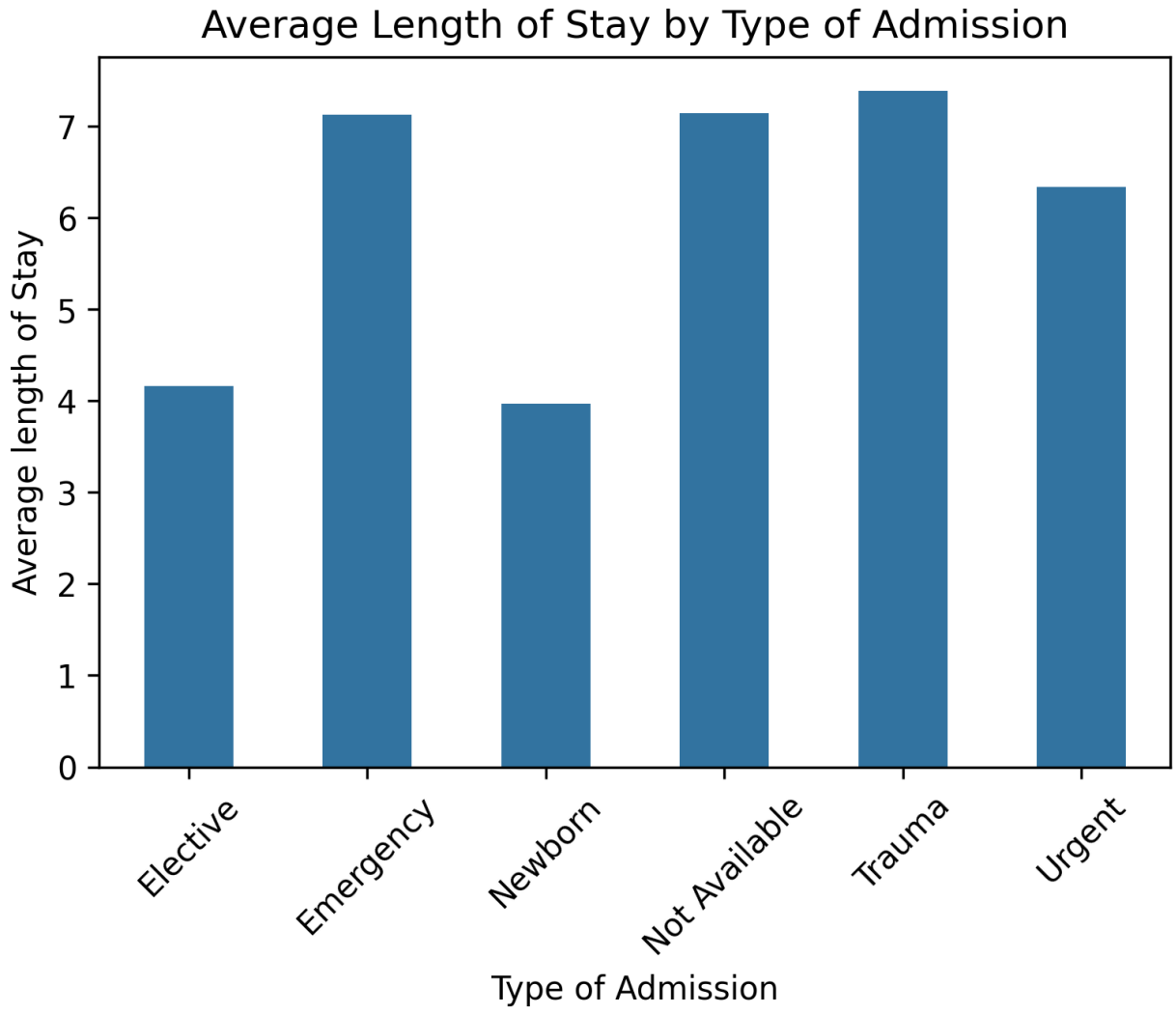
Data Cleaning Steps Taken:

1. Created a new dataframe with specific columns deemed important to the prediction of length of stay. These columns are Length of Stay, Age Group, Gender, Type of Admission, CCSR Diagnosis Description, CCSR Procedure Description, APR DRG Description, APR Severity of Illness Description, APR Risk of Mortality, APR Medical Surgical Description, Emergency Department Indicator, APR MDC Description.
2. Tidy the Columns: Strip extra space, lowercase, and remove symbols. All columns renamed to this format “length_of_stay” instead of “Length of Stay”
3. Drop nulls from the Length of Stay column (target value). There were 0 rows with null length of stay

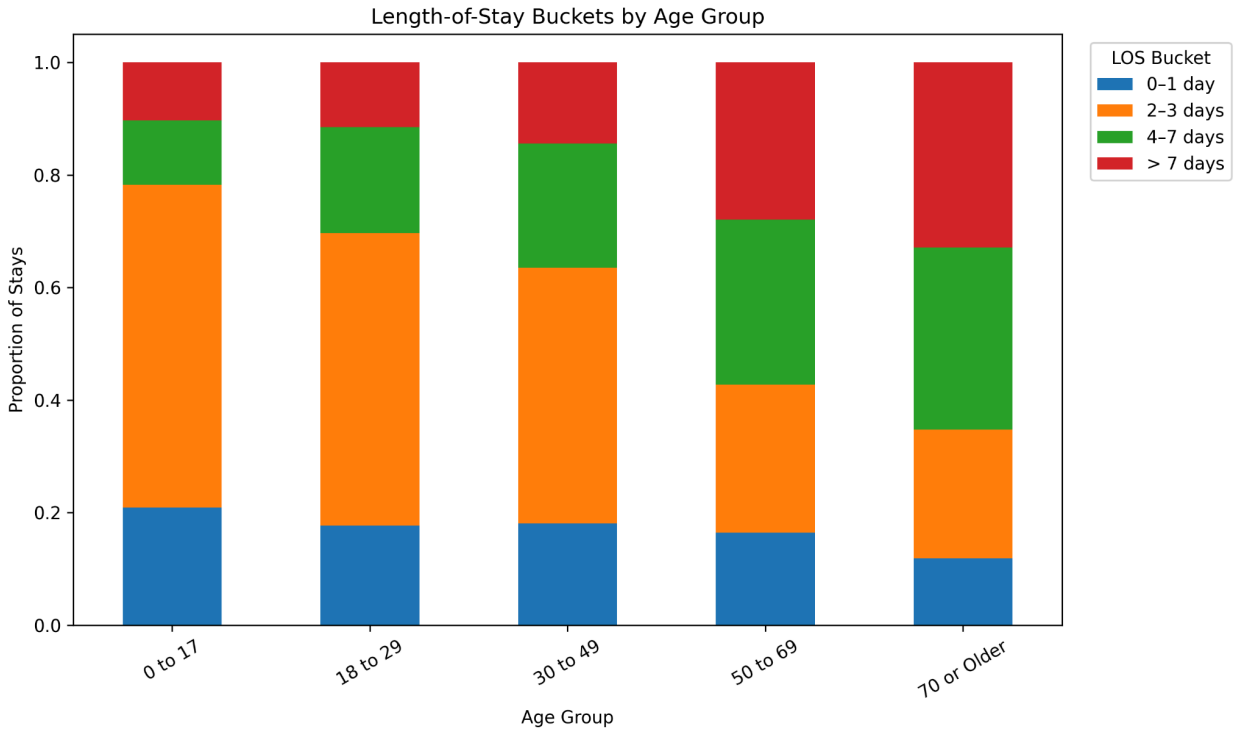
4. Convert Length of Stay column from String type to Numeric type and drop all leftover non numeric rows. 1561 rows were dropped during conversion to numeric
5. Checked if any other columns needed their data types converted. There were none.
6. Dropped all other rows with null values. There were 2100027 rows before dropping nulls and 1522692 rows after dropping nulls.
7. Checked unique values and head() to ensure cleaning working properly
8. Save cleaned data as a new CSV file named “hospital_cleaned.csv”

Data Analysis Visualizations:

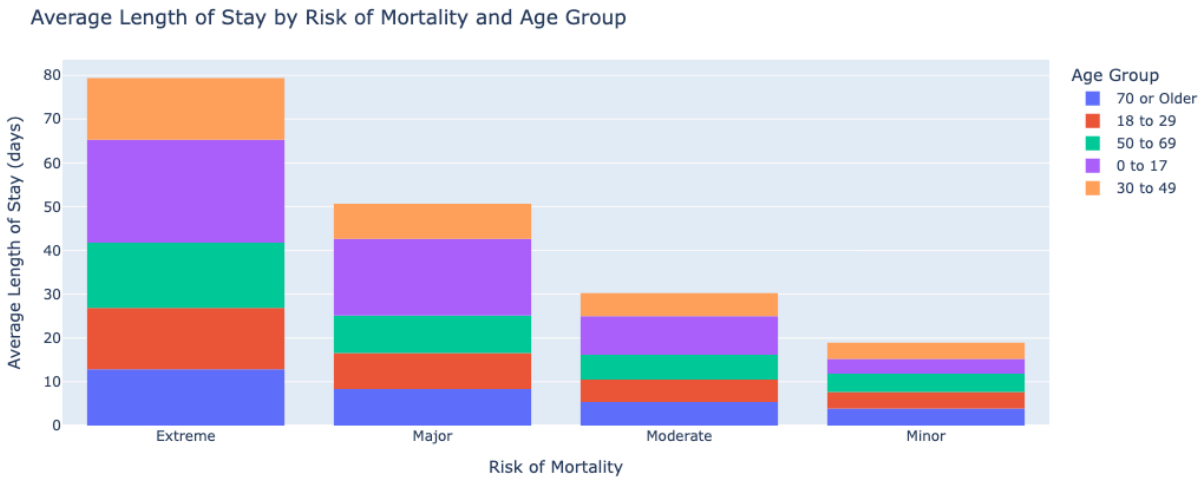
These visualizations show relationships between the target variable, length of stay, and the various features.



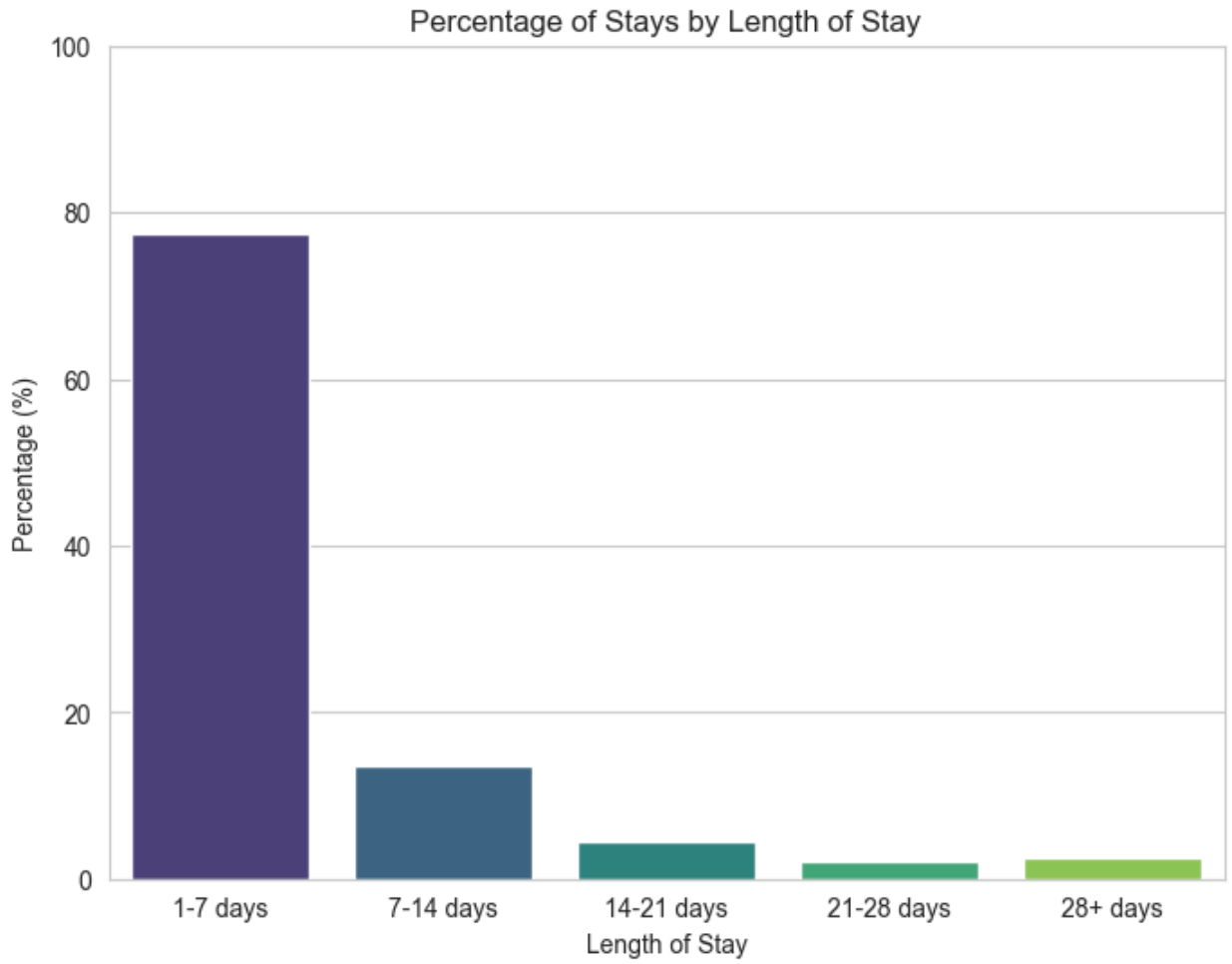
This graph shows that when type of admission is elective or newborn, there is typically a lower length of stay. Additionally, emergency or trauma admissions have a longer average length of stay.



This graph shows the proportion of each age group in 4 different length of stay categories (0-1 day, 2-3 days, 4-7 days, or more than 7 days). This shows that older age groups typically have longer length of stay.



This graph shows that the more extreme the risk of mortality, the longer they will stay in the hospital for all age groups.



This graph shows the percentage of data that is in each range of length of stay (1-7 days, 7-14 days, 14-21 days, 21-28 days, and more than 28 days). This clearly reveals that there is a data imbalance and that the outlier values may be affecting the models' accuracy. This graph encouraged us to test the models with a dataset that only includes rows with a length of stay of 21 days or less. For the following model descriptions, results will be shown for the "hospital_cleaned.csv" (original cleaned data) and/or "hospital_los_21.csv" (cleaned data with length of stay values of 21 days or less).

What Are "Classic" Machine Learning (ML) Models/Algorithms?

These are traditional models that make predictions by looking for patterns and relationships in the data. They rely on mathematical rules and statistics to find the best possible fit between the input features and the target values. Unlike more complex models like neural networks, these algorithms are usually faster to train and easier to understand. They work well when the data is clean, not too large, and the relationships are somewhat straightforward.

Why Did We Test These Models?

We knew we wouldn't ultimately use these models due to their lack of customization and adaptability. We still wanted to test these to set a baseline with how the classic ML algorithms performed with our data.

Classic ML Results:

LinearRegression → MSE: 58.338, R^2: 0.200
RandomForest → MSE: 52.599, R^2: 0.278
GradientBoosting → MSE: 53.689, R^2: 0.263
SVR —> Error: Took long time to run
BayesianRidge → MSE: 58.338, R^2: 0.200
GPR → Error: MemoryError

What Is A Neural Network?

A neural network is a type of computer program that tries to learn how to make decisions by looking at examples. It's inspired by how the human brain works. Instead of being told exactly what to do, the network looks at a lot of data and finds patterns. For example, if you show it enough pictures of cats and dogs, it can learn to tell the difference on its own. It does this through layers of tiny units (called "neurons") that pass information to each other and adjust over time to get better at making predictions.

Why Did We Use It?

We chose a neural network as our primary model for predicting the length of stay due to its ability to understand complex patterns, quality performance with large data, flexibility regarding the type of data (ex: integers, text), and skill in figuring out which features are more important.

First Neural Network Results:

- Features: Age Group, Gender, Type of Admission, APR Severity of Illness Description, APR Risk of Mortality, APR Medical Surgical Description, Emergency Department Indicator
- One Hot Encoding of all features: Splits each features unique values into separate columns and indicates which value using a 1 to show presence and a 0 to show absence.
- Built Sequential model, so each layer has one input and one output. Our model had an input layer, 2 dense hidden layers, and a dense output layer. The first hidden layer had 64 neurons while the second had 32 neurons.
- We also utilized early stopping to prevent overfitting, so the model stops training and restores the best weights when it begins 'memorizing' the training data
- Implemented validation split of 20%, meaning 20% of the training data was used to test the model at the end of each epoch.

- Evaluated the model with Mean Squared Error and Mean Absolute Error
- The best settings from our experiments for this neural network was running for 10 epochs with a batch size of 32, and a patience for early stopping of 3. This was the original settings, and all of our changes worsened the model's performance
- Hospital_cleaned.csv → `Test MSE: 52.611, Test MAE: 3.741`

Second Neural Networks Results:

- We kept the same structure as the one above for this neural network but continuously tweaked and refined various things to achieve the best possible settings
- Here is what type of things we changed
    - Number of epochs
    - Features
    - Batch size
    - Layers and layer settings in our sequential model
- Hospital_cleaned.csv → `Test MSE: 38.479, Test MAE: 3.065`

Third Neural Networks Results:

- We kept the same structure as the one above for this neural network but experimented with numerically encoding certain categories
- 3 columns were changed to numeric instead of categorical, which were "age_group", "apr_risk_of_mortality", and "apr_severity_of_illness". However, only converting "age_group" to numeric produced the best results for numeric encoding.
- This was the best model when there was no numeric encoding, batch size of 32, less than 21 days dataset, all features used, one input layer, two layers.Dense(64, activation='relu'),one layers.Dense(32, activation='relu'), and one output layer.
- Hospital_cleaned.csv → `Test MSE: 37.994, Test MAE: 3.083`
- Hospital_los_21.csv → `Test MSE: 9.644 , Test MAE: 2.055`

What is XG boost?

XGBoost is a powerful type of machine learning model that makes smart predictions by combining many simple decisions. It builds a series of small models (like decision trees), where each one learns from the mistakes of the ones before it. Over time, this process helps the overall model get more accurate. It's especially good at handling messy or complex data and is known for being both fast and accurate, which is why it's popular in real-world projects and competitions.

Why Did We Test It?

We tested XG boost due to its reliability, consistency, and good reputation for being known as an accurate and powerful type of AI model. Though we achieved better results with neural networks, XG boost closely rivalled our best neural network's accuracy.

XG Boost Model Results:

- Results Recap: In the first go with less features (top 7 most important) we got our worst score in MAE and MSE. The consistent theme throughout all of our models proved true when we added more features and got a better score. When converting features to numeric datatypes we surprisingly got worse results than leaving the features as is with the same settings in both tests. After tweaking the settings for a while, we tested again with extra features and numeric data types where possible and got worse than just leaving at default
- Critical steps we did to improve results: One hot encoding, set various conditions to optimal values such as n estimators, learning rate, max depth, and early stopping rounds
- Hospital_cleaned.csv → `Test MSE : 38.487, Test RMSE: 6.204, Test MAE : 3.151`
- Hospital_los_21.csv → `Test MSE : 9.779, Test RMSE: 3.127, Test MAE : 2.103`

What is a RNN?

Recurrent Neural Networks (RNNs) are a special type of neural network designed to work with data that comes in a sequence — like sentences, time series, or audio. What makes RNNs unique is that they remember information from earlier in the sequence and use it to help understand what comes next. This "memory" helps the model make better predictions when the order or context of the data matters.

Why Did We Test It?

We knew the RNN was not going to give the best results because our data was not sequential or fully text based. We were curious to see how a sequential based model could perform with our data, so we tried it anyway.

Recurrent Neural Networks Results:

- Features: Age Group, Gender, Type of Admission, CCSR Diagnosis Description, CCSR Procedure Description, APR DRG Description, APR Severity of Illness Description, APR Risk of Mortality, APR Medical Surgical Description, Emergency Department Indicator, APR MDC Description
- One Hot encoding of all features

- Built Sequential model with an input layer, LSTM layer, a Dense layer with 32 neurons, and Dense layer for output
- This model takes in the data as a sequence of different time points so that earlier data inputs has an effect on later ones
- We utilized early stopping and validation split again as well as tracked the Mean Absolute Error
- The best results so far is with the original settings (only one tested)
- Hospital_cleaned.csv → `Test MAE: 3.084`

Conclusion

In conclusion, we were able to predict the length of stay for a patient in a hospital with a margin of error of 2.055 days using the third neural network when no features were encoded numerically and the 21 and less dataset was used. Also, there was a batch size of 32, one input layer, two layers.Dense(64, activation='relu'), one layers.Dense(32, activation='relu'), and one output layer. Predicting the length of stay for patients in a hospital is important to manage resources such as bed capacity and staff scheduling as well as improving overall patient satisfaction and care.