HealthCare Data Analysis Final Project Report
On
Prediction of readmission for diabetes patients

Shatabdi Pal

(PSU ID# 952821988)
Portland State University

CS 510 Introduction to Healthcare Data Analysis

14th June 2024

Abstract

Project Title: Predictive Modeling of Hospital Readmissions for Diabetes Patients

Overview

This project aims to predict hospital readmission rates for diabetes patients by applying advanced data analysis techniques and machine learning algorithms. It will involve two main phases: Exploratory Data Analysis (EDA) and Machine Learning Prediction.

Objectives:

The project's primary focus is to conduct a comprehensive exploratory data analysis (EDA) to uncover patterns and address data quality issues, such as outliers and missing values, in a dataset on diabetes patient readmissions. This will be followed by implementing and evaluating two machine learning models, chosen based on the dataset and problem context, to predict readmissions. These models will be trained, tested, and validated using preprocessed data, with performance assessed through metrics like accuracy, precision, recall, and F1-score. The insights and recommendations derived from this analysis will provide valuable guidance for improving healthcare outcomes**.**

Tools & Technologies used:

Programming Language: Python

IDE: Jupyter Notebook, Google Colaboratory

Visualization: Python (Matplotlib and Seaborn)

Models: Logistic Regression, Decision Tree, Random Forest

Libraries: NumPy, Pandas, Seaborn, Matplotlib, Scikit-learn

Dataset: Diabetes 130-US Hospitals for Years 1999-2008

Part 1: Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) involves visually and statistically examining a dataset to extract meaningful insights and understand its underlying patterns and characteristics. EDA provides a foundation for further analysis and decision-making processes by summarizing key features and relationships within the data.

A. Data Understanding and Preprocessing

1. Dataset Description:

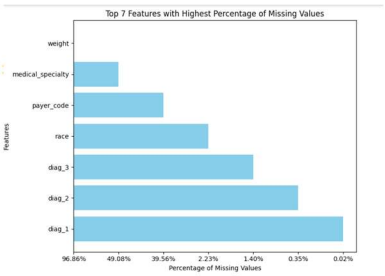
- encounter_id: Unique identifier of an encounter - **Numerical** (Identifier)
- patient_nbr: Unique identifier of a patient - **Numerical** (Identifier)

- race: Caucasian, Asian, African American, Hispanic, and other - **Categorical** (Nominal)
- gender: male, female, and unknown/invalid - **Categorica**l (Nominal)
- age: Grouped in 10-year intervals: (0, 10), (10, 20), …, (90, 100) - **Categorical** (Ordinal)
- weight: Weight in pounds ranges 0 - > 200 - **Categorical** (Ordinal)
- admission_type_id: Integer identifier corresponding to 8 distinct values - **Numerical** (Identifier)
- discharge_disposition_id: Integer identifier corresponding to 26 distinct values - **Numerical** (Identifier)
- admission_source_id: Integer identifier corresponding to 17 distinct values - **Numerical** (Identifier)
- time_in_hospital: Integer number of days between admission and discharge - **Numerical** (Continuous)
- payer_code: Integer identifier corresponding to 18 distinct values - **Numerical** (Identifier)
- medical_specialty: name of a specialty of the admitting physician, corresponding to 73 distinct values - **Categorical** (Nominal)
- num_lab_procedures: Number of lab tests performed during the encounter - **Numerical** (Continuous)
- num_procedures: Number of procedures (other than lab tests) performed during the encounter - **Numerical** (Continuous)
- num_medications: Number of distinct generic names administered during the encounter - **Numerical** (Continuous)
- number_outpatient: Number of outpatient visits of the patient in the year preceding the encounter - **Numerical** (Continuous)
- number_emergency: Number of emergency visits of the patient in the year preceding the encounter - **Numerical** (Continuous)
- number_inpatient: Number of inpatient visits of the patient in the year preceding the encounter - **Numerical** (Continuous)
- diag_1: The primary diagnosis (coded as first three digits of ICD9); 717 distinct values - **Categorical** (Nominal)
- diag_2: Secondary diagnosis (coded as first three digits of ICD9); 749 distinct values - **Categorical** (Nominal)
- diag_3: Additional secondary diagnosis (coded as first three digits of ICD9); 790 distinct values - **Categorical** (Nominal)
- number_diagnoses: Number of diagnoses with 16 distinct values - **Numerical** (Ordinal)
- max_glu_serum: Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured - **Categorical** (Nominal)
- A1C result: Indicates the range of the result or if the test was not taken. Values: “>8,” “>7,” “normal,” and “none” - **Categorical** (Nominal)
- Drug lists: 22 different medical drugs, with distinct values ['No,' 'Steady,' 'Up,' 'Down'] - **Categorical** (Nominal)
- Change: Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change” - **Categorica**l (Nominal)
- diabetesMed: Indicates if any diabetic medication was prescribed. Values: “yes” and “no” - **Categorical** (Nominal)
- readmitted: Days to inpatient readmission. Values: “0” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission - **Categorical** (Nominal)

- The missing values in the given dataset are denoted by “?”.

2. **Columns with missing values and Percentage:**

```
Columns with missing data, their count, and respective missing percentages
                Column  Missing Count Missing Percentage
0                 race           2273              2.23%
1               weight          98569             96.86%
2          payer_code          40256             39.56%
3   medical_specialty          49949             49.08%
4               diag_1             21              0.02%
5               diag_2            358              0.35%
6               diag_3           1423              1.40%
```

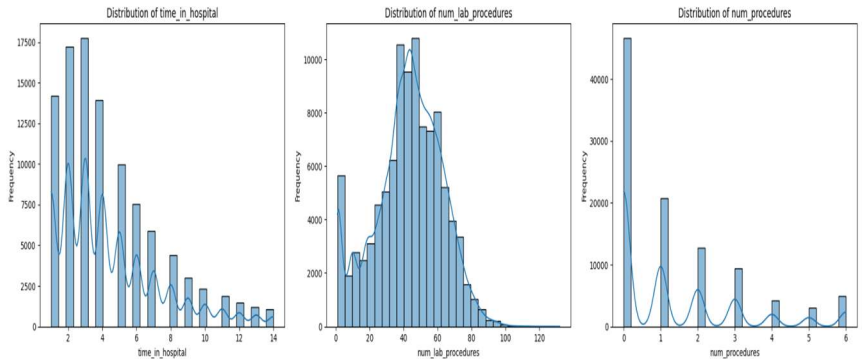


3. **Handling of Missing Values:**
   The data cleaning process is thorough and meticulous. To handle missing values, we need to identify and appropriately deal with the records where data is missing. In this dataset, missing values are represented by question marks ('?'). Here are steps to clean the data:

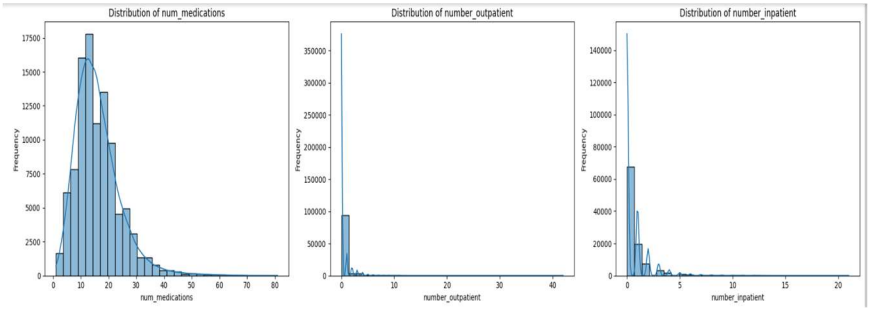
   - Weight has 96.86%, medical_specialty has 49.08% and payer_code 39.56% missing values. We can drop the 'weight,' 'payer_code,' and 'medical_specialty' columns.
   - For diag_1, diag_2, and diag_3, we will use the most common value to fill the missing value
   - A meticulous process addressed missing values in the 'race' column. The number of instances to impute for each race category was calculated based on the percentages: 53% Caucasian, 12% African American, and 1% Hispanic. This involved determining the exact counts for each category and randomly assigning these values to the missing entries. The process ensured that 1204 entries were replaced with 'Caucasian,' 272 with 'African American,' and 22 with 'Hispanic,' while the remaining missing values were categorized as 'Other.' This imputation method fills the missing values proportionally, maintaining the dataset's integrity and diversity.

B. Visualization: (Relationship between variables)

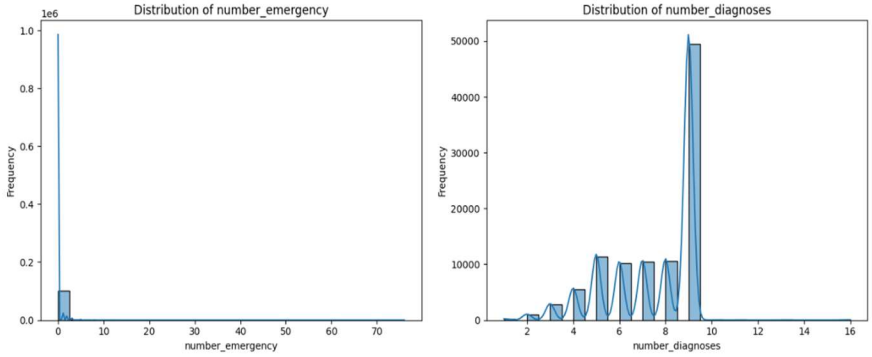
Distribution of Variables (Numerical Features):



The histograms reveal that most patients have short hospital stays, with a few outliers having more extended stays. The number of lab procedures has a normal distribution, indicating a standard range of tests, and the number of procedures is highly skewed, showing that most patients did not undergo any procedures. These patterns highlight variability in hospital stays and procedural interventions among patients.



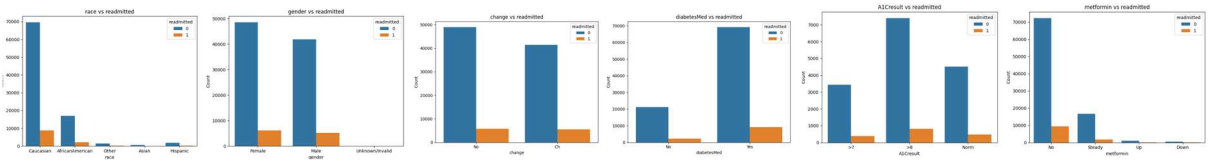
The histograms for 'num medications,' 'number outpatient visits,' and 'number inpatient stay' all exhibit right-skewed distributions, indicating most patients have lower counts for these variables, with few having high counts. This suggests a higher frequency of patients with fewer medications, outpatient visits, and inpatient stays. The data is not normally distributed, highlighting variability in these aspects of patient care.



The histograms for 'number emergency' and 'number diagnoses' show right-skewed distributions, with most patients having few emergency visits and a prominent peak around three diagnoses, indicating it is the most common. These patterns suggest that patients frequently have lower counts of emergencies and a few diagnoses.

These distributions suggest that most patients have low counts for hospital stays, medications, visits, and emergencies, with diagnoses peaking around three and lab procedures following a normal distribution. This indicates a prevalence of lower patient care counts and variability across different measures.

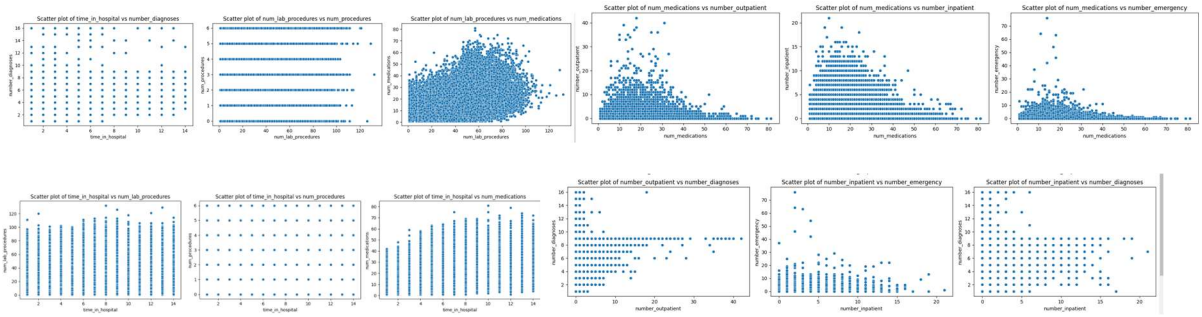
Bar plot of Categorical Features:



Overall Insights:

The charts reveal critical insights into hospital readmission rates based on various factors. Caucasian and African American individuals have the highest readmission counts, with disparities evident among different races. Gender differences in readmission rates are minimal. Regular metformin usage and being on diabetes medication seem to be associated with lower readmission rates. Patients not experiencing medication changes also show higher non-readmission rates. These insights highlight the importance of addressing racial disparities, medication management, and healthcare policies to improve patient outcomes.

Scatter plot of Continuous Variables:

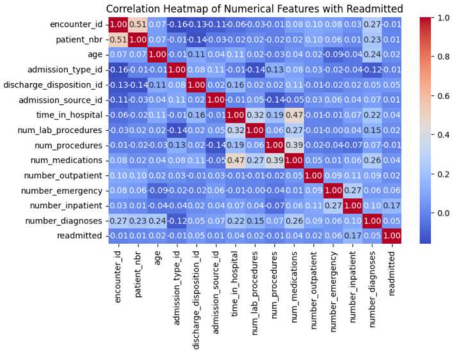


Here are some observations that can be made from the scatter plot:

- For the scatter plot of "Number of Outpatient Visits vs. Number of Diagnoses," there doesn't seem to be a significant increase in outpatient visits as the number of diagnoses increases. This suggests that the number of diagnoses may not strongly correlate with outpatient visits.
- In the plot of "Number of Inpatient Visits vs. Number of Emergency Visits," a downward trend indicates that fewer inpatient services are used as emergency visits increase. This suggests a potential relationship between emergency visits and the need for inpatient services.
- Regarding "Number of Inpatient Visits vs. Number of Diagnoses," there isn't a clear trend or correlation between these two variables.
- In the scatter plot of "Number of Medications vs. Number of Outpatient Visits," outpatient visits appear to increase as the number of medications increases, suggesting a potential positive correlation.

- The plot of "Number of Medications vs. Number of Inpatient Visits" shows clustering at the lower end of both axes, indicating that lower medication numbers are associated with fewer inpatient visits.
- In the scatter plot of "Number of Medications vs. Number of Emergency Visits," there seems to be an increase in emergency visits as the number of medications increases, but with high variability.
- Overall, while the scatter plots suggest some potential trends or correlations, they are not consistently strong across all variables.

Correlation between variables:



The correlation heatmap in the image provides insights into the relationships between different numerical features and their correlation with patient readmissions. The diagonal squares (from top left to bottom right) are all dark red, indicating perfect positive correlations with themselves (as expected).

Strong Positive Correlations

encounter_id and patient_nbr (0.51): Strong correlation due to multiple encounters per patient.

num_lab_procedures and time_in_hospital (0.32): More lab procedures correlate with more extended hospital stays, indicating severe conditions.

num_medications and num_procedures (0.39): Higher procedures are associated with more medications, suggesting complex health issues.

num_medications and time_in_hospital (0.47): Longer stays result in more medications, reflecting complex conditions.

num_medication and number_diagnoses (0.265): More diagnoses correlate with more medication, indicating complex cases.

Negative Correlations

No solid negative correlations were observed.

Impact on Readmission Predictions

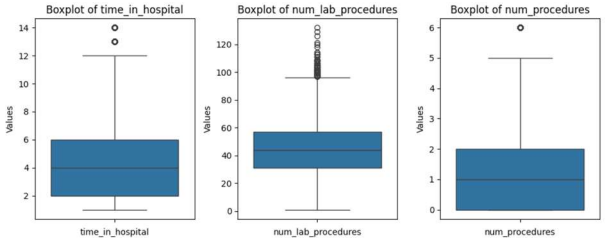
Weak correlations with readmitted suggest numerical variables are poor individual predictors.

The highest correlation is number_inpatient with readmission, which is only 0.17.

Advanced models capturing interactions and nonlinearities are needed for accurate predictions.

Presence of outliers:

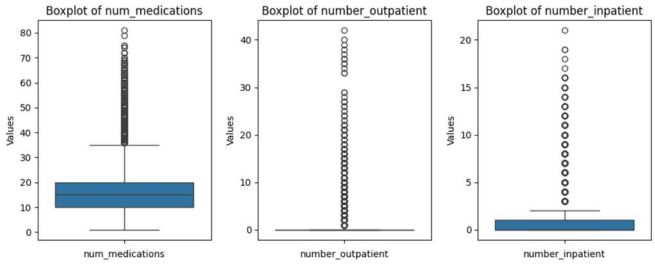
These boxplots provide valuable insights into the distribution and variability of different variables related to hospital stays and procedures. Here's how these observations might impact the analysis:



- **Time in Hospital:** One outlier indicates an unusually long stay, which could skew statistical measures and affect the interpretation of hospitalization trends.

- **Number of Lab Procedures**: Multiple high and low outliers suggest extreme values that might distort the overall picture of procedural frequency and influence trend analysis.

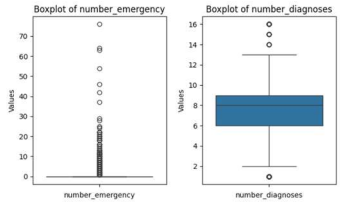
- **Number of Procedures:** One outlier with a notably high count may skew statistical measures and lead to inaccurate conclusions about procedural frequency.



- **Number of Medications:** Several outliers above the upper whisker could distort statistical measures and affect interpretations of medication usage trends.

- **Number of Outpatient Visits**: Many outliers above the upper whisker suggest potential extremes that could impact the analysis of outpatient visit patterns.

- **Number of Inpatient Visits:** A few outliers above the upper whisker indicate potentially extreme cases that could distort the analysis of inpatient visit trends.

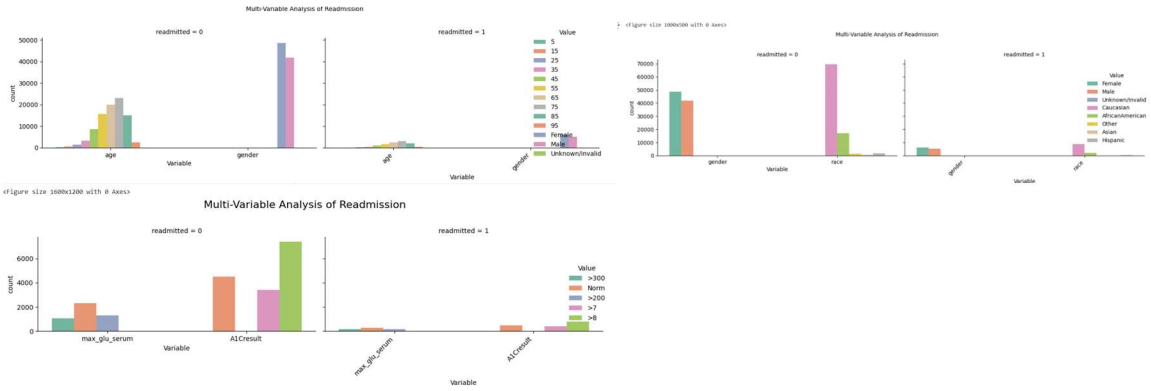


- **Number of Emergency Visits:** Several outliers above the upper whisker may skew analyses of emergency visit patterns and influence predictive modeling.

- **Number of Diagnoses:** A few outliers above the upper whisker could distort analyses of diagnosis frequency and predictions based on diagnostic patterns.

- Overall, these observations enhance the ability to analyze healthcare data comprehensively, leading to improved patient care, efficient resource utilization, and informed decision-making in healthcare management. Addressing these outliers through a thorough investigation to determine their validity and impact on the analysis is crucial for ensuring accurate interpretations and conclusions. Outliers can skew statistical measures, distort trends, and affect predictive modeling, potentially leading to incorrect decisions or interpretations if not adequately addressed.

Multivariable Relationship:



The three multivariable plots analyze the factors associated with readmission rates using age, gender, race, max_glu_serum, and A1C results. Here are the observations from each plot:

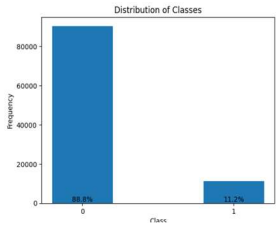
- **Age and Gender:** Elderly patients (65-85) dominate the non-admitted category, with more females than males. Readmissions are significantly fewer across all groups.
- **Race and Gender:** Caucasians are the predominant race among non-readmitted patients, and females are more prevalent. Readmissions are rare but occur more in Caucasians and, to a lesser extent, in African Americans.
- **Max Glu Serum and A1C Result:** Most non-readmitted patients have higher A1C results (>8), with fewer patients having extremely high max glucose serum levels. Readmissions are low across these metrics but show some variation.

These observations suggest that age, race, gender, and diabetic metrics (max_glu_serum and A1C result) have some influence on readmission rates, with elderly Caucasians showing higher

counts of non-readmission and slightly varied influences of glucose and A1C levels on readmission.

Class imbalance:

After categorizing patients into class 1 if they were readmitted in less than 30 days, class ">30" if readmitted in more than 30 days, and class 0 for no record of readmission, an imbalance within the dataset becomes evident. The plot below highlights a substantial disparity, with significantly fewer observations belonging to class 1 than those in class 0.



The bar for class '0' is much taller than for class '1', indicating a significant imbalance. Class '0' has approximately 88.9%, and Class '1' has 11.2% instances.

Part 2: Machine Learning Prediction

2.1. Feature Engineering: Transformations, Scaling, or encoding methods.

Transformation:

Data transformation is utilized to approximate normality to tackle non-normal distributions and differing standard deviations among groups. The skewness of highly skewed variables is evaluated to guide the choice of suitable transformations. Common transformations, such as square root for count data or logarithmic for size data, are frequently favored due to their prevalent usage in research. In this instance, we applied the square root transformation to five variables with skewness exceeding 1**.**

Encoding:

Age Features: The values of age are given like [0–10),[10–20), and [20–30) The age feature is not categorical but ordinal, and hence we should convert it to numbers and treat it as a numeric feature. We are converting the age feature to a numeric feature like below, [0–10) → 5, [10–20) → 15, [20–30) — -> 25

Non-numerical variables: Numerical features are the most accessible features in machine learning. All non-numerical columns are converted into binary features (0 and 1) using one-hot encoding, making them suitable for analysis.

Readmitted Feature: The 'readmitted' variable is redefined for binary classification: 0 for "No" or "> 30 days" and 1 for "< 30 days." This simplifies the problem into two classes, making it easier to analyze.

Scaling:

First, the summary statistics for the numerical features are calculated to understand their distributions. Then, we visualized these distributions using histograms to assess their spread and skewness. After identifying variations in scale among the features, we performed feature scaling using MinMaxScaler to normalize the data to a range of [0, 1]. This normalization process involved subtracting the minimum value and dividing by the range for each numerical variable, ensuring consistency in scale for machine learning algorithms.

Dropping Columns: We drop some features, like 'encounter_id' and 'patient_nbr,' that are unimportant in our analysis.

Creating New Features and Dropping Redundant Ones:

After creating three features, such as Health_index, Severity of Disease, and Number of Changes of Medication, redundant columns are removed.

Health_index: If the frequency of a person's visit to the hospital is high, then we can think of that person as being less healthy, and less healthy patients tend to be readmitted quickly. Let's create a health_index variable. The higher the health_index, the lesser the chance that a person will be readmitted (indirectly proportional)

Health_index = (1 / (number_emergency + number_inpatient + number_outpatient))

The severity of Disease: The severity of disease is high if a patient spends a lot of time in the hospital and undergoes many complicated tests, so let's create the severity of disease as one of the features. To get a probabilistic interpretation, let's divide it by total values.

severity_of_disease = (time_in_hospital + num_procedures + num_medications + num_lab_procedures + number_of_diagnoses)

Number of Changes: Research has found that patients who experience frequent changes(up/down) in the proportion of medications tend to readmit, so we have engineered a new variable called 'number_of_changes.' This captures the number of drugs whose proportion has changed.

Feature Selection:

The below screenshot shows the final features for model training:

```
Data columns (total 25 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   race                      101766 non-null  object
 1   gender                    101766 non-null  object
 2   age                       101766 non-null  float64
 3   admission_type_id         101766 non-null  float64
 4   discharge_disposition_id  101766 non-null  float64
 5   admission_source_id       101766 non-null  float64
 6   num_lab_procedures        101766 non-null  float64
 7   diag_1                    101766 non-null  object
 8   diag_2                    101766 non-null  object
 9   diag_3                    101766 non-null  object
 10  number_diagnoses          101766 non-null  float64
 11  max_glu_serum             5346 non-null    object
 12  A1Cresult                 17018 non-null   object
 13  change                    101766 non-null  object
 14  diabetesMed               101766 non-null  object
 15  readmitted                101766 non-null  float64
 16  time_in_hospital_sqrt     101766 non-null  float64
 17  num_procedures_sqrt       101766 non-null  float64
 18  num_medications_log       101766 non-null  float64
 19  number_outpatient_sqrt    101766 non-null  float64
 20  number_emergency_sqrt     101766 non-null  float64
 21  number_inpatient_sqrt     101766 non-null  float64
 22  Health_index              101766 non-null  int64
 23  severity_of_disease       101766 non-null  float64
 24  number_of_changes         101766 non-null  int64
```

2.2 Model Choice:

I have chosen Logistic Regression, Decision Tree, and Random Forest to predict hospital re-admission. The SMOTE technique resolves the issue since our target variable has class imbalance problems.

The reasons for choosing those models are as follows:

Logistic Regression

- **Probabilistic Outputs:** Logistic regression outputs probabilities, which are beneficial for understanding the likelihood of readmission and can help healthcare providers assess risk levels.
- **Interpretability:** The model is straightforward and interpretable. The coefficients provide clear insights into how each predictor affects the odds of readmission.
- **Handling Linear Relationships:** Logistic regression is a practical choice if the predictors have a linear relationship with the log odds of readmission.
- **Regularization:** The model can be regularized to prevent overfitting and ensure it generalizes well to new data.

Decision Trees

- **Non-linear Relationships:** Decision trees can model non-linear relationships between predictors and the target variable, which might be present in complex healthcare data.
- **Easy Interpretation**: Decision trees are easy to visualize and interpret, making it clear how decisions are made based on different predictor values. This can be valuable for explaining the model to healthcare professionals.

- **No Need for Assumptions:** Unlike logistic regression, decision trees do not assume linearity or a specific data distribution, providing flexibility in handling diverse datasets.

Random Forests

- **Handling Complexity:** Random forests can capture complex interactions between predictors and are robust to overfitting due to the ensemble approach of combining multiple decision trees.
- **Variable Importanc**e: They can rank predictors by importance, helping to identify which factors are most influential in predicting readmission. This can guide clinical focus and interventions.
- **Robustness:** Random forests are less sensitive to outliers and can effectively handle missing data, which is common in medical datasets.
- **Dimensionality Reduction:** They can manage high-dimensional data and reduce it by identifying the most significant variables, improving model performance and interpretability.

Addressing Class Imbalance with SMOTE

- **Synthetic Minority Over-sampling Technique (SMOTE)**: This technique balances the dataset by generating synthetic examples for the minority class (readmitted patients), improving model performance.
- **Improved Performance**: Ensures models learn characteristics of both readmitted and non-admitted patients, leading to more accurate and reliable predictions.

2.3 Split the data to train and test:

The data is split by separating the features (X) from the target variable (y). Following this, the classes are balanced using SMOTE, with synthetic examples generated for the minority class. Subsequently, the balanced data is divided into training and testing sets using a 70-30 ratio, whereby 70% of the data is allocated for training the models, and 30% is reserved for testing their performance.

2.4 Performance Metrics

To evaluate models predicting hospital readmissions, I have used several performance metrics:

Confusion Matrix: This provides a detailed breakdown of the model's predictions, highlighting true positives, true negatives, false positives, and false negatives. It's essential for understanding the types of errors the model makes.

Accuracy Score: This measures the proportion of correctly classified instances among the total cases, giving a general sense of the model's effectiveness.

Precision measures the proportion of actual positive cases out of all predicted positives. High precision is crucial in healthcare to minimize false alarms and ensure that predicted readmissions are likely accurate.

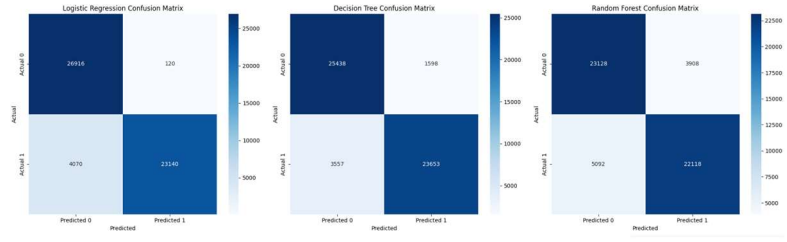
Recall: Also known as sensitivity, this measures the proportion of actual positives that were correctly identified. It's critical in healthcare to capture as many true readmissions as possible for timely intervention.

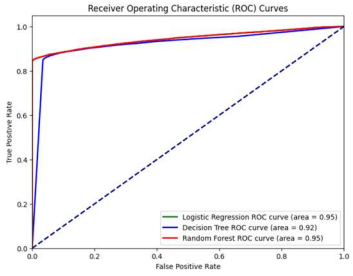
F1 Score: This is the harmonic mean of precision and recall, providing a balanced measure that accounts for false positives and false negatives. It's beneficial with imbalanced class distributions.

ROC Curve and AUC: The ROC curve plots the true positive rate against the false positive rate at various thresholds, showing the tradeoff between sensitivity and specificity. The AUC summarizes this curve, indicating the model's ability to distinguish between positive and negative cases. This is valuable in healthcare, where the impacts of false positives and false negatives differ significantly.

These metrics collectively offer a comprehensive evaluation of the model's performance in predicting hospital readmissions.

2.5 Interpretation of the results of these metrics for model effectiveness:

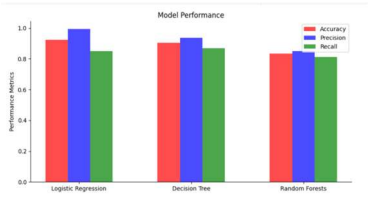




	Model	Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	0.922759	0.994841	0.850423	0.916980
1	Decision Tree	0.904970	0.936715	0.869276	0.901737
2	Random Forest	0.834089	0.849842	0.812863	0.830941

- Logistic Regression and Decision Tree have higher accuracy, indicating they correctly classify more instances than Random Forest.
- Logistic Regression has the highest precision, meaning it has the lowest rate of false positives among the three models.
- Decision Tree has the highest recall, indicating it identifies a higher proportion of actual positives.
- Logistic Regression has the highest F1 Score, balancing precision and recall effectively.
- The AUC values indicate that all models have strong discriminative ability, with Logistic Regression and Random Forest performing slightly better than the Decision Tree.
- The ROC curves show that all three models have high AUC values (around 0.95 for Logistic Regression and Random Forest and 0.92 for Decision Tree), indicating good overall performance.

2.6 Performance comparison chosen model.



Logistic Regression is the most balanced model, with the highest F1 score and competitive accuracy, precision, and recall. It's a strong candidate if the balance between false positives and false negatives is crucial. Decision Tree has a slightly higher recall than Logistic Regression, making it better at identifying true positives but at the cost of more false positives. Random Forest has the lowest performance across most metrics, particularly accuracy and precision. However, its AUC is still strong, suggesting it might be more robust across different thresholds. Based on these metrics, logistic regression is the most effective model for this classification task overall.

Conclusion:

This project aimed to reduce hospital readmission rates for diabetic patients by developing a predictive risk model using machine learning. The study found that logistic regression outperformed other models like decision trees and random forest. Key recommendations include mandatory A1C and glucose tests, reevaluating medication regimes, and ensuring follow-up visits. Challenges include handling heterogeneous data, missing values, and incorporating factors like access to care. Despite these challenges, the model promises to improve patient outcomes and reduce healthcare costs.

References:

https://towardsdatascience.com/predicting-hospital-readmission-for-patients-with-diabetes-using-scikit-learn-a2e359b15f0

https://saurabhraj5162.medium.com/diabetes-130-us-hospitals-for-years-1999-2008-hospital-readmission-823ff48272f9

https://medium.com/analytics-vidhya/diabetes-130-us-hospitals-for-years-1999-2008-e18d69beea4d

https://www.kaggle.com/code/iabhishekofficial/prediction-on-hospital-readmission/notebook

Final_Project_HCDA

June 12, 2024

1 Python Code Appendix:

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

2 Data Loading

[2]:
```
df = pd.read_csv('diabetic_data-1.csv')
print(df.shape)
```

```
(101766, 50)
```

3 Data Exploration

4 Count and distinct values of each features

[3]:
```
# Get distinct count for each feature
#distinct_counts = df.nunique()
for column in df.columns:
    distinct_values = df[column].unique()
    distinct_count = len(distinct_values)
    print(f"Feature: {column}")
    print(f"Distinct Count: {distinct_count}")
    print(f"Distinct Values: {distinct_values}")
    print()
```

```
Feature: encounter_id
Distinct Count: 101766
Distinct Values: [  2278392    149190     64410 … 443854148 443857166
443867222]

Feature: patient_nbr
Distinct Count: 71518
Distinct Values: [  8222157  55629189  86047875 … 140199494 120975314
```

```
175429310]

Feature: race
Distinct Count: 6
Distinct Values: ['Caucasian' 'AfricanAmerican' '?' 'Other' 'Asian' 'Hispanic']

Feature: gender
Distinct Count: 3
Distinct Values: ['Female' 'Male' 'Unknown/Invalid']

Feature: age
Distinct Count: 10
Distinct Values: ['[0-10)' '[10-20)' '[20-30)' '[30-40)' '[40-50)' '[50-60)'
'[60-70)'
 '[70-80)' '[80-90)' '[90-100)']

Feature: weight
Distinct Count: 10
Distinct Values: ['?' '[75-100)' '[50-75)' '[0-25)' '[100-125)' '[25-50)'
'[125-150)'
 '[175-200)' '[150-175)' '>200']

Feature: admission_type_id
Distinct Count: 8
Distinct Values: [6 1 2 3 4 5 8 7]

Feature: discharge_disposition_id
Distinct Count: 26
Distinct Values: [25  1  3  6  2  5 11  7 10  4 14 18  8 13 12 16 17 22 23  9 20
15 24 28
 19 27]

Feature: admission_source_id
Distinct Count: 17
Distinct Values: [ 1  7  2  4  5  6 20  3 17  8  9 14 10 22 11 25 13]

Feature: time_in_hospital
Distinct Count: 14
Distinct Values: [ 1  3  2  4  5 13 12  9  7 10  6 11  8 14]

Feature: payer_code
Distinct Count: 18
Distinct Values: ['?' 'MC' 'MD' 'HM' 'UN' 'BC' 'SP' 'CP' 'SI' 'DM' 'CM' 'CH'
'PO' 'WC' 'OT'
 'OG' 'MP' 'FR']

Feature: medical_specialty
Distinct Count: 73
```

```
Distinct Values: ['Pediatrics-Endocrinology' '?' 'InternalMedicine'
 'Family/GeneralPractice' 'Cardiology' 'Surgery-General' 'Orthopedics'
 'Gastroenterology' 'Surgery-Cardiovascular/Thoracic' 'Nephrology'
 'Orthopedics-Reconstructive' 'Psychiatry' 'Emergency/Trauma'
 'Pulmonology' 'Surgery-Neuro' 'Obsterics&Gynecology-GynecologicOnco'
 'ObstetricsandGynecology' 'Pediatrics' 'Hematology/Oncology'
 'Otolaryngology' 'Surgery-Colon&Rectal' 'Pediatrics-CriticalCare'
 'Endocrinology' 'Urology' 'Psychiatry-Child/Adolescent'
 'Pediatrics-Pulmonology' 'Neurology' 'Anesthesiology-Pediatric'
 'Radiology' 'Pediatrics-Hematology-Oncology' 'Psychology' 'Podiatry'
 'Gynecology' 'Oncology' 'Pediatrics-Neurology' 'Surgery-Plastic'
 'Surgery-Thoracic' 'Surgery-PlasticwithinHeadandNeck' 'Ophthalmology'
 'Surgery-Pediatric' 'Pediatrics-EmergencyMedicine'
 'PhysicalMedicineandRehabilitation' 'InfectiousDiseases' 'Anesthesiology'
 'Rheumatology' 'AllergyandImmunology' 'Surgery-Maxillofacial'
 'Pediatrics-InfectiousDiseases' 'Pediatrics-AllergyandImmunology'
 'Dentistry' 'Surgeon' 'Surgery-Vascular' 'Osteopath'
 'Psychiatry-Addictive' 'Surgery-Cardiovascular' 'PhysicianNotFound'
 'Hematology' 'Proctology' 'Obstetrics' 'SurgicalSpecialty' 'Radiologist'
 'Pathology' 'Dermatology' 'SportsMedicine' 'Speech' 'Hospitalist'
 'OutreachServices' 'Cardiology-Pediatric' 'Perinatology'
 'Neurophysiology' 'Endocrinology-Metabolism' 'DCPTEAM' 'Resident']

Feature: num_lab_procedures
Distinct Count: 118
Distinct Values: [ 41  59  11  44  51  31  70  73  68  33  47  62  60  55  49
75  45  29
  35  42  66  36  19  64  25  53  52  87  27  37  46  28  48  72  10   2
  65  67  40  54  58  57  43  32  83  34  39  69  38  56  22  96  78  61
  88  50   1  18  82   9  63  24  71  77  81  76  90  93   3 103  13  80
  85  16  15  12  30  23  17  21  79  26   5  95  97  84  14  74 105  86
  98  20   6  94   8 102 100   7  89  91  92   4 101  99 114 113 111 129
 107 108 106 104 109 120 132 121 126 118]

Feature: num_procedures
Distinct Count: 7
Distinct Values: [0 5 1 6 2 3 4]

Feature: num_medications
Distinct Count: 75
Distinct Values: [ 1 18 13 16  8 21 12 28 17 11 15 31  2 23 19  7 20 14 10 22  9
27 25  4
 32  6 30 26 24 33  5 39  3 29 61 40 46 41 36 34 35 50 43 42 37 51 38 45
 54 52 49 62 55 47 44 53 48 57 59 56 60 63 58 70 67 64 69 65 68 66 81 79
 75 72 74]

Feature: number_outpatient
Distinct Count: 39
```

```
Distinct Values: [ 0  2  1  5  7  9  3  8  4 12 11  6 20 15 10 13 14 16 21 35 17
29 36 18
 19 27 22 24 42 39 34 26 33 25 23 28 37 38 40]

Feature: number_emergency
Distinct Count: 33
Distinct Values: [ 0  1  2  4  3  9  5  7  6  8 22 25 10 13 42 16 11 28 15 14 18
12 21 20
 19 46 76 37 64 63 54 24 29]

Feature: number_inpatient
Distinct Count: 21
Distinct Values: [ 0  1  2  3  6  5  4  7  8  9 15 10 11 14 12 13 17 16 21 18
19]

Feature: diag_1
Distinct Count: 717
Distinct Values: ['250.83' '276' '648' '8' '197' '414' '428' '398' '434' '250.7'
'157'
 '518' '999' '410' '682' '402' '737' '572' 'V57' '189' '786' '427' '996'
 '277' '584' '462' '473' '411' '174' '486' '998' '511' '432' '626' '295'
 '196' '250.6' '618' '182' '845' '423' '808' '250.4' '722' '403' '250.11'
 '784' '707' '440' '151' '715' '997' '198' '564' '812' '38' '590' '556'
 '578' '250.32' '433' 'V58' '569' '185' '536' '255' '250.13' '599' '558'
 '574' '491' '560' '244' '250.03' '577' '730' '188' '824' '250.8' '332'
 '562' '291' '296' '510' '401' '263' '438' '70' '250.02' '493' '642' '625'
 '571' '738' '593' '250.42' '807' '456' '446' '575' '250.41' '820' '515'
 '780' '250.22' '995' '235' '250.82' '721' '787' '162' '724' '282' '514'
 'V55' '281' '250.33' '530' '466' '435' '250.12' 'V53' '789' '566' '822'
 '191' '557' '733' '455' '711' '482' '202' '280' '553' '225' '154' '441'
 '250.81' '349' '?' '962' '592' '507' '386' '156' '200' '728' '348' '459'
 '426' '388' '607' '337' '82' '531' '596' '288' '656' '573' '492' '220'
 '516' '210' '922' '286' '885' '958' '661' '969' '250.93' '227' '112'
 '404' '823' '532' '416' '346' '535' '453' '250' '595' '211' '303'
 '250.01' '852' '218' '782' '540' '457' '285' '431' '340' '550' '54' '351'
 '601' '723' '555' '153' '443' '380' '204' '424' '241' '358' '694' '331'
 '345' '681' '447' '290' '158' '579' '436' '335' '309' '654' '805' '799'
 '292' '183' '78' '851' '458' '586' '311' '892' '305' '293' '415' '591'
 '794' '803' '79' '655' '429' '278' '658' '598' '729' '585' '444' '604'
 '727' '214' '552' '284' '680' '708' '41' '644' '481' '821' '413' '437'
 '968' '756' '632' '359' '275' '512' '781' '420' '368' '522' '294' '825'
 '135' '304' '320' '250.31' '669' '868' '496' '250.43' '826' '567' '3'
 '203' '53' '251' '565' '161' '495' '49' '250.1' '297' '663' '576' '355'
 '850' '287' '250.2' '611' '840' '350' '726' '537' '620' '180' '366' '783'
 '11' '751' '716' '250.3' '199' '464' '580' '836' '664' '283' '813' '966'
 '289' '965' '184' '480' '608' '333' '972' '212' '117' '788' '924' '959'
 '621' '238' '785' '714' '942' '250.23' '710' '47' '933' '508' '478' '844'
 '7' '736' '233' '42' '250.5' '397' '395' '201' '421' '253' '250.92' '600'
```

```
 '494' '977' '39' '659' '312' '614' '647' '652' '646' '274' '861' '425'
 '527' '451' '485' '217' '250.53' '442' '970' '193' '160' '322' '581'
 '475' '623' '374' '582' '568' '465' '801' '237' '376' '150' '461' '913'
 '226' '617' '987' '641' '298' '790' '336' '362' '228' '513' '383' '746'
 '353' '911' '506' '873' '155' '860' '534' '802' '141' 'V45' '396' '310'
 '341' '242' '719' '239' '533' '616' '519' '301' 'V66' '5' '989' '230'
 '385' '300' '853' '871' '570' '848' '463' '9' '934' '250.21' '236' '361'
 '594' '501' '810' '643' '430' '528' '205' '791' '983' '992' '490' '172'
 '171' '622' '306' '863' '864' '474' '660' '759' '356' '634' '967' '551'
 '695' '187' '732' '747' '323' '308' '370' '252' '152' '846' '164' '365'
 '718' '48' '266' '720' '94' '344' '797' '170' '878' '904' 'V56' '882'
 '843' '709' '973' '454' '686' '939' '487' '229' '991' '483' '357' '692'
 '796' '693' '935' '936' '800' '920' 'V26' '261' '307' '262' '250.9' '831'
 '145' '223' 'V71' '839' '685' 'V54' '35' '34' '179' '964' '136' '324'
 '389' '815' '334' '143' '526' '588' '192' 'V67' '394' '917' '88' '219'
 '325' '792' '717' '994' '990' '793' '207' '637' '195' '373' '847' '827'
 '31' '891' '814' 'V60' '703' '865' '352' '627' '378' '342' '886' '369'
 '745' '705' '816' '541' '986' '610' '633' '640' '753' '173' '835' '379'
 '445' '272' '382' '945' '619' '881' '250.52' '866' '405' '916' '215'
 '893' '75' '671' '928' '906' '897' '725' '867' '115' '890' '734' '521'
 '674' '470' '834' '146' '696' '524' '980' '691' '384' '142' '879'
 '250.51' '246' '208' '448' '955' '653' '149' '245' '735' '883' '854'
 '952' '838' '194' 'V43' '163' '216' '147' '354' '27' '477' '318' '880'
 '921' '377' '471' '683' '175' '602' '250.91' '982' '706' '375' '417'
 '131' '347' '870' '148' '862' '61' '817' '914' '360' '684' '314' 'V63'
 '36' '57' '240' '915' '971' '795' '988' '452' '963' '327' '731' '842'
 'V25' '645' '665' '110' '944' '603' '923' '412' '363' '957' '976' '698'
 '299' '700' '273' '974' '97' '529' '66' '98' '605' '941' '52' '806' '84'
 '271' '837' '657' '895' '338' '523' '542' '114' '543' '372' 'V70' 'E909'
 '583' 'V07' '422' '615' '279' '500' '903' '919' '875' '381' '804' '704'
 '23' '58' '649' '832' '133' '975' '833' '391' '690' '10' 'V51']

Feature: diag_2
Distinct Count: 749
Distinct Values: ['?' '250.01' '250' '250.43' '157' '411' '492' '427' '198'
'403' '288'
 '998' '507' '174' '425' '456' '401' '715' '496' '428' '585' '250.02'
 '410' '999' '996' '135' '244' '41' '571' '276' '997' '599' '424' '491'
 '553' '707' '286' '440' '493' '242' '70' 'V45' '250.03' '357' '511' '196'
 '396' '197' '414' '250.52' '577' '535' '413' '285' '53' '780' '518' '150'
 '566' '250.6' '867' '486' 'V15' '8' '788' '340' '574' '581' '228' '530'
 '250.82' '786' '294' '567' '785' '512' '305' '729' '250.51' '280' '648'
 '560' '618' '444' '38' 'V10' '578' '277' '781' '250.42' '278' '426' '584'
 '462' '402' '153' '272' '733' '34' '881' '203' '250.41' '250.13' '293'
 '245' '250.12' '558' '787' '342' '573' '626' '303' '250.53' '458' '710'
 '415' 'V42' '284' '569' '759' '682' '112' '292' '435' '290' '250.93'
 '642' '536' '398' '319' '711' 'E878' '446' '255' 'V44' '250.7' '784'
 '300' '562' '162' '287' '447' '789' '790' '591' '200' '154' '304' '117'
```

'847' '852' '250.83' '250.11' '816' '575' '416' '412' '441' '515' '372'
'482' '382' 'V65' '572' '283' '78' '250.81' '576' '432' '595' '295' 'V12'
'204' '466' '721' '434' '590' '271' '813' '368' '227' '783' '250.5' '258'
'253' '309' '250.91' '519' '333' '459' '250.92' '250.4' '179' '420' '345'
'433' '661' '537' '205' '722' '405' '437' '714' '211' 'E812' '263' '202'
'397' '250.23' 'E932' '201' '301' '723' '614' '568' '861' 'V57' '724'
'189' '297' '453' 'E888' '730' '354' '451' '738' 'E939' '805' 'V43' '155'
'910' '218' '358' '220' 'E937' '583' '958' '794' '564' '436' '250.22'
'620' '621' '331' '617' '596' '314' '378' '250.8' '625' '478' '731' '172'
'404' '681' '470' '279' '281' '531' '443' '799' '337' '311' '719' 'E944'
'423' 'E870' '465' 'E849' '782' '481' '480' 'V23' '199' '79' '438' '348'
'42' 'E950' '473' '627' '726' '54' '490' '317' '332' '508' '369' '600'
'349' '485' '208' '922' '431' '296' 'E934' '753' 'E935' '386' '728' '607'
'E915' '344' '716' '289' '191' '873' '850' '611' '377' '352' '616' 'V17'
'136' '455' '933' 'E885' '860' '513' '603' '484' '223' 'V72' '291' '151'
'V58' '550' '510' '891' '185' '592' '791' '138' '598' '336' '362' '217'
'825' '298' '821' 'E880' '343' '429' 'E879' '579' '225' '250.9' 'V49'
'696' '233' '658' '969' '275' '250.1' '601' '704' '808' 'E890' 'V18'
'920' '380' '570' 'E817' '359' '812' '274' 'V14' '324' '758' 'V66' '911'
'E931' 'E924' '593' '792' '727' 'V46' '394' '532' 'V64' '557' '864' '718'
'E942' '807' '604' '924' '820' '580' '273' '241' '282' '824' 'V61' '646'
'701' '736' '565' '383' '250.2' 'E947' '452' '872' '905' 'E930' '921'
'131' '448' '389' '421' '214' '705' '494' '752' '623' '9' '299' '959'
'365' '967' 'E858' '40' '691' '909' '5' '814' '746' '250.31' '556' '680'
'745' '351' '306' '110' '695' '552' '346' '918' '882' '947' '520' '188'
'31' '356' '737' 'V08' '322' '182' '517' '974' 'E929' 'V53' '912' '252'
'608' '516' 'E933' '94' '702' '923' '594' '647' '111' '934' '430' '487'
'709' '796' '156' '977' '915' '756' '840' '341' '259' '693' '725' 'V62'
'528' '683' '953' '457' '501' 'E900' 'V09' '522' '919' '461' '506' '193'
'483' 'E936' '717' '802' '335' 'V54' '320' '945' '906' '239' '454' '826'
'823' 'E941' '226' '795' '684' '844' '250.33' '308' '615' '588' '712'
'663' '706' '833' '741' '713' '533' 'E884' '586' '555' '755' 'E928' '742'
'869' '962' 'V11' '543' '373' '870' '913' '152' '810' '965' '907' '908'
'995' '845' '474' '442' '751' '323' '472' '464' '686' '250.32' '540'
'251' '811' '652' '659' '851' '422' '815' '307' '325' '463' '992' '692'
'521' '917' 'E965' '524' '916' 'E813' '173' '238' '137' '514' '312' '837'
'355' '980' '622' '475' '500' '754' '261' '801' '868' '968' '381' '11'
'250.21' '694' '610' '734' 'E814' '310' '130' '246' '892' '846' '634'
'75' 'E927' 'E905' '183' '379' 'E917' '163' 'E868' '495' '747' '989'
'E854' '240' '832' '605' '602' '644' 'V16' '35' 'V70' '376' '266' 'E918'
'619' '477' '656' '46' '883' '171' 'V13' '698' '842' 'E850' '800' '269'
'664' 'E887' '952' '164' 'E881' '527' '685' '366' '836' '27' 'V63' '865'
'793' '232' '990' '52' '831' '327' '542' '806' '972' '862' 'E829' 'E919'
'944' 'E916' '963' '316' '645' '347' 'V85' '374' 'V02' '748' '256' '186'
'866' '975' '96' '395' '262' 'E819' '654' '994' '318' 'E826' '879' '674'
'641' '822' '145' '797' '353' 'E938' 'E816' '948' '987' '99' '192'
'250.3' 'E906' '534' '115' 'E818' 'E980' '360' '338' '529' '871' '750'
'212' '302' '955' '141' '88' 'V25' '215' '350' 'V50' 'V03' 'E853' 'E968'

```
 'E882' '140' '703' '991' '893' 'E821' '235' 'V69' '670' '195' 'V55' '388'
 '268' '894' '114' '260' '853' '7' '880' 'V86' '180' 'E945' '523' '863'
 '649' '270' '665' '460' '942' '364' '66' 'E883' '123' '884' 'V60' '843'
 '927']

Feature: diag_3
Distinct Count: 790
Distinct Values: ['?' '255' 'V27' '403' '250' 'V45' '38' '486' '996' '197'
'250.6' '427'
 '627' '414' '416' '714' '428' '582' 'V43' '250.01' '263' '250.42' '276'
 '482' '401' '250.41' '585' '781' '278' '998' '568' '682' '618' '250.02'
 '305' '707' '496' '599' '715' '424' '518' '553' '794' '411' 'V42' '531'
 '511' '490' '562' '250.8' '250.7' '250.52' '784' '491' '581' '420' '8'
 '724' '730' '789' '131' '250.82' '999' '41' '493' '250.03' '753' '786'
 '529' 'E888' '425' '595' '303' '560' '711' '492' '332' '296' '438' '362'
 '250.4' '654' '244' 'V70' '737' '625' '681' '250.51' '404' 'V10' '810'
 '280' '440' '785' '588' '569' '272' '997' '250.43' '918' '584' '54' '788'
 '426' '722' '250.92' '196' '461' '535' '787' '891' '284' '458' '648'
 '780' '182' '285' '593' '413' '664' '564' '201' '356' 'V15' '292' '782'
 '473' '455' 'E932' '357' '348' '294' '250.23' '459' 'E878' '437' '733'
 '507' '525' '250.53' '397' '572' '805' '453' '331' '736' '402' '591'
 '576' '465' '533' '703' '349' '315' '658' '608' '578' '716' '382' '300'
 '282' '571' '536' '596' '287' '644' 'V11' '558' 'E885' '162' '198' '218'
 '412' '396' 'V14' '570' '433' 'E934' '882' '288' '577' '443' '729' '836'
 '295' '799' '281' '304' '153' '410' '616' '250.83' '601' '291' '75' '512'
 '660' '250.5' '598' '337' '574' '653' 'V58' '311' '415' '386' '602' '790'
 '112' '873' '620' '436' '70' '155' '138' '663' '530' '710' '42' '342'
 '250.91' 'E884' '515' '307' '704' '728' '731' '583' '238' '441' '293'
 '573' '532' '290' '594' '319' '250.13' '250.12' '519' '346' '380' '135'
 '642' '698' '924' '905' 'E933' '555' '309' 'E879' '286' '565' '752' '580'
 '446' '444' '344' '252' '35' '813' '394' '301' '575' '258' 'V17' '802'
 '435' '746' 'V12' '709' '881' 'E935' '139' '250.81' '718' '365' '202'
 '334' '185' '398' 'V44' '517' 'E849' '614' '466' '626' '250.9' '368'
 '605' '883' '289' '478' '617' '429' '442' 'V25' '866' '610' '557' '959'
 'E942' '94' '920' '345' '313' '379' '79' '516' '586' '821' '600' '242'
 '373' '592' 'V64' '487' '253' '706' 'E947' '117' '340' 'E950' '656'
 'E949' '590' 'V09' '250.22' '934' '694' '203' '250.93' '995' '726' '923'
 '958' '275' 'E929' '211' 'V18' 'V66' '199' '665' '53' '279' '522' '791'
 '890' '456' 'E938' 'E816' '122' '721' 'V65' '136' '480' '423' 'E920'
 '793' '647' '537' '351' '845' '336' '274' '719' '945' '434' '494' '227'
 '157' '208' '174' 'V57' '812' '734' '150' 'V23' '447' '692' '228' 'V16'
 '756' '405' 'E928' '823' '552' '528' '389' '240' '454' '792' '366' 'E939'
 '907' '270' '310' '266' '387' 'E931' '783' '245' '607' '355' 'E930' '705'
 '372' '369' '611' '283' 'V46' '110' '867' 'E956' '251' '250.2' '820'
 '712' '695' '567' '343' '723' 'V08' '273' '623' '807' '451' '495' '701'
 '34' 'V53' '314' '472' 'E945' '11' '189' '534' '354' '333' 'V54' '277'
 '659' '708' '452' '655' '816' '670' '621' '246' '953' '865' 'E817' '646'
 '151' '378' '78' '298' '840' '641' '521' '745' '619' '912' '506' 'E904'
```

```
 '259' 'E870' 'E980' '383' '204' '696' '566' '727' '47' 'E943' '358' '191'
 '965' '921' '432' '27' 'E861' '758' '477' '524' '751' '652' '556' '188'
 '825' '919' '732' '908' '951' '962' '685' 'E850' 'E944' '527' '341' '693'
 '250.1' 'V49' '860' '323' 'V55' '579' '508' '969' '205' '462' 'E880'
 '680' '697' '826' '200' '457' '717' '738' '742' '735' '235' '308' '725'
 '241' '824' '464' '260' '917' '239' '661' '892' '261' 'E883' '943' '744'
 'E936' '796' '318' '967' '350' '854' 'E905' '9' '741' 'E941' '170' '643'
 '317' '759' '909' 'V22' '831' '713' '180' '801' '360' '359' '501' '335'
 '250.11' '306' '811' '690' 'V02' '271' '214' '847' '543' 'V63' '906'
 '842' '686' '445' '808' '861' 'E852' '220' 'E887' 'E858' '915' '970'
 '256' '747' '395' '243' '815' '481' '5' 'E927' '297' '299' '851' '864'
 '922' '384' 'E876' '225' '158' 'E937' '871' '88' '966' 'E917' 'E812'
 'V62' 'E924' '604' '233' 'E916' '377' '797' 'V72' '172' '7' '421' '852'
 'E819' '972' '916' '956' '3' 'E965' '173' '193' '154' '347' '862' '250.3'
 '987' '470' '262' 'E855' '161' '115' '179' '910' '312' '17' '460' '265'
 '66' '163' 'V60' '870' 'E906' '514' '944' '844' '417' '152' '183' '991'
 '216' '385' '164' '935' '510' '814' '485' '850' '250.21' 'E919' '872'
 '195' '431' '597' '933' '171' '884' '156' '868' '483' 'E815' '542' 'V61'
 '853' '374' 'E881' 'E882' 'E822' '192' '754' '327' '523' '500' 'V85'
 '992' '657' '684' '603' 'E826' '550' '913' '376' '755' '361' '186' '720'
 '250.31' '674' '911' 'E813' '226' '365.44' 'E818' '146' '955' 'E894'
 '475' 'V13' '880' '930' 'E915' '381' '132' '353' '795' '893' 'V01' 'E853'
 '863' '540' 'E828' '430' '800' 'E865' '148' 'E946' '822' '879' '848'
 'V86' 'V03' '338' '989' '388' 'E966' '111' 'E922' '123' '757' 'E901'
 '141' '268' 'E892' '649' '702' '948' '223' '484' 'E886' '838' '928' '236'
 '624' '837' 'E987' 'V07' '841' '622' 'E912' 'E955' '463' 'V06' 'E864'
 '217' '877' '391' 'E825' '952' '669' '875' 'E900' '215' '538' '980' '834'
 '448' '175' '49' '876' '230' '57' 'E854' '942' '14' '750' '370' '671'
 '971']

Feature: number_diagnoses
Distinct Count: 16
Distinct Values: [ 1  9  6  7  5  8  3  4  2 16 12 13 15 10 11 14]

Feature: max_glu_serum
Distinct Count: 4
Distinct Values: [nan '>300' 'Norm' '>200']

Feature: A1Cresult
Distinct Count: 4
Distinct Values: [nan '>7' '>8' 'Norm']

Feature: metformin
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Up' 'Down']

Feature: repaglinide
Distinct Count: 4
```

```
Distinct Values: ['No' 'Up' 'Steady' 'Down']

Feature: nateglinide
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Down' 'Up']

Feature: chlorpropamide
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Down' 'Up']

Feature: glimepiride
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Down' 'Up']

Feature: acetohexamide
Distinct Count: 2
Distinct Values: ['No' 'Steady']

Feature: glipizide
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Up' 'Down']

Feature: glyburide
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Up' 'Down']

Feature: tolbutamide
Distinct Count: 2
Distinct Values: ['No' 'Steady']

Feature: pioglitazone
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Up' 'Down']

Feature: rosiglitazone
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Up' 'Down']

Feature: acarbose
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Up' 'Down']

Feature: miglitol
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Down' 'Up']

Feature: troglitazone
Distinct Count: 2
```

```
Distinct Values: ['No' 'Steady']

Feature: tolazamide
Distinct Count: 3
Distinct Values: ['No' 'Steady' 'Up']

Feature: examide
Distinct Count: 1
Distinct Values: ['No']

Feature: citoglipton
Distinct Count: 1
Distinct Values: ['No']

Feature: insulin
Distinct Count: 4
Distinct Values: ['No' 'Up' 'Steady' 'Down']

Feature: glyburide-metformin
Distinct Count: 4
Distinct Values: ['No' 'Steady' 'Down' 'Up']

Feature: glipizide-metformin
Distinct Count: 2
Distinct Values: ['No' 'Steady']

Feature: glimepiride-pioglitazone
Distinct Count: 2
Distinct Values: ['No' 'Steady']

Feature: metformin-rosiglitazone
Distinct Count: 2
Distinct Values: ['No' 'Steady']

Feature: metformin-pioglitazone
Distinct Count: 2
Distinct Values: ['No' 'Steady']

Feature: change
Distinct Count: 2
Distinct Values: ['No' 'Ch']

Feature: diabetesMed
Distinct Count: 2
Distinct Values: ['No' 'Yes']

Feature: readmitted
Distinct Count: 3
```

```
Distinct Values: ['NO' '>30' '<30']
```

[4]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 50 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   encounter_id              101766 non-null  int64
 1   patient_nbr               101766 non-null  int64
 2   race                      101766 non-null  object
 3   gender                    101766 non-null  object
 4   age                       101766 non-null  object
 5   weight                    101766 non-null  object
 6   admission_type_id         101766 non-null  int64
 7   discharge_disposition_id  101766 non-null  int64
 8   admission_source_id       101766 non-null  int64
 9   time_in_hospital          101766 non-null  int64
 10  payer_code                101766 non-null  object
 11  medical_specialty         101766 non-null  object
 12  num_lab_procedures        101766 non-null  int64
 13  num_procedures            101766 non-null  int64
 14  num_medications           101766 non-null  int64
 15  number_outpatient         101766 non-null  int64
 16  number_emergency          101766 non-null  int64
 17  number_inpatient          101766 non-null  int64
 18  diag_1                    101766 non-null  object
 19  diag_2                    101766 non-null  object
 20  diag_3                    101766 non-null  object
 21  number_diagnoses          101766 non-null  int64
 22  max_glu_serum             5346 non-null    object
 23  A1Cresult                 17018 non-null   object
 24  metformin                 101766 non-null  object
 25  repaglinide               101766 non-null  object
 26  nateglinide               101766 non-null  object
 27  chlorpropamide            101766 non-null  object
 28  glimepiride               101766 non-null  object
 29  acetohexamide             101766 non-null  object
 30  glipizide                 101766 non-null  object
 31  glyburide                 101766 non-null  object
 32  tolbutamide               101766 non-null  object
 33  pioglitazone              101766 non-null  object
 34  rosiglitazone             101766 non-null  object
 35  acarbose                  101766 non-null  object
 36  miglitol                  101766 non-null  object
 37  troglitazone              101766 non-null  object
```

```
 38  tolazamide                101766 non-null  object
 39  examide                   101766 non-null  object
 40  citoglipton               101766 non-null  object
 41  insulin                   101766 non-null  object
 42  glyburide-metformin       101766 non-null  object
 43  glipizide-metformin       101766 non-null  object
 44  glimepiride-pioglitazone  101766 non-null  object
 45  metformin-rosiglitazone   101766 non-null  object
 46  metformin-pioglitazone    101766 non-null  object
 47  change                    101766 non-null  object
 48  diabetesMed               101766 non-null  object
 49  readmitted                101766 non-null  object
dtypes: int64(13), object(37)
memory usage: 38.8+ MB
None
```

5 Data Preprocessing

6 Grouping into Class 0 and 1

[5]:
```
#count the number of readdmission by each type
df.groupby('readmitted').size()
```

[5]:
```
readmitted
<30    11357
>30    35545
NO     54864
dtype: int64
```

[6]:
```
# Define the label function
def label(x):
    if x == 'NO' or x == '>30':
        return 0
    elif x == '<30':
        return 1
    else:
        return None  # Handle unexpected values

# Apply the label function to the 'readmitted' column
df['readmitted'] = df['readmitted'].apply(label)

# Display the modified DataFrame to verify changes
print(df.head())
```

```
   encounter_id  patient_nbr             race  gender      age weight  \
0       2278392      8222157        Caucasian  Female   [0-10)      ?
1        149190     55629189        Caucasian  Female  [10-20)      ?
```

```
2         64410     86047875  AfricanAmerican  Female  [20-30)      ?
3        500364     82442376        Caucasian    Male  [30-40)      ?
4         16680     42519267        Caucasian    Male  [40-50)      ?

   admission_type_id  discharge_disposition_id  admission_source_id  \
0                  6                        25                    1
1                  1                         1                    7
2                  1                         1                    7
3                  1                         1                    7
4                  1                         1                    7

   time_in_hospital  … citoglipton insulin  glyburide-metformin  \
0                 1  …          No      No                   No
1                 3  …          No      Up                   No
2                 2  …          No      No                   No
3                 2  …          No      Up                   No
4                 1  …          No  Steady                   No

   glipizide-metformin  glimepiride-pioglitazone  metformin-rosiglitazone  \
0                   No                        No                       No
1                   No                        No                       No
2                   No                        No                       No
3                   No                        No                       No
4                   No                        No                       No

   metformin-pioglitazone  change diabetesMed readmitted
0                      No      No          No          0
1                      No      Ch         Yes          0
2                      No      No         Yes          0
3                      No      Ch         Yes          0
4                      No      Ch         Yes          0

[5 rows x 50 columns]
```

[7]:
```
#count the number of readdmission by each type
df.groupby('readmitted').size()
```

[7]:
```
readmitted
0    90409
1    11357
dtype: int64
```

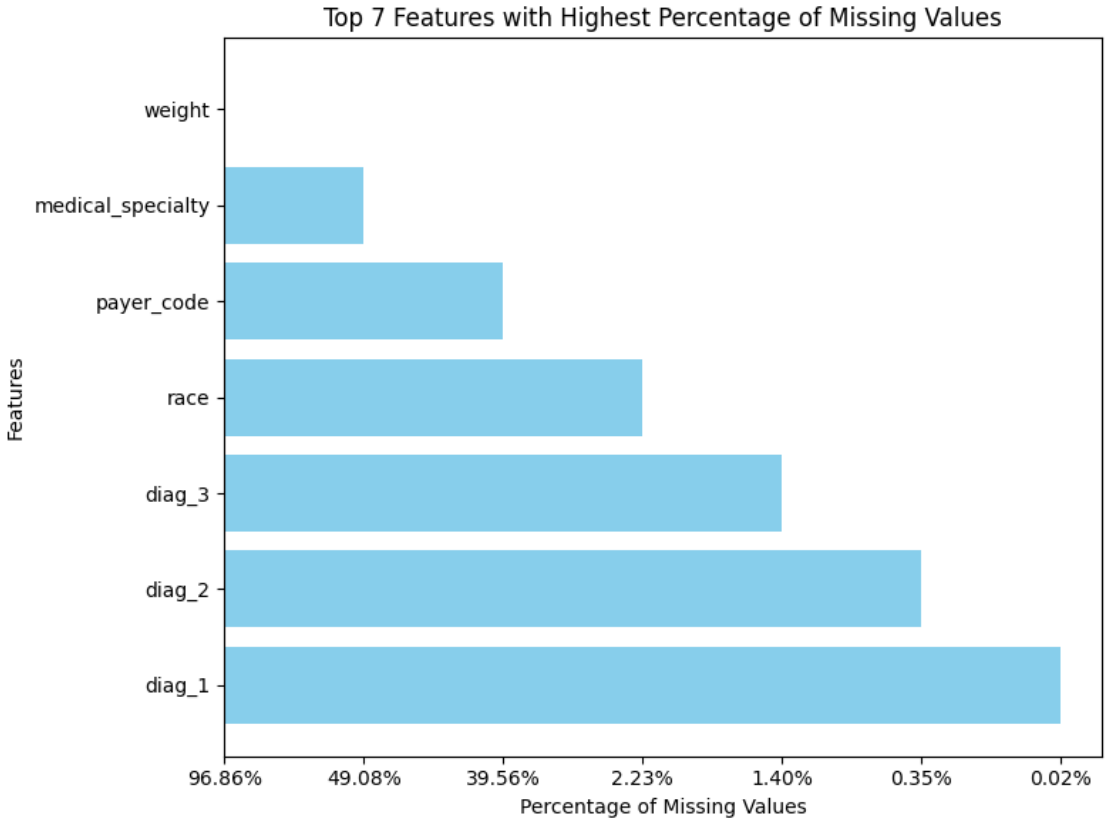
7 Finding Missing features percentage

[8]:
```
# Checking for missing values in the dataset where missing values are␣
 ↪represented as '?'
# Initialize a dictionary to store the information
missing_info = {
    'Column': [],
    'Missing Count': [],
    'Missing Percentage': []
}

# Loop through the columns to calculate missing values
for col in df.columns:
    if df[col].dtype == object:
        missing_count = df[col][df[col] == '?'].count()
        if missing_count > 0:
            missing_percentage = (missing_count / len(df[col])) * 100
            missing_info['Column'].append(col)
            missing_info['Missing Count'].append(missing_count)
            missing_info['Missing Percentage'].append(f"{missing_percentage:.
 ↪2f}%")

# Create a DataFrame from the dictionary
missing_df = pd.DataFrame(missing_info)

#Display columns with missing data and their respective count and missing␣
 ↪percentages
print("Columns with missing data, their count, and respective missing␣
 ↪percentages:")
print(missing_df)
```

```
Columns with missing data, their count, and respective missing percentages:
              Column  Missing Count Missing Percentage
0               race           2273              2.23%
1             weight          98569             96.86%
2         payer_code          40256             39.56%
3  medical_specialty          49949             49.08%
4             diag_1             21              0.02%
5             diag_2            358              0.35%
6             diag_3           1423              1.40%
```

8 Missing value plot

[9]:
```
# Select top 7 features with highest percentage of missing values
top_7_missing = missing_df.sort_values(by='Missing Percentage',␣
 ↪ascending=False).head(7)

# Visualize the percentage of missing values for the top 7 features
plt.figure(figsize=(8, 6))
plt.barh(top_7_missing['Column'], top_7_missing['Missing Percentage'],␣
 ↪color='skyblue')
plt.xlabel('Percentage of Missing Values')
plt.ylabel('Features')
plt.title('Top 7 Features with Highest Percentage of Missing Values')
plt.gca().invert_yaxis()  # Invert y-axis to display highest percentage at the␣
 ↪top
plt.tight_layout()
plt.show()
```



9 Age Feature

[10]:
```
df.groupby('age').size()
```

[10]:
```
age
[0-10)        161
[10-20)       691
[20-30)      1657
[30-40)      3775
[40-50)      9685
[50-60)     17256
[60-70)     22483
[70-80)     26068
[80-90)     17197
[90-100)     2793
dtype: int64
```

[11]:
```
age_id = {'[0-10)':5,
'[10-20)': 15,
'[20-30)':25,
'[30-40)':35,
'[40-50)':45,
'[50-60)':55,
'[60-70)':65,
'[70-80)':75,
'[80-90)':85,
'[90-100)':95}

df['age'] = df['age'].apply(lambda x: age_id[x])
```

10 Dealing the columns with missing data

11 Dropping weight, payer_code,medical_specialty

[12]:
```
df = df.drop(['weight', 'payer_code','medical_specialty'], axis = 1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 47 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   encounter_id              101766 non-null  int64
 1   patient_nbr               101766 non-null  int64
 2   race                      101766 non-null  object
 3   gender                    101766 non-null  object
```

```
 4   age                       101766 non-null  int64
 5   admission_type_id         101766 non-null  int64
 6   discharge_disposition_id  101766 non-null  int64
 7   admission_source_id       101766 non-null  int64
 8   time_in_hospital          101766 non-null  int64
 9   num_lab_procedures        101766 non-null  int64
 10  num_procedures            101766 non-null  int64
 11  num_medications           101766 non-null  int64
 12  number_outpatient         101766 non-null  int64
 13  number_emergency          101766 non-null  int64
 14  number_inpatient          101766 non-null  int64
 15  diag_1                    101766 non-null  object
 16  diag_2                    101766 non-null  object
 17  diag_3                    101766 non-null  object
 18  number_diagnoses          101766 non-null  int64
 19  max_glu_serum             5346 non-null    object
 20  A1Cresult                 17018 non-null   object
 21  metformin                 101766 non-null  object
 22  repaglinide               101766 non-null  object
 23  nateglinide               101766 non-null  object
 24  chlorpropamide            101766 non-null  object
 25  glimepiride               101766 non-null  object
 26  acetohexamide             101766 non-null  object
 27  glipizide                 101766 non-null  object
 28  glyburide                 101766 non-null  object
 29  tolbutamide               101766 non-null  object
 30  pioglitazone              101766 non-null  object
 31  rosiglitazone             101766 non-null  object
 32  acarbose                  101766 non-null  object
 33  miglitol                  101766 non-null  object
 34  troglitazone              101766 non-null  object
 35  tolazamide                101766 non-null  object
 36  examide                   101766 non-null  object
 37  citoglipton               101766 non-null  object
 38  insulin                   101766 non-null  object
 39  glyburide-metformin       101766 non-null  object
 40  glipizide-metformin       101766 non-null  object
 41  glimepiride-pioglitazone  101766 non-null  object
 42  metformin-rosiglitazone   101766 non-null  object
 43  metformin-pioglitazone    101766 non-null  object
 44  change                    101766 non-null  object
 45  diabetesMed               101766 non-null  object
 46  readmitted                101766 non-null  int64
dtypes: int64(15), object(32)
memory usage: 36.5+ MB
```

12 # Feature diag_1 , diag_2 , diag_3 imputation by most common value

[13]:
```
from collections import Counter
diag_1 = Counter(list(df['diag_1'])).most_common(1)[0][0]
diag_2 = Counter(list(df['diag_2'])).most_common(1)[0][0]
diag_3 = Counter(list(df['diag_3'])).most_common(1)[0][0]
df['diag_1'] = df['diag_1'].apply(lambda x : diag_1 if x == '?' else x)
df['diag_2'] = df['diag_2'].apply(lambda x : diag_2 if x == '?' else x)
df['diag_3'] = df['diag_3'].apply(lambda x : diag_3 if x == '?' else x)
```

13 Imputation Race Feature

[14]:
```
# Count the frequency of each type in the 'race' column
race_freq = df['race'].value_counts()

# Print the frequency
print(race_freq)
```

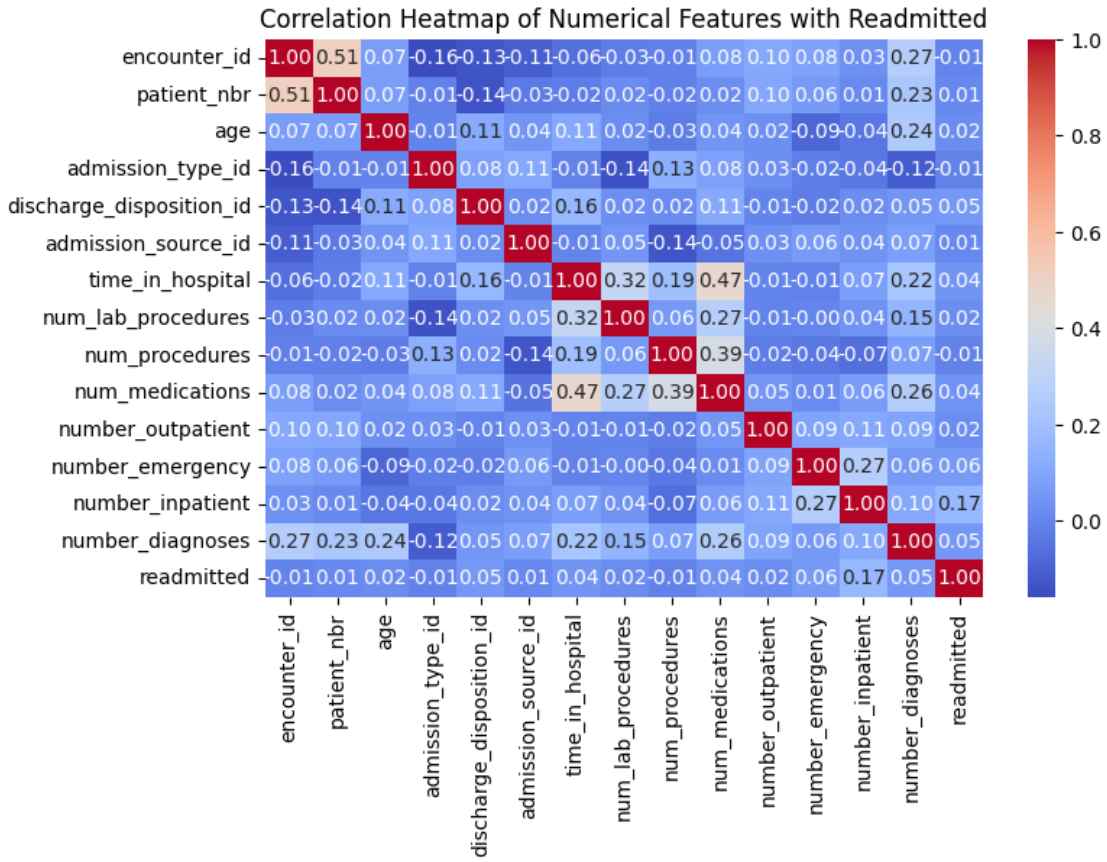
```
race
Caucasian          76099
AfricanAmerican    19210
?                   2273
Hispanic            2037
Other               1506
Asian                641
Name: count, dtype: int64
```

[15]:
```
# Print the number of instances to be imputed for each race category
print('Caucasian will be imputed for:', int(2273 * 0.53))
print('AfricanAmerican will be imputed for:', int(2273 * 0.12))
print('Hispanic will be imputed for:', int(2273 * 0.01))

# Replace missing values in 'race' column with specified percentages
df.loc[df['race'] == '?', 'race'] = 'Caucasian'
df.loc[df['race'] == '?', 'race'] = 'AfricanAmerican'
df.loc[df['race'] == '?', 'race'] = 'Hispanic'

# Check unique values in the 'race' column to ensure all missing values are␣
↪imputed
print(df['race'].unique())
```

```
Caucasian will be imputed for: 1204
AfricanAmerican will be imputed for: 272
Hispanic will be imputed for: 22
['Caucasian' 'AfricanAmerican' 'Other' 'Asian' 'Hispanic']
```

14 Correlation between columns

[16]:
```
#Identify Numerical and Non-Numerical Columns
numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
non_numerical_columns = df.select_dtypes(include=['object', 'string',␣
 ↪'category']).columns

# Create DataFrames for Numerical and Non-Numerical Data
numerical_df = df[numerical_columns].copy()
non_numerical_df = df[non_numerical_columns].copy()

# Add the 'readmitted' column to both DataFrames
numerical_df['readmitted'] = df['readmitted']
non_numerical_df['readmitted'] = df['readmitted']

#Calculate Correlation Matrices for Numerical Columns
numerical_corr = numerical_df.corr()

# Analyze Non-Numerical Columns Against `readmitted`
# This can be done using cross-tabulation for categorical columns
non_numerical_analysis = {col: pd.crosstab(df[col], df['readmitted']) for col␣
 ↪in non_numerical_columns}

# Plot Heatmaps for Numerical Data Correlation
plt.figure(figsize=(8, 5))
sns.heatmap(numerical_corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap of Numerical Features with Readmitted')
plt.show()

# Display analysis for non-numerical columns
for col, analysis in non_numerical_analysis.items():
    print(f'Analysis for {col} against readmitted:')
    print(analysis)
    print('\n')
```




```
Analysis for race against readmitted:
readmitted           0     1
race
AfricanAmerican  17055  2155
Asian              576    65
Caucasian        69592  8780
Hispanic          1825   212
Other             1361   145


Analysis for gender against readmitted:
readmitted           0     1
gender
Female           48556  6152
Male             41850  5205
Unknown/Invalid      3     0


Analysis for diag_1 against readmitted:
readmitted   0   1
```

```
diag_1
10            1   0
11           10   0
110           2   0
112          59  14
114           1   0
…           ..  ..
V63           7   1
V66           2   0
V67           1   0
V70           1   0
V71           9   0

[716 rows x 2 columns]


Analysis for diag_2 against readmitted:
readmitted    0   1
diag_2
11            2   1
110           6   2
111           1   0
112         173  28
114           0   1
…           …  ..
V69           1   0
V70           7   0
V72          13   0
V85         147  22
V86           2   0

[748 rows x 2 columns]


Analysis for diag_3 against readmitted:
readmitted    0   1
diag_3
11            1   1
110          18   2
111           0   1
112         184  22
115           1   0
…           …  ..
V66          17   1
V70           2   0
V72           7   1
V85          87   9
V86           3   0
```

```
[789 rows x 2 columns]


Analysis for max_glu_serum against readmitted:
readmitted        0    1
max_glu_serum
>200           1300  185
>300           1083  181
Norm           2302  295


Analysis for A1Cresult against readmitted:
readmitted     0    1
A1Cresult
>7          3429  383
>8          7405  811
Norm        4508  482


Analysis for metformin against readmitted:
readmitted      0     1
metformin
Down          506    69
No          72360  9418
Steady      16564  1782
Up            979    88


Analysis for repaglinide against readmitted:
readmitted       0      1
repaglinide
Down            42      3
No           89075  11152
Steady        1202    182
Up              90     20


Analysis for nateglinide against readmitted:
readmitted       0      1
nateglinide
Down            10      1
No           89786  11277
Steady         590     78
Up              23      1


Analysis for chlorpropamide against readmitted:
```

```
readmitted          0      1
chlorpropamide
Down                1      0
No              90328  11352
Steady             74      5
Up                  6      0


Analysis for glimepiride against readmitted:
readmitted       0      1
glimepiride
Down           169     25
No           85748  10827
Steady        4202    468
Up             290     37


Analysis for acetohexamide against readmitted:
readmitted         0      1
acetohexamide
No             90408  11357
Steady             1      0


Analysis for glipizide against readmitted:
readmitted      0     1
glipizide
Down          475    85
No          79175  9905
Steady      10088  1268
Up            671    99


Analysis for glyburide against readmitted:
readmitted      0      1
glyburide
Down          512     52
No          80891  10225
Steady       8279    995
Up            727     85


Analysis for tolbutamide against readmitted:
readmitted       0      1
tolbutamide
No           90387  11356
Steady          22      1
```

```
Analysis for pioglitazone against readmitted:
readmitted        0      1
pioglitazone
Down            100     18
No            83855  10583
Steady         6249    727
Up              205     29


Analysis for rosiglitazone against readmitted:
readmitted         0      1
rosiglitazone
Down              82      5
No             84709  10692
Steady          5459    641
Up               159     19


Analysis for acarbose against readmitted:
readmitted      0      1
acarbose
Down            2      1
No          90129  11329
Steady        270     25
Up              8      2


Analysis for miglitol against readmitted:
readmitted      0      1
miglitol
Down            3      2
No          90375  11353
Steady         29      2
Up              2      0


Analysis for troglitazone against readmitted:
readmitted        0      1
troglitazone
No            90406  11357
Steady            3      0


Analysis for tolazamide against readmitted:
readmitted      0      1
tolazamide
No          90373  11354
```

```
Steady         35      3
Up              1      0


Analysis for examide against readmitted:
readmitted      0      1
examide
No          90409  11357


Analysis for citoglipton against readmitted:
readmitted       0      1
citoglipton
No           90409  11357


Analysis for insulin against readmitted:
readmitted      0     1
insulin
Down        10520  1698
No          42627  4756
Steady      27416  3433
Up           9846  1470


Analysis for glyburide-metformin against readmitted:
readmitted               0      1
glyburide-metformin
Down                     5      1
No                   89781  11279
Steady                 615     77
Up                       8      0


Analysis for glipizide-metformin against readmitted:
readmitted               0      1
glipizide-metformin
No                   90397  11356
Steady                  12      1


Analysis for glimepiride-pioglitazone against readmitted:
readmitted                    0      1
glimepiride-pioglitazone
No                        90408  11357
Steady                        1      0
```

```
Analysis for metformin-rosiglitazone against readmitted:
readmitted                    0      1
metformin-rosiglitazone
No                        90407  11357
Steady                        2      0


Analysis for metformin-pioglitazone against readmitted:
readmitted                    0      1
metformin-pioglitazone
No                        90408  11357
Steady                        1      0


Analysis for change against readmitted:
readmitted      0     1
change
Ch          41453  5558
No          48956  5799


Analysis for diabetesMed against readmitted:
readmitted       0     1
diabetesMed
No           21157  2246
Yes          69252  9111
```

15 Plot of categorical columns

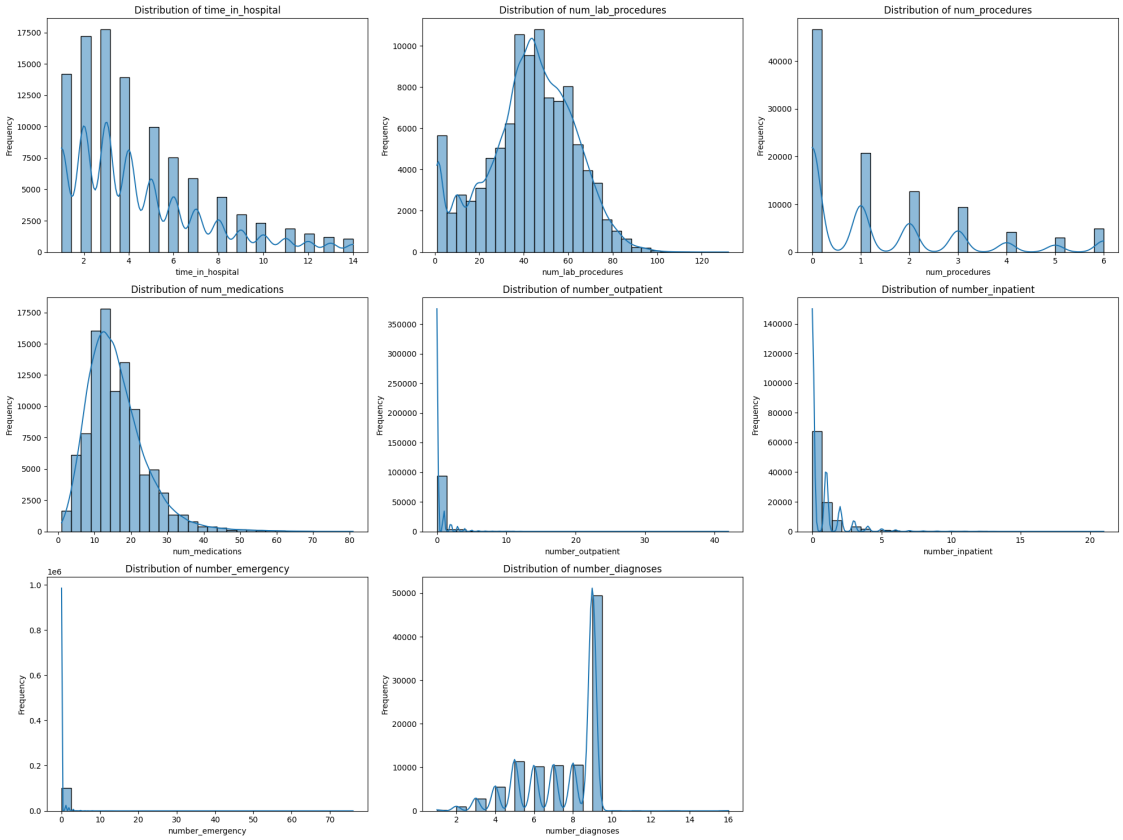
[17]:
```
#Identify Non-Numerical Columns
non_numerical_columns = df.select_dtypes(include=['object', 'string',␣
 ↪'category']).columns

# Add the 'readmitted' Column to Non-Numerical DataFrame
non_numerical_df = df[non_numerical_columns].copy()
non_numerical_df['readmitted'] = df['readmitted']

# Plot Non-Numerical Data in Subgraphs
num_cols = len(non_numerical_columns)
num_plots_per_row = 3
num_rows = (num_cols // num_plots_per_row) + (num_cols % num_plots_per_row > 0)

fig, axes = plt.subplots(num_rows, num_plots_per_row, figsize=(20, 5 *␣
 ↪num_rows))
```

```
# Flatten axes array if more than one row
if num_rows > 1:
    axes = axes.flatten()
else:
    axes = [axes]

for ax, col in zip(axes, non_numerical_columns):
    sns.countplot(data=non_numerical_df, x=col, hue='readmitted', ax=ax)
    ax.set_title(f'{col} vs readmitted')
    ax.set_xlabel(col)
    ax.set_ylabel('Count')
    ax.legend(title='readmitted')

# Remove any empty subplots
for i in range(num_cols, len(axes)):
    fig.delaxes(axes[i])

plt.tight_layout()
plt.show()
```


race vs readmitted
gender vs readmitted
diag_1 vs readmitted
diag_2 vs readmitted
diag_3 vs readmitted
max_glu_serum vs readmitted
A1Cresult vs readmitted
metformin vs readmitted
repaglinide vs readmitted
nateglinide vs readmitted
chlorpropamide vs readmitted
glimepiride vs readmitted
acetohexamide vs readmitted
glipizide vs readmitted
glyburide vs readmitted
tolbutamide vs readmitted
pioglitazone vs readmitted
rosiglitazone vs readmitted
acarbose vs readmitted
miglitol vs readmitted
troglitazone vs readmitted
tolazamide vs readmitted
examide vs readmitted
citoglipton vs readmitted
insulin vs readmitted
glyburide-metformin vs readmitted
glipizide-metformin vs readmitted
glimepiride-pioglitazone vs readmitted
metformin-rosiglitazone vs readmitted
metformin-pioglitazone vs readmitted
change vs readmitted
diabetesMed vs readmitted
readmitted
Count

16 Plot of numerical_columns

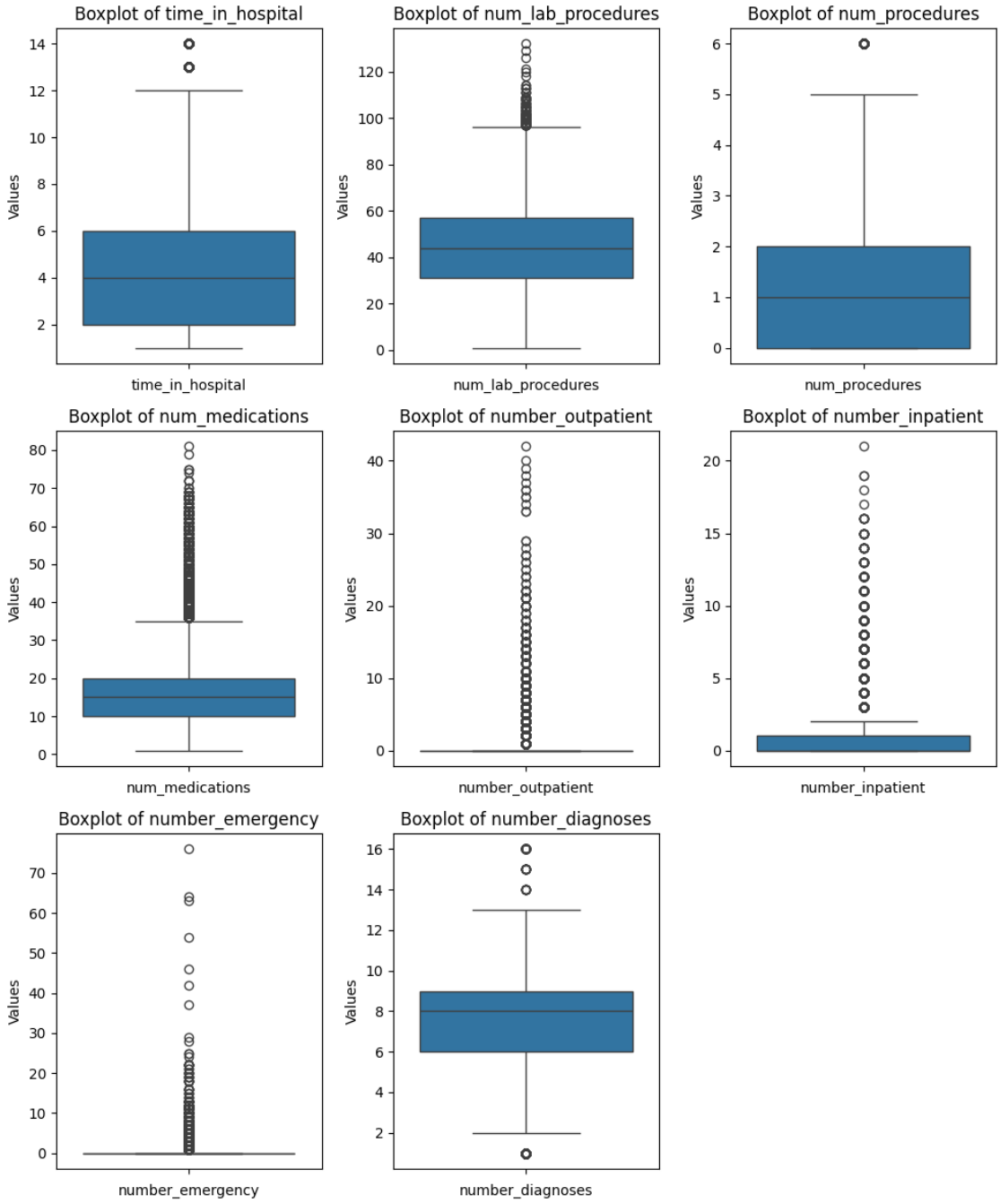
[18]:
```
# List of features to analyze
features = ['time_in_hospital', 'num_lab_procedures', 'num_procedures',␣
 ↪'num_medications',
            'number_outpatient', 'number_inpatient', 'number_emergency',␣
 ↪'number_diagnoses']

# Plot Histograms in Subplots
num_features = len(features)
num_plots_per_row = 3
num_rows = (num_features // num_plots_per_row) + (num_features %␣
 ↪num_plots_per_row > 0)

fig, axes = plt.subplots(num_rows, num_plots_per_row, figsize=(20, 5 *␣
 ↪num_rows))

# Flatten axes array if more than one row
if num_rows > 1:
    axes = axes.flatten()
else:
    axes = [axes]

for ax, feature in zip(axes, features):
    sns.histplot(df[feature], bins=30, kde=True, ax=ax)
    ax.set_title(f'Distribution of {feature}')
    ax.set_xlabel(feature)
    ax.set_ylabel('Frequency')

# Remove any empty subplots
for i in range(num_features, len(axes)):
    fig.delaxes(axes[i])

plt.tight_layout()
plt.show()
```



17 Outlier detection

[19]:
```
# List of features to analyze
features = ['time_in_hospital', 'num_lab_procedures', 'num_procedures',␣
 ↪'num_medications',
            'number_outpatient', 'number_inpatient', 'number_emergency',␣
 ↪'number_diagnoses']

#Plot Boxplots in Subplots
num_features = len(features)
num_plots_per_row = 3
num_rows = (num_features // num_plots_per_row) + (num_features %␣
 ↪num_plots_per_row > 0)

fig, axes = plt.subplots(num_rows, num_plots_per_row, figsize=(10, 4 *␣
 ↪num_rows))

# Flatten axes array if more than one row
if num_rows > 1:
    axes = axes.flatten()
```

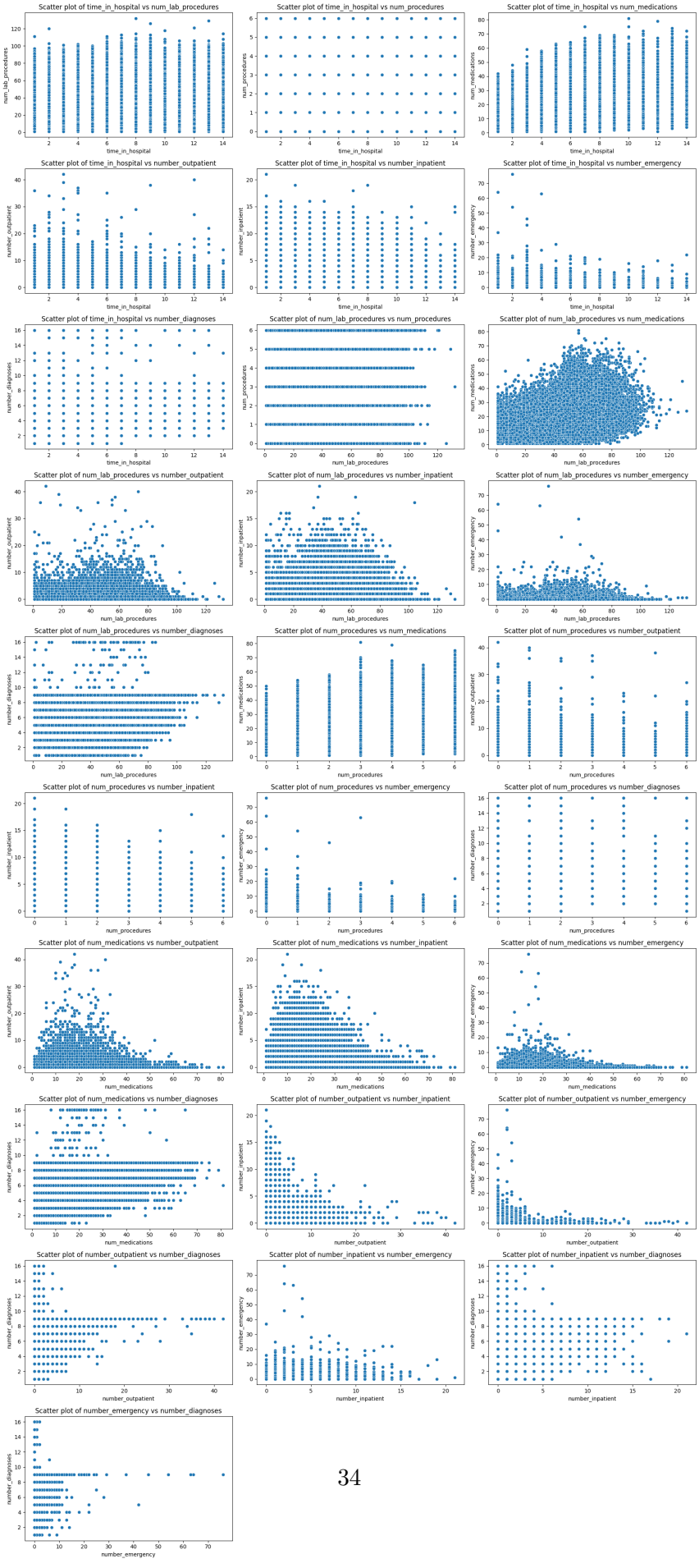
```
else:
    axes = [axes]

for ax, feature in zip(axes, features):
    sns.boxplot(y=df[feature], ax=ax)
    ax.set_title(f'Boxplot of {feature}')
    ax.set_xlabel(feature)
    ax.set_ylabel('Values')

# Remove any empty subplots
for i in range(num_features, len(axes)):
    fig.delaxes(axes[i])

plt.tight_layout()
plt.show()
```



18 Scatter plots for continuous variables

[20]:
```
continuous_vars = ['time_in_hospital', 'num_lab_procedures', 'num_procedures',␣
↪'num_medications',
```

```
                    'number_outpatient', 'number_inpatient', 'number_emergency',␣
 ↪'number_diagnoses']

# Calculate the number of continuous variable pairs
num_pairs = len(continuous_vars) * (len(continuous_vars) - 1) // 2

# Calculate the number of rows needed (3 plots per row)
rows = (num_pairs + 2) // 3

# Create subplots
fig, axes = plt.subplots(rows, 3, figsize=(18, 4 * rows))
axes = axes.flatten()  # Flatten the 2D array of axes for easy indexing

pair_idx = 0
for i in range(len(continuous_vars)):
    for j in range(i + 1, len(continuous_vars)):
        if pair_idx < len(axes):
            ax = axes[pair_idx]
            sns.scatterplot(x=df[continuous_vars[i]], y=df[continuous_vars[j]],␣
 ↪ax=ax)
            ax.set_title(f'Scatter plot of {continuous_vars[i]} vs␣
 ↪{continuous_vars[j]}')
            ax.set_xlabel(continuous_vars[i])
            ax.set_ylabel(continuous_vars[j])
            pair_idx += 1

# Remove any unused subplots
for k in range(pair_idx, len(axes)):
    fig.delaxes(axes[k])

plt.tight_layout()
plt.show()
```

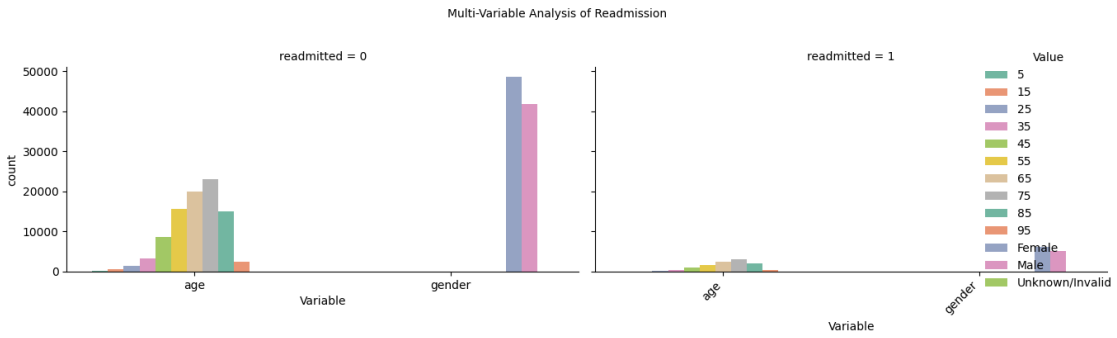

Scatter plot of time_in_hospital vs num_lab_procedures
Scatter plot of time_in_hospital vs num_procedures
Scatter plot of time_in_hospital vs num_medications
Scatter plot of time_in_hospital vs number_outpatient
Scatter plot of time_in_hospital vs number_inpatient
Scatter plot of time_in_hospital vs number_emergency
Scatter plot of time_in_hospital vs number_diagnoses
Scatter plot of num_lab_procedures vs num_procedures
Scatter plot of num_lab_procedures vs num_medications
Scatter plot of num_lab_procedures vs number_outpatient
Scatter plot of num_lab_procedures vs number_inpatient
Scatter plot of num_lab_procedures vs number_emergency
Scatter plot of num_lab_procedures vs number_diagnoses
Scatter plot of num_procedures vs num_medications
Scatter plot of num_procedures vs number_outpatient
Scatter plot of num_procedures vs number_inpatient
Scatter plot of num_procedures vs number_emergency
Scatter plot of num_procedures vs number_diagnoses
Scatter plot of num_medications vs number_outpatient
Scatter plot of num_medications vs number_inpatient
Scatter plot of num_medications vs number_emergency
Scatter plot of num_medications vs number_diagnoses
Scatter plot of number_outpatient vs number_inpatient
Scatter plot of number_outpatient vs number_emergency
Scatter plot of number_outpatient vs number_diagnoses
Scatter plot of number_inpatient vs number_emergency
Scatter plot of number_inpatient vs number_diagnoses
Scatter plot of number_emergency vs number_diagnoses

19 Multivariable relationships

20 Readmission Rates by Age, Gender

[21]:
```
# Select relevant columns
columns_of_interest = ['age','gender','readmitted']
subset_df = df[columns_of_interest]

# Melt dataframe to long format for visualization
subset_df_melted = subset_df.melt(id_vars='readmitted', var_name='Variable',␣
 ↪value_name='Value')

# Create catplot
plt.figure(figsize=(10, 5))
sns.catplot(data=subset_df_melted, x='Variable', hue='Value', col='readmitted',␣
 ↪kind='count',
            height=4, aspect=1.5, palette='Set2', col_order=[0, 1])

# Set titles and labels
plt.suptitle('Multi-Variable Analysis of Readmission', fontsize=10, y=1.02)
plt.xlabel('Variable')
plt.ylabel('Count')

# Rotate x-axis labels
plt.xticks(rotation=45, ha='right')

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```

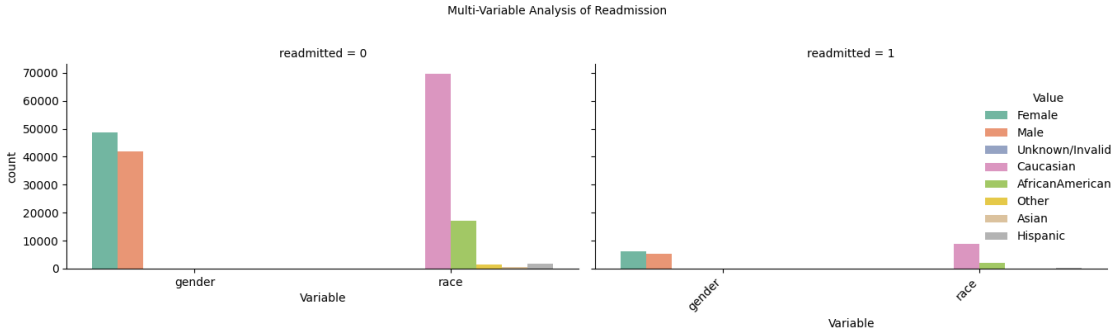
```
<Figure size 1000x500 with 0 Axes>
```



21 Readmission Rates by Gender, and Race

[22]:
```
# Select relevant columns
columns_of_interest = ['gender','race','readmitted']
subset_df = df[columns_of_interest]

# Melt dataframe to long format for visualization
subset_df_melted = subset_df.melt(id_vars='readmitted', var_name='Variable',␣
 ↪value_name='Value')

# Create catplot
plt.figure(figsize=(10, 5))
sns.catplot(data=subset_df_melted, x='Variable', hue='Value', col='readmitted',␣
 ↪kind='count',
            height=4, aspect=1.5, palette='Set2', col_order=[0, 1])

# Set titles and labels
plt.suptitle('Multi-Variable Analysis of Readmission', fontsize=10, y=1.02)
plt.xlabel('Variable')
plt.ylabel('Count')

# Rotate x-axis labels
plt.xticks(rotation=45, ha='right')

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```

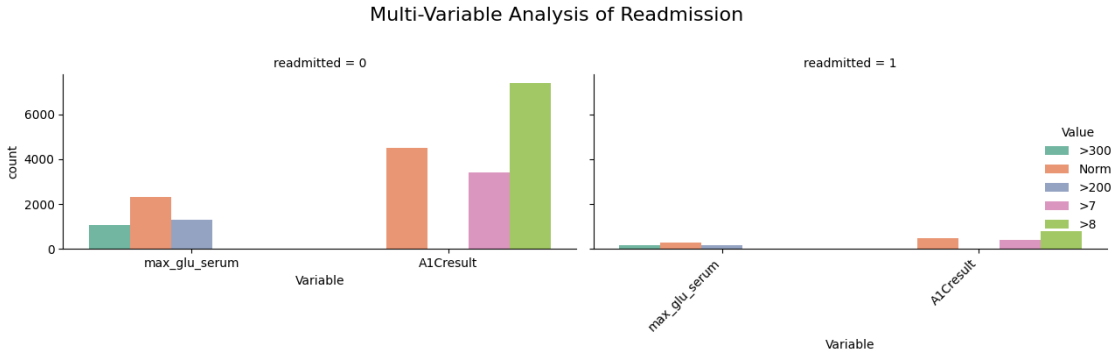
```
<Figure size 1000x500 with 0 Axes>
```



22 Readmission Rates by max_glu_serum, A1Cresult

[23]:
```
# Select relevant columns
columns_of_interest = ['max_glu_serum', 'A1Cresult', 'readmitted']
subset_df = df[columns_of_interest]

# Melt dataframe to long format for visualization
subset_df_melted = subset_df.melt(id_vars='readmitted', var_name='Variable',␣
 ↪value_name='Value')

# Create catplot
plt.figure(figsize=(16, 12))
sns.catplot(data=subset_df_melted, x='Variable', hue='Value', col='readmitted',␣
 ↪kind='count',
            height=4, aspect=1.5, palette='Set2', col_order=[0, 1])

# Set titles and labels
plt.suptitle('Multi-Variable Analysis of Readmission', fontsize=16, y=1.02)
plt.xlabel('Variable')
plt.ylabel('Count')

# Rotate x-axis labels
plt.xticks(rotation=45, ha='right')

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```

```
<Figure size 1600x1200 with 0 Axes>
```



23 # Feature Engineering

[24]:
```
print(df.head())
```

```
   encounter_id  patient_nbr             race  gender  age  admission_type_id  \
0       2278392      8222157        Caucasian  Female    5                  6
1        149190     55629189        Caucasian  Female   15                  1
2         64410     86047875  AfricanAmerican  Female   25                  1
3        500364     82442376        Caucasian    Male   35                  1
4         16680     42519267        Caucasian    Male   45                  1

   discharge_disposition_id  admission_source_id  time_in_hospital  \
0                        25                    1                 1
1                         1                    7                 3
2                         1                    7                 2
3                         1                    7                 2
4                         1                    7                 1

   num_lab_procedures  …  citoglipton  insulin  glyburide-metformin  \
0                  41  …           No       No                   No
1                  59  …           No       Up                   No
2                  11  …           No       No                   No
3                  44  …           No       Up                   No
4                  51  …           No   Steady                   No

   glipizide-metformin  glimepiride-pioglitazone metformin-rosiglitazone  \
0                   No                        No                      No
1                   No                        No                      No
2                   No                        No                      No
3                   No                        No                      No
4                   No                        No                      No

  metformin-pioglitazone change  diabetesMed readmitted
0                     No     No           No          0
1                     No     Ch          Yes          0
2                     No     No          Yes          0
3                     No     Ch          Yes          0
4                     No     Ch          Yes          0

[5 rows x 47 columns]
```

24 Transformation of positively skewed plot

[25]:
```
from scipy.stats import skew

# Select only numeric columns
```

```
numeric_df = df.select_dtypes(include=['number'])

# Check skewness of each numeric variable
skewness = numeric_df.apply(skew)
print("Skewness before transformation:\n", skewness)

# Identify variables with skewness > 1
high_skew_columns = skewness[skewness > 1].index.tolist()
print("Variables with skewness > 1:", high_skew_columns)
```

```
Skewness before transformation:
 encounter_id                0.699131
patient_nbr                 0.471274
age                        -0.630530
admission_type_id           1.591961
discharge_disposition_id    2.563029
admission_source_id         1.029920
time_in_hospital            1.133982
num_lab_procedures         -0.236540
num_procedures              1.316395
num_medications             1.326653
number_outpatient           8.832829
number_emergency           22.855245
number_inpatient            3.614086
number_diagnoses           -0.876733
readmitted                  2.467034
dtype: float64
Variables with skewness > 1: ['admission_type_id', 'discharge_disposition_id',
'admission_source_id', 'time_in_hospital', 'num_procedures', 'num_medications',
'number_outpatient', 'number_emergency', 'number_inpatient', 'readmitted']
```

[26]:
```
# Apply square-root transformation for count data and log transformation for␣
 ↪size data
for col in high_skew_columns:
    if col in ['time_in_hospital', 'num_procedures', 'number_outpatient',␣
 ↪'number_inpatient', 'number_emergency']:
        df[f'{col}_sqrt'] = np.sqrt(df[col])
    elif col in ['num_medications']:
        df[f'{col}_log'] = np.log1p(df[col])  # np.log1p is equivalent to log(1␣
 ↪+ x)

# Check skewness after transformation
transformed_skewness = df[[f'{col}_sqrt' for col in high_skew_columns if␣
 ↪f'{col}_sqrt' in df.columns] +
                          [f'{col}_log' for col in high_skew_columns if␣
 ↪f'{col}_log' in df.columns]].apply(skew)
print("Skewness after transformation:\n", transformed_skewness)
```

```
# Display the transformed dataframe
print(df)
```

```
Skewness after transformation:
 time_in_hospital_sqrt     0.473596
num_procedures_sqrt        0.405462
number_outpatient_sqrt     2.739159
number_emergency_sqrt      3.679552
number_inpatient_sqrt      1.300642
num_medications_log       -0.485321
dtype: float64
        encounter_id  patient_nbr             race  gender  age  \
0            2278392      8222157        Caucasian  Female    5
1             149190     55629189        Caucasian  Female   15
2              64410     86047875  AfricanAmerican  Female   25
3             500364     82442376        Caucasian    Male   35
4              16680     42519267        Caucasian    Male   45
…                …            …                …     …  …
101761     443847548    100162476  AfricanAmerican    Male   75
101762     443847782     74694222  AfricanAmerican  Female   85
101763     443854148     41088789        Caucasian    Male   75
101764     443857166     31693671        Caucasian  Female   85
101765     443867222    175429310        Caucasian    Male   75

        admission_type_id  discharge_disposition_id  admission_source_id  \
0                       6                        25                    1
1                       1                         1                    7
2                       1                         1                    7
3                       1                         1                    7
4                       1                         1                    7
…                     …                         …                    …
101761                  1                         3                    7
101762                  1                         4                    5
101763                  1                         1                    7
101764                  2                         3                    7
101765                  1                         1                    7

        time_in_hospital  num_lab_procedures  …  metformin-pioglitazone  \
0                      1                  41  …                      No
1                      3                  59  …                      No
2                      2                  11  …                      No
3                      2                  44  …                      No
4                      1                  51  …                      No
…                    …                   …  …                       …
101761                 3                  51  …                      No
101762                 5                  33  …                      No
```

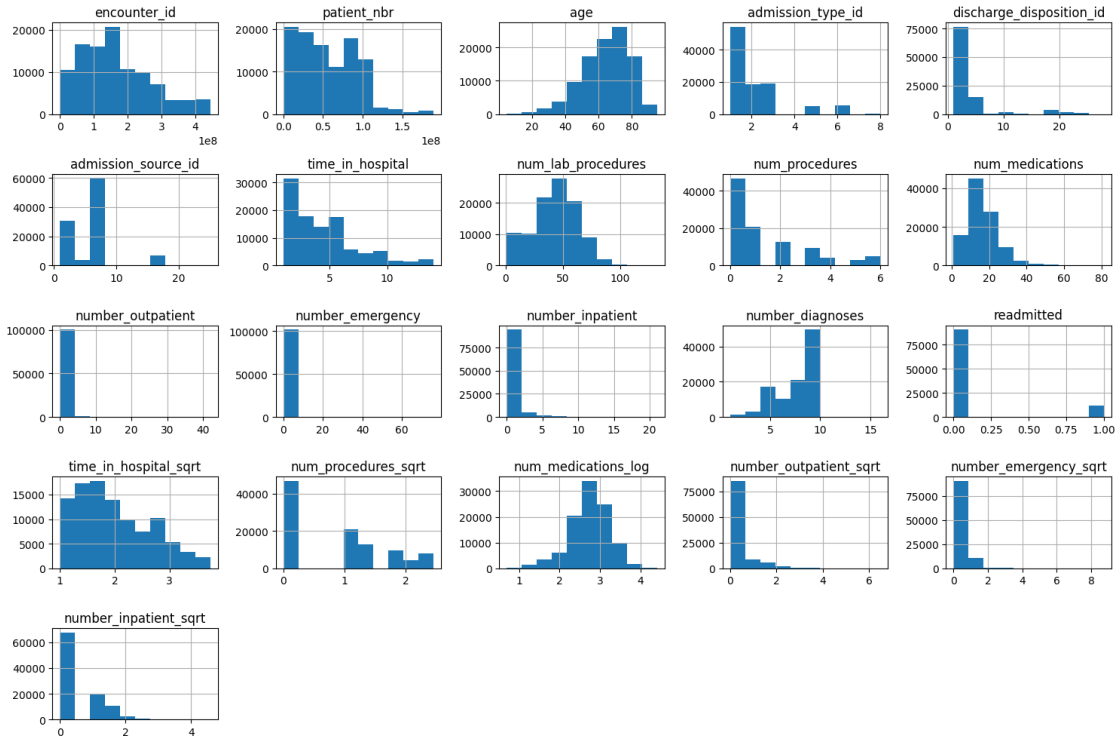
```
101763                1                  53  …                      No
101764               10                  45  …                      No
101765                6                  13  …                      No

        change  diabetesMed  readmitted  time_in_hospital_sqrt  \
0           No           No           0               1.000000
1           Ch          Yes           0               1.732051
2           No          Yes           0               1.414214
3           Ch          Yes           0               1.414214
4           Ch          Yes           0               1.000000
…        …           …          …                     …
101761      Ch          Yes           0               1.732051
101762      No          Yes           0               2.236068
101763      Ch          Yes           0               1.000000
101764      Ch          Yes           0               3.162278
101765      No           No           0               2.449490

       num_procedures_sqrt num_medications_log number_outpatient_sqrt  \
0                 0.000000            0.693147               0.000000
1                 0.000000            2.944439               0.000000
2                 2.236068            2.639057               1.414214
3                 1.000000            2.833213               0.000000
4                 0.000000            2.197225               0.000000
…                    …                 …                     …
101761            0.000000            2.833213               0.000000
101762            1.732051            2.944439               0.000000
101763            0.000000            2.302585               1.000000
101764            1.414214            3.091042               0.000000
101765            1.732051            1.386294               0.000000

        number_emergency_sqrt number_inpatient_sqrt
0                         0.0                   0.0
1                         0.0                   0.0
2                         0.0                   1.0
3                         0.0                   0.0
4                         0.0                   0.0
…                        …                     …
101761                    0.0                   0.0
101762                    0.0                   1.0
101763                    0.0                   0.0
101764                    0.0                   1.0
101765                    0.0                   0.0

[101766 rows x 53 columns]
```

25 Normalization

[27]:
```
from sklearn.preprocessing import MinMaxScaler
 #Check summary statistics
print(df.describe())

# Visualize distributions
df.hist(bins=10, figsize=(15, 10))
plt.tight_layout()
plt.show()

# Assuming we determine normalization is required based on the inspection
scaler = MinMaxScaler()


# Fit and transform the numerical columns
df[numerical_columns] = scaler.fit_transform(df[numerical_columns])

# Display the normalized dataframe
print(df)
```

```
       encounter_id   patient_nbr            age  admission_type_id  \
count  1.017660e+05  1.017660e+05  101766.000000      101766.000000
mean   1.652016e+08  5.433040e+07      65.967022           2.024006
std    1.026403e+08  3.869636e+07      15.940838           1.445403
min    1.252200e+04  1.350000e+02       5.000000           1.000000
25%    8.496119e+07  2.341322e+07      55.000000           1.000000
50%    1.523890e+08  4.550514e+07      65.000000           1.000000
75%    2.302709e+08  8.754595e+07      75.000000           3.000000
max    4.438672e+08  1.895026e+08      95.000000           8.000000

       discharge_disposition_id  admission_source_id  time_in_hospital  \
count             101766.000000        101766.000000     101766.000000
mean                   3.715642             5.754437          4.395987
std                    5.280166             4.064081          2.985108
min                    1.000000             1.000000          1.000000
25%                    1.000000             1.000000          2.000000
50%                    1.000000             7.000000          4.000000
75%                    4.000000             7.000000          6.000000
max                   28.000000            25.000000         14.000000

       num_lab_procedures  num_procedures  num_medications  …  \
count       101766.000000   101766.000000    101766.000000  …
mean            43.095641        1.339730        16.021844  …
std             19.674362        1.705807         8.127566  …
min              1.000000        0.000000         1.000000  …
25%             31.000000        0.000000        10.000000  …
50%             44.000000        1.000000        15.000000  …
```

```
75%             57.000000        2.000000        20.000000  …
max            132.000000        6.000000        81.000000  …

       number_emergency  number_inpatient  number_diagnoses     readmitted  \
count     101766.000000     101766.000000     101766.000000  101766.000000
mean           0.197836          0.635566          7.422607       0.111599
std            0.930472          1.262863          1.933600       0.314874
min            0.000000          0.000000          1.000000       0.000000
25%            0.000000          0.000000          6.000000       0.000000
50%            0.000000          0.000000          8.000000       0.000000
75%            0.000000          1.000000          9.000000       0.000000
max           76.000000         21.000000         16.000000       1.000000

       time_in_hospital_sqrt  num_procedures_sqrt  num_medications_log  \
count          101766.000000        101766.000000        101766.000000
mean                1.981832             0.810287             2.722373
std                 0.684350             0.826542             0.489285
min                 1.000000             0.000000             0.693147
25%                 1.414214             0.000000             2.397895
50%                 2.000000             1.000000             2.772589
75%                 2.449490             1.414214             3.044522
max                 3.741657             2.449490             4.406719

       number_outpatient_sqrt  number_emergency_sqrt  number_inpatient_sqrt
count           101766.000000          101766.000000          101766.000000
mean                 0.229104               0.138917               0.437246
std                  0.562914               0.422540               0.666623
min                  0.000000               0.000000               0.000000
25%                  0.000000               0.000000               0.000000
50%                  0.000000               0.000000               0.000000
75%                  0.000000               0.000000               1.000000
max                  6.480741               8.717798               4.582576

[8 rows x 21 columns]
```



	encounter_id	patient_nbr	race	gender	age	\
0	0.005105	0.043387	Caucasian	Female	0.000000	
1	0.000308	0.293553	Caucasian	Female	0.111111	
2	0.000117	0.454072	AfricanAmerican	Female	0.222222	
3	0.001099	0.435046	Caucasian	Male	0.333333	
4	0.000009	0.224372	Caucasian	Male	0.444444	
…	…	…	…	…	…	
101761	0.999956	0.528554	AfricanAmerican	Male	0.777778	
101762	0.999956	0.394159	AfricanAmerican	Female	0.888889	
101763	0.999971	0.216824	Caucasian	Male	0.777778	
101764	0.999977	0.167246	Caucasian	Female	0.888889	
101765	1.000000	0.925735	Caucasian	Male	0.777778	

	admission_type_id	discharge_disposition_id	admission_source_id	\
0	0.714286	0.888889	0.000000	
1	0.000000	0.000000	0.250000	
2	0.000000	0.000000	0.250000	
3	0.000000	0.000000	0.250000	
4	0.000000	0.000000	0.250000	
…	…	…	…	
101761	0.000000	0.074074	0.250000	
101762	0.000000	0.111111	0.166667	
101763	0.000000	0.000000	0.250000	
101764	0.142857	0.074074	0.250000	

```
101765           0.000000                  0.000000             0.250000

        time_in_hospital  num_lab_procedures  …  metformin-pioglitazone  \
0               0.000000            0.305344  …                      No
1               0.153846            0.442748  …                      No
2               0.076923            0.076336  …                      No
3               0.076923            0.328244  …                      No
4               0.000000            0.381679  …                      No
…                  …                   …  …                    …
101761          0.153846            0.381679  …                      No
101762          0.307692            0.244275  …                      No
101763          0.000000            0.396947  …                      No
101764          0.692308            0.335878  …                      No
101765          0.384615            0.091603  …                      No

        change  diabetesMed  readmitted  time_in_hospital_sqrt  \
0           No           No         0.0               1.000000
1           Ch          Yes         0.0               1.732051
2           No          Yes         0.0               1.414214
3           Ch          Yes         0.0               1.414214
4           Ch          Yes         0.0               1.000000
…        …           …          …                      …
101761      Ch          Yes         0.0               1.732051
101762      No          Yes         0.0               2.236068
101763      Ch          Yes         0.0               1.000000
101764      Ch          Yes         0.0               3.162278
101765      No           No         0.0               2.449490

       num_procedures_sqrt num_medications_log number_outpatient_sqrt  \
0                 0.000000            0.693147               0.000000
1                 0.000000            2.944439               0.000000
2                 2.236068            2.639057               1.414214
3                 1.000000            2.833213               0.000000
4                 0.000000            2.197225               0.000000
…                    …                   …                     …
101761            0.000000            2.833213               0.000000
101762            1.732051            2.944439               0.000000
101763            0.000000            2.302585               1.000000
101764            1.414214            3.091042               0.000000
101765            1.732051            1.386294               0.000000

        number_emergency_sqrt number_inpatient_sqrt
0                         0.0                  0.0
1                         0.0                  0.0
2                         0.0                  1.0
3                         0.0                  0.0
4                         0.0                  0.0
…                         …                    …
```

```
101761                    0.0                    0.0
101762                    0.0                    1.0
101763                    0.0                    0.0
101764                    0.0                    1.0
101765                    0.0                    0.0

[101766 rows x 53 columns]
```

26 Dropping few columns not necessary for analysis

[28]:
```
# Assuming df is your DataFrame
columns_to_drop = ['encounter_id', 'patient_nbr']

# Drop the specified columns
df.drop(columns=columns_to_drop, inplace=True)

# Verify the changes
print(df.head())
```

```
             race  gender       age  admission_type_id  \
0       Caucasian  Female  0.000000           0.714286
1       Caucasian  Female  0.111111           0.000000
2  AfricanAmerican  Female  0.222222           0.000000
3       Caucasian    Male  0.333333           0.000000
4       Caucasian    Male  0.444444           0.000000

   discharge_disposition_id  admission_source_id  time_in_hospital  \
0                  0.888889                 0.00          0.000000
1                  0.000000                 0.25          0.153846
2                  0.000000                 0.25          0.076923
3                  0.000000                 0.25          0.076923
4                  0.000000                 0.25          0.000000

   num_lab_procedures  num_procedures  num_medications  …  \
0            0.305344        0.000000           0.0000  …
1            0.442748        0.000000           0.2125  …
2            0.076336        0.833333           0.1500  …
3            0.328244        0.166667           0.1875  …
4            0.381679        0.000000           0.0875  …

   metformin-pioglitazone  change  diabetesMed readmitted  \
0                      No      No           No        0.0
1                      No      Ch          Yes        0.0
2                      No      No          Yes        0.0
3                      No      Ch          Yes        0.0
4                      No      Ch          Yes        0.0
```

```
  time_in_hospital_sqrt num_procedures_sqrt  num_medications_log  \
0              1.000000            0.000000             0.693147
1              1.732051            0.000000             2.944439
2              1.414214            2.236068             2.639057
3              1.414214            1.000000             2.833213
4              1.000000            0.000000             2.197225

  number_outpatient_sqrt number_emergency_sqrt number_inpatient_sqrt
0               0.000000                   0.0                   0.0
1               0.000000                   0.0                   0.0
2               1.414214                   0.0                   1.0
3               0.000000                   0.0                   0.0
4               0.000000                   0.0                   0.0

[5 rows x 51 columns]
```

27 Creating New Features

[29]:
```
# Calculate Health_index
df['Health_index'] = 1 / (df['number_emergency'] + df['number_inpatient'] +␣
 ↪df['number_outpatient'])

# If there are cases where the denominator is zero (resulting in division by␣
 ↪zero), replace with NaN
df['Health_index'].replace({0: pd.NA}, inplace=True)

# Calculate severity_of_disease
df['severity_of_disease'] = df['time_in_hospital'] + df['num_procedures'] +␣
 ↪df['num_medications'] + df['num_lab_procedures'] + df['number_diagnoses']


# Print the DataFrame to verify the changes
print(df.head())
```

```
             race  gender       age  admission_type_id  \
0       Caucasian  Female  0.000000           0.714286
1       Caucasian  Female  0.111111           0.000000
2  AfricanAmerican  Female  0.222222           0.000000
3       Caucasian    Male  0.333333           0.000000
4       Caucasian    Male  0.444444           0.000000

   discharge_disposition_id  admission_source_id  time_in_hospital  \
0                  0.888889                 0.00          0.000000
1                  0.000000                 0.25          0.153846
2                  0.000000                 0.25          0.076923
3                  0.000000                 0.25          0.076923
4                  0.000000                 0.25          0.000000
```

```
   num_lab_procedures  num_procedures  num_medications  …  diabetesMed  \
0            0.305344        0.000000           0.0000  …           No
1            0.442748        0.000000           0.2125  …          Yes
2            0.076336        0.833333           0.1500  …          Yes
3            0.328244        0.166667           0.1875  …          Yes
4            0.381679        0.000000           0.0875  …          Yes

   readmitted  time_in_hospital_sqrt num_procedures_sqrt num_medications_log  \
0         0.0               1.000000            0.000000             0.693147
1         0.0               1.732051            0.000000             2.944439
2         0.0               1.414214            2.236068             2.639057
3         0.0               1.414214            1.000000             2.833213
4         0.0               1.000000            0.000000             2.197225

  number_outpatient_sqrt  number_emergency_sqrt number_inpatient_sqrt  \
0               0.000000                    0.0                   0.0
1               0.000000                    0.0                   0.0
2               1.414214                    0.0                   1.0
3               0.000000                    0.0                   0.0
4               0.000000                    0.0                   0.0

  Health_index severity_of_disease
0          inf            0.305344
1          inf            1.342428
2         10.5            1.469926
3          inf            1.159334
4          inf            0.735846

[5 rows x 53 columns]
```

[30]:
```
from tqdm import tqdm

drugList =['metformin', 'repaglinide', 'nateglinide',
       'chlorpropamide', 'glimepiride', 'acetohexamide', 'glipizide',
       'glyburide', 'tolbutamide', 'pioglitazone', 'rosiglitazone', 'acarbose',
       'miglitol', 'troglitazone', 'tolazamide', 'examide', 'citoglipton',
       'insulin', 'glyburide-metformin', 'glipizide-metformin',
       'glimepiride-pioglitazone', 'metformin-rosiglitazone',
       'metformin-pioglitazone']

number_of_changes = []
for i in tqdm(range(len(df))):
    changeCount = 0
    for col in drugList:
        if df.iloc[i][col] in ['Down', 'Up']:
            changeCount += 1
```

```
    number_of_changes.append(changeCount)

# Add the 'number_of_changes' column to the DataFrame
df['number_of_changes'] = number_of_changes

# Print the DataFrame to verify the changes
print(df.head())
```

```
100%|      | 101766/101766 [04:06<00:00, 413.56it/s]
```

	race	gender	age	admission_type_id	\
0	Caucasian	Female	0.000000	0.714286	
1	Caucasian	Female	0.111111	0.000000	
2	AfricanAmerican	Female	0.222222	0.000000	
3	Caucasian	Male	0.333333	0.000000	
4	Caucasian	Male	0.444444	0.000000	

	discharge_disposition_id	admission_source_id	time_in_hospital	\
0	0.888889	0.00	0.000000	
1	0.000000	0.25	0.153846	
2	0.000000	0.25	0.076923	
3	0.000000	0.25	0.076923	
4	0.000000	0.25	0.000000	

	num_lab_procedures	num_procedures	num_medications	…	readmitted	\
0	0.305344	0.000000	0.0000	…	0.0	
1	0.442748	0.000000	0.2125	…	0.0	
2	0.076336	0.833333	0.1500	…	0.0	
3	0.328244	0.166667	0.1875	…	0.0	
4	0.381679	0.000000	0.0875	…	0.0	

	time_in_hospital_sqrt	num_procedures_sqrt	num_medications_log	\
0	1.000000	0.000000	0.693147	
1	1.732051	0.000000	2.944439	
2	1.414214	2.236068	2.639057	
3	1.414214	1.000000	2.833213	
4	1.000000	0.000000	2.197225	

	number_outpatient_sqrt	number_emergency_sqrt	number_inpatient_sqrt	\
0	0.000000	0.0	0.0	
1	0.000000	0.0	0.0	
2	1.414214	0.0	1.0	
3	0.000000	0.0	0.0	
4	0.000000	0.0	0.0	

	Health_index	severity_of_disease	number_of_changes
0	inf	0.305344	0
1	inf	1.342428	1

```
2         10.5            1.469926                0
3          inf            1.159334                1
4          inf            0.735846                0

[5 rows x 54 columns]
```

[31]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 54 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   race                      101766 non-null  object
 1   gender                    101766 non-null  object
 2   age                       101766 non-null  float64
 3   admission_type_id         101766 non-null  float64
 4   discharge_disposition_id  101766 non-null  float64
 5   admission_source_id       101766 non-null  float64
 6   time_in_hospital          101766 non-null  float64
 7   num_lab_procedures        101766 non-null  float64
 8   num_procedures            101766 non-null  float64
 9   num_medications           101766 non-null  float64
 10  number_outpatient         101766 non-null  float64
 11  number_emergency          101766 non-null  float64
 12  number_inpatient          101766 non-null  float64
 13  diag_1                    101766 non-null  object
 14  diag_2                    101766 non-null  object
 15  diag_3                    101766 non-null  object
 16  number_diagnoses          101766 non-null  float64
 17  max_glu_serum             5346 non-null    object
 18  A1Cresult                 17018 non-null   object
 19  metformin                 101766 non-null  object
 20  repaglinide               101766 non-null  object
 21  nateglinide               101766 non-null  object
 22  chlorpropamide            101766 non-null  object
 23  glimepiride               101766 non-null  object
 24  acetohexamide             101766 non-null  object
 25  glipizide                 101766 non-null  object
 26  glyburide                 101766 non-null  object
 27  tolbutamide               101766 non-null  object
 28  pioglitazone              101766 non-null  object
 29  rosiglitazone             101766 non-null  object
 30  acarbose                  101766 non-null  object
 31  miglitol                  101766 non-null  object
 32  troglitazone              101766 non-null  object
```

```
 33  tolazamide                101766 non-null  object
 34  examide                   101766 non-null  object
 35  citoglipton               101766 non-null  object
 36  insulin                   101766 non-null  object
 37  glyburide-metformin       101766 non-null  object
 38  glipizide-metformin       101766 non-null  object
 39  glimepiride-pioglitazone  101766 non-null  object
 40  metformin-rosiglitazone   101766 non-null  object
 41  metformin-pioglitazone    101766 non-null  object
 42  change                    101766 non-null  object
 43  diabetesMed               101766 non-null  object
 44  readmitted                101766 non-null  float64
 45  time_in_hospital_sqrt     101766 non-null  float64
 46  num_procedures_sqrt       101766 non-null  float64
 47  num_medications_log       101766 non-null  float64
 48  number_outpatient_sqrt    101766 non-null  float64
 49  number_emergency_sqrt     101766 non-null  float64
 50  number_inpatient_sqrt     101766 non-null  float64
 51  Health_index              101766 non-null  float64
 52  severity_of_disease       101766 non-null  float64
 53  number_of_changes         101766 non-null  int64
dtypes: float64(21), int64(1), object(32)
memory usage: 41.9+ MB
```

28 Dropping all the medication columns after creating new features

[32]:
```
# Range of column indices to drop
col_indices_to_drop = range(19, 42)

# Drop the columns by index range
df = df.drop(df.columns[col_indices_to_drop], axis=1)
print(df.info())
#df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 31 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   race                      101766 non-null  object
 1   gender                    101766 non-null  object
 2   age                       101766 non-null  float64
 3   admission_type_id         101766 non-null  float64
 4   discharge_disposition_id  101766 non-null  float64
 5   admission_source_id       101766 non-null  float64
 6   time_in_hospital          101766 non-null  float64
```

```
 7   num_lab_procedures        101766 non-null  float64
 8   num_procedures            101766 non-null  float64
 9   num_medications           101766 non-null  float64
 10  number_outpatient         101766 non-null  float64
 11  number_emergency          101766 non-null  float64
 12  number_inpatient          101766 non-null  float64
 13  diag_1                    101766 non-null  object
 14  diag_2                    101766 non-null  object
 15  diag_3                    101766 non-null  object
 16  number_diagnoses          101766 non-null  float64
 17  max_glu_serum             5346 non-null    object
 18  A1Cresult                 17018 non-null   object
 19  change                    101766 non-null  object
 20  diabetesMed               101766 non-null  object
 21  readmitted                101766 non-null  float64
 22  time_in_hospital_sqrt     101766 non-null  float64
 23  num_procedures_sqrt       101766 non-null  float64
 24  num_medications_log       101766 non-null  float64
 25  number_outpatient_sqrt    101766 non-null  float64
 26  number_emergency_sqrt     101766 non-null  float64
 27  number_inpatient_sqrt     101766 non-null  float64
 28  Health_index              101766 non-null  float64
 29  severity_of_disease       101766 non-null  float64
 30  number_of_changes         101766 non-null  int64
dtypes: float64(21), int64(1), object(9)
memory usage: 24.1+ MB
None
```

[33]:
```
# Replace infinite values with NaN
df['Health_index'].replace([np.inf, -np.inf], np.nan, inplace=True)

# Option 1: Fill NaN values with a specific value (e.g., the mean of the column)
df['Health_index'].fillna(df['Health_index'].mean(), inplace=True)
# Convert 'Health_index' column to integer
df['Health_index'] = df['Health_index'].astype(int)
print(df.head())
```

```
             race  gender       age  admission_type_id  \
0       Caucasian  Female  0.000000           0.714286
1       Caucasian  Female  0.111111           0.000000
2  AfricanAmerican  Female  0.222222           0.000000
3       Caucasian    Male  0.333333           0.000000
4       Caucasian    Male  0.444444           0.000000

   discharge_disposition_id  admission_source_id  time_in_hospital  \
0                  0.888889                 0.00          0.000000
1                  0.000000                 0.25          0.153846
2                  0.000000                 0.25          0.076923
```

```
3                  0.000000                  0.25          0.076923
4                  0.000000                  0.25          0.000000

   num_lab_procedures  num_procedures  num_medications  …  readmitted  \
0            0.305344        0.000000           0.0000  …         0.0
1            0.442748        0.000000           0.2125  …         0.0
2            0.076336        0.833333           0.1500  …         0.0
3            0.328244        0.166667           0.1875  …         0.0
4            0.381679        0.000000           0.0875  …         0.0

   time_in_hospital_sqrt  num_procedures_sqrt num_medications_log  \
0               1.000000             0.000000            0.693147
1               1.732051             0.000000            2.944439
2               1.414214             2.236068            2.639057
3               1.414214             1.000000            2.833213
4               1.000000             0.000000            2.197225

  number_outpatient_sqrt number_emergency_sqrt  number_inpatient_sqrt  \
0               0.000000                   0.0                    0.0
1               0.000000                   0.0                    0.0
2               1.414214                   0.0                    1.0
3               0.000000                   0.0                    0.0
4               0.000000                   0.0                    0.0

  Health_index severity_of_disease number_of_changes
0           19            0.305344                 0
1           19            1.342428                 1
2           10            1.469926                 0
3           19            1.159334                 1
4           19            0.735846                 0

[5 rows x 31 columns]
```

[34]:
```python
def find_and_remove_redundant_columns(df):
    redundant_columns = set()
    suffixes = ['_log', '_sqrt']

    # Identify original columns that have corresponding transformed columns
    for suffix in suffixes:
        for col in df.columns:
            if col.endswith(suffix):
                base_col = col.replace(suffix, '')
                if base_col in df.columns:
                    redundant_columns.add(base_col)

    # Create a new DataFrame excluding the redundant columns
    df_cleaned = df.drop(columns=redundant_columns)
```

```
    return df_cleaned, list(redundant_columns)

# Example usage
df_cleaned, redundant_columns = find_and_remove_redundant_columns(df)
df = df_cleaned.copy()
print("Redundant columns:", redundant_columns)
```

```
Redundant columns: ['num_medications', 'number_outpatient', 'time_in_hospital',
'number_inpatient', 'num_procedures', 'number_emergency']
```

[35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 25 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   race                      101766 non-null  object
 1   gender                    101766 non-null  object
 2   age                       101766 non-null  float64
 3   admission_type_id         101766 non-null  float64
 4   discharge_disposition_id  101766 non-null  float64
 5   admission_source_id       101766 non-null  float64
 6   num_lab_procedures        101766 non-null  float64
 7   diag_1                    101766 non-null  object
 8   diag_2                    101766 non-null  object
 9   diag_3                    101766 non-null  object
 10  number_diagnoses          101766 non-null  float64
 11  max_glu_serum             5346 non-null    object
 12  A1Cresult                 17018 non-null   object
 13  change                    101766 non-null  object
 14  diabetesMed               101766 non-null  object
 15  readmitted                101766 non-null  float64
 16  time_in_hospital_sqrt     101766 non-null  float64
 17  num_procedures_sqrt       101766 non-null  float64
 18  num_medications_log       101766 non-null  float64
 19  number_outpatient_sqrt    101766 non-null  float64
 20  number_emergency_sqrt     101766 non-null  float64
 21  number_inpatient_sqrt     101766 non-null  float64
 22  Health_index              101766 non-null  int64
 23  severity_of_disease       101766 non-null  float64
 24  number_of_changes         101766 non-null  int64
dtypes: float64(14), int64(2), object(9)
memory usage: 19.4+ MB
```

29 Separating features and label

[36]:
```
# Get the list of all columns in the DataFrame
feature_set = list(df.columns)
print("Feature set:", feature_set)

# Define the target variable y
y = df['readmitted']

# Define the feature set X by dropping the 'readmitted' column
X = df.drop(columns='readmitted')

# Print the shapes of y and X to confirm
print("y shape:", y.shape)
print("X shape:", X.shape)
```

```
Feature set: ['race', 'gender', 'age', 'admission_type_id',
'discharge_disposition_id', 'admission_source_id', 'num_lab_procedures',
'diag_1', 'diag_2', 'diag_3', 'number_diagnoses', 'max_glu_serum', 'A1Cresult',
'change', 'diabetesMed', 'readmitted', 'time_in_hospital_sqrt',
'num_procedures_sqrt', 'num_medications_log', 'number_outpatient_sqrt',
'number_emergency_sqrt', 'number_inpatient_sqrt', 'Health_index',
'severity_of_disease', 'number_of_changes']
y shape: (101766,)
X shape: (101766, 24)
```

[37]:
```
# Step 1: Identify Non-Numerical Columns
non_numerical_columns = df.select_dtypes(include=['object', 'string',␣
 ↪'category']).columns

# Step 2: Perform One-Hot Encoding
df_encoded = pd.get_dummies(df, columns=non_numerical_columns, drop_first=True)

# Display the DataFrame with one-hot encoded features
print(df_encoded.head())
```

```
        age  admission_type_id  discharge_disposition_id  admission_source_id  \
0  0.000000           0.714286                  0.888889                 0.00
1  0.111111           0.000000                  0.000000                 0.25
2  0.222222           0.000000                  0.000000                 0.25
3  0.333333           0.000000                  0.000000                 0.25
4  0.444444           0.000000                  0.000000                 0.25

   num_lab_procedures  number_diagnoses  readmitted  time_in_hospital_sqrt  \
0            0.305344          0.000000         0.0               1.000000
1            0.442748          0.533333         0.0               1.732051
2            0.076336          0.333333         0.0               1.414214
3            0.328244          0.400000         0.0               1.414214
```

```
4            0.381679          0.266667         0.0                1.000000

   num_procedures_sqrt  num_medications_log  …  diag_3_V70  diag_3_V72  \
0             0.000000             0.693147  …       False       False
1             0.000000             2.944439  …       False       False
2             2.236068             2.639057  …       False       False
3             1.000000             2.833213  …       False       False
4             0.000000             2.197225  …       False       False

   diag_3_V85  diag_3_V86  max_glu_serum_>300  max_glu_serum_Norm  \
0       False       False               False               False
1       False       False               False               False
2       False       False               False               False
3       False       False               False               False
4       False       False               False               False

   A1Cresult_>8  A1Cresult_Norm  change_No  diabetesMed_Yes
0         False           False       True            False
1         False           False      False             True
2         False           False       True             True
3         False           False      False             True
4         False           False      False             True

[5 rows x 2278 columns]
```

30 Logistic Regression Model

[38]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score, f1_score, recall_score, precision_score, roc_auc_score,␣
 ↪roc_curve
from imblearn.over_sampling import SMOTE

# Step 1: Separate features and target variable
X = df_encoded.drop('readmitted', axis=1)  # Features
y = df_encoded['readmitted']  # Target

# Step 2: Perform SMOTE to handle class imbalance
smote = SMOTE(random_state=42)
X_smote, y_smote = smote.fit_resample(X, y)

# Step 3: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_smote, y_smote,␣
 ↪test_size=0.3, random_state=42)
```

```
# Step 4: Train a logistic regression model
log_reg = LogisticRegression(max_iter=500, random_state=42)
log_reg.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[38]:
```
LogisticRegression(max_iter=500, random_state=42)
```

[39]:
```
# Step 5: Make predictions
y_pred = log_reg.predict(X_test)
y_pred_prob_lr = log_reg.predict_proba(X_test)[:, 1]

# Step 6: Evaluate the model
# Confusion Matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Classification Report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Confusion Matrix:
[[26916   120]
 [ 4070 23140]]

Classification Report:
              precision    recall  f1-score   support

         0.0       0.87      1.00      0.93     27036
         1.0       0.99      0.85      0.92     27210

    accuracy                           0.92     54246
   macro avg       0.93      0.92      0.92     54246
weighted avg       0.93      0.92      0.92     54246
```

[40]:
```
# Accuracy Score
accuracy_lr = accuracy_score(y_test, y_pred)
```

```
print("\nAccuracy Score:", accuracy_lr)

# Precision, Recall, F1 Score
precision_lr = precision_score(y_test, y_pred)
recall_lr = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
print("\nPrecision:", precision_lr)
print("Recall:", recall_lr)
print("F1 Score:", f1)
```

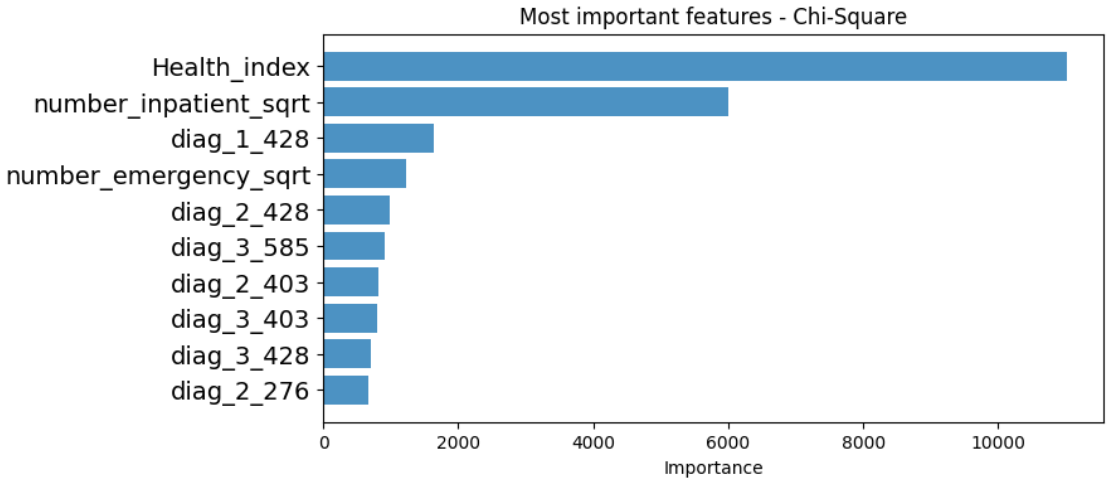
```
Accuracy Score: 0.9227592817903624

Precision: 0.9948409286328461
Recall: 0.850422638735759
F1 Score: 0.9169803843867644
```

31 #Finding Top 10 features using Chi-Square Test

[41]:
```
from sklearn.feature_selection import SelectKBest, chi2

# Assuming you have your X_train and y_train defined

# Apply SelectKBest with chi2 to get the top features
k = 10  # Number of top features to select
selector = SelectKBest(score_func=chi2, k=k)
selector.fit(X_train, y_train)

# Get the feature names and their scores
feature_names = X_train.columns
feature_scores = selector.scores_

# Create DataFrame with feature names and their importance
most_imp_features = pd.DataFrame([f for f in zip(feature_names,␣
 ↪feature_scores)], columns=["Feature", "Importance"])

# Get top k most important features
most_imp_features = most_imp_features.nlargest(k, "Importance")

# Sort values by Importance
most_imp_features.sort_values(by="Importance", inplace=True)
print(most_imp_features)
# Plotting
plt.figure(figsize=(8, 4))
plt.barh(range(len(most_imp_features)), most_imp_features.Importance,␣
 ↪align='center', alpha=0.8)
```

```
plt.yticks(range(len(most_imp_features)), most_imp_features.Feature,␣
  ↪fontsize=14)
plt.xlabel('Importance')
plt.title('Most important features - Chi-Square')
plt.show()
```

	Feature	Importance
868	diag_2_276	667.306057
1750	diag_3_428	710.520090
1730	diag_3_403	797.042901
976	diag_2_403	817.670095
1877	diag_3_585	909.910168
996	diag_2_428	992.151468
10	number_emergency_sqrt	1227.128904
296	diag_1_428	1630.244679
11	number_inpatient_sqrt	5997.817235
12	Health_index	11009.335733



32 Decision Tree Classifier

[42]:
```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=28, criterion = "entropy",␣
  ↪min_samples_split=10)
dtree.fit(X_train, y_train)
```

[42]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=28, min_samples_split=10)
```

[43]:
```
dtree_pred = dtree.predict(X_test)
y_pred_prob_dtree = dtree.predict_proba(X_test)[:, 1]
```

```
pd.crosstab(pd.Series(y_test, name = 'Actual'), pd.Series(dtree_pred, name =␣
 ↪'Predict'), margins = True)
```

[43]:
Predict	0.0	1.0	All
Actual			
0.0	7658	6616	14274
1.0	993	882	1875
All	8651	7498	16149

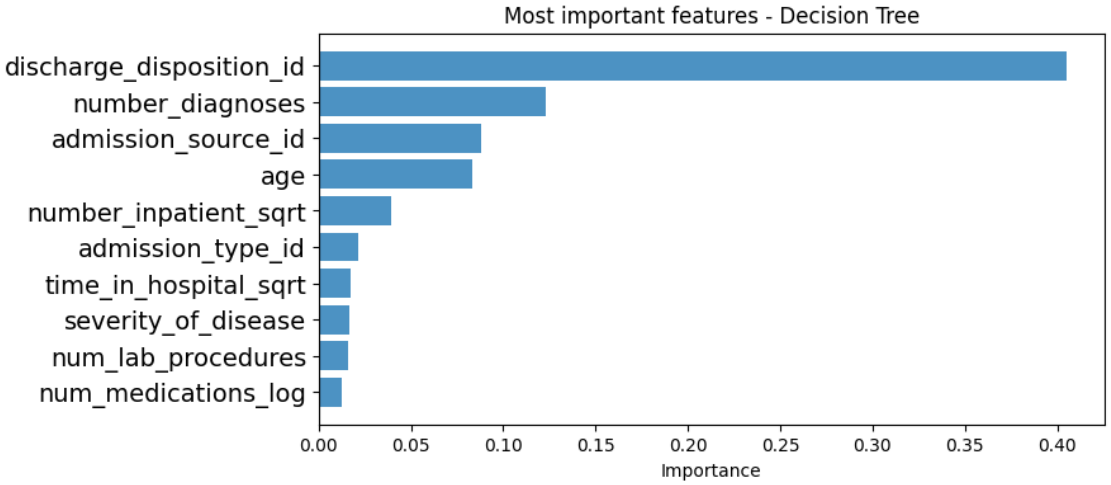
[44]:
```
print("Accuracy is {0:.2f}".format(accuracy_score(y_test, dtree_pred)))
print("Precision is {0:.2f}".format(precision_score(y_test, dtree_pred)))
print("Recall is {0:.2f}".format(recall_score(y_test, dtree_pred)))

accuracy_dtree = accuracy_score(y_test, dtree_pred)
precision_dtree = precision_score(y_test, dtree_pred)
recall_dtree = recall_score(y_test, dtree_pred)
```

```
Accuracy is 0.90
Precision is 0.94
Recall is 0.87
```

[45]:
```
# Create list of top most features based on importance
feature_names = X_train.columns
feature_imports = dtree.feature_importances_
most_imp_features = pd.DataFrame([f for f in␣
 ↪zip(feature_names,feature_imports)], columns=["Feature", "Importance"]).
 ↪nlargest(10, "Importance")
most_imp_features.sort_values(by="Importance", inplace=True)
print(most_imp_features)
plt.figure(figsize=(8,4))
plt.barh(range(len(most_imp_features)), most_imp_features.Importance,␣
 ↪align='center', alpha=0.8)
plt.yticks(range(len(most_imp_features)), most_imp_features.Feature,␣
 ↪fontsize=14)
plt.xlabel('Importance')
plt.title('Most important features - Decision Tree')
plt.show()
```

	Feature	Importance
8	num_medications_log	0.012717
4	num_lab_procedures	0.015848
13	severity_of_disease	0.016494
6	time_in_hospital_sqrt	0.017113
1	admission_type_id	0.021262
11	number_inpatient_sqrt	0.039384
0	age	0.083570
3	admission_source_id	0.087793

```
5            number_diagnoses    0.123105
2   discharge_disposition_id     0.405217
```



33 Random Forest

[46]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 10, max_depth=25, criterion =␣
 ↪"gini", min_samples_split=10)
rf.fit(X_train, y_train)
```

[46]:
```
RandomForestClassifier(max_depth=25, min_samples_split=10, n_estimators=10)
```

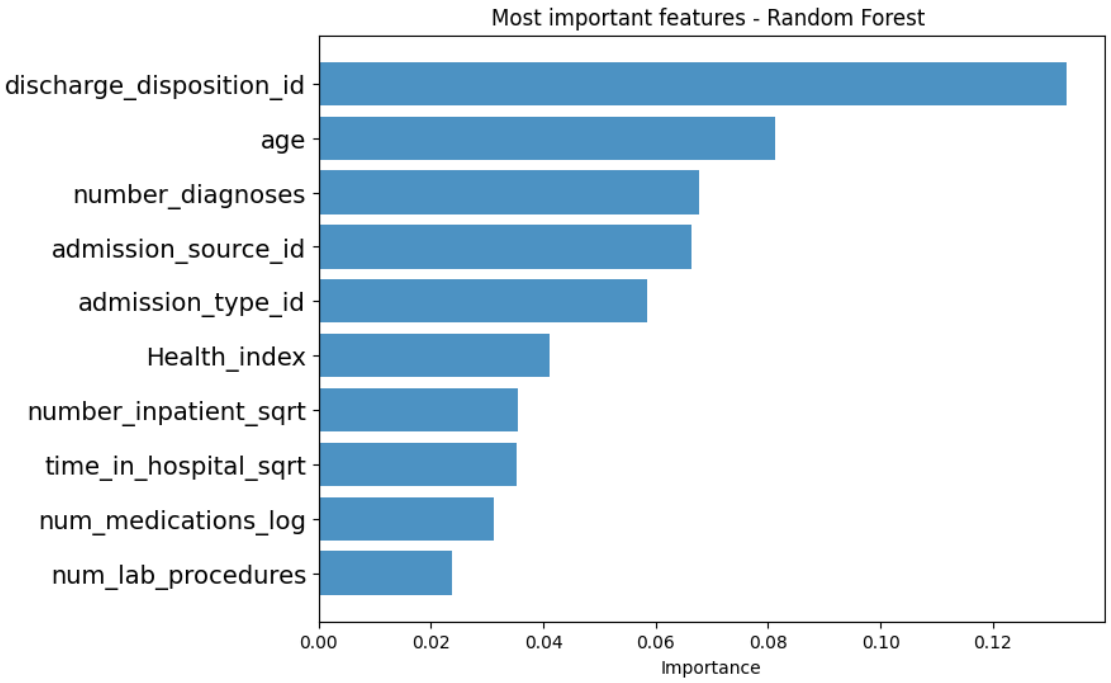
[47]:
```
rf_prd = rf.predict(X_test)
y_pred_prob_rf = log_reg.predict_proba(X_test)[:, 1]
pd.crosstab(pd.Series(y_test, name = 'Actual'), pd.Series(rf_prd, name =␣
 ↪'Predict'), margins = True)
```

[47]:
```
Predict   0.0   1.0    All
Actual
0.0      7413  6861  14274
1.0       959   916   1875
All      8372  7777  16149
```

[48]:
```
print("Accuracy is {0:.2f}".format(accuracy_score(y_test, rf_prd)))
print("Precision is {0:.2f}".format(precision_score(y_test, rf_prd)))
print("Recall is {0:.2f}".format(recall_score(y_test, rf_prd)))

accuracy_rf = accuracy_score(y_test, rf_prd)
precision_rf = precision_score(y_test, rf_prd)
recall_rf = recall_score(y_test, rf_prd)
```

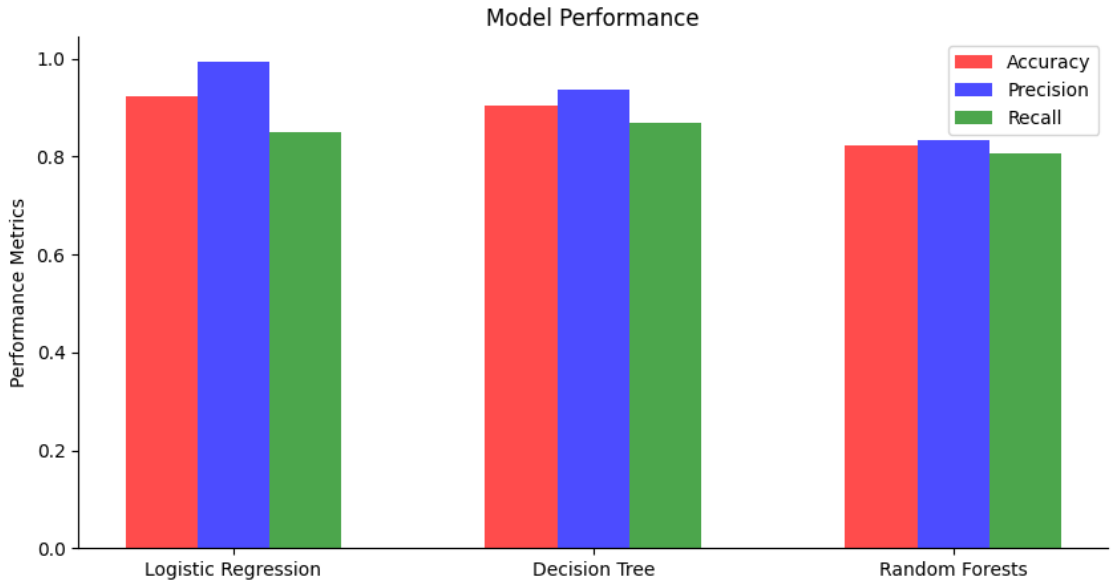
```
Accuracy is 0.82
Precision is 0.83
Recall is 0.81
```

[49]:
```
# Create list of top most features based on importance
feature_names = X_train.columns
feature_imports = rf.feature_importances_
most_imp_features = pd.DataFrame([f for f in␣
 ↪zip(feature_names,feature_imports)], columns=["Feature", "Importance"]).
 ↪nlargest(10, "Importance")
most_imp_features.sort_values(by="Importance", inplace=True)
plt.figure(figsize=(8,6))
plt.barh(range(len(most_imp_features)), most_imp_features.Importance,␣
 ↪align='center', alpha=0.8)
plt.yticks(range(len(most_imp_features)), most_imp_features.Feature,␣
 ↪fontsize=14)
plt.xlabel('Importance')
plt.title('Most important features - Random Forest ')
plt.show()
```



34 Model Comparision

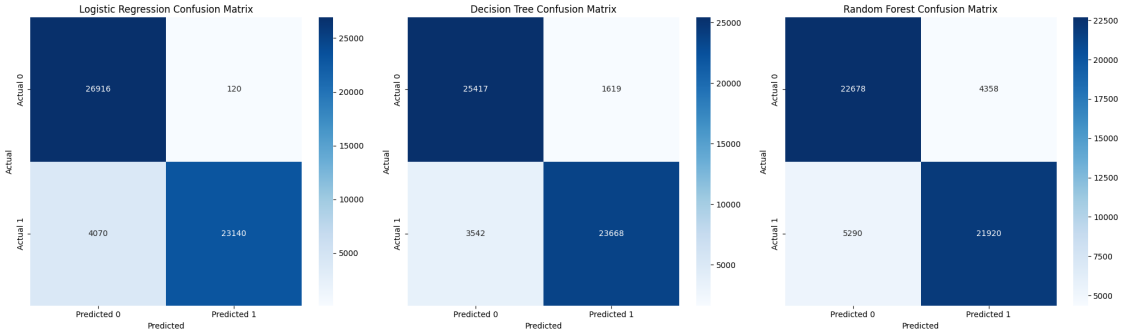
[50]:
```
import matplotlib.pyplot as plt
import numpy as np

plt.figure(figsize=(10, 5))

models = ['Logistic Regression', 'Decision Tree', 'Random Forests']

# Accuracy
accuracy = [accuracy_lr, accuracy_dtree, accuracy_rf]
plt.bar(np.arange(len(models)), accuracy, align='center', width=0.2, alpha=0.7,␣
 ↪color='red', label='Accuracy')

# Precision
precision = [precision_lr, precision_dtree, precision_rf]
plt.bar(np.arange(len(models)) + 0.2, precision, align='center', width=0.2,␣
 ↪alpha=0.7, color='blue', label='Precision')

# Recall
recall = [recall_lr, recall_dtree, recall_rf]
plt.bar(np.arange(len(models)) + 0.4, recall, align='center', width=0.2,␣
 ↪alpha=0.7, color='green', label='Recall')

plt.xticks(np.arange(len(models)) + 0.2, models)
plt.ylabel('Performance Metrics')
plt.title('Model Performance')
plt.legend()

# Removing the axis on the top and right of the plot window
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['top'].set_visible(False)

plt.show()
```



35 Confusion matrix for the models

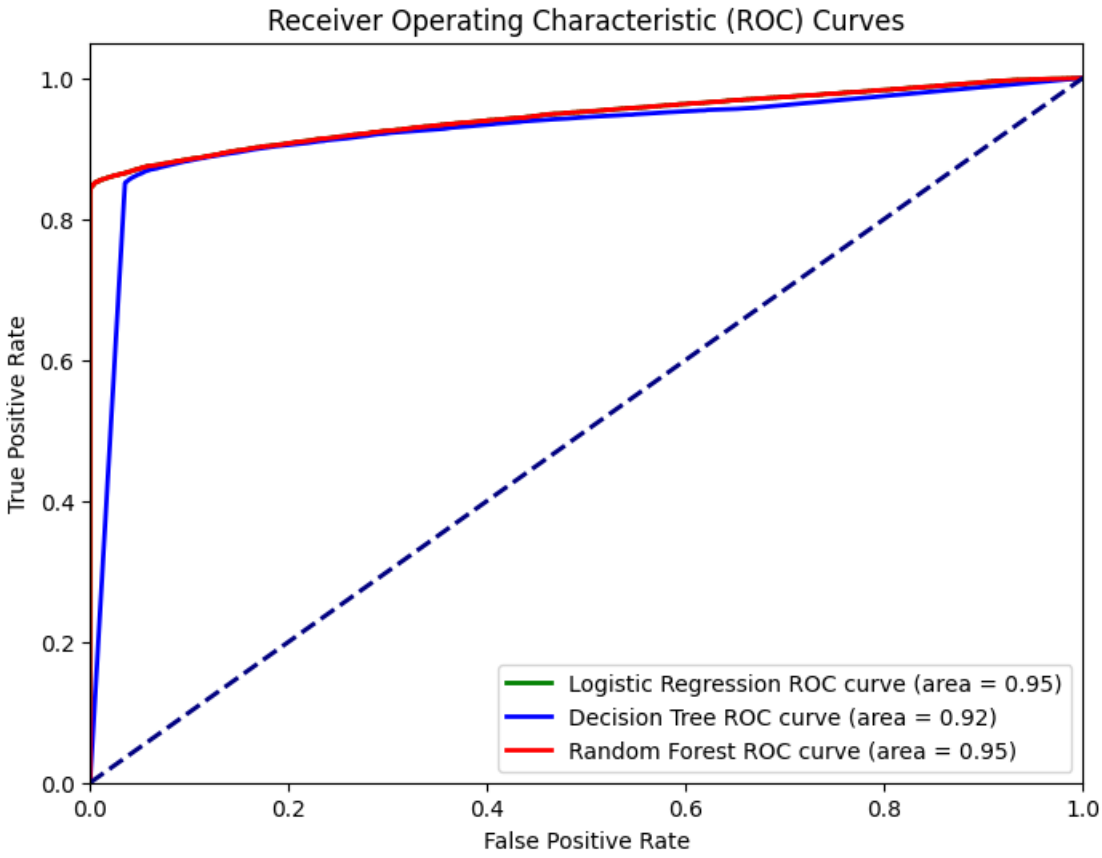
[51]:
```
# Generate confusion matrices
conf_matrix_lr = confusion_matrix(y_test, y_pred)
conf_matrix_dtree = confusion_matrix(y_test, dtree_pred )
conf_matrix_rf = confusion_matrix(y_test, rf_prd)

# Convert confusion matrices to DataFrames for better visualization
conf_matrix_lr_df = pd.DataFrame(conf_matrix_lr, index=[f'Actual {i}' for i in␣
 ↪range(len(conf_matrix_lr))], columns=[f'Predicted {i}' for i in␣
 ↪range(len(conf_matrix_lr[0]))])
conf_matrix_dtree_df = pd.DataFrame(conf_matrix_dtree, index=[f'Actual {i}' for␣
 ↪i in range(len(conf_matrix_dtree))], columns=[f'Predicted {i}' for i in␣
 ↪range(len(conf_matrix_dtree[0]))])
conf_matrix_rf_df = pd.DataFrame(conf_matrix_rf, index=[f'Actual {i}' for i in␣
 ↪range(len(conf_matrix_rf))], columns=[f'Predicted {i}' for i in␣
 ↪range(len(conf_matrix_rf[0]))])

# Plot confusion matrices side by side
fig, axes = plt.subplots(1, 3, figsize=(20, 6))

sns.heatmap(conf_matrix_lr_df, annot=True, fmt='d', cmap='Blues', ax=axes[0])
axes[0].set_title('Logistic Regression Confusion Matrix')
axes[0].set_ylabel('Actual')
axes[0].set_xlabel('Predicted')
```

```
sns.heatmap(conf_matrix_dtree_df, annot=True, fmt='d', cmap='Blues', ax=axes[1])
axes[1].set_title('Decision Tree Confusion Matrix')
axes[1].set_ylabel('Actual')
axes[1].set_xlabel('Predicted')

sns.heatmap(conf_matrix_rf_df, annot=True, fmt='d', cmap='Blues', ax=axes[2])
axes[2].set_title('Random Forest Confusion Matrix')
axes[2].set_ylabel('Actual')
axes[2].set_xlabel('Predicted')

plt.tight_layout()
plt.show()
```



[52]:
```
from sklearn.metrics import roc_curve, auc

# Calculate ROC curve and AUC for Logistic Regression
fpr_lr, tpr_lr, _ = roc_curve(y_test, y_pred_prob_lr)
roc_auc_lr = auc(fpr_lr, tpr_lr)

# Calculate ROC curve and AUC for Decision Tree
fpr_dtree, tpr_dtree, _ = roc_curve(y_test, y_pred_prob_dtree)
roc_auc_dtree = auc(fpr_dtree, tpr_dtree)

# Calculate ROC curve and AUC for Random Forest
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_pred_prob_rf)
roc_auc_rf = auc(fpr_rf, tpr_rf)

# Plotting the ROC curves
plt.figure(figsize=(8, 6))

plt.plot(fpr_lr, tpr_lr, color='green', lw=2, label='Logistic Regression ROC␣
 ↪curve (area = %0.2f)' % roc_auc_lr)
plt.plot(fpr_dtree, tpr_dtree, color='blue', lw=2, label='Decision Tree ROC␣
 ↪curve (area = %0.2f)' % roc_auc_dtree)
```

```
plt.plot(fpr_rf, tpr_rf, color='red', lw=2, label='Random Forest ROC curve␣
↪(area = %0.2f)' % roc_auc_rf)

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curves')
plt.legend(loc="lower right")
plt.show()
```



[53]:
```
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,␣
↪recall_score, f1_score
import pandas as pd

# Define lists to store metric values
models = ['Logistic Regression', 'Decision Tree', 'Random Forest']
accuracies = []
```

```
precisions = []
recalls = []
f1_scores = []

# Calculate metrics for each model
for model, pred in zip(models, [y_pred, dtree_pred, rf_prd]):
    accuracies.append(accuracy_score(y_test, pred))
    precisions.append(precision_score(y_test, pred))
    recalls.append(recall_score(y_test, pred))
    f1_scores.append(f1_score(y_test, pred))

# Create a DataFrame to display metrics
metrics_df = pd.DataFrame({
    'Model': models,
    'Accuracy': accuracies,
    'Precision': precisions,
    'Recall': recalls,
    'F1 Score': f1_scores
})

# Print the DataFrame
print(metrics_df)
```

```
                Model  Accuracy  Precision    Recall  F1 Score
0  Logistic Regression  0.922759   0.994841  0.850423  0.916980
1        Decision Tree  0.904859   0.935975  0.869827  0.901690
2        Random Forest  0.822144   0.834158  0.805586  0.819623
```

[55]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
↪texlive-plain-generic
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 libapache-pom-java libcommons-logging-java
  libcommons-parent-java libfontbox-java libfontenc1 libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
  libruby3.0 libsynctex2 libteckit0 libtexlua53 libtexluajit2 libwoff1
  libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby
  ruby-net-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
```

```
  texlive-binaries texlive-latex-base texlive-latex-extra
  texlive-latex-recommended texlive-pictures tipa xfonts-encodings
  xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java
  poppler-utils ghostscript fonts-japanese-mincho | fonts-ipafont-mincho
  fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai
  fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper gv
  | postscript-viewer perl-tk xpdf | pdf-viewer xzdec
  texlive-fonts-recommended-doc texlive-latex-base-doc python3-pygments
  icc-profiles libfile-which-perl libspreadsheet-parseexcel-perl
  texlive-latex-extra-doc texlive-latex-recommended-doc texlive-luatex
  texlive-pstricks dot2tex prerex texlive-pictures-doc vprerex
  default-jre-headless tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono
  fonts-texgyre fonts-urw-base35 libapache-pom-java libcommons-logging-java
  libcommons-parent-java libfontbox-java libfontenc1 libgs9 libgs9-common
  libidn12 libijs-0.35 libjbig2dec0 libkpathsea6 libpdfbox-java libptexenc1
  libruby3.0 libsynctex2 libteckit0 libtexlua53 libtexluajit2 libwoff1
  libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby
  ruby-net-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0
  rubygems-integration t1utils teckit tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic texlive-xetex tipa xfonts-encodings xfonts-utils
0 upgraded, 54 newly installed, 0 to remove and 45 not upgraded.
Need to get 182 MB of archives.
After this operation, 571 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.6 [751 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
```

```
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.6 [5,031 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.5 [50.1 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:25 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:28 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ruby-webrick all
1.7.0-3 [51.8 kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.5 [5,113 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
```

```
Get:34 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:43 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 182 MB in 4s (44.4 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78,
<> line 54.)
debconf: falling back to frontend: Readline
```

```
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 121913 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.6_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.6) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.6_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.6) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
```

```
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/17-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/18-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/19-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/20-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/21-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/22-ruby3.0_3.0.2-7ubuntu2.5_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.5) …
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/23-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/24-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/25-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/26-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/27-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/28-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
```

```
Preparing to unpack …/29-libruby3.0_3.0.2-7ubuntu2.5_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.5) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/30-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/31-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/32-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/33-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/34-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/35-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/36-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/37-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/38-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/39-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/40-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/41-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/42-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/43-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
```

```
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/44-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/45-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/46-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/47-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/48-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/49-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/50-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/51-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/52-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/53-texlive-xetex_2021.20220204-1_all.deb …
Unpacking texlive-xetex (2021.20220204-1) …
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
Setting up tex-common (6.17) …
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
78.)
debconf: falling back to frontend: Readline
update-language: texlive-base not installed and configured, doing nothing!
```

```
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up ruby-webrick (1.7.0-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.6) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.6) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
debconf: unable to initialize frontend: Dialog
```

```
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line
78.)
debconf: falling back to frontend: Readline
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
Setting up tipa (2:1.3-21) …
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.5) …
Setting up ruby3.0 (3.0.2-7ubuntu2.5) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.4) …
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link
```

[ ]:
```
!jupyter nbconvert --to pdf /content/drive/MyDrive/Colab\ Notebooks/
↪Final_Project_HCDA.ipynb
```