Name: Tanay Patil

Problem Statement: Hospital Readmission Prediction

GitHub Link: https://github.com/tanaypatil14/Hospital-Readmission-Prediction

Problem Statement

Management of lifestyle diseases in hospitalized patients has a significant bearing on outcome, in terms of both morbidity and mortality. Hospital readmissions is a critical challenge in healthcare, reflecting potential inadequacies in post-discharge care and significantly burdening healthcare systems with increased costs and adverse patient outcomes. Proactively identifying patients at elevated risk of readmission can empower healthcare providers to implement targeted interventions and optimize care pathways. This project aims to develop a robust predictive model leveraging patient data to accurately assess the probability of 30-day readmissions. By integrating advanced machine learning techniques, the model aspires to serve as a decision-support tool, enhancing clinical workflows, reducing preventable readmissions, and ultimately contributing to improved healthcare quality and resource utilization.

The main object for this problem is to predict whether a patient is likely to be readmitted to hospital based on the previous details of the patient.

Dataset Description

The dataset contains 17 features and 25000 rows in the dataset. The Column readmitted is the target dataset. The description of each feature is as below:

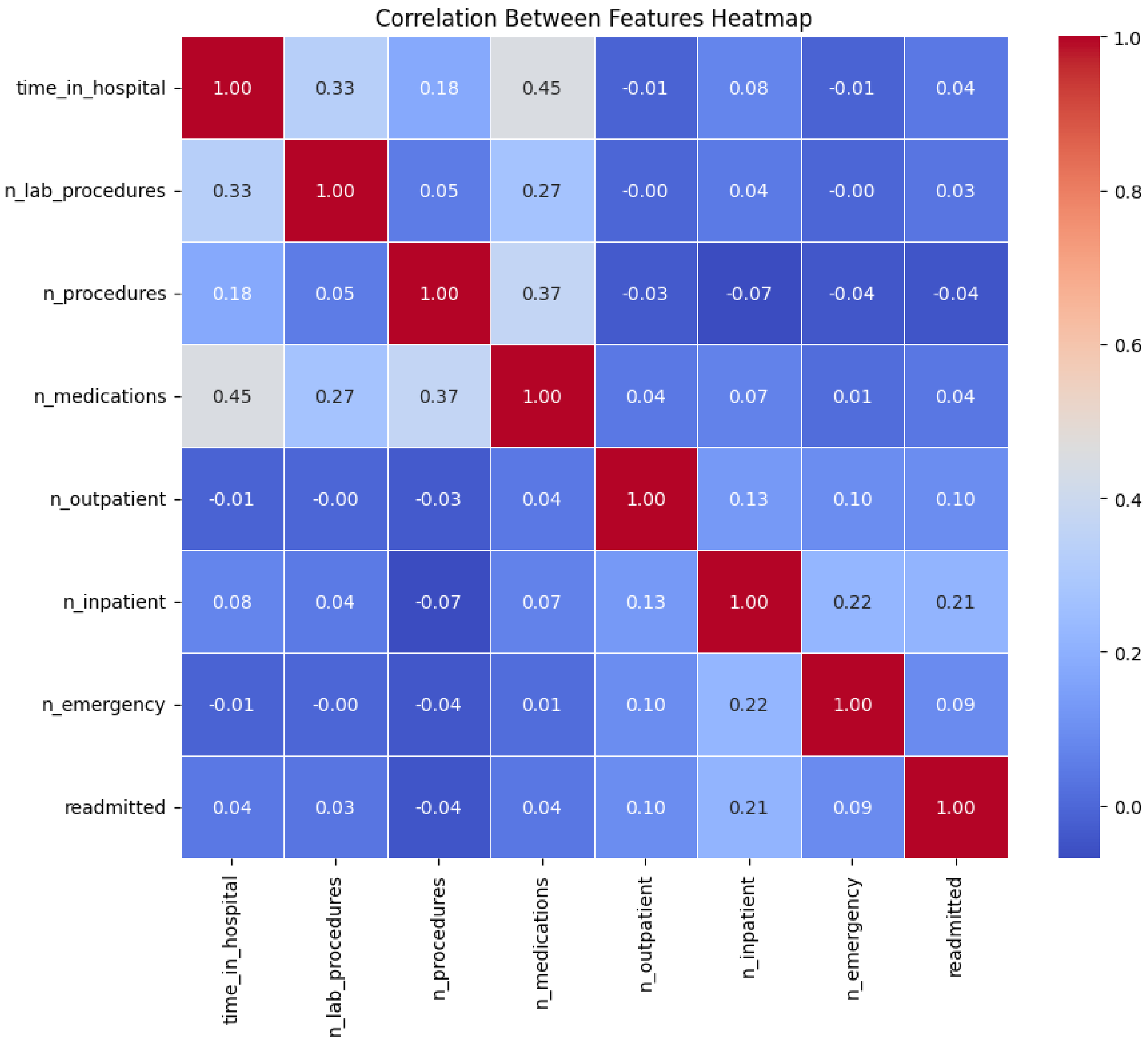
Target attribute: "readmitted" (discrete variable: 2 classes)

- "age" - age bracket of the patient
- "time_in_hospital" - days (from 1 to 14)
- "n_procedures" - number of procedures performed during the hospital stay
- "n_lab_procedures" - number of laboratory procedures performed during the hospital stay
- "n_medications" - number of medications administered during the hospital stay
- "n_outpatient" - number of outpatient visits in the year before a hospital stay

- "n_inpatient" - number of inpatient visits in the year before the hospital stay
- "n_emergency" - number of visits to the emergency room in the year before the hospital stay
- "medical_specialty" - the specialty of the admitting physician
- "diag_1" - primary diagnosis (Circulatory, Respiratory, Digestive, etc.)
- "diag_2" - secondary diagnosis
- "diag_3" - additional secondary diagnosis
- "glucose_test" - whether the glucose serum came out as high (> 200), normal, or not performed
- "A1Ctest" - whether the A1C level of the patient came out as high (> 7%), normal, or not performed
- "change" - whether there was a change in the diabetes medication ('yes' or 'no')
- "diabetes_med" - whether a diabetes medication was prescribed ('yes' or 'no')
- "readmitted" - if the patient was readmitted at the hospital ('yes' or 'no')

Exploratory Data Analysis

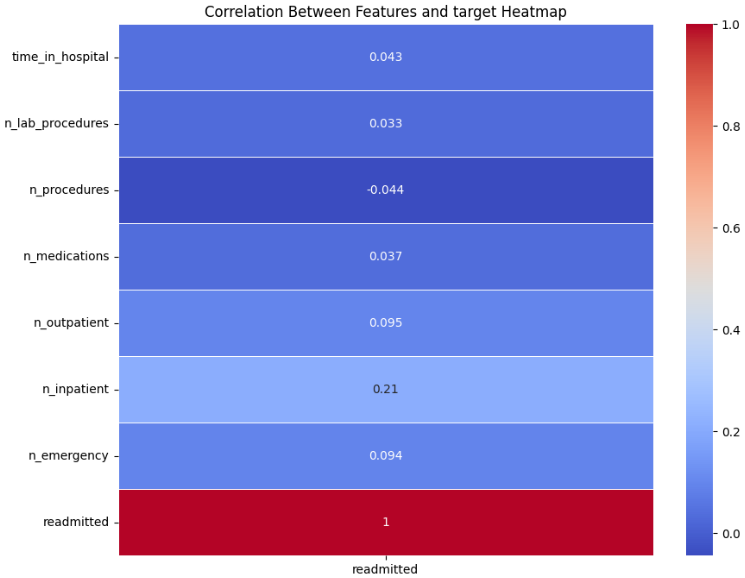
Checking correlation between the features:



Checking correlation between the features because highly correlated features often carry similar information. Including both in the model can lead to redundancy, unnecessarily increasing model complexity without adding significant predictive value.

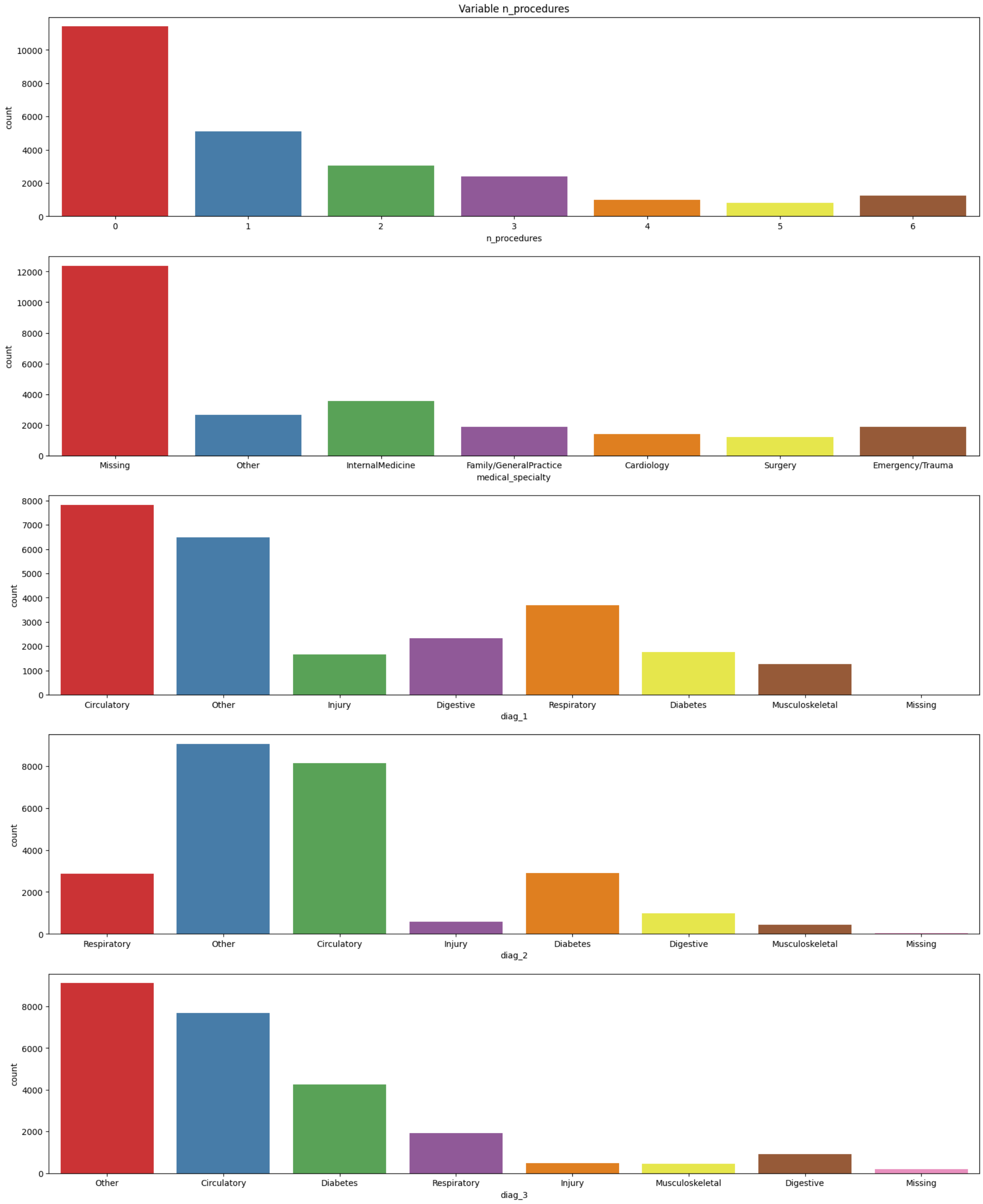
As we can see from the above heatmap there is not a strong correlation between the features, hence all the features can be considered as independent.

Checking correlation of features with respect to target variable



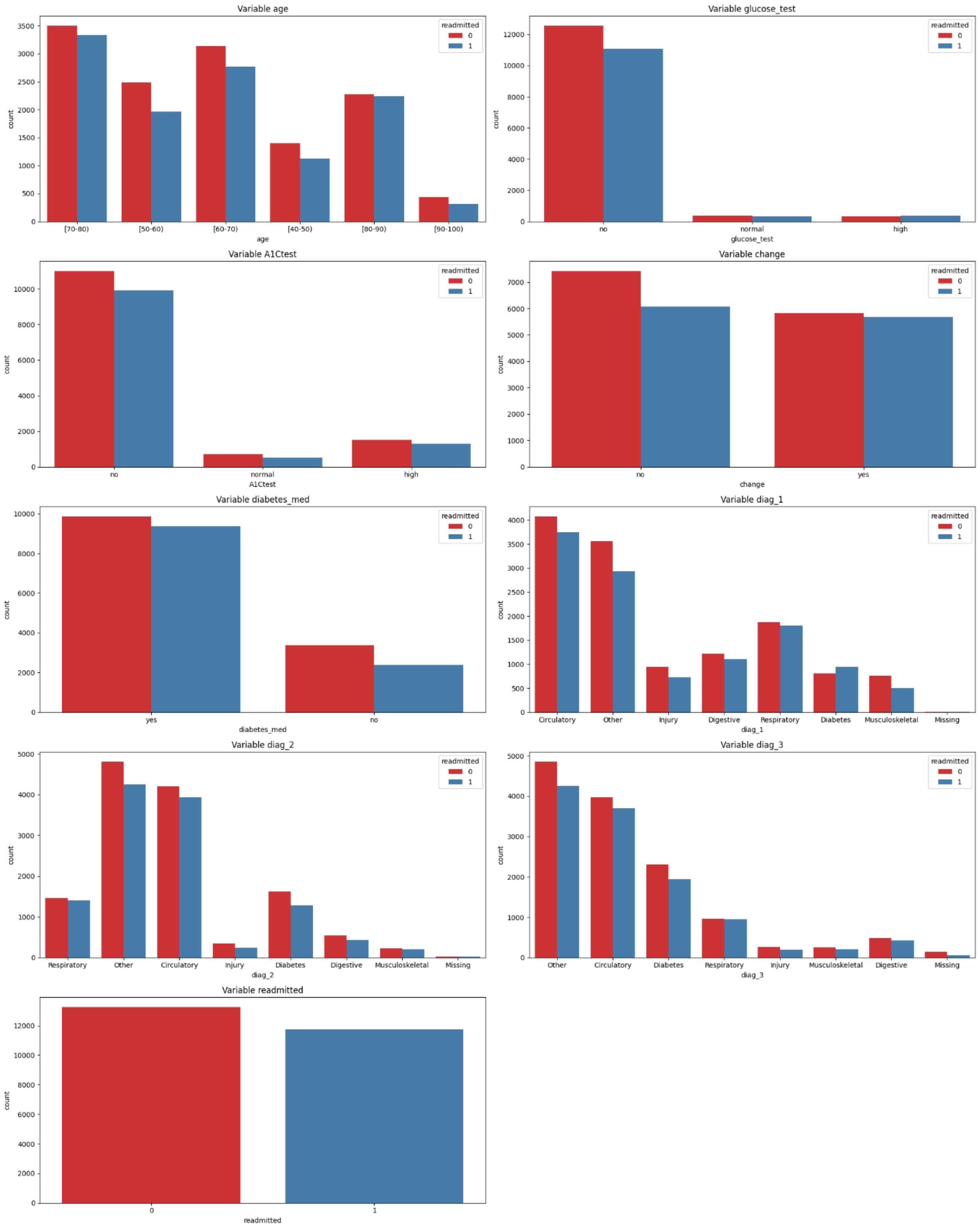
Its clear that the features like n_ipatient, n_emergency and n_outpatient show a strong correlation with the target variable.

EDA on categorical data



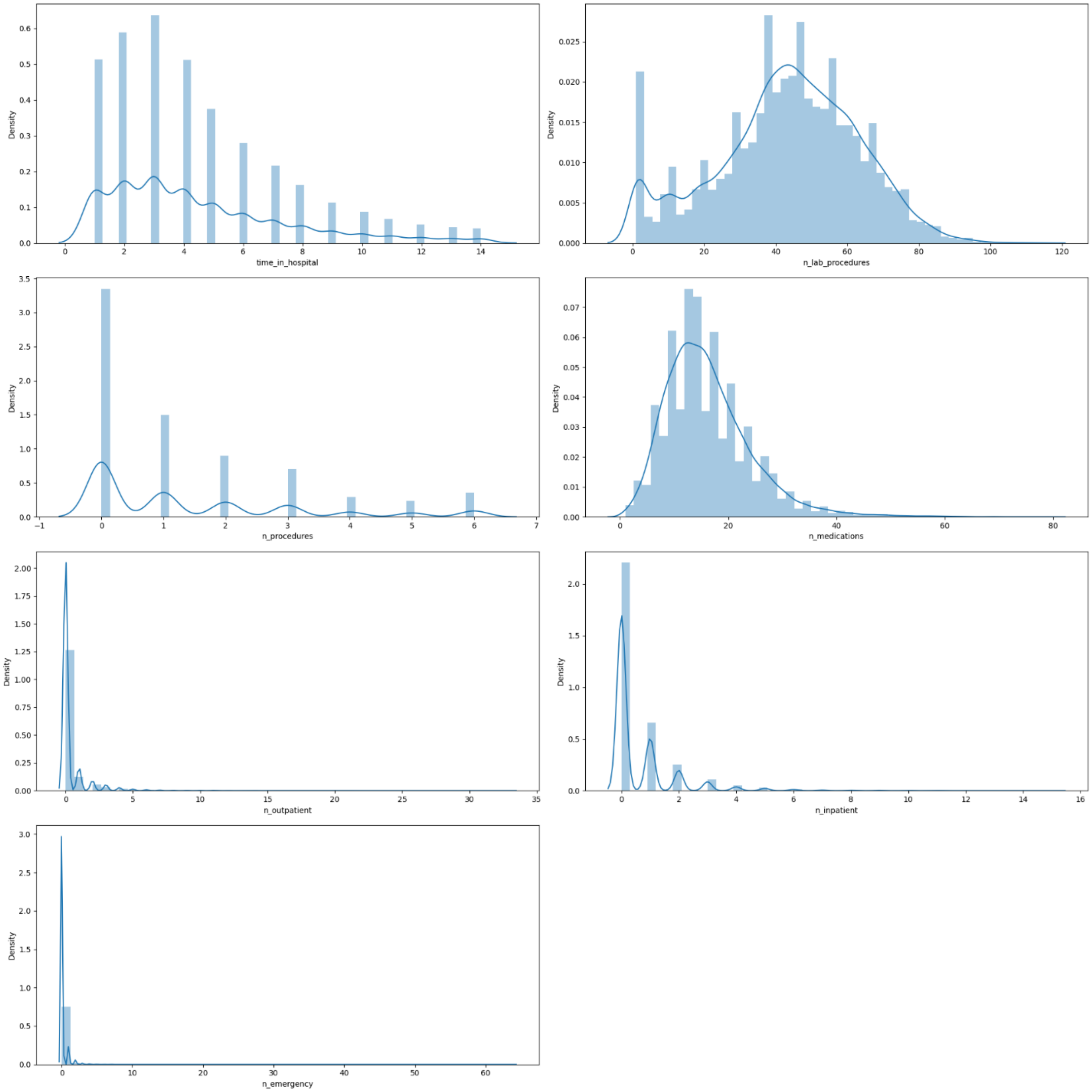
Upon examining the categorical variables, we notice that the n_procedures variable shows a decrease in data as the number increases. The medical_speciality variable has a significant amount of missing data, and the diagnosis variable contains various diagnoses distributed across three different values.

EDA on numerical data



Upon reviewing the age data, it is evident that the majority of patients in the dataset are older, with a significant concentration of individuals in the higher age brackets. This suggests a potential age-related pattern in the dataset, where older individuals are more frequently represented. In contrast, the glucose test data shows a large number of missing values, with most records lacking glucose test results. A similar issue is observed with the A1C test data, where a considerable portion of the records does not include A1C test information, which could affect the comprehensiveness of the analysis for these health indicators. Despite these missing values, the target variable (which likely represents the outcome or condition being predicted) appears to be well-balanced, with the two classes being almost evenly distributed. This indicates that, for the most part, the dataset does not exhibit an inherent bias towards either class of the target variable, which is crucial for training balanced machine learning models.

Generating distplots to check the distribution the data.

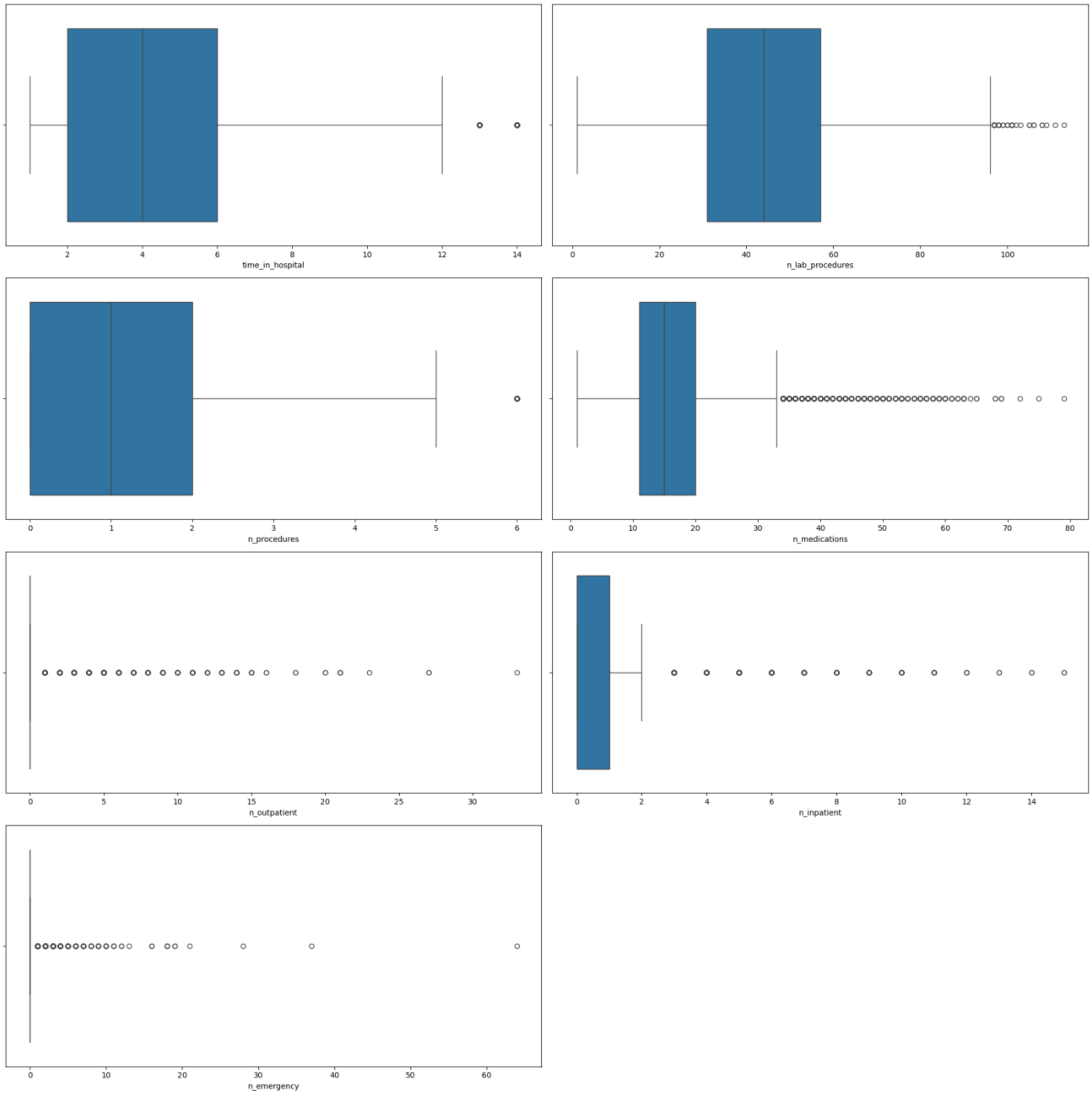


Skewness and Distribution of Data

In this dataset, we observe that the distribution is highly **left-skewed**, meaning the majority of the data points are concentrated towards the lower values, with fewer data points as the value increases. This indicates that the data tends to have a long tail on the right side, with most observations falling within a smaller range of values. Additionally, as the values increase, the frequency or number of data points decreases significantly, suggesting that larger values are less common. This pattern is often indicative of a

long-tailed distribution, where extreme or outlier values are present but infrequent, leading to an imbalance in the data distribution.

Outlier Detection through Box Plots



Box plots were used to identify outliers in the dataset, and the results indicate a significant presence of outliers across several variables. The identification of many outliers suggests that certain features exhibit extreme values that could potentially distort statistical analyses and model performance. Upon further investigation these outliers are not data errors but they are rare and valid occurrences.

Encoding of categorical data

To transform the categorical variables into a numerical format, **label encoding** was applied. This technique assigns a unique integer value to each category in a given categorical feature. Specifically, each distinct category or label is mapped to a corresponding integer, enabling the model to interpret categorical data as numeric inputs.

Feature Generation

Total Visits (total_visits)

The total_visits feature is created by summing the values from the n_outpatient, n_inpatient, and n_emergency columns. This gives an aggregate count of all types of visits for each patient, which helps in understanding the overall healthcare utilization of a patient. By combining these different visit types, this feature provides a comprehensive measure of a patient's total visits to healthcare facilities.

Chronic Condition Index (chronic_condition_index)

The chronic_condition_index is calculated by adding the n_procedures and n_medications variables. This feature aims to capture the potential severity or complexity of a patient's health status by combining the number of medical procedures and medications they are receiving. A higher score could indicate a more complex medical condition, which may be relevant for predicting outcomes or healthcare needs.

Emergency-Inpatient Ratio (emergency_inpatient_ratio)

The emergency_inpatient_ratio is calculated by dividing the number of emergency visits (n_emergency) by the sum of inpatient visits (n_inpatient) and 1 (to avoid division by zero). This ratio helps to understand the relationship between emergency visits and inpatient visits. A higher ratio could indicate that a patient frequently seeks emergency care instead of being admitted to the hospital, which could point to specific health concerns or treatment patterns.

Unique Diagnoses Count (unique_diagnoses_count)

The unique_diagnoses_count feature is created by counting the number of unique diagnoses across the columns diag_1, diag_2, and diag_3. This provides a measure of the diversity of diagnoses a patient has, which could be an important indicator of their overall health condition. A higher count of unique diagnoses might suggest multiple health issues or complex medical conditions.

Scaling of Data

To ensure that the features are on a comparable scale and to improve the performance of machine learning models, the **StandardScaler** was applied to scale the data. The **StandardScaler** standardizes the features by transforming them into a distribution with a mean of 0 and a standard deviation of 1. This scaling method is particularly useful when the data includes features with different units or ranges, as it eliminates the bias that can be introduced by variables with larger ranges. By centering the data around 0 and scaling it to unit variance, the StandardScaler helps prevent certain features from dominating the model due to their larger magnitude. This is especially important for algorithms that are sensitive to feature scaling, such as linear regression, support vector machines, and neural networks.

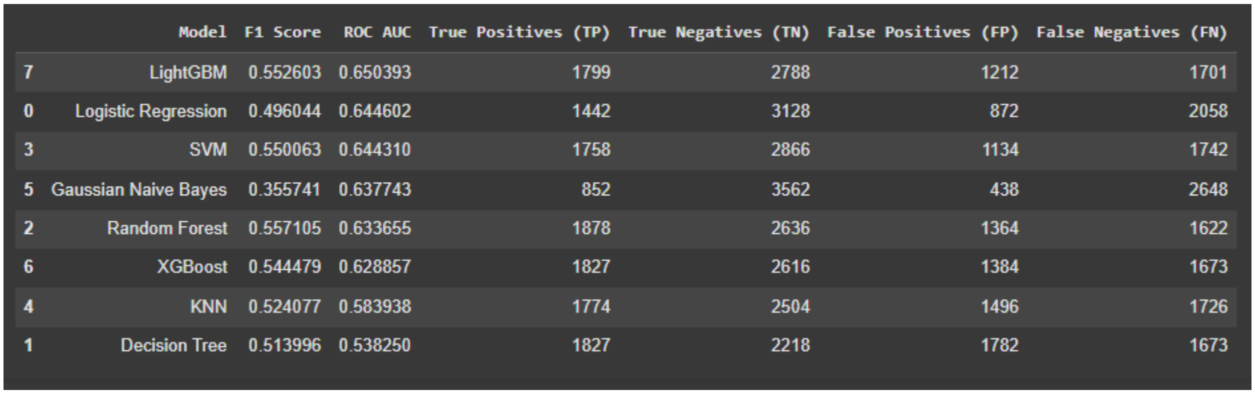
Modelling

Splitting of Data

To split the data into training and testing sets, the **train_test_split** function from sklearn.model_selection was used. The data was divided with 70% allocated for training and 30% for testing, ensuring a balanced approach for model evaluation. The random_state parameter was set to 42 to ensure reproducibility of the split across different runs.

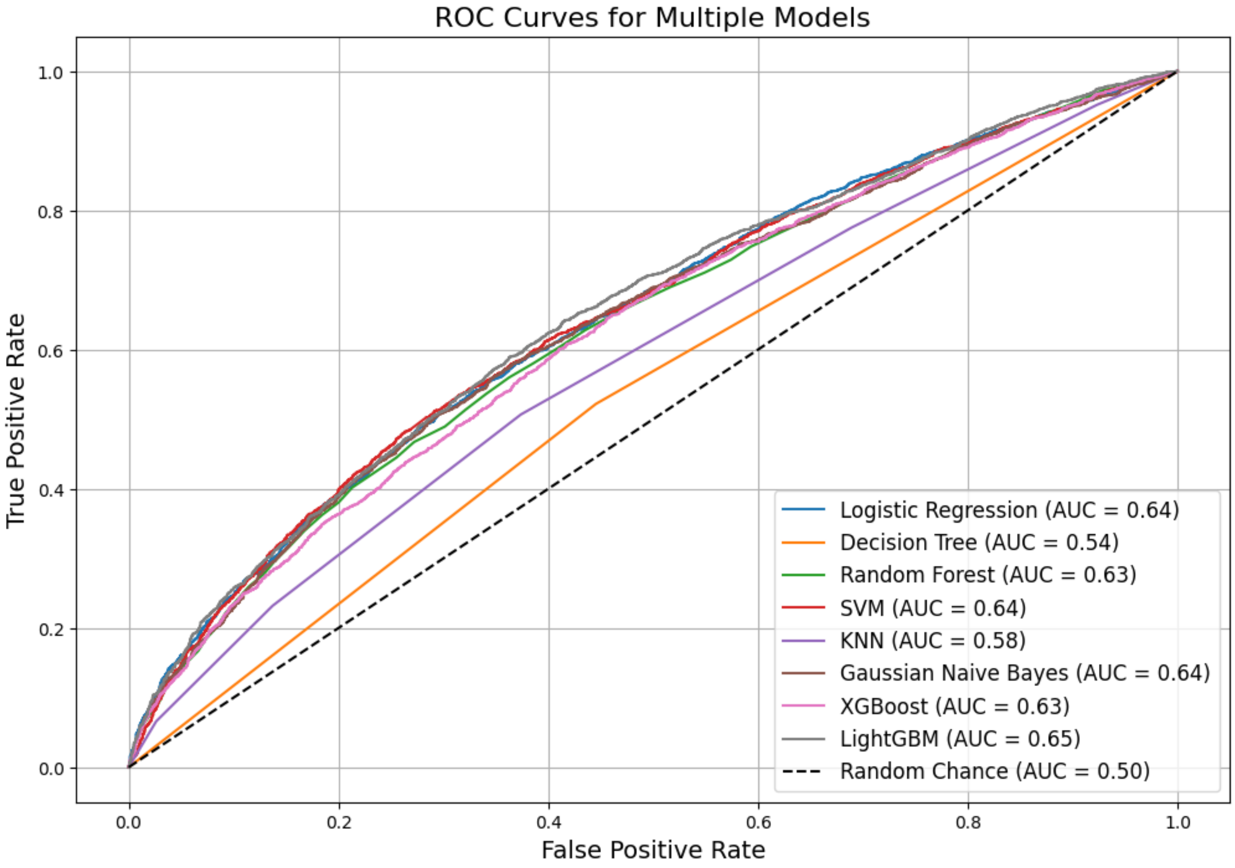
Multiple machine learning models were utilized for classification, including **Logistic Regression**, **Decision Tree**, **Random Forest**, **Support Vector Machine (SVM)**, **K-Nearest Neighbors (KNN)**, **Gaussian Naive Bayes**, **XGBoost**, and **LightGBM**. Each model was trained on the data and evaluated using performance metrics such as F1 score, ROC AUC score, and confusion matrix to determine the best-performing model.

Model Evaluation Metrices



	Model	F1 Score	ROC AUC	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
7	LightGBM	0.552603	0.650393	1799	2788	1212	1701
0	Logistic Regression	0.496044	0.644602	1442	3128	872	2058
3	SVM	0.550063	0.644310	1758	2866	1134	1742
5	Gaussian Naive Bayes	0.355741	0.637743	852	3562	438	2648
2	Random Forest	0.557105	0.633655	1878	2636	1364	1622
6	XGBoost	0.544479	0.628857	1827	2616	1384	1673
4	KNN	0.524077	0.583938	1774	2504	1496	1726
1	Decision Tree	0.513996	0.538250	1827	2218	1782	1673

ROC Curve graph for all the experimented models



The table below summarizes the evaluation metrics for a range of machine learning models, including **F1 Score**, **ROC AUC**, and the confusion matrix components (True Positives, True Negatives, False Positives, and False Negatives). These metrics were used to assess the performance of each model in terms of its ability to correctly classify the data, with a focus on balancing precision and recall (F1 Score) and the overall discrimination ability (ROC AUC).

Upon review, **LightGBM** emerged as the top-performing model, achieving the highest **F1 Score** of 0.55 and **ROC AUC** of 0.65. This indicates that LightGBM was most effective at making accurate predictions while minimizing both false positives and false negatives, surpassing all other models. Other models such as **Random Forest**, **SVM**, and **Logistic Regression** followed closely but showed slightly lower performance in terms of both **F1 Score** and **ROC AUC**. Models like **Gaussian Naive Bayes** and **Decision Tree** had relatively lower scores, suggesting they were less effective in accurately classifying the data. The performance of LightGBM suggests that it is particularly well-suited for this classification task, likely due to its ability to handle complex relationships and provide robust predictions.

Difficulties

Limited Predictive Power of Basic Features

Initially, the model's performance with the default features was not satisfactory. These features alone were insufficient in capturing the complex relationships between patient characteristics and their likelihood of readmission. The results from models trained with just the default features showed poor predictive accuracy, likely because these features did not capture all the nuances and critical factors influencing readmissions.

Many of the basic features in the dataset, such as age were not sufficiently informative by themselves. These features did not provide enough discriminative power to distinguish between patients who would be readmitted and those who would not. This issue became particularly evident when evaluating model performance using standard metrics such as accuracy, where the imbalance in readmission rates caused poor predictions for the minority class (readmitted patients).

Solution

To address the limitations of the default features, significant efforts were made to engineer new, more informative features that would better capture the risk of readmission such as number of total visits and Chronic Condition Index which are aggregated features.

The emergency-inpatient ratio was constructed to reflect the balance between emergency care and inpatient care, which could provide insights into the patient's overall health management.

Similarly, the unique diagnoses count feature was derived from the diagnostic codes to capture the diversity of a patient's health conditions. This feature helped identify patients with complex or multiple diagnoses, which might increase their likelihood of readmission.