Springboard Final Capstone Report:

Predicting the Length of Stay in Hospital among Schizophrenic and Other Psychotic Disorders Patients Using Machine Learning

By **Ranjana Roka**

1. Problem Statement

The length of hospital stay (LOS) is a critical measure in healthcare, particularly in psychiatry, as it impacts both the quality of care and resource allocation. The length of stay is defined as: **"The total number of patient days at an acute level and/or other than acute care level (excluding leave of absence days)".** Prolonged hospitalizations among patients with schizophrenia and psychotic disorders can increase healthcare costs and affect patients' mental health outcomes. This project aims to apply machine learning techniques to predict the LOS of such patients, enabling healthcare professionals to manage resources and develop personalized discharge plans proactively.

Predicting LOS is crucial for hospital resource management, personalized care planning, and improving patient outcomes. Early identification of patients with potentially extended stays could help healthcare providers plan interventions and allocate resources more efficiently. This capstone project applies machine learning techniques to predict LOS for patients diagnosed with schizophrenia and psychotic disorders.

2. Data Wrangling

In this project, data wrangling was a crucial step in cleaning and preparing the dataset for further analysis. The dataset used in this project is a large open healthcare dataset provided by the New York State Statewide Planning and Research Cooperative System (SPARCS) containing 2.3 million de-identified patient records, which includes various patient, demographic, and clinical features relevant to length of stay. I sourced the dataset from the following link:

https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/tg3i-cinn/data_preview

Initially, the dataset was loaded using the pandas library from a CSV file and explored using various Python built-in features. we get an overview of the dataset, including the number of rows (2135260), columns (33), data types. The dataset includes both numerical and categorical features such as age group, gender, race, ethnicity, diagnosis, severity of illness, type of admission etc with the target variable length of stay. All necessary data wrangling steps were performed to prepare the data for analysis and modeling.

1. Column Filtering: Columns with over 50% missing values and irrelevant columns (e.g., Birth Weight, Operating Certificate Number, Emergency Department Indicator) were dropped.

2. Handling Missing Values: Missing values in numeric columns were filled with the median, and categorical columns were mapped to numerical labels for ease of processing.

3. Encoding Categorical Variables: Non-numeric features, such as Gender, Age Group, Race, and Ethnicity, were converted to numerical values using mappings.

After that, the final check was done to ensure that there were no remaining missing values and that the dataset was clean for further analysis.

3. Exploratory Data Analysis

After the data had been collected and cleaned, the next step that I performed was data exploration and analysis. This helps me to understand the data's structure, visualize relationships between variables, and detect patterns that might contribute to length of stay. Visualizations and correlation matrices were employed to understand feature relationships, with the following findings.

a. ***Top Diagnoses and DRG:*** Plots of the top diagnoses and diagnosis-related groups highlighted the frequent clinical issues related to the LOS.

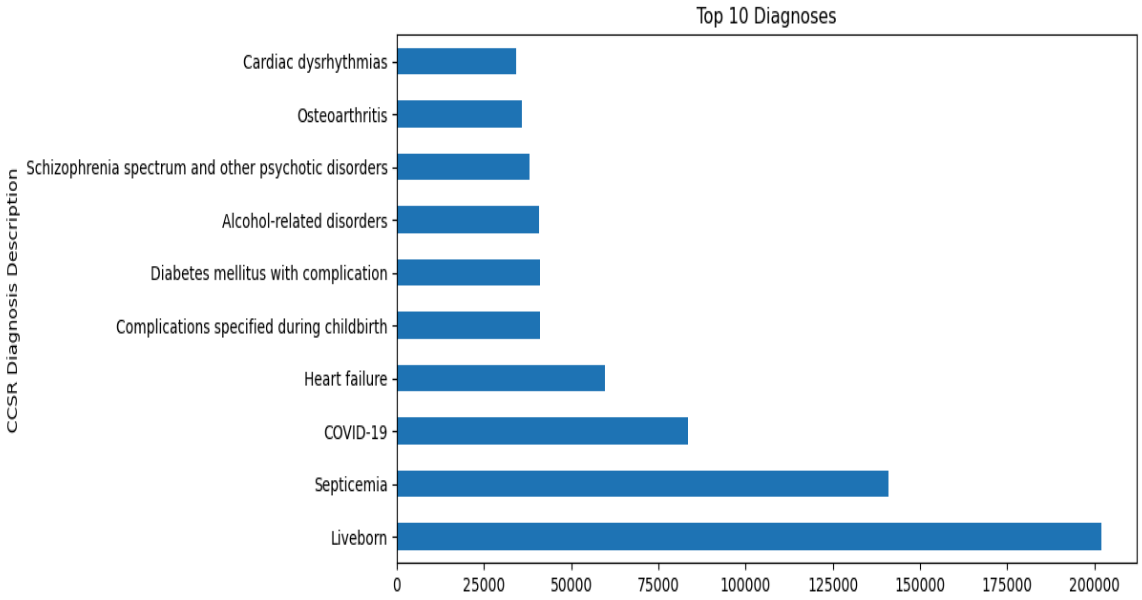


Figure 1. Bar plot of top 10 diagnosis

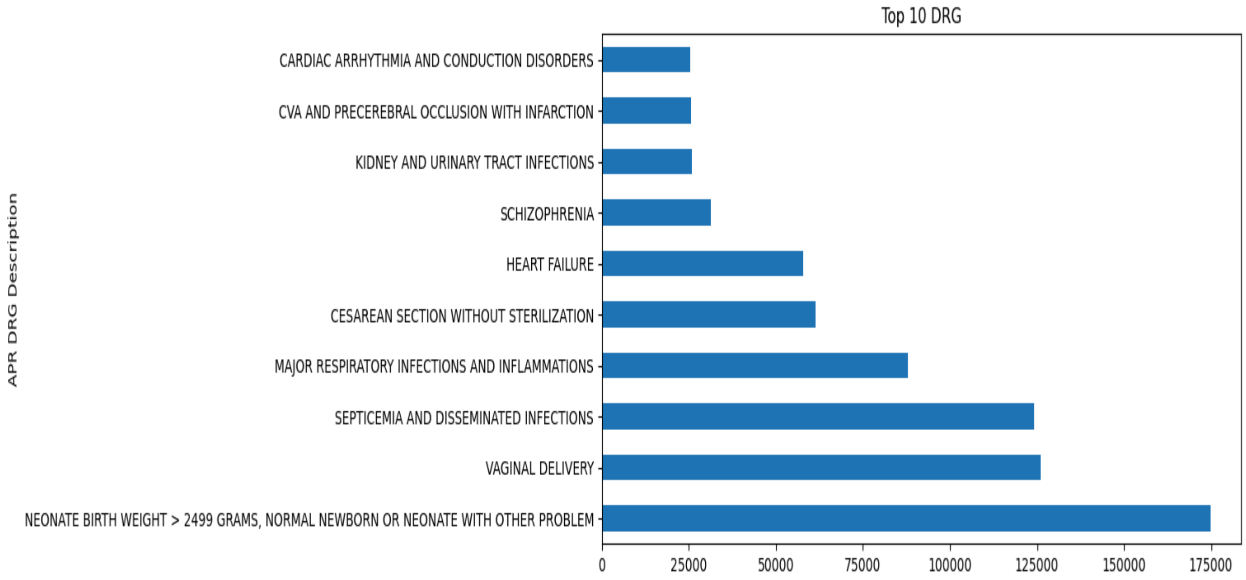


Figure 2. Bar plot of top 10 diagnosis related group

After identifying the top 10 diagnoses, I filtered the dataset to include only rows where the "CCSR Diagnosis Description" was "Schizophrenia spectrum and other psychotic disorders." This created a dataset focused solely on schizophrenia patients, aligning with the main topic I aimed to study. Next, I separated numeric and categorical columns within this filtered dataset and ensured no missing values by verifying both types of columns. After completing the preprocessing steps, I confirmed that all missing values had been addressed, preparing the dataset for the next phase: applying machine learning algorithms to predict the length of stay for patients with schizophrenia and psychotic disorders.

b. Distribution Analysis

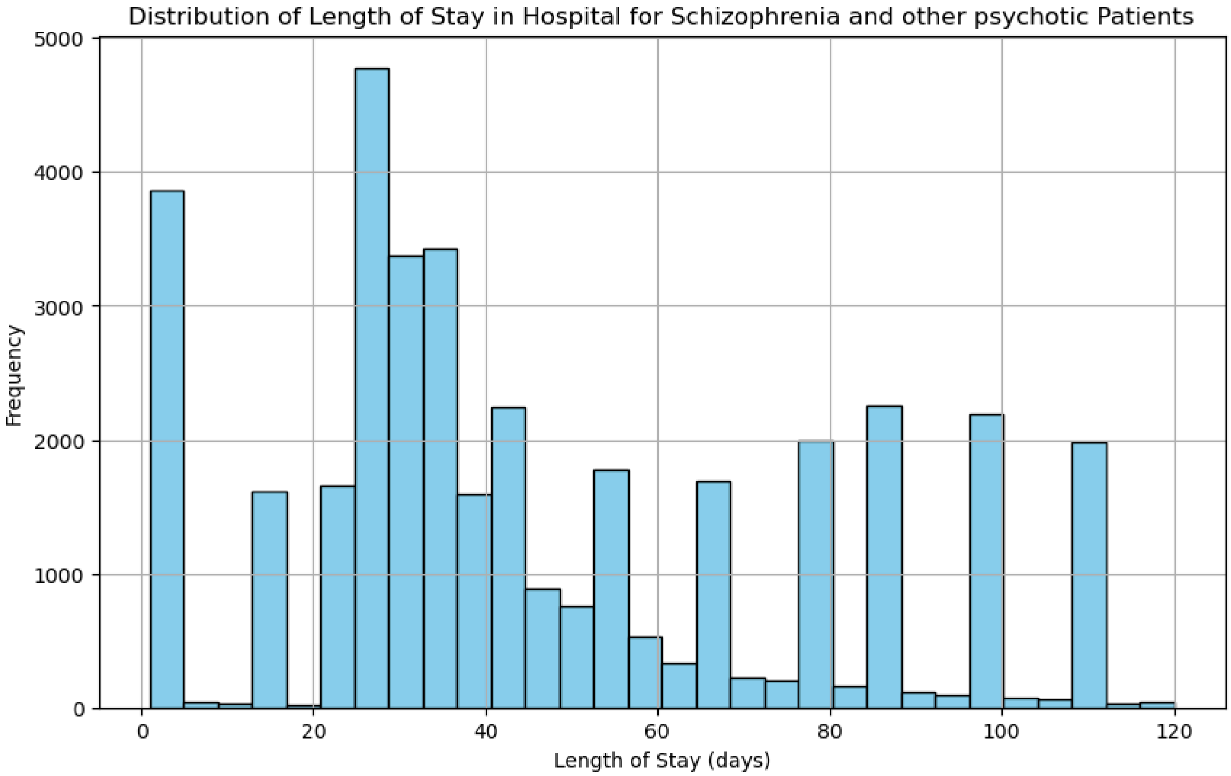


Figure 3: Histogram illustrating length of stay distribution

This visualization provides useful insights into how long patients with these conditions typically remain in the hospital. The X-axis represents the number of days patients with schizophrenia and other psychotic disorders stayed in the hospital. The length of stay ranges from 0 to over 120 days, grouped into bins. The Y-axis shows the number of patients (or occurrences) for each bin of days in the X-axis. It represents the frequency of patients staying for a certain duration in the hospital.

The histogram demonstrates that most patients with schizophrenia or psychotic disorders tend to stay in the hospital for around 20 to 40 days, with a gradual decrease in the number of patients as the length of stay increases. There are a few patients with stays extending beyond 100 days.

c. Correlation Analysis:

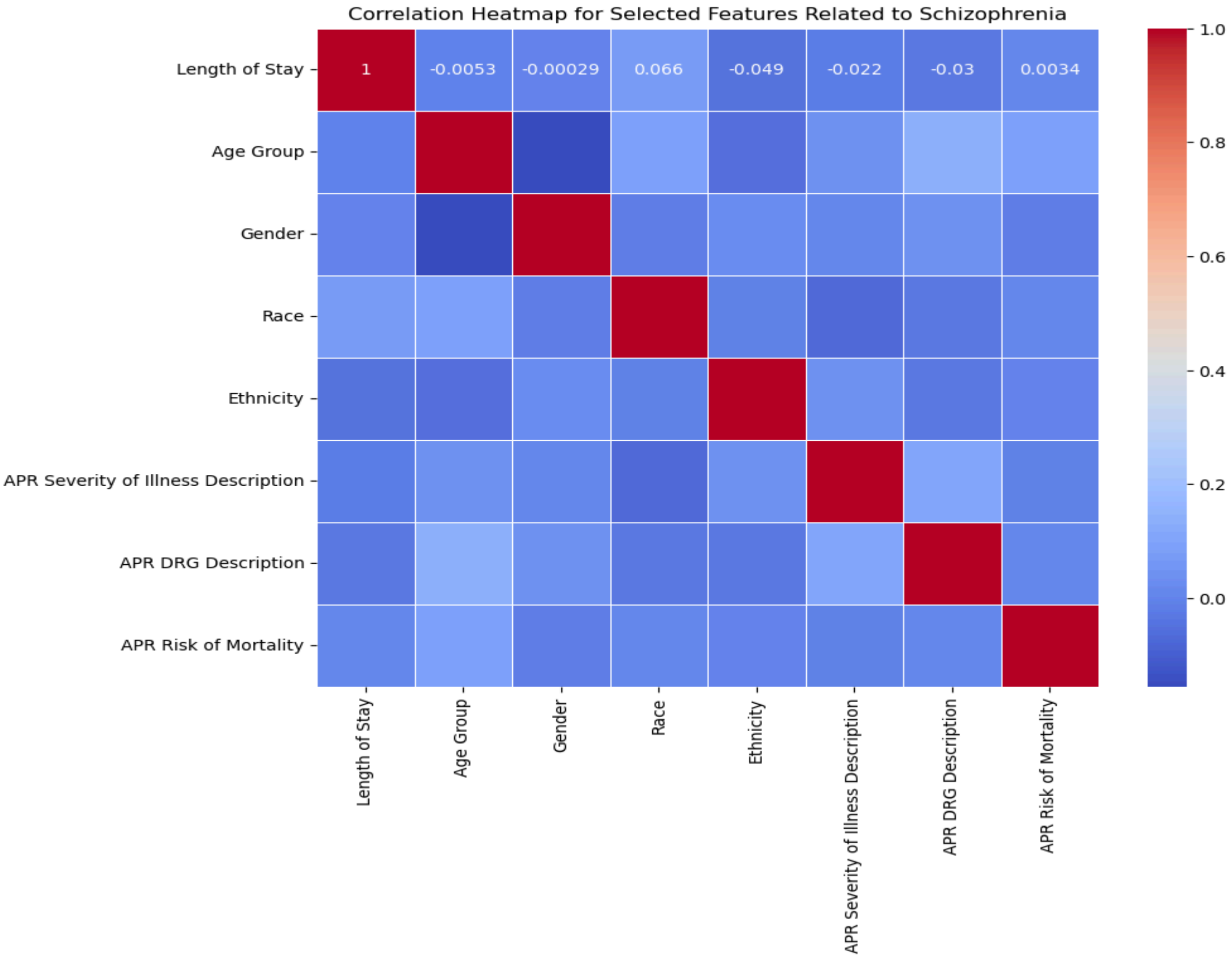
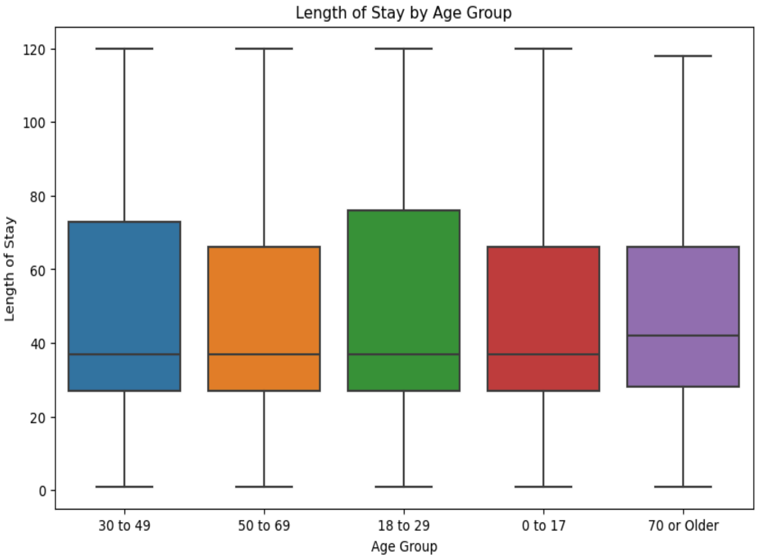
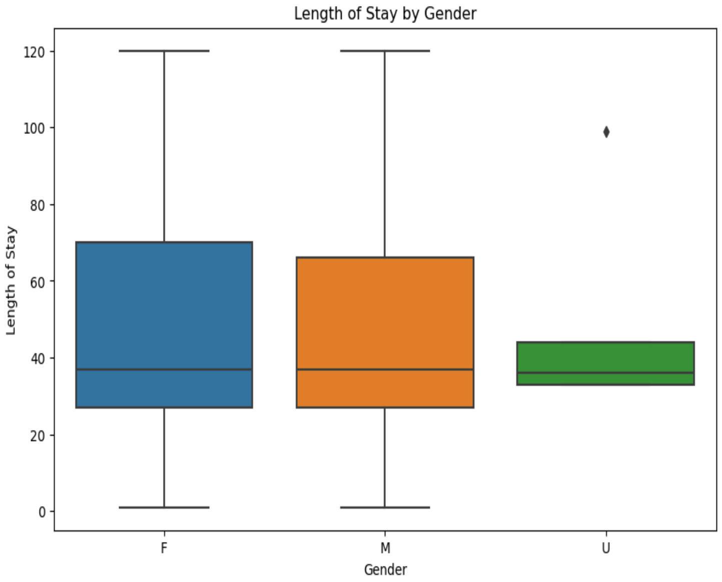


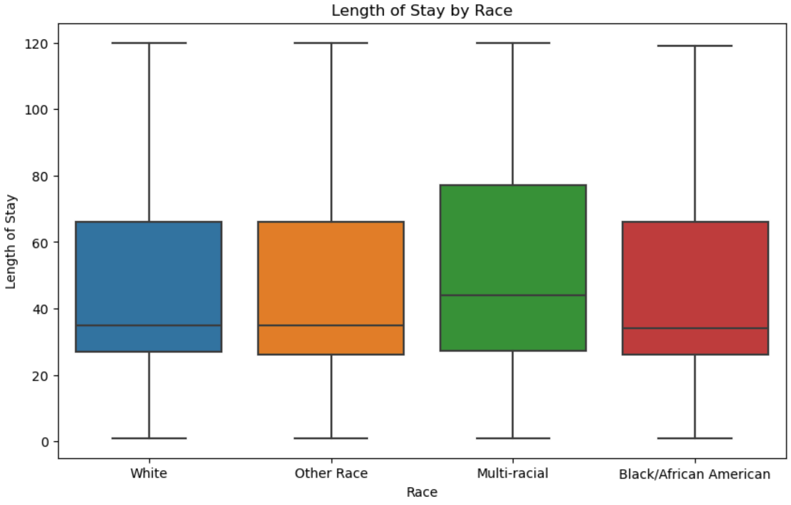
Figure 3. Visualizing the Correlation Matrix with Heatmap

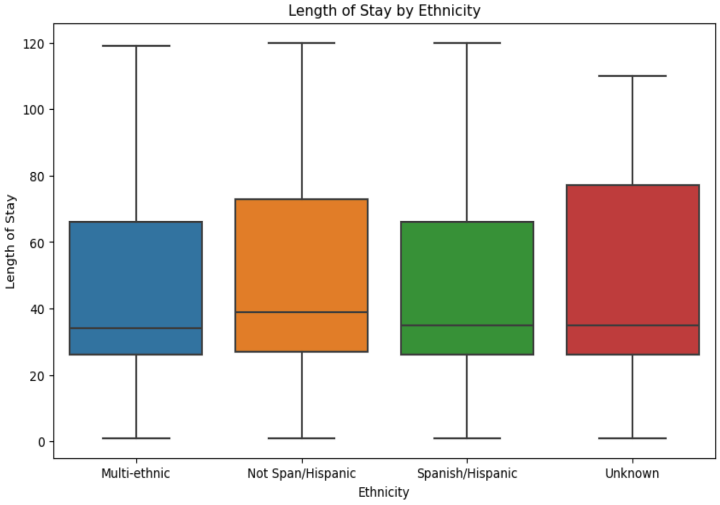
A heatmap visually represents the correlation matrix, making it easier to identify strong and weak correlations at a glance. This heatmap shows that there is very little correlation between Length of Stay and the other variables, except for some slight relationships between features like APR DRG Description and APR Risk of Mortality. For predicting Length of Stay, the correlations indicate that these particular features may not be strong linear predictors, but non-linear relationships might still exist, which could be explored through other machine learning techniques.

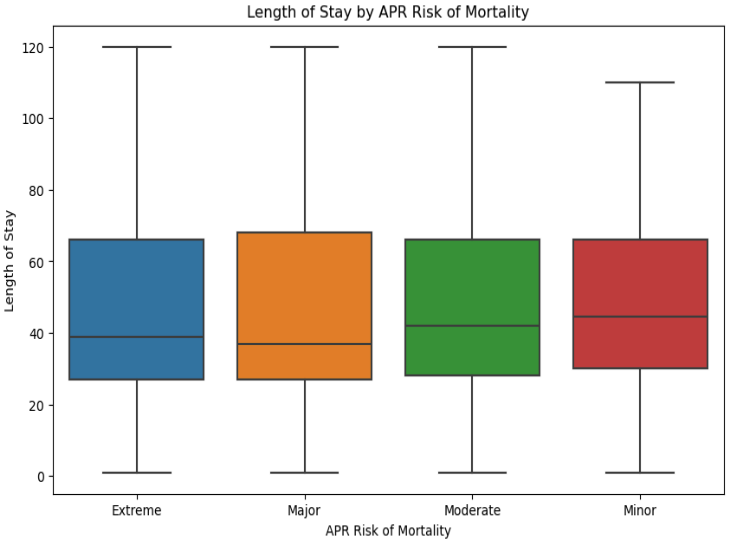
Feature Selection: I created a subset DataFrame containing only the relevant columns and generated a boxplot to visualize the distribution of the length of stay.

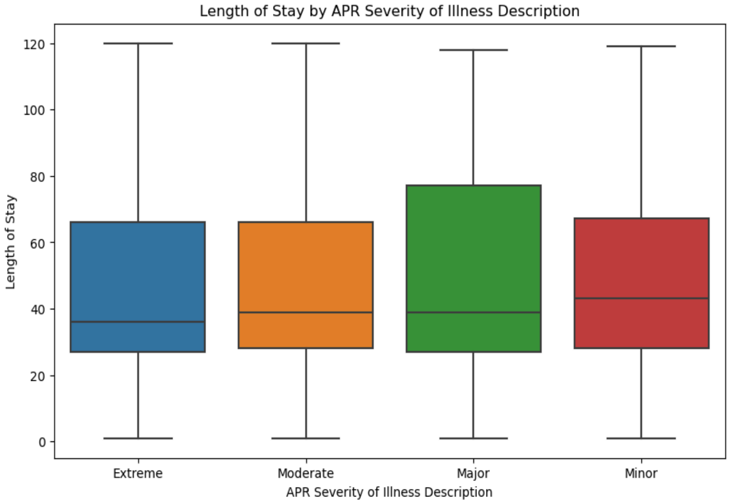


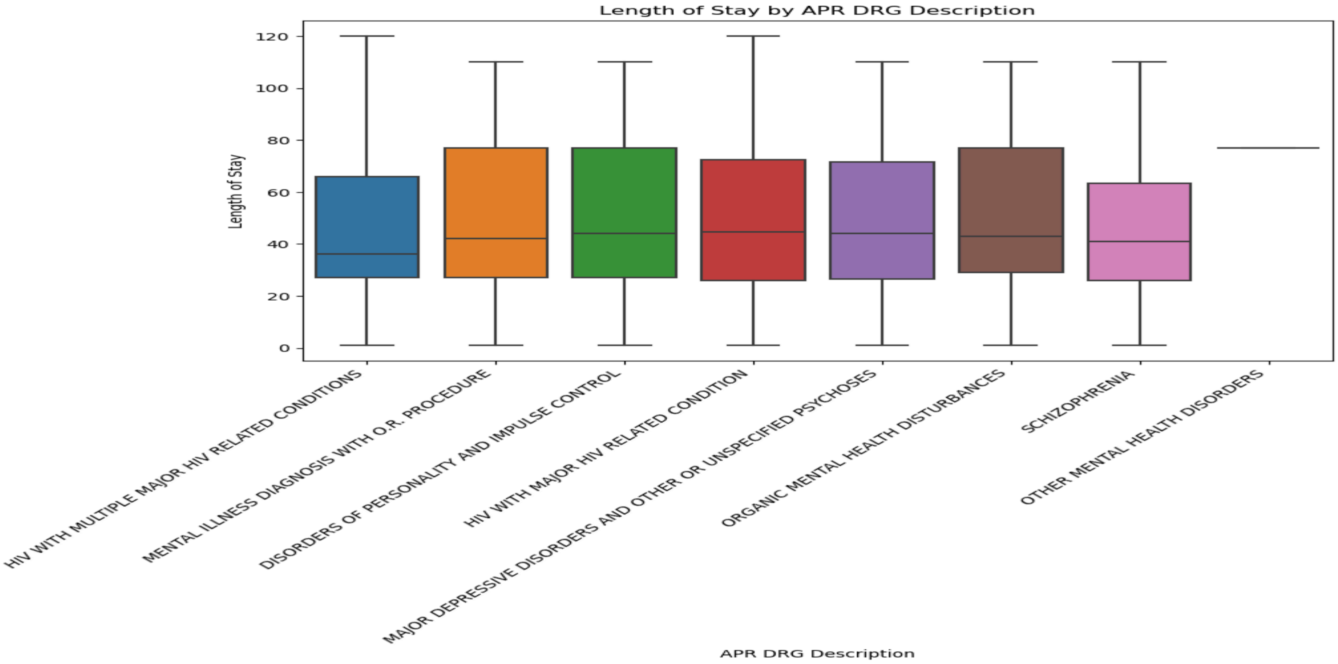












This boxplot shows the distribution of length of stay (LOS) across different features. It indicates that factors like age, gender, race, ethnicity, diagnosis-related groups, risk of mortality, and illness severity do not show strong individual influences on LOS in this dataset. This reinforces the need to explore complex, multi-factor interactions in predictive modeling.

Modeling and Evaluation

Modeling Process

To build predictive models for hospital length of stay (LOS), I explored various algorithms and evaluated their performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Here's a step-by-step breakdown of my approach:

1. Splitting the Data

- I began by defining the features (X) and the target variable (y). The Length of Stay column was set as the target variable.
- Then, I split the data into training and testing sets with an 80-20 split to ensure that we have sufficient data for both model training and evaluation.

2. Data Scaling

- Since we have features with varying scales, I applied Standard Scaling to the numerical columns. This step standardizes the features, which is especially helpful for algorithms sensitive to feature scales.
- I used StandardScaler from Sklearn to transform the training set and applied the same transformation to the test set to maintain consistency.

3. Model Selection and Training

- I experimented with different regression models, including:
    - **Linear Regression**
    - **Decision Tree Regressor**
    - **Random Forest Regressor**
    - **Gradient Boosting Regressor**

- To automate the process, I placed these models in a dictionary and defined a function, fit_and_score, to train each model on the training set and evaluate it on the test set. This function also calculated the R² score for each model.

4. Model Performance Comparison

After training, I compared the models based on their R² scores:

- ○ **Linear Regression**: 0.02
- ○ **Decision Tree Regressor**: 0.55
- ○ **Random Forest Regressor**: 0.76
- ○ **Gradient Boosting Regressor**: 0.41

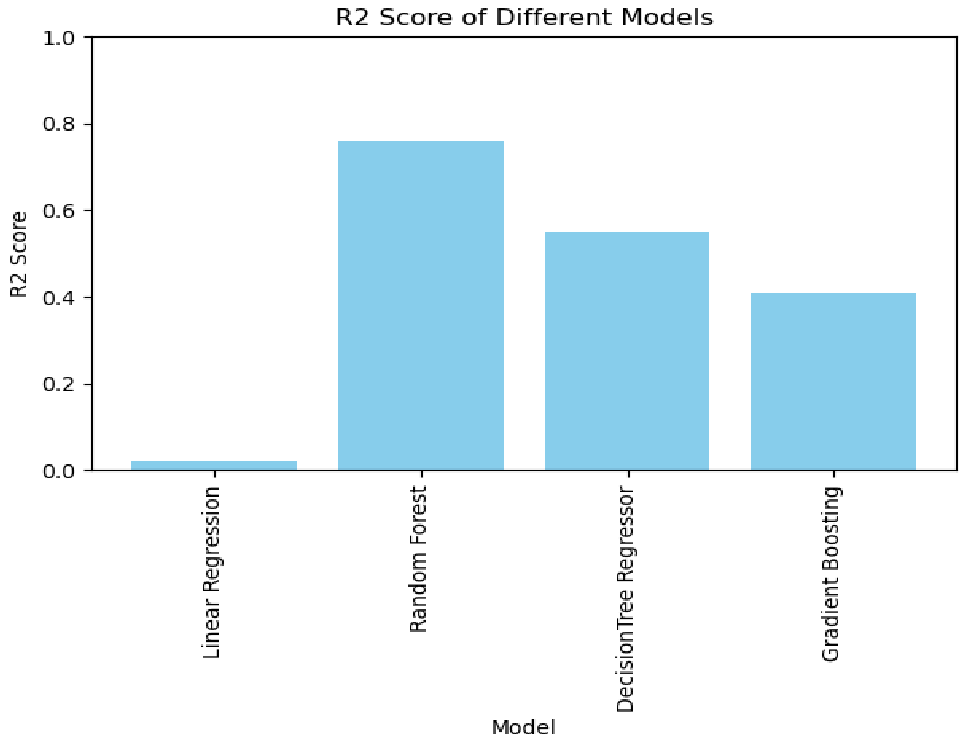


Figure 5: Comparison of R2 score of different models

The **Random Forest Regressor** performed the best with an R² score of 0.76, indicating that it captured the most variance in the LOS data compared to the other models.

5. Evaluating Models with Error Metrics

Beyond R², I also calculated MAE, MSE, and RMSE for each model to get a more comprehensive understanding of prediction errors.

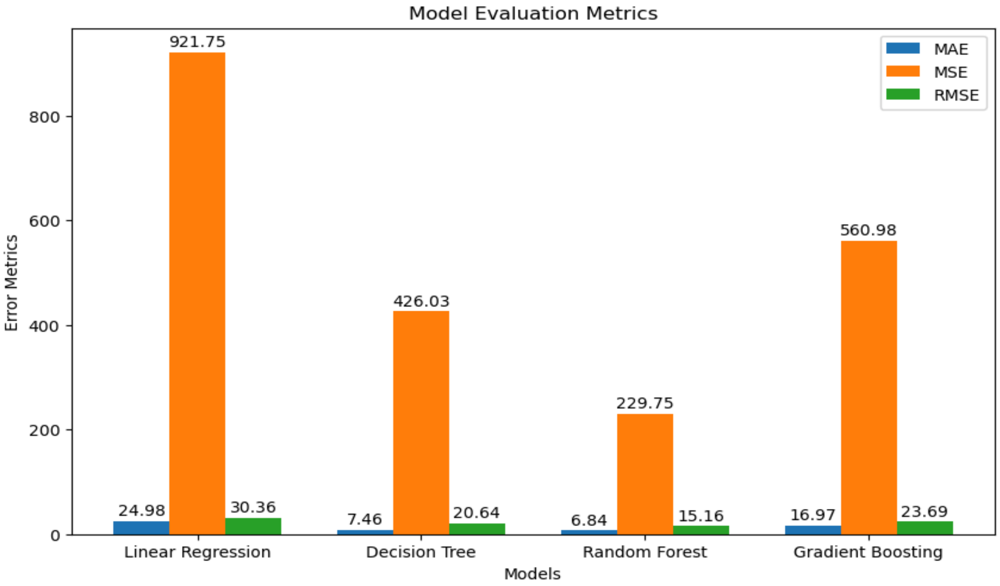


Figure 6: Evaluation model metrics

These metrics helped highlight the accuracy of each model, with Random Forest again showing the lowest error values, reinforcing its strong performance.

- **MAE**: 6.84
- **MSE**: 229.75
- **RMSE**: 15.16

Hyperparameter Tuning

Given the promising results from the Random Forest and Gradient Boosting models, I further tuned their hyperparameters to improve performance.

- I used RandomizedSearchCV to search across a parameter grid for each model. This approach allowed me to quickly explore a range of parameter combinations without exhaustive computation.
- After tuning:
    - The **Random Forest Regressor** achieved the best parameters with an MSE of approximately -301.91 (negated for scoring).
    - The **Gradient Boosting Regressor** achieved an MSE of approximately -370.40.
- I retrieved the best parameters and evaluated the tuned models on the test set, where Random Forest continued to outperform.

Final Model Evaluation

- Using the best hyperparameters, I made final predictions on the test set and calculated MAE, MSE, and RMSE again:
    - **Tuned Random Forest**:
        - **MAE**: 9.16
        - **MSE**: 280.12
        - **RMSE**: 16.73
    - **Tuned Gradient Boosting**:
        - **MAE**: 11.44
        - **MSE**: 347.60
        - **RMSE**: 18.64

The tuned Random Forest showed better performance in terms of lower errors, making it the final chosen model for this problem.

Evaluating Prediction Accuracy

- To provide additional insight into model accuracy, I calculated the absolute error for each prediction and determined how many predictions were “close” to the actual LOS.
- I defined a threshold of 10 days for "close predictions" and found that about 73.31% of predictions fell within this range.

Conclusion

In summary, after testing multiple models, performing hyperparameter tuning, and evaluating the errors, the Random Forest Regressor was identified as the best model for predicting LOS. It consistently demonstrated higher accuracy across all metrics and showed a substantial percentage of close predictions within a reasonable error margin.