2023

DS 7333 | Quantifying the World

CASE STUDY 2 | DIABETES & HOSPITAL READMISSION

JOEY HERNANDEZ
DANIEL CHANG

Introduction

Background

Diabetes, a chronic metabolic disorder characterized by elevated blood sugar levels, is a significant global health concern. The management of diabetes often involves hospitalization, especially for individuals experiencing acute complications or undergoing significant treatment adjustments. One critical aspect of diabetes care is the prevention of readmission into a hospital within a short period after the initial discharge. Readmissions can indicate various issues, such as inadequate treatment during the initial hospitalization, complications arising post-discharge, or a lack of effective outpatient care.

Objective and Scope

The primary objective of this report is to develop a predictive model to anticipate whether a patient with diabetes will be readmitted to a hospital within 30 days following their initial discharge. Achieving this objective can offer several advantages, including improved patient care, reduced healthcare costs, and enhanced resource allocation within healthcare facilities.

To accomplish our goal, we will employ logistic regression, a widely used statistical technique for binary classification. Logistic regression is well-suited for this task as it allows us to model the probability of readmission based on a set of relevant predictor variables. By analyzing these predictors, we can gain insights into the factors contributing to readmission risk among diabetic patients. Additionally, we will utilize imputation techniques to handle missing data, ensuring that our analysis is robust and representative of the patient population.

Data Source

This case study will utilize two datasets, “diabetic_data” and “IDs_mapping”. The data was provided to us in the Case 2 Study Module and is in the form of two separate csv files. When combined the data contains 101766 observations and 53 features.

Data Inspection

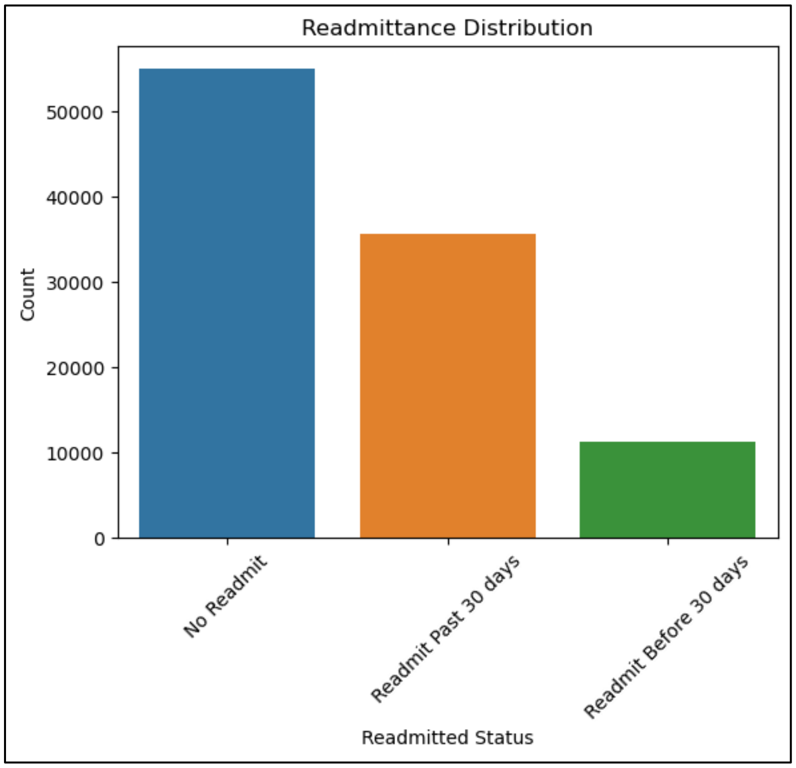
Before creating any models or analysis with the data our first step was to inspect our data to better understand data types (such as “int”, “cat”, “object”, etc.), distributions of values, identification of missing values, duplicated data, and outliers. This step is vital in understanding how we should approach any types of transformations or adjustments to the modeling and analysis process of our data.

Target Variable Inspection

To gain insight into the distribution of our classification target variable, we performed a thorough examination of the data. The target variable in this analysis "readmitted" represents the status of no readmission, readmission within 30 days or greater than 30 days of the initial hospitalization for patients. It is a critical factor in our predictive model, as it helps determine whether a patient falls into one of two categories: "no readmittance" or "less than 30 days readmittance." Which is the objective classification problem for our case study.

Initially, we visualized the distribution of the target variable in its original form. This initial examination revealed a substantial class imbalance between the three categories. Most instances were labeled as "no readmittance," while a considerably smaller portion represented "less than 30 days readmittance", and "greater than 30 days readmittance". This significant imbalance posed a challenge for our predictive modeling, as models may tend to be biased toward the majority class, potentially leading to suboptimal performance.

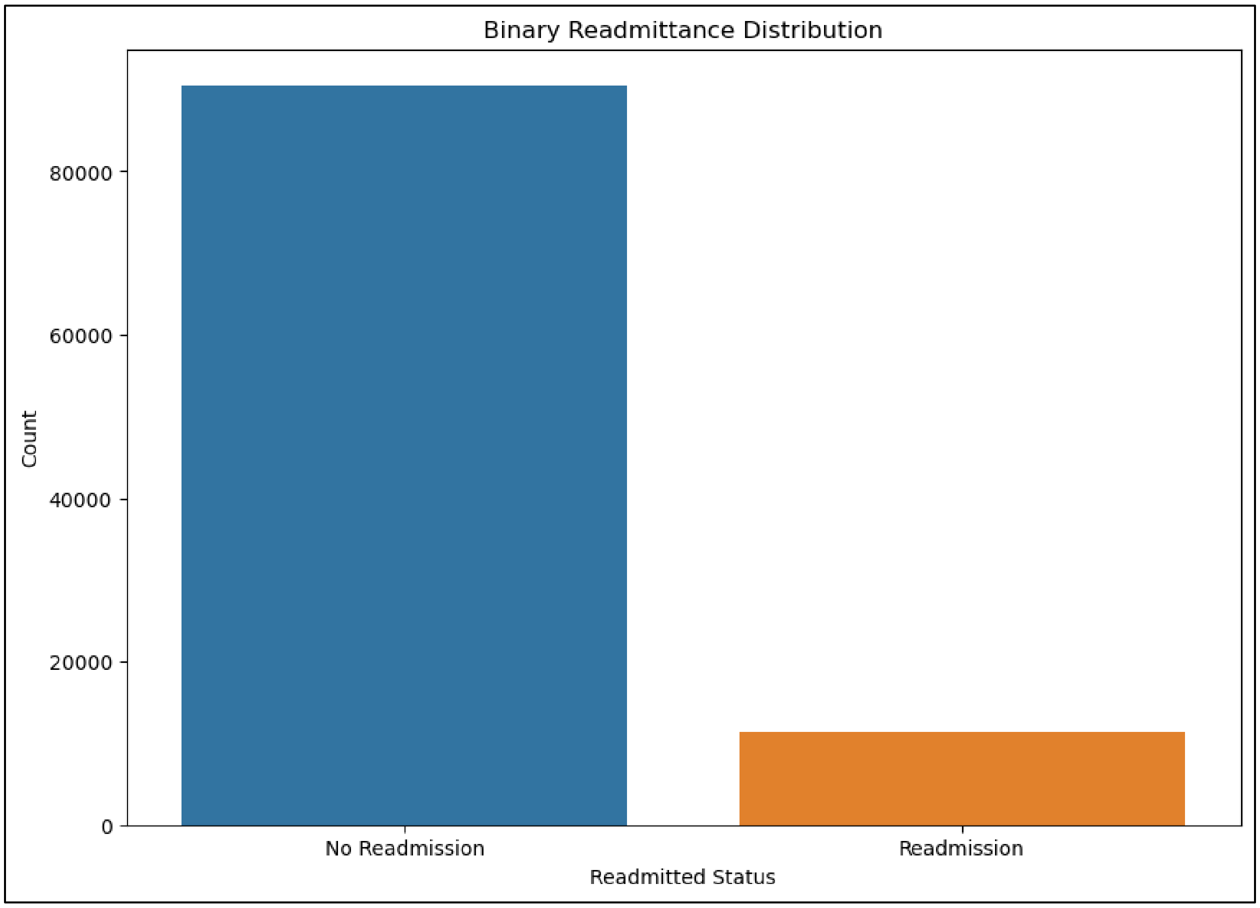
***Figure 1:** Count Plot Illustrating the Distribution of the Target Variable across the classes "No" ">30," "<30"*



***Description**: The count plot indicates that out of all the readmitted status. Most were not readmitted.*

The first amendment to our data and thus the scope of our project began with removal of the 'greater than 30 days readmittance'. The objective of our data as provided to us by the client specifically mentioned the desire to predict patients that would be readmitted to the hospital within 30 days, thus this change in the data allows us to remove information from our data that may otherwise bias our prediction of what type of readmittance is occurring.

Figure 2: *Count Plot Illustrating the new Distribution of the Target Variable across the binary values of "No Readmittance" and "Readmittance", where Readmittance is only occurrences where the patient was readmitted <30 days.*



Description*: The count plot indicates that out of all the readmitted statuses (after converting them to binary). Our "No Readmission" count is far higher than the "readmission" count.*

Patient Data Privacy And Information Inclusion/Exclusion

Respecting patient data privacy is of paramount importance in healthcare research, and this study is no exception. We recognize that certain demographic and clinical variables, such as race, gender, and age, can be invaluable in identifying potential contributors to the prediction of patient readmission. However, it is essential to emphasize that in our data analysis and modeling process, stringent measures have been implemented to safeguard patient privacy.

All patient identifiers, including but not limited to names, addresses, and specific identification numbers, have been rigorously excluded from our dataset. Additionally, any data attributes that could be used to directly infer individual patient identities or sensitive personal information have been carefully removed or anonymized to ensure the utmost protection of patient privacy.

The information used in our predictive model is limited to objective and de-identified data points relevant to the healthcare context. This means that our analysis is focused solely on variables that contribute to the accurate prediction of patient readmission risk, without compromising the confidentiality and privacy of individual patients. Our commitment to data privacy aligns with the highest ethical standards and legal requirements, ensuring that the insights derived from this study are both valuable and ethically sound.

Missing Data

In the process of preparing and cleaning the dataset for our analysis, we encountered various missing data points across different attributes. Our approach to handling these missing values is guided by thorough investigations into the nature and potential implications of the missing data.

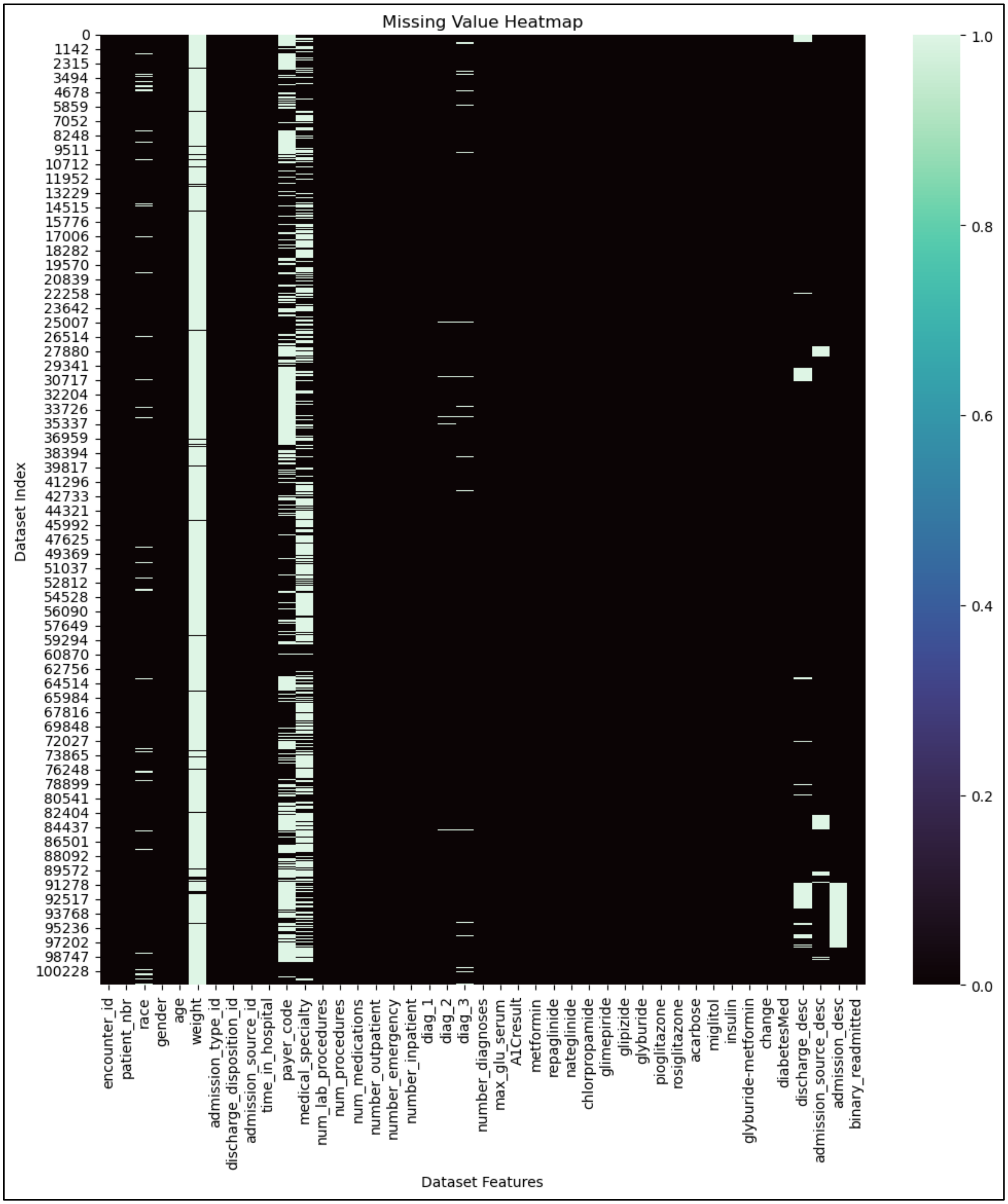
1. **Race:** In the 'race' attribute, we identified missing values denoted by "?". After further investigation, including visual analysis and cross-tabulations, we determined that these missing values were Missing Completely at Random (MCAR). As a result, we chose to proceed with imputation by replacing the missing values with the most frequent category. This approach allows us to maintain the integrity of the dataset while addressing the missing data issue.
2. **Weight:** The 'weight' attribute exhibited a significant percentage of missing values, exceeding 90%. Similar to the 'race' attribute, we investigated the nature of this missing data and confirmed that it was MCAR. However, due to the substantial extent of missing values and the limited potential usefulness of this feature, we made the decision to remove the 'weight' attribute from our dataset.
3. **Payer Code:** Missing values represented by "?" in the 'payer_code' attribute were also identified as MCAR upon thorough investigation. In this case, instead of imputation or removal, we chose to re-label the missing values and include them as a separate category within the 'payer_code' feature. This approach ensures that we retain valuable information while handling the missing data appropriately.
4. **Medical Specialty:** For the 'medical_specialty' attribute, there was no discernible relationship between the values in our dataset and the missingness of medical specialty values. To address this, we proceeded with inputting the missing values by assigning them a new category labeled as "other." This imputation strategy helps preserve the overall structure of the data while accounting for the missing information.

5. **Diagnosis Codes (diag_1, diag_2, diag_3):** While these attributes did contain missing values, it is worth noting that some of these gaps may be attributed to patients not having a specific diagnosis to report. Given the inconsequential amount of missing data in the diagnosis codes, we opted for a conservative approach by removing the null values from these attributes.

Our data preprocessing efforts regarding missing data aim to ensure that the resulting dataset is as informative and representative as possible while mitigating the potential bias introduced by the missing values. These carefully considered strategies allow us to maintain the integrity of the data and facilitate meaningful analyses for our predictive model of patient readmission.

Figure 3: *A heatmap representing the distribution of missing values across the dataset. Darker areas indicate the absence of data points, while lighter regions denote complete information.*

Understanding the pattern of missingness is crucial for effective data preprocessing and predictive modeling.



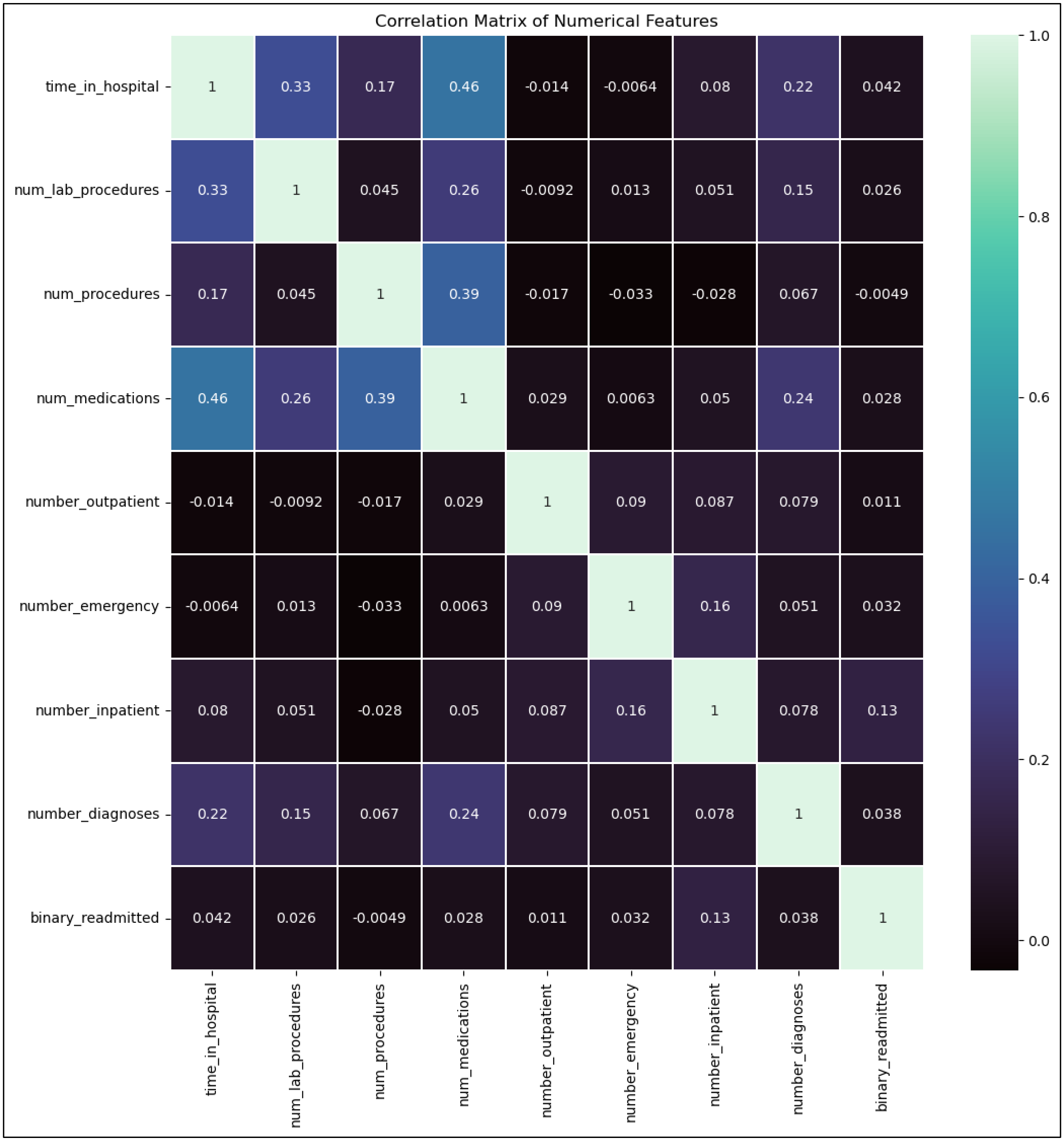
***Description**: Here is the heatmap of missing values. The lighter color represents a missing value, while the darker color represents a non-missing value. Right away, we can see that the weight column has the most number of missing values.*

Correlation Plot (Original Target Variable)

Prioritizing the preprocessing steps for the target variable in the step prior to this was important because it allowed us to assess integrity and accuracy of subsequent analyses. By addressing the target variable's distribution and more narrow scope, the resulting correlation values can be better trusted to either accurately reflect the underlying relationships between variables or understand what limitations may arise from the less than desirable target variable distribution. Failure to preprocess and identify data discrepancies in the target variable could lead to misinterpretations, as correlations might be influenced by skewedness, outliers, or nonlinearities within the target data.

Performing a correlation heatmap provides a visually informative representation of the relationships between variables within a dataset. By illustrating the strength and direction of linear associations, the heatmap becomes an indispensable tool for uncovering patterns and dependencies that might not be immediately apparent from individual variable analyses. Each cell in the heatmap corresponds to a pair of variables, with the color gradient indicating the magnitude of correlation. This enables the rapid identification of high and low correlation values, highlighting potential areas of interest for further investigation.

***Figure 4 (next page):** A heatmap illustrating the correlation among integer-type features in the dataset. The color intensity reflects the strength and direction of associations between variables. Analyzing feature correlations is essential for identifying potential patterns and dependencies that can inform our predictive modeling efforts.*



	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	number_diagnoses	binary_readmitted
time_in_hospital	1	0.33	0.17	0.46	-0.014	-0.0064	0.08	0.22	0.042
num_lab_procedures	0.33	1	0.045	0.26	-0.0092	0.013	0.051	0.15	0.026
num_procedures	0.17	0.045	1	0.39	-0.017	-0.033	-0.028	0.067	-0.0049
num_medications	0.46	0.26	0.39	1	0.029	0.0063	0.05	0.24	0.028
number_outpatient	-0.014	-0.0092	-0.017	0.029	1	0.09	0.087	0.079	0.011
number_emergency	-0.0064	0.013	-0.033	0.0063	0.09	1	0.16	0.051	0.032
number_inpatient	0.08	0.051	-0.028	0.05	0.087	0.16	1	0.078	0.13
number_diagnoses	0.22	0.15	0.067	0.24	0.079	0.051	0.078	1	0.038
binary_readmitted	0.042	0.026	-0.0049	0.028	0.011	0.032	0.13	0.038	1

Modeling

Lasso (L1 Regularization) Logistic Regression Model

To build an effective predictive model for patient readmission within 30 days of initial hospitalization, we began with a Lasso Logistic Regression approach. Lasso (Least Absolute Shrinkage and Selection Operator) is a regularization technique that helps prevent overfitting by penalizing the absolute values of the regression coefficients. In our analysis, we utilized Lasso Logistic Regression as a starting point due to its ability to perform feature selection by driving some coefficient estimates to zero, thereby simplifying the model and enhancing its interpretability.

Our model-building process included several crucial steps:

1. Grid Search for Optimal Alpha Value: We performed a grid search to determine the best alpha value for the Lasso Logistic Regression model. Alpha controls the strength of the L1 regularization penalty, and finding the optimal value is essential for achieving the right balance between model complexity and performance.

2. Train-Test Split: To assess the model's performance effectively, we divided the dataset into training and testing subsets. The training data were used to train the model, while the testing data served as an independent dataset for evaluating its generalization performance.

3. Feature Scaling and Dummitizing: Proper scaling of the features is crucial for logistic regression models. We standardized the numerical features to have zero mean and unit variance to ensure that all features contributed equally to the model. Additionally, we employed one-hot encoding (dummitizing) for categorical variables to convert them into a format suitable for the logistic regression model.

4. Applying Threshold to Classification: In binary classification problems like ours, a probability threshold is applied to determine the predicted class labels. We experimented with various threshold values to optimize the trade-off between sensitivity and specificity, tailoring the model to prioritize either minimizing false positives or false negatives, depending on the clinical context.

By implementing these steps, we aimed to develop a robust Lasso Logistic Regression model capable of predicting patient readmission within 30 days. This model not only provides predictive accuracy but also offers interpretability, allowing us to identify the most influential factors contributing to the likelihood of readmission. In the subsequent sections, we will detail the results of our model evaluation and discuss its implications for improving patient care and healthcare resource allocation.

<u>Results:</u>

Optimal Alpha (Lambda): The grid search procedure identified the best alpha value for Lasso regularization as 0.0100. This parameter controls the strength of regularization, striking a balance between model complexity and performance.

Test Set Accuracy: The Lasso Logistic Regression model achieved an accuracy of 0.6104 on the test dataset. This metric reflects the proportion of correctly classified instances, indicating that the model correctly predicted the readmission outcomes for approximately 61.04% of the test cases.

User-Defined Threshold: In binary classification, the choice of threshold for converting predicted probabilities into class labels can significantly impact the model's performance. In this case, a user-defined threshold of 0.65 was applied to the model's predicted probabilities to determine the classification labels.

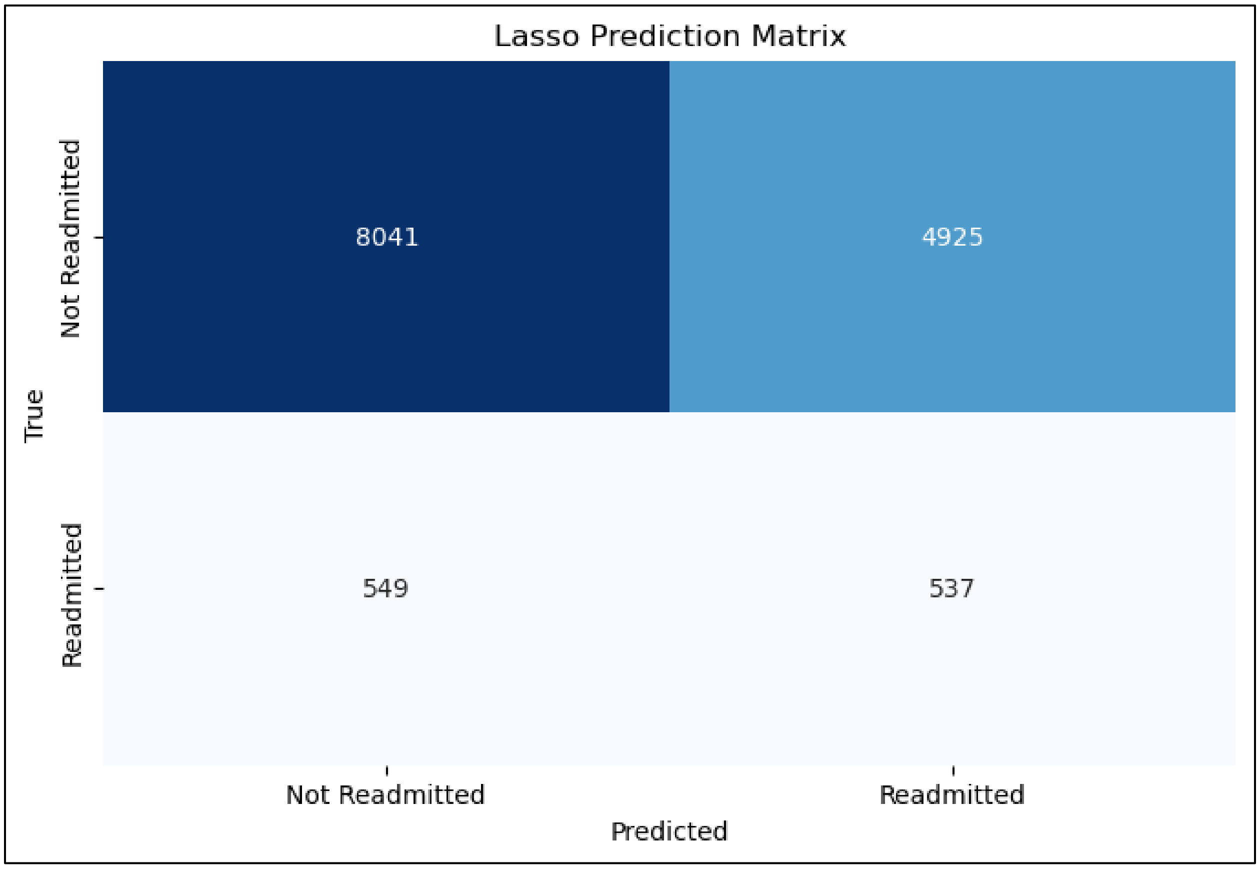
Classification Report with Custom Threshold:

- **Precision:** The precision for class "0" (no readmittance) was notably high at 0.94. This indicates that when the model predicted "no readmittance," it was accurate 94% of the time. Intuitively this makes sense given that the majority class of our data was “no readmittance” so it would be expected that the model would be bias to predicting “no readmittance”.
- **Recall:** The recall for class "0" was 0.62, indicating that the model captured 62% of the actual "no readmittance" cases. For class "1" (less than 30 days readmittance), the recall was 0.49, suggesting that the model identified 49% of the actual cases.
- **F1-Score:** The F1-score, which balances precision and recall, was 0.75 for class "0" and 0.16 for class "1."
- **Support:** The support represents the number of instances in each class within the test dataset. In this case, class "0" had significantly more instances (12,966) than class "1" (1,086).

Macro and Weighted Averages: The macro-average F1-score was 0.46, reflecting the overall balance between precision and recall across both classes. The weighted-average F1-score, which accounts for class imbalances, was 0.70. These averages provide a comprehensive view of the model's performance across all classes.

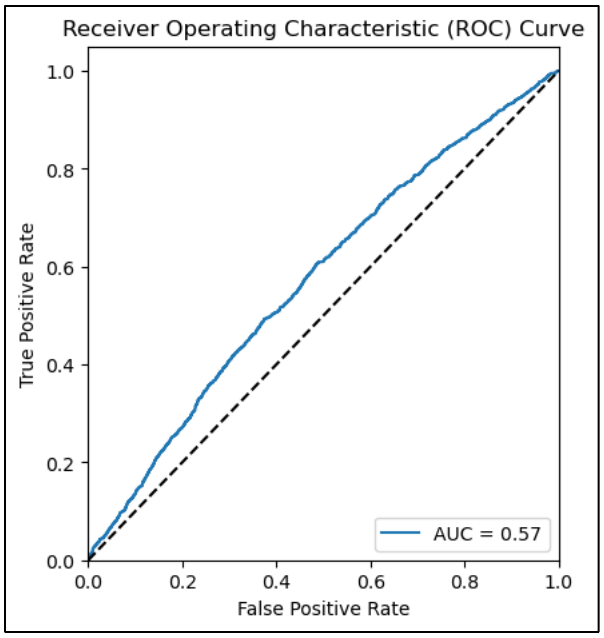
In summary, the Lasso Logistic Regression model, with an optimized alpha value and a user-defined threshold, demonstrated reasonable accuracy in predicting patient readmission within 30 days. It exhibited strong precision for the "no readmittance" class, indicating a high degree of confidence in negative predictions. However, there is room for improvement in recall and F1-score, particularly for the "less than 30 days readmittance" class, which represents cases where early readmission is crucial to identify. These results provide a baseline for our predictive modeling efforts, and further analysis and model refinements will be discussed in subsequent sections.

Figure 5 : *A heatmap of a confusion matrix.*



Description: This is a confusion matrix that was generated from our LASSO Logistic Regression.

Figure 6 : *Receiver Operating Characteristics(ROC) Curve.*



Description: This is a ROC Curve Plot with the False Positive Rate on the x-axis and True Positive Rate on the y-axis.

Figure 7 : *Classification Report of LASSO Logistic Regression*

Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
Not Readmitted	0.94	0.62	0.75	12966
Readmitted	0.10	0.49	0.16	1086
Accuracy			0.61	14052
Macro Average		0.56	0.46	14052
Weighted Average		0.61	0.70	14052

Description: Our classification report shows that our accuracy is 0.61 and our Macro Average F-1 Score is 0.46.

Lasso Logistic Regression Model – Under Sampling

For our second Lasso Logistic Regression model, we followed a similar procedure to the previous model but introduced a key modification: under sampling. Under sampling is a technique used to address class imbalance, which is particularly relevant in our binary classification problem due to the significant disproportion between the "no readmittance" and "less than 30 days readmittance" classes.

The steps in building this model included:

1. Grid Search for Optimal Alpha Value: We conducted a grid search to identify the optimal alpha value for Lasso Logistic Regression, maintaining the importance of regularization in feature selection and model simplification.

2. Train-Test Split: We again divided the dataset into training and testing subsets, ensuring an independent evaluation of model performance.

3. Feature Scaling and Dummitizing: Consistent with the previous model, we standardized numerical features and applied one-hot encoding to categorical variables for compatibility with logistic regression.

4. Under sampling: To mitigate the effects of class imbalance, we employed under sampling. This technique involved randomly selecting a subset of the majority class (in this case, "no readmittance") to balance the class distribution. By reducing the number of instances in the majority class, we aimed to ensure that the model did not disproportionately favor this class during training.

The introduction of under sampling in this Lasso Logistic Regression model allowed us to address the class imbalance issue more effectively. It enabled the model to learn from a more balanced dataset, potentially leading to better generalization and predictive performance, especially for the minority class ("less than 30 days readmittance"). In the subsequent sections, we will present the results of this model, including its evaluation and impact on predictive accuracy for patient readmission within 30 days.

Results

Optimal Alpha (Lambda): The grid search determined the best alpha value for Lasso regularization to be 0.0100, consistent with our first model.

Test Set Accuracy: The Lasso Logistic Regression model with under-sampling achieved an accuracy of 0.6600 on the test dataset. This metric indicates that the model correctly predicted the readmission outcomes for approximately 66% of the test cases.

User-Defined Threshold: To classify predictions into classes, a user-defined threshold of 0.98 was applied to the model's predicted probabilities.

Classification Report with Custom Threshold:

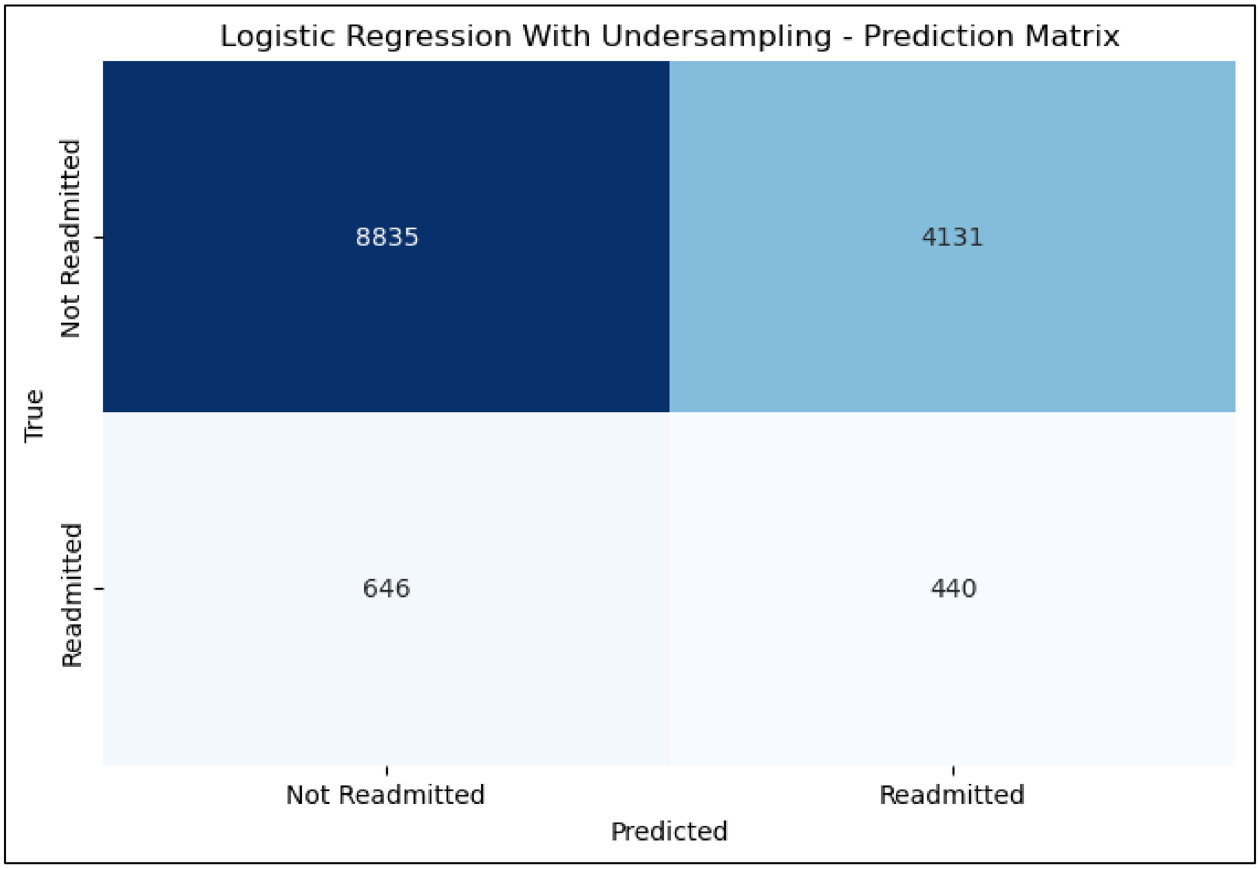
- **Precision:** The precision for class "0" (no readmittance) remained high at 0.93, indicating that when the model predicted "no readmittance," it was accurate 93% of the time. For class "1" (less than 30 days readmittance), the precision was 0.10, reflecting that the model's positive predictions for this class had a lower precision rate.

- **Recall:** The recall for class "0" was 0.68, suggesting that the model captured 68% of the actual "no readmittance" cases. The recall for class "1" was 0.41, indicating that the model identified 41% of the actual cases of "less than 30 days readmittance."
- **F1-Score:** The F1-score for class "0" improved to 0.79, demonstrating a better balance between precision and recall compared to our first model. However, the F1-score for class "1" remained at 0.16.
- **Support:** As in the previous model, class "0" had a substantially larger number of instances (12,966) compared to class "1" (1,086).

Macro and Weighted Averages: The macro-average F1-score was 0.47, reflecting the overall balance between precision and recall across both classes. The weighted-average F1-score, accounting for class imbalances, improved to 0.74.

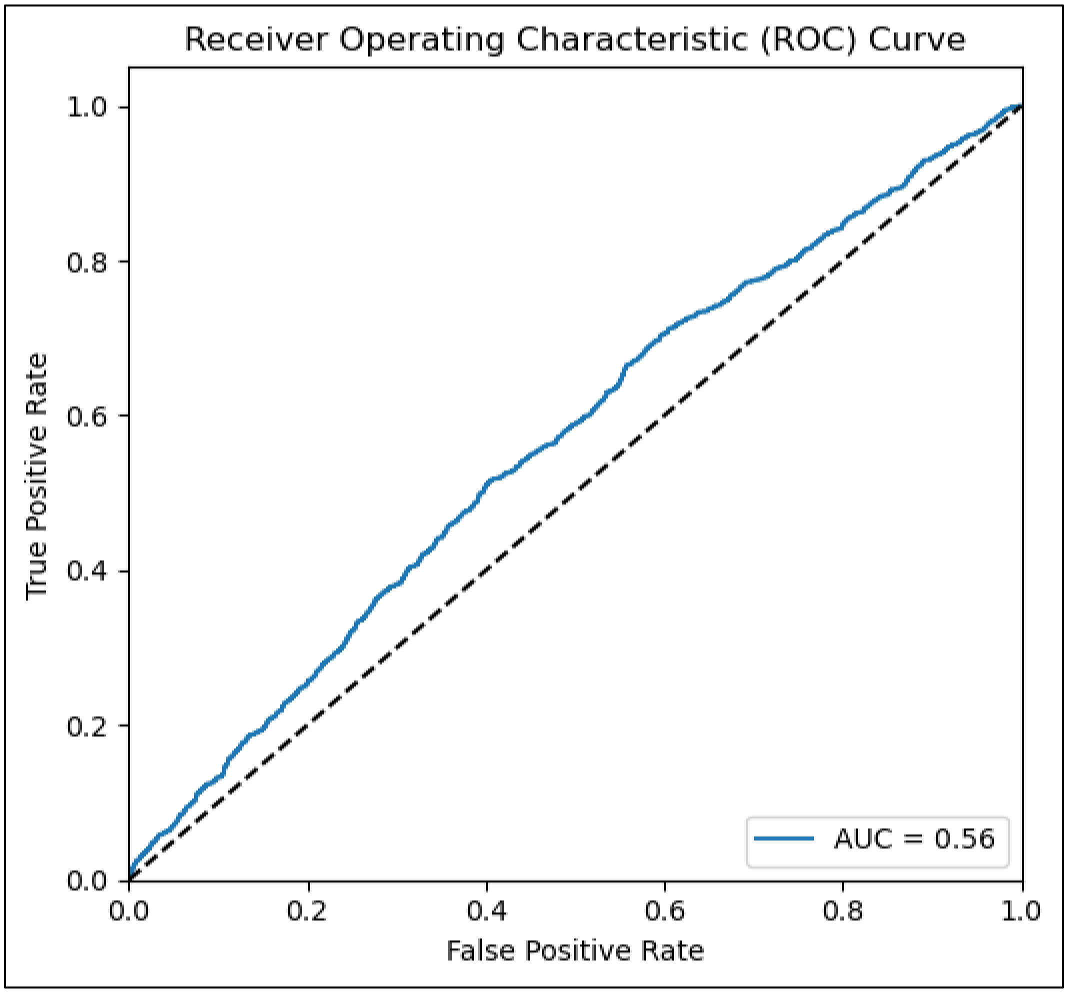
In summary, the Lasso Logistic Regression model with undersampling showed an improvement in accuracy compared to our first model. While precision remained high for class "0," the model's ability to identify cases of "less than 30 days readmittance" improved, as reflected in the higher recall for this class and a slightly improved F1-score. The undersampling technique helped address class imbalance, leading to a more balanced model. However, there is still room for further enhancement, especially in improving the model's ability to identify cases of early readmission. These results provide valuable insights for our ongoing analysis and model refinement efforts.

Figure 8 : *Confusion Matrix of Logistic Regression with UnderSampling.*



Description: *Here is the confusion matrix of our Logistic Regression model with UnderSampling. It seems somewhat similar to our LASSO model.*

Figure 9 : *Receiver Operating Characteristics(ROC) Curve of Logistic Regression with UnderSampling.*



Description: *This is a ROC Curve Plot with the False Positive Rate on the x-axis and True Positive Rate on the y-axis.*

Figure 10 : *Classification Report of Logistic Regression with UnderSampling.*

Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
Not Readmitted	0.93	0.68	0.79	12966
Readmitted	0.10	0.41	0.16	1086
Accuracy			0.66	14052
Macro Average	0.51	0.54	0.47	14052
Weighted Average	0.87	0.66	0.74	14052

Description: *Classification Report for Logistic Regression with UnderSampling. From the report, we can see that our model has an improved accuracy of 0.66.*

Random Forest

In addition to Lasso Logistic Regression, we explored the application of a Random Forest classifier as our third predictive model for patient readmission within 30 days of initial hospitalization. The Random Forest algorithm is an ensemble learning method that leverages the collective decision-making of multiple decision trees. This approach often yields robust and accurate predictions, making it a valuable tool for complex classification tasks like ours.

Our strategy for implementing the Random Forest model consisted of the following steps:

1. Hyperparameter Tuning: To optimize the performance of the Random Forest model, we conducted hyperparameter tuning. This involved exploring various settings, such as the number of trees in the forest and the maximum depth of each tree. We aimed to identify the combination of hyperparameters that would result in the most effective model.

2. Train-Test Split: Similar to our previous models, we divided the dataset into training and testing subsets. This separation allowed us to train the Random Forest model on one portion of the data and evaluate its performance on an independent dataset, ensuring that the model generalizes well to new, unseen instances.

3. Feature Importance Analysis: One of the advantages of Random Forest is its ability to provide insight into feature importance. We conducted a feature importance analysis to identify which variables had the most significant influence on predicting patient readmission. This analysis can be valuable for healthcare providers and decision-makers to prioritize interventions and allocate resources effectively.

4. Model Evaluation: We rigorously assessed the performance of the Random Forest model using a range of evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. These metrics allowed us to gauge the model's predictive accuracy and its ability to balance sensitivity and specificity in predicting readmission outcomes.

5. Class Weight: In contrast to the under sampling technique utilized in the logistic regression model Random Forest Classifier has a parameter" class_weight" where we can set class weight equal to 'balanced'. This is a more convenient option compared to under/overs sampling because it doesn't require modifying the training dataset's size. Instead, it automatically calculates class weights inversely proportional to the class frequencies in the training data. In other words, it assigns higher weights to the minority class and lower weights to the majority class.

By incorporating the Random Forest model into our analysis, we aimed to leverage its ensemble capabilities to capture complex relationships in the data and provide a more accurate and robust prediction of patient readmission within 30 days. The results of our Random Forest model will be discussed in subsequent sections, shedding light on its effectiveness in enhancing patient care and healthcare resource allocation.

<u>**Results:**</u>

Our third model employed the Random Forest classifier with class weights set to 'balanced' to address the class imbalance issue. Here are the results obtained from this model:

Test Set Accuracy: The Random Forest model with balanced class weights achieved an accuracy of 0.6050 on the test dataset. This metric indicates that the model correctly predicted the readmission outcomes for approximately 60.50% of the test cases.

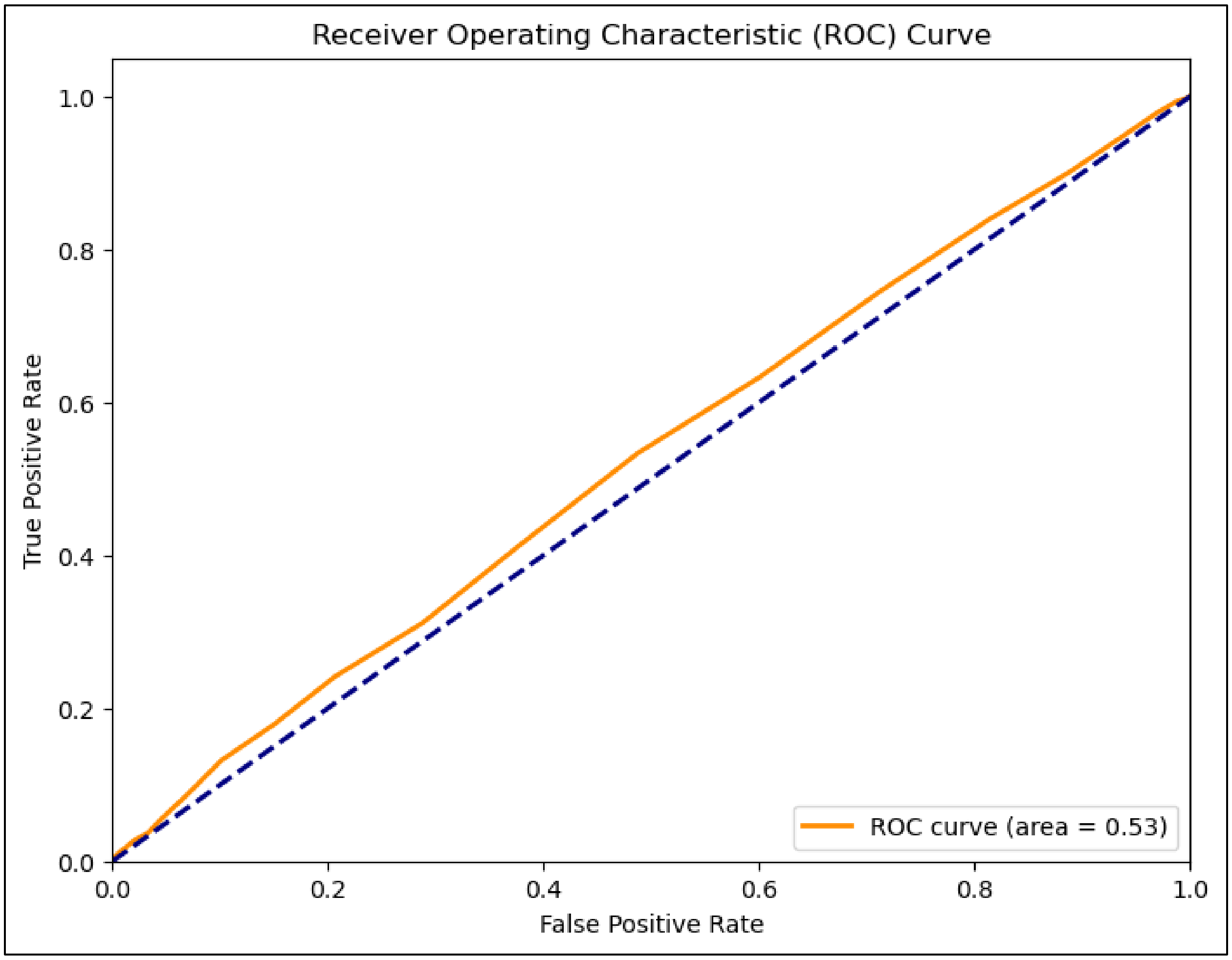
Classification Report:

- **Precision:** The precision for class "0" (no readmittance) remained high at 0.93, indicating that when the model predicted "no readmittance," it was accurate 93% of the time. However, the precision for class "1" (less than 30 days readmittance) was lower at 0.08, reflecting a lower precision rate for positive predictions.
- **Recall:** The recall for class "0" was 0.62, suggesting that the model captured 62% of the actual "no readmittance" cases. For class "1," the recall was 0.41, indicating that the model identified 41% of the actual cases of "less than 30 days readmittance."
- **F1-Score:** The F1-score for class "0" was 0.74, demonstrating a good balance between precision and recall. However, the F1-score for class "1" remained low at 0.14.
- **Support:** As in the previous models, class "0" had a significantly larger number of instances (12,966) compared to class "1" (1,086).

Macro and Weighted Averages: The macro-average F1-score was 0.44, reflecting the overall balance between precision and recall across both classes. The weighted-average F1-score, accounting for class imbalances, was 0.70.

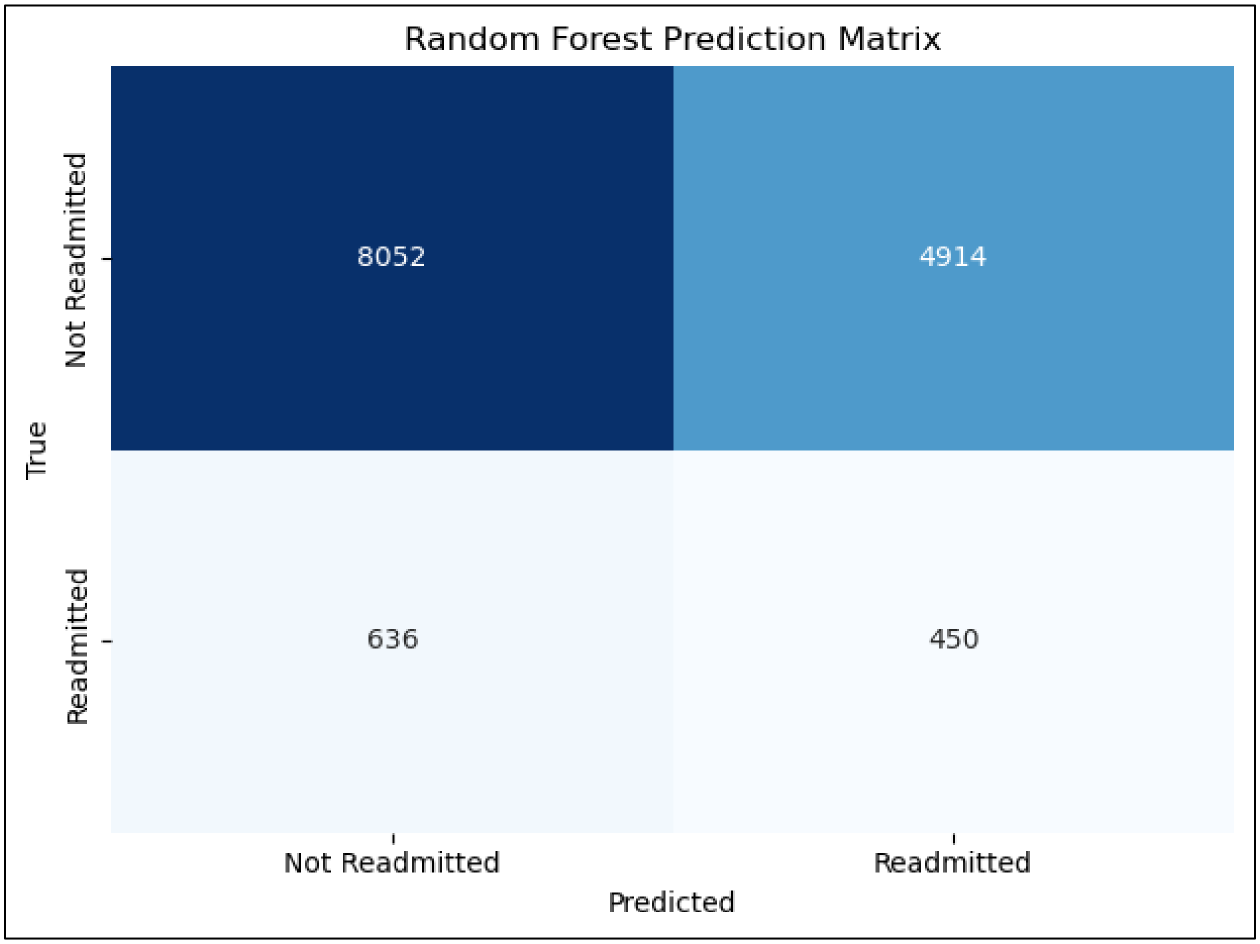
In summary, the Random Forest Classifier with balanced class weights exhibited a moderate accuracy in predicting patient readmission within 30 days. The model demonstrated high precision for class "0" (no readmittance), but the ability to identify cases of "less than 30 days readmittance" remained limited, as reflected in the lower recall and F1-score for class "1." While balanced class weights helped mitigate class imbalance, further model refinement may be needed to improve its performance in identifying early readmission cases. These results provide insights for ongoing analysis and potential model enhancements.

Figure 11(Next Page) : *Receiver Operating Characteristics(ROC) Curve of Random Forest.*



Description: This is a ROC Curve Plot with the False Positive Rate on the x-axis and True Positive Rate on the y-axis.

Figure 12(Next Page) :*Confusion Matrix of Random Forest.*



***Description**: Here is the confusion matrix of our Random Forest. It seems somewhat similar to our LASSO model.*

***Figure 13 :** Classification Report of Random Forest.*

Classification Report				
	Precision	**Recall**	**F1- Score**	**Support**
Not Readmitted	0.93	0.62	0.74	12966
Readmitted	0.08	0.41	0.14	1086
Accuracy			0.61	14052
Macro Average	0.51	0.52	0.44	14052
Weighted Average	0.86	0.61	0.70	14052

***Description**: Classification Report for Random Forest. From the report, we can see that our model has an improved accuracy of 0.61.*

Conclusion

In this comprehensive analysis, we delved into the critical task of predicting patient readmission within 30 days of their initial hospitalization, a significant concern in the management of diabetes and other chronic conditions. Our study revolved around the evaluation of three distinct predictive models: Lasso Logistic Regression, Lasso Logistic Regression with Under sampling, and Random Forest Classifier with Balanced Class Weights. Each model brought its unique strengths and considerations to the table, contributing to our understanding of this complex problem.

The Lasso Logistic Regression model served as a foundational step, showcasing the importance of regularization in feature selection and model simplification. Despite reasonable accuracy, this initial model revealed the challenges of striking a balance between precision and recall, particularly for cases of early readmission. Feature engineering and custom thresholding were essential steps to fine-tune its performance.

The introduction of under sampling in our second model marked an effective attempt to mitigate class imbalance, providing a more balanced dataset for training. This adjustment yielded improvements in overall accuracy and recall for cases of early readmission, although there is still room for enhancement in precision and F1-score.

Our third model, the Random Forest Classifier with balanced class weights, showcased the robustness of ensemble learning in addressing complex classification tasks. While achieving a moderate level of accuracy, this model upheld high precision for class "0" predictions but struggled to effectively identify cases of "less than 30 days readmittance." Balancing class weights helped, yet further optimization is needed to bridge the gap between precision and recall for class "1."

In conclusion, the prediction of patient readmission within a 30-day window is a multifaceted challenge with significant clinical implications. Our models provide a foundation for understanding this problem, and their results shed light on the intricacies of balancing accuracy, precision, and recall, especially concerning early readmission cases.

Recommendations

Based on our analysis and findings, we propose several recommendations to enhance the predictive modeling of patient readmission within 30 days of initial hospitalization, with a focus on improving model performance and the practical utility of the predictions:

1. Feature Engineering and Selection:

- **Feature Engineering:** Continue to explore feature engineering techniques to create meaningful and predictive variables. Consider interactions between variables, time-based features, and domain-specific variables that may impact readmission risk.
- **Dimensionality Reduction:** Given the dimensionality of the dataset, consider employing dimensionality reduction techniques, or feature selection algorithms, to reduce noise and enhance model interpretability.

2. Dual Threshold Model:

- Develop a dual-threshold model that assigns probabilities or likelihood scores to the likelihood of readmission. This model can provide more nuanced insights by distinguishing between patients with a higher risk of early readmission and those with a lower risk. Implementing two thresholds allows healthcare providers to prioritize interventions more effectively.

3. Handling Class Imbalance:

- Continue experimenting with techniques to address class imbalance. Undersampling and oversampling methods, such as Synthetic Minority Over-sampling Technique (SMOTE), may further improve the model's ability to identify early readmission cases.

4. SME Collaboration:

- Collaborate closely with healthcare professionals to clinically validate the model's predictions. Incorporate domain expertise to refine the model's features, thresholds, and evaluation metrics to align better with the practical needs of healthcare providers.

5. Interpretability and Explainability:

- Prioritize the development of model interpretability and explainability techniques. Transparent models are essential for healthcare practitioners to understand and trust model predictions.

6. Data Quality and Collection:

- Invest in improving data quality by addressing missing data issues and conducting regular data audits. Collect additional relevant variables that might not be part of the current dataset but could have significant predictive power.

By implementing these recommendations, healthcare providers and data scientists can work together to develop more accurate, reliable, and clinically relevant models for predicting patient readmission. These efforts can ultimately contribute to improved patient care, better allocation of healthcare resources, and enhanced outcomes in the management of chronic conditions such as diabetes.

Appendix

Import Pacakges

In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

Import Data

Diabetic data (main set)

In [2]:

```
pd.set_option("display.max_columns", None)

diabetic_data = pd.read_csv('diabetic_data.csv')
diabetic_data.head()
```

Out[2]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	a
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	25	
1	149190	55629189	Caucasian	Female	[10-20)	?	1	1	
2	64410	86047875	AfricanAmerican	Female	[20-30)	?	1	1	
3	500364	82442376	Caucasian	Male	[30-40)	?	1	1	
4	16680	42519267	Caucasian	Male	[40-50)	?	1	1	

ID dataset (3 of them)

In [3]:

```
id_data = pd.read_csv('IDs_mapping.csv', header = None)
id_data.head()
```

Out[3]:

	0	1
0	admission_type_id	description
1	1	Emergency
2	2	Urgent
3	3	Elective
4	4	Newborn

In [4]:
```
# dataset one - admission_type
empty_rows = id_data[id_data.isna().all(axis=1)].index.tolist()

id_admission_type = id_data.iloc[:empty_rows[0]]
id_admission_type.columns = id_admission_type.iloc[0] # get the columsn to be first row
id_admission_type = id_admission_type.drop(id_admission_type.index[0]) # remove the firs
id_admission_type = id_admission_type.reset_index(drop=True) # reset index

id_admission_type.head(20)
```

Out[4]:

	admission_type_id	description
0	1	Emergency
1	2	Urgent
2	3	Elective
3	4	Newborn
4	5	Not Available
5	6	NaN
6	7	Trauma Center
7	8	Not Mapped

In [5]:
```
# dataset two - discharge id
id_discharge = id_data.iloc[empty_rows[0]+ 1 : empty_rows[1]]
id_discharge.columns = id_discharge.iloc[0]
id_discharge = id_discharge.drop(id_discharge.index[0])
id_discharge = id_discharge.reset_index(drop = True)

id_discharge.head(31)
```

Out[5]:

10	discharge_disposition_id	description
0	1	Discharged to home
1	2	Discharged/transferred to another short term h...
2	3	Discharged/transferred to SNF
3	4	Discharged/transferred to ICF
4	5	Discharged/transferred to another type of inpa...
5	6	Discharged/transferred to home with home healt...
6	7	Left AMA
7	8	Discharged/transferred to home under care of H...
8	9	Admitted as an inpatient to this hospital
9	10	Neonate discharged to another hospital for neo...
10	11	Expired
11	12	Still patient or expected to return for outpat...
12	13	Hospice / home
13	14	Hospice / medical facility
14	15	Discharged/transferred within this institution...

15	16	Discharged/transferred/referred another instit...
16	17	Discharged/transferred/referred to this instit...
17	18	NaN
18	19	Expired at home. Medicaid only, hospice.
19	20	Expired in a medical facility. Medicaid only, ...
20	21	Expired, place unknown. Medicaid only, hospice.
21	22	Discharged/transferred to another rehab fac in...
22	23	Discharged/transferred to a long term care hos...
23	24	Discharged/transferred to a nursing facility c...
24	25	Not Mapped
25	26	Unknown/Invalid
26	30	Discharged/transferred to another Type of Heal...
27	27	Discharged/transferred to a federal health car...
28	28	Discharged/transferred/referred to a psychiatr...
29	29	Discharged/transferred to a Critical Access Ho...

In [6]:
```
# dataset three - admission source id
id_admission_source = id_data.iloc[empty_rows[1]+1 : ]
id_admission_source.columns = id_admission_source.iloc[0]
id_admission_source = id_admission_source.drop(id_admission_source.index[0])
id_admission_source = id_admission_source.reset_index(drop = True)
id_admission_source.head(26)
```

Out[6]:

42	admission_source_id	description
0	1	Physician Referral
1	2	Clinic Referral
2	3	HMO Referral
3	4	Transfer from a hospital
4	5	Transfer from a Skilled Nursing Facility (SNF)
5	6	Transfer from another health care facility
6	7	Emergency Room
7	8	Court/Law Enforcement
8	9	Not Available
9	10	Transfer from critial access hospital
10	11	Normal Delivery
11	12	Premature Delivery
12	13	Sick Baby
13	14	Extramural Birth
14	15	Not Available
15	17	NaN
16	18	Transfer From Another Home Health Agency

17	19	Readmission to Same Home Health Agency
18	20	Not Mapped
19	21	Unknown/Invalid
20	22	Transfer from hospital inpt/same fac reslt in...
21	23	Born inside this hospital
22	24	Born outside this hospital
23	25	Transfer from Ambulatory Surgery Center
24	26	Transfer from Hospice

Merge Datasets

In [7]:
```
id_admission_type['admission_type_id'] = id_admission_type['admission_type_id'].astype(i
id_discharge['discharge_disposition_id'] = id_discharge['discharge_disposition_id'].asty
id_admission_source['admission_source_id'] = id_admission_source['admission_source_id'].
```

In [8]:
```
df = pd.merge(diabetic_data, id_admission_type, how ='inner', on = 'admission_type_id')
df = pd.merge(df, id_discharge, how = 'inner', on = 'discharge_disposition_id') # descri
df = pd.merge(df, id_admission_source, how = 'inner', on= 'admission_source_id') # descr
```

In [9]:
```
df.rename(columns={'description':'admission_desc',
                  'description_x':'discharge_desc',
                  'description_y':'admission_source_desc'}, inplace = True)
```

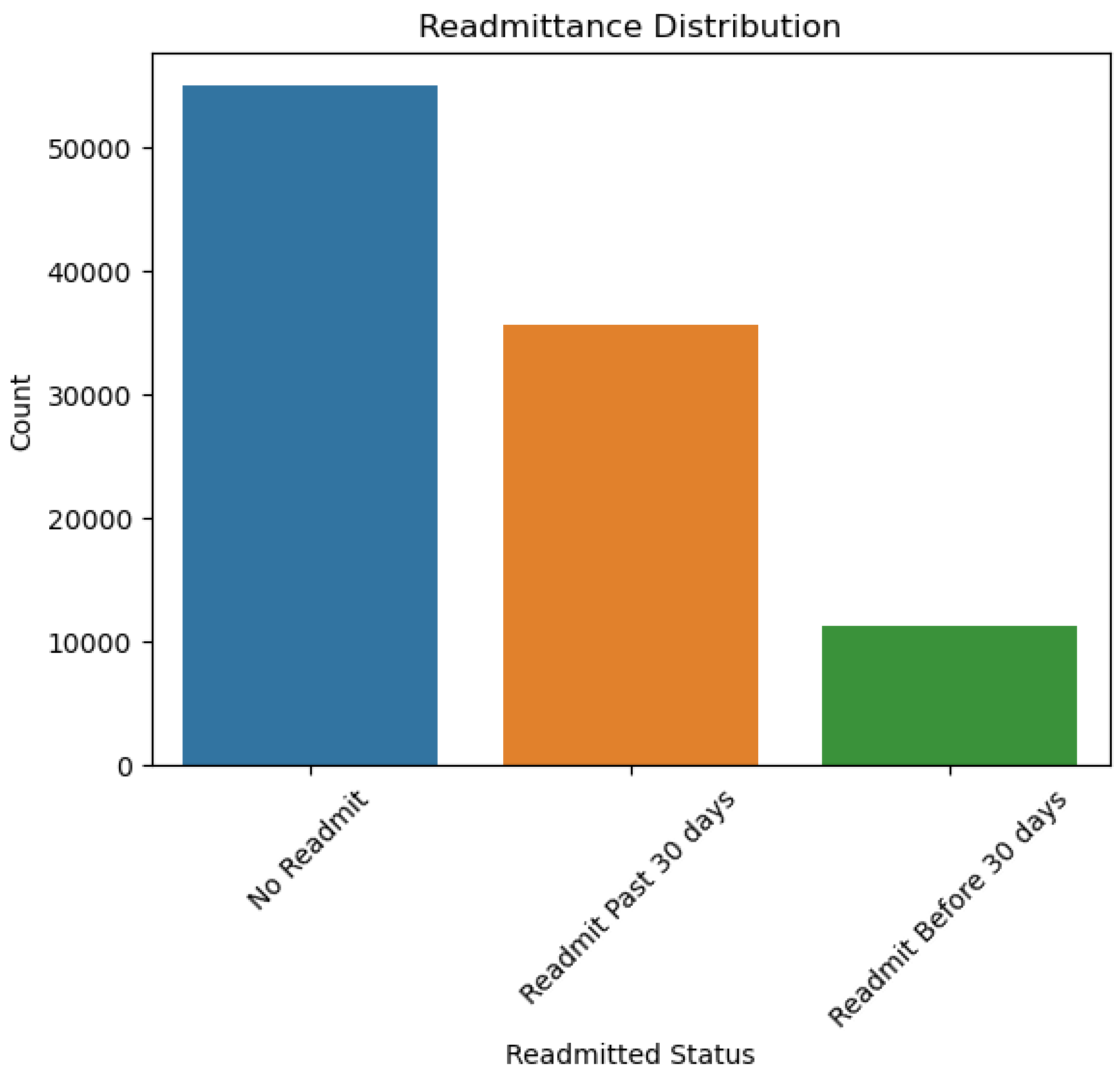
In [10]:
```
df.head()
```

Out[10]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	a
0	2278392	8222157	Caucasian	Female	[0-10)	?	6	25	
1	1968528	720936	Caucasian	Female	[70-80)	?	6	25	
2	2223336	558360	AfricanAmerican	Female	[60-70)	?	6	25	
3	2298006	2519748	Caucasian	Male	[60-70)	?	6	25	
4	2356308	608841	AfricanAmerican	Female	[50-60)	?	6	25	

Target Variable Inspection

In [11]:
```
custom_x = ['No Readmit', 'Readmit Past 30 days','Readmit Before 30 days']

ax = sns.countplot(df, x = 'readmitted')
ax.set_xticklabels(custom_x)
plt.xticks(rotation = 45)
plt.title('Readmittance Distribution')
plt.xlabel('Readmitted Status')
plt.ylabel('Count')
plt.show()
```

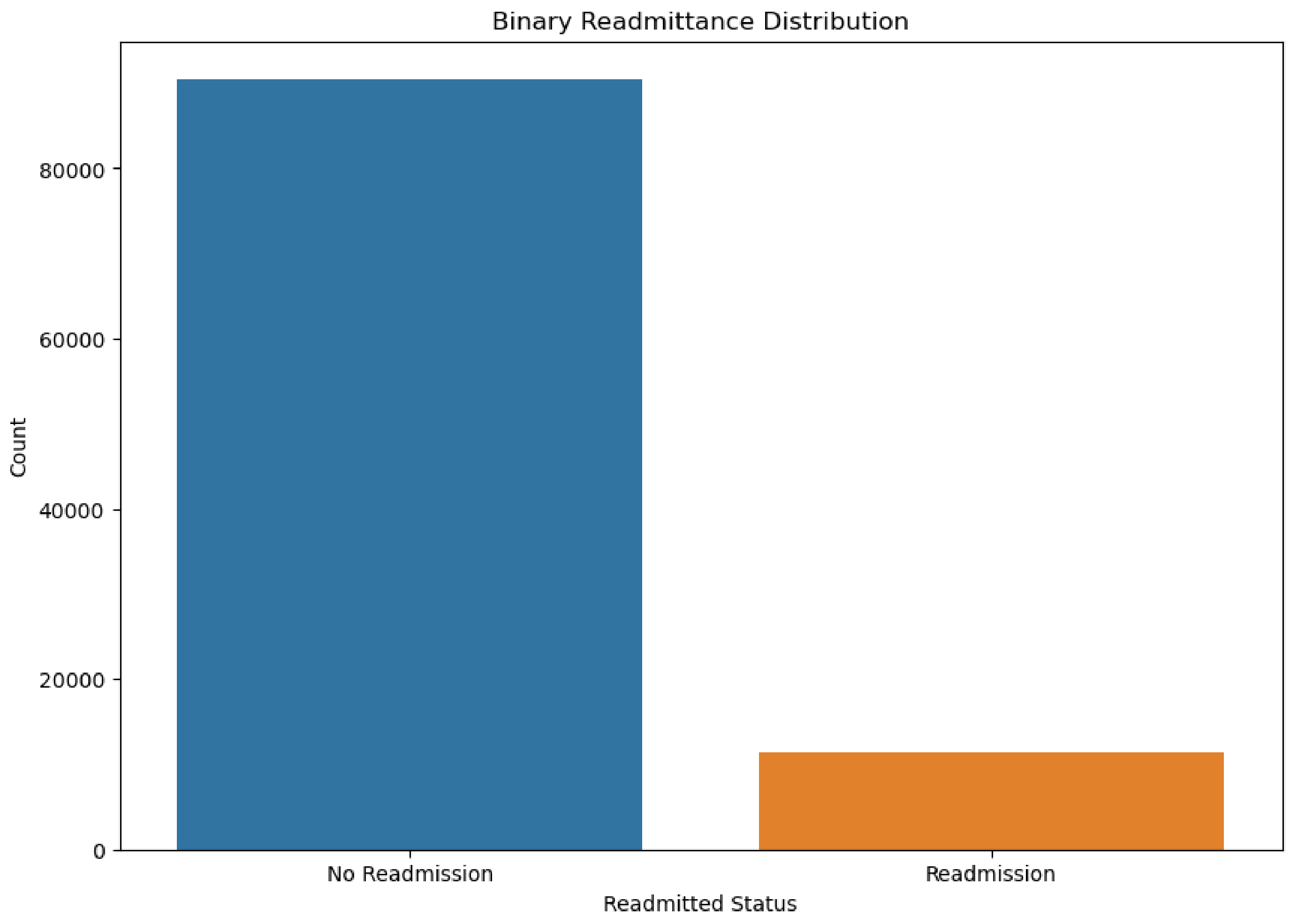


Binary Target Variable

- Becuase we are looking to classify readmittance within 30 days, we will make this a binary type of target instead of multiclass.
- This will look like "readmitted_30? = Yes or No"

In [12]:
```
# df['binary_readmitted'] = df['readmitted'].apply(lambda x: 'yes' if x != 'NO' else 'no
df['binary_readmitted'] = df['readmitted'].apply(lambda x: 1 if x == '<30' else 0)
```

In [13]:
```
custom_x = ['No Readmission','Readmission']

plt.figure(figsize=(10,7))
ax = sns.countplot(df, x = 'binary_readmitted')
ax.set_xticklabels(custom_x)
plt.title('Binary Readmittance Distribution')
plt.xlabel('Readmitted Status')
plt.ylabel('Count')
plt.show()
```



Missing Values

Examining Value Counts

In [14]:
```
value_counts_dict = {}
for column in df.columns:
    value_counts_dict[column] = df[column].value_counts()

for column, i in value_counts_dict.items():
    print(f'Column: {column}')
    print(i)
    print('\n')
```

```
Column: encounter_id
2278392      1
217273374    1
217188990    1
217165278    1
217146714    1
            ..
48648786     1
48582588     1
48525978     1
48513912     1
370756376    1
Name: encounter_id, Length: 101766, dtype: int64
```

```
Column: patient_nbr
88785891     40
43140906     28
1660293      23
23199021     23
88227540     23
             ..
112244895     1
5899338       1
108997866     1
3452139       1
87775947      1
Name: patient_nbr, Length: 71518, dtype: int64


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


Column: gender
Female             54708
Male               47055
Unknown/Invalid        3
Name: gender, dtype: int64


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


Column: weight
?            98569
[75-100)      1336
[50-75)        897
[100-125)      625
[125-150)      145
[25-50)         97
[0-25)          48
[150-175)       35
[175-200)       11
>200             3
Name: weight, dtype: int64


Column: admission_type_id
1    53990
3    18869
2    18480
```

```
6     5291
5     4785
8      320
7       21
4       10
Name: admission_type_id, dtype: int64


Column: discharge_disposition_id
1     60234
3     13954
6     12902
18     3691
2      2128
22     1993
11     1642
5      1184
25      989
4       815
7       623
23      412
13      399
14      372
28      139
8       108
15       63
24       48
9        21
17       14
16       11
19        8
10        6
27        5
12        3
20        2
Name: discharge_disposition_id, dtype: int64


Column: admission_source_id
7     57494
1     29565
17     6781
4      3187
6      2264
2      1104
5       855
3       187
20      161
9       125
8        16
22       12
10        8
14        2
11        2
25        2
13        1
Name: admission_source_id, dtype: int64


Column: time_in_hospital
3     17756
2     17224
1     14208
4     13924
5      9966
6      7539
```

```
7      5859
8      4391
9      3002
10     2342
11     1855
12     1448
13     1210
14     1042
Name: time_in_hospital, dtype: int64


Column: payer_code
?     40256
MC    32439
HM     6274
SP     5007
BC     4655
MD     3532
CP     2533
UN     2448
CM     1937
OG     1033
PO      592
DM      549
CH      146
WC      135
OT       95
MP       79
SI       55
FR        1
Name: payer_code, dtype: int64


Column: medical_specialty
?                                   49949
InternalMedicine                    14635
Emergency/Trauma                     7565
Family/GeneralPractice               7440
Cardiology                           5352
                                    ...
Surgery-PlasticwithinHeadandNeck        1
Neurophysiology                         1
Speech                                  1
Psychiatry-Addictive                    1
Pediatrics-InfectiousDiseases           1
Name: medical_specialty, Length: 73, dtype: int64


Column: num_lab_procedures
1      3208
43     2804
44     2496
45     2376
38     2213
       ...
118       1
121       1
126       1
132       1
129       1
Name: num_lab_procedures, Length: 118, dtype: int64


Column: num_procedures
0    46652
1    20742
```

```
2    12717
3     9443
6     4954
4     4180
5     3078
Name: num_procedures, dtype: int64


Column: num_medications
13    6086
12    6004
11    5795
15    5792
14    5707
      ...
75       2
70       2
79       1
81       1
74       1
Name: num_medications, Length: 75, dtype: int64


Column: number_outpatient
0     85027
1      8547
2      3594
3      2042
4      1099
5       533
6       303
7       155
8        98
9        83
10       57
11       42
13       31
12       30
14       28
15       20
16       15
17        8
21        7
20        7
22        5
18        5
27        3
24        3
19        3
23        2
25        2
26        2
29        2
36        2
33        2
35        2
42        1
34        1
40        1
38        1
37        1
28        1
39        1
Name: number_outpatient, dtype: int64
```

```
Column: number_emergency
0     90383
1      7677
2      2042
3       725
4       374
5       192
6        94
7        73
8        50
10       34
9        33
11       23
13       12
12       10
22        6
16        5
18        5
19        4
20        4
14        3
15        3
21        2
25        2
28        1
42        1
76        1
37        1
29        1
46        1
64        1
63        1
54        1
24        1
Name: number_emergency, dtype: int64


Column: number_inpatient
0     67630
1     19521
2      7566
3      3411
4      1622
5       812
6       480
7       268
8       151
9       111
10       61
11       49
12       34
13       20
14       10
15        9
16        6
19        2
21        1
18        1
17        1
Name: number_inpatient, dtype: int64


Column: diag_1
428    6862
414    6581
786    4016
```

```
410    3614
486    3508
       ...
640       1
314       1
817       1
61        1
915       1
Name: diag_1, Length: 717, dtype: int64


Column: diag_2
276    6752
428    6662
250    6071
427    5036
401    3736
       ...
256       1
752       1
316       1
963       1
917       1
Name: diag_2, Length: 749, dtype: int64


Column: diag_3
250     11555
401      8289
276      5175
428      4577
427      3955
        ...
744         1
732         1
E945        1
122         1
193         1
Name: diag_3, Length: 790, dtype: int64


Column: number_diagnoses
9     49474
5     11393
8     10616
7     10393
6     10161
4      5537
3      2835
2      1023
1       219
16       45
10       17
13       16
11       11
15       10
12        9
14        7
Name: number_diagnoses, dtype: int64


Column: max_glu_serum
None    96420
Norm     2597
>200     1485
>300     1264
```

```
Name: max_glu_serum, dtype: int64


Column: A1Cresult
None    84748
>8       8216
Norm     4990
>7       3812
Name: A1Cresult, dtype: int64


Column: metformin
No        81778
Steady    18346
Up         1067
Down        575
Name: metformin, dtype: int64


Column: repaglinide
No        100227
Steady      1384
Up           110
Down          45
Name: repaglinide, dtype: int64


Column: nateglinide
No        101063
Steady       668
Up            24
Down          11
Name: nateglinide, dtype: int64


Column: chlorpropamide
No        101680
Steady        79
Up             6
Down           1
Name: chlorpropamide, dtype: int64


Column: glimepiride
No        96575
Steady     4670
Up          327
Down        194
Name: glimepiride, dtype: int64


Column: acetohexamide
No        101765
Steady         1
Name: acetohexamide, dtype: int64


Column: glipizide
No        89080
Steady    11356
Up          770
Down        560
Name: glipizide, dtype: int64


Column: glyburide
```

```
No        91116
Steady     9274
Up          812
Down        564
Name: glyburide, dtype: int64


Column: tolbutamide
No        101743
Steady        23
Name: tolbutamide, dtype: int64


Column: pioglitazone
No        94438
Steady     6976
Up          234
Down        118
Name: pioglitazone, dtype: int64


Column: rosiglitazone
No        95401
Steady     6100
Up          178
Down         87
Name: rosiglitazone, dtype: int64


Column: acarbose
No        101458
Steady       295
Up            10
Down           3
Name: acarbose, dtype: int64


Column: miglitol
No        101728
Steady        31
Down           5
Up             2
Name: miglitol, dtype: int64


Column: troglitazone
No        101763
Steady         3
Name: troglitazone, dtype: int64


Column: tolazamide
No        101727
Steady        38
Up             1
Name: tolazamide, dtype: int64


Column: examide
No    101766
Name: examide, dtype: int64


Column: citoglipton
No    101766
Name: citoglipton, dtype: int64
```

```
Column: insulin
No        47383
Steady    30849
Down      12218
Up        11316
Name: insulin, dtype: int64


Column: glyburide-metformin
No        101060
Steady       692
Up             8
Down           6
Name: glyburide-metformin, dtype: int64


Column: glipizide-metformin
No        101753
Steady        13
Name: glipizide-metformin, dtype: int64


Column: glimepiride-pioglitazone
No        101765
Steady         1
Name: glimepiride-pioglitazone, dtype: int64


Column: metformin-rosiglitazone
No        101764
Steady         2
Name: metformin-rosiglitazone, dtype: int64


Column: metformin-pioglitazone
No        101765
Steady         1
Name: metformin-pioglitazone, dtype: int64


Column: change
No    54755
Ch    47011
Name: change, dtype: int64


Column: diabetesMed
Yes    78363
No     23403
Name: diabetesMed, dtype: int64


Column: readmitted
NO     54864
>30    35545
<30    11357
Name: readmitted, dtype: int64


Column: discharge_desc
Emergency        53990
Elective         18869
Urgent           18480
Not Available     4785
Not Mapped          320
Trauma Center        21
Newborn              10
Name: discharge_desc, dtype: int64


Column: admission_source_desc
Discharged to home
                     60234
Discharged/transferred to SNF
                     13954
Discharged/transferred to home with home health service
                     12902
Discharged/transferred to another short term hospital
                      2128
Discharged/transferred to another rehab fac including rehab units of a hospital .
                      1993
Expired
                      1642
Discharged/transferred to another type of inpatient care institution
                      1184
Not Mapped
                       989
Discharged/transferred to ICF
                       815
Left AMA
                       623
Discharged/transferred to a long term care hospital.
                       412
Hospice / home
                       399
Hospice / medical facility
                       372
Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part u
nit of a hospital      139
Discharged/transferred to home under care of Home IV provider
                       108
Discharged/transferred within this institution to Medicare approved swing bed
                        63
Discharged/transferred to a nursing facility certified under Medicaid but not certified
under Medicare.         48
Admitted as an inpatient to this hospital
                        21
Discharged/transferred/referred to this institution for outpatient services
                        14
Discharged/transferred/referred another institution for outpatient services
                        11
Expired at home. Medicaid only, hospice.
                         8
Neonate discharged to another hospital for neonatal aftercare
                         6
Discharged/transferred to a federal health care facility.
                         5
Still patient or expected to return for outpatient services
                         3
Expired in a medical facility. Medicaid only, hospice.
                         2
Name: admission_source_desc, dtype: int64


Column: admission_desc
 Emergency Room                                                 57494
 Physician Referral                                             29565
Transfer from a hospital                                         3187
 Transfer from another health care facility                      2264
Clinic Referral                                                  1104
```

```
 Transfer from a Skilled Nursing Facility (SNF)               855
HMO Referral                                                  187
 Not Mapped                                                   161
 Not Available                                                125
 Court/Law Enforcement                                         16
 Transfer from hospital inpt/same fac reslt in a sep claim     12
 Transfer from critial access hospital                          8
 Extramural Birth                                               2
Normal Delivery                                                 2
 Transfer from Ambulatory Surgery Center                        2
 Sick Baby                                                      1
Name: admission_desc, dtype: int64


Column: binary_readmitted
0    90409
1    11357
Name: binary_readmitted, dtype: int64
```

Features that may be redundant or contain redundant values

- citoglipton - single value
- examide - single value
- Gender - remove the unknown gender
- metformin-pioglitazone - binary
- metformin-rosiglitazone - binary
- glimepiride-pioglitazone - binary
- glipizide-metformin - binary
- tolazamide - binary
- troglitazone - binary
- tolbutamide - binary
- acetohexamide - binary

In [15]:
```
df = df[df['gender']!='Unknown/Invalid'] # taking out the 'unknown/invalid' gender obser

# dropping observations where the values were insignificant
df.drop(['citoglipton','examide','metformin-pioglitazone',
        'metformin-rosiglitazone','glimepiride-pioglitazone',
        'glipizide-metformin','tolazamide','troglitazone',
        'tolbutamide','acetohexamide', 'readmitted'], axis = 1, inplace = True)
```

In [16]:
```
# some patients have been back multiple times. We are trying to determine readmittance,
# when patients have come back by keeping the first return visit rather than every singl
df = df.drop_duplicates(subset = 'patient_nbr', keep = 'first')

print(df.shape) # looking at how the shape has been reduced thus far
```

```
(71515, 43)
```

Hidden N/A Values

- **race:** contain ?
  - Upon further investigation (visually and with cross tabulations) we have determined this data to be MCAR (Missing Completely at Random)

- We will proceed with imputation of the most frequent category

- **weight:** > 90% missing values
  - Upon further investigation (visually and with cross tabulations) we have determined this data to be MCAR (Missing Completely at Random)
  - We will proceed with the removal of this feature.

- **payer_code:** contain ?
  - Upon Futher Investigation (visually and with cross tabulations) we have determined that this data is MCAR.
  - We will not impute these values, or remove them as there is a significant amount missing, instead we will re-label the missing values and include them as a feature in our data.

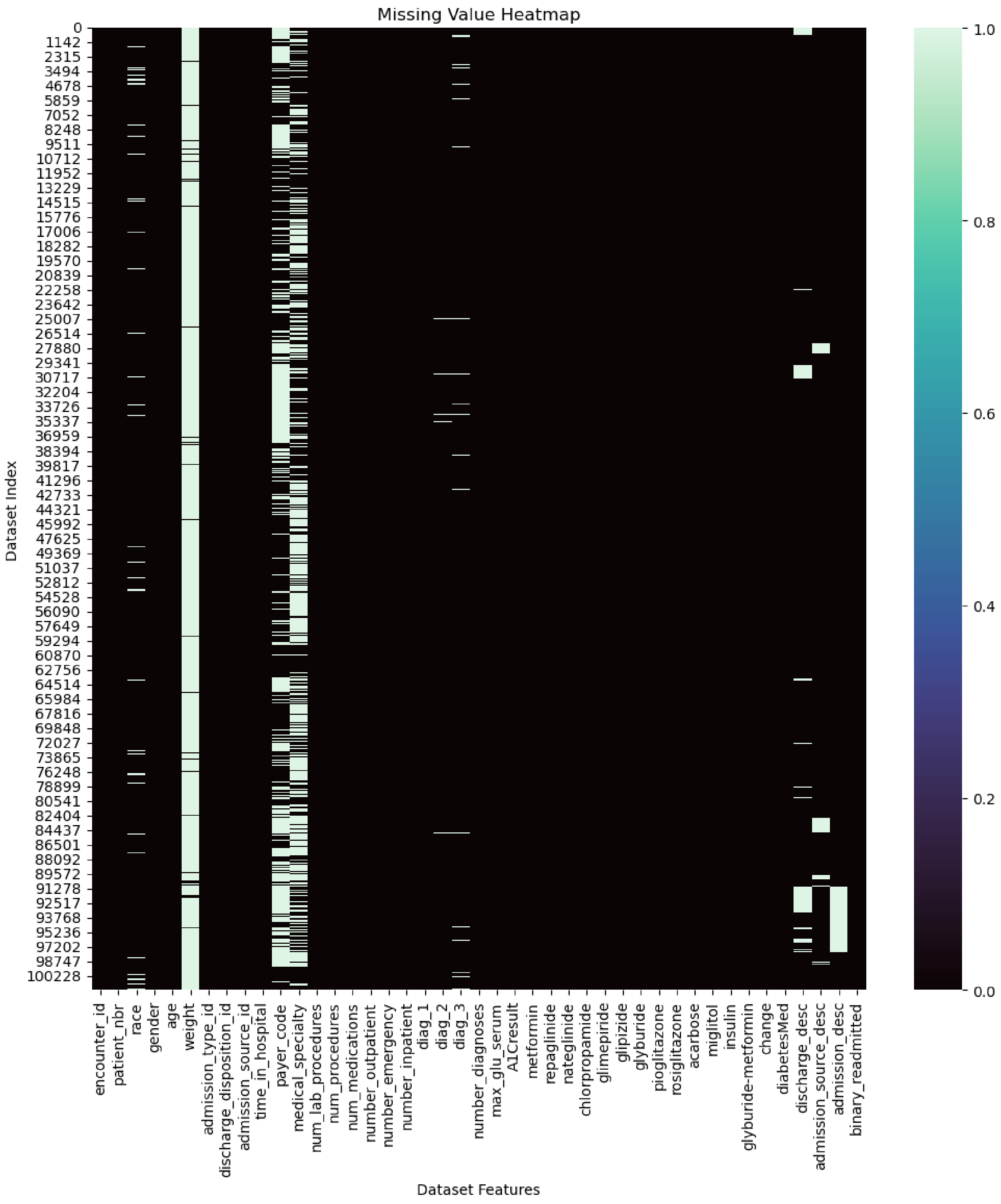
- **medical_specialty:** contain ?
  - There is seemingly no relationship between the values in our data and the missingness of our medical specialty values.
  - We will proceed with imputing the missing values as their own value of "other".

- **diag-1,2,3:** all contain '?'
  - There is an inconsequential amount of missing data in the Diagnosis 1,2,3. We will proceed with removing the null values.

In [17]:
```
# we are replacing '?' with NaN so that we can visually see where the missing data is ha
df['race'].replace('?', np.nan, inplace = True)
df['weight'].replace('?', np.nan, inplace =True)
df['payer_code'].replace('?', np.nan, inplace = True)
df['medical_specialty'].replace('?', np.nan, inplace = True)
df['diag_1'].replace('?', np.nan, inplace = True)
df['diag_2'].replace('?', np.nan, inplace = True)
df['diag_3'].replace('?', np.nan, inplace = True)
```

Heatmap for Missing Values

In [18]:
```
plt.figure(figsize=(12,12))
sns.heatmap(df.isnull(), cbar = True, cmap = 'mako')
plt.title('Missing Value Heatmap')
plt.xlabel('Dataset Features')
plt.ylabel('Dataset Index')
plt.show()
```



Discharge Desc. & Admission Desc.

- The heatmap and cross tabulations show that there is some relationship between the missing values in Discharge Desc. & Admission Desc.
- This relationship somewhats makes sense intuitively becuase if there is missing information regarding Admission, it would make sense that Discharge missing information could be related.
- Ultimately this relationship is only a partial correlation and isn't quite considered MNAR (missing not at random). We will proceed with filling the NaN values for these as "other".

```
In [19]: cont_table = pd.crosstab(df['discharge_desc'].isna(), df['admission_desc'].isna())
         cont_table
```

Out[19]:

admission_desc	False	True
discharge_desc		
False	64622	2307
True	2004	2582

```
In [20]: df[(df['discharge_desc'] == 'NAN') & (df['admission_type_id'] == 6)]
```

Out[20]:

encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_so

Payer Code Vs Medical Specialty

- Looking at the heatmap above it appears that there are some instances where payer code and medical specialty flip between missing observations.
- to further explore if maybe the missingness can be determined by the missingness of the other we will create a contingency table.

```
In [21]: cont_table = pd.crosstab(df['payer_code'].isnull(), df['medical_specialty'].isnull())
         cont_table
```

Out[21]:

medical_specialty	False	True
payer_code		
False	19288	21474
True	18202	12551

From this table, we can observe that when 'payer_code' is not missing (False), 'medical_specialty' can be either not missing (False) or missing (True). Similarly, when 'payer_code' is missing (True), 'medical_specialty' can also be either not missing (False) or missing (True).

There doesn't seem to be a strong relationship indicating that when one feature is missing, the other is not missing, or vice versa. In other words, the missingness of 'payer_code' and 'medical_specialty' does not appear to be dependent on each other.

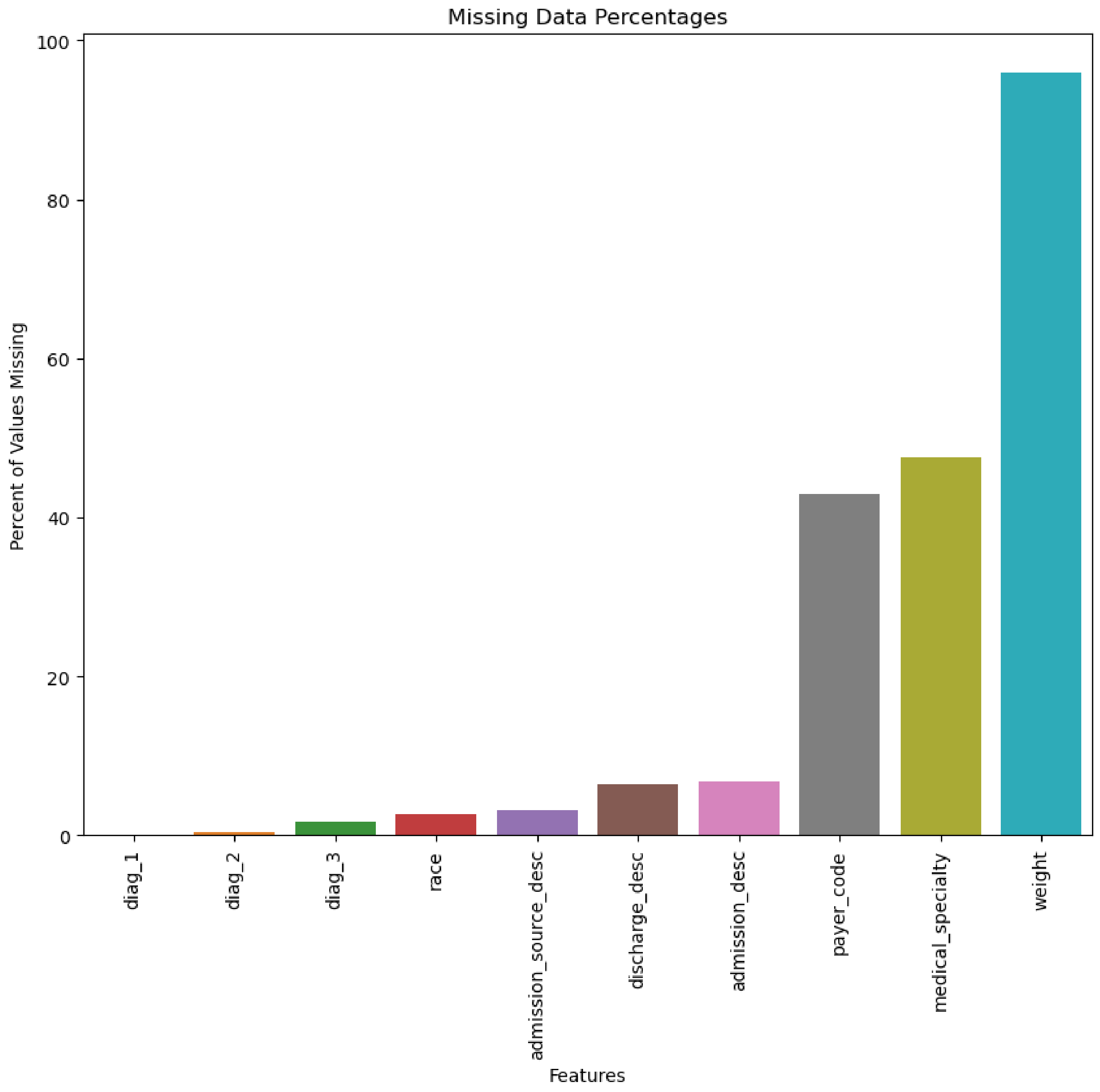
Define Percent Missing

```
In [22]: def percent_missing(df):
             percent_nan = 100*df.isnull().sum()/len(df)
             percent_nan = percent_nan[percent_nan > 0].sort_values()
             return(percent_nan)
         percent_nan = percent_missing(df)
         print(round(percent_nan,2))

diag_1                     0.02
diag_2                     0.41
diag_3                     1.68
race                       2.67
admission_source_desc      3.22
```

```
discharge_desc            6.41
admission_desc            6.84
payer_code               43.00
medical_specialty        47.58
weight                   96.02
dtype: float64
```

Percent Missing Bar Plot

In [23]:
```
plt.figure(figsize = (10,8))
sns.barplot(x= percent_nan.index, y = percent_nan)
plt.title('Missing Data Percentages')
plt.xlabel('Features')
plt.ylabel('Percent of Values Missing')
plt.xticks(rotation = 90)
plt.show()
```

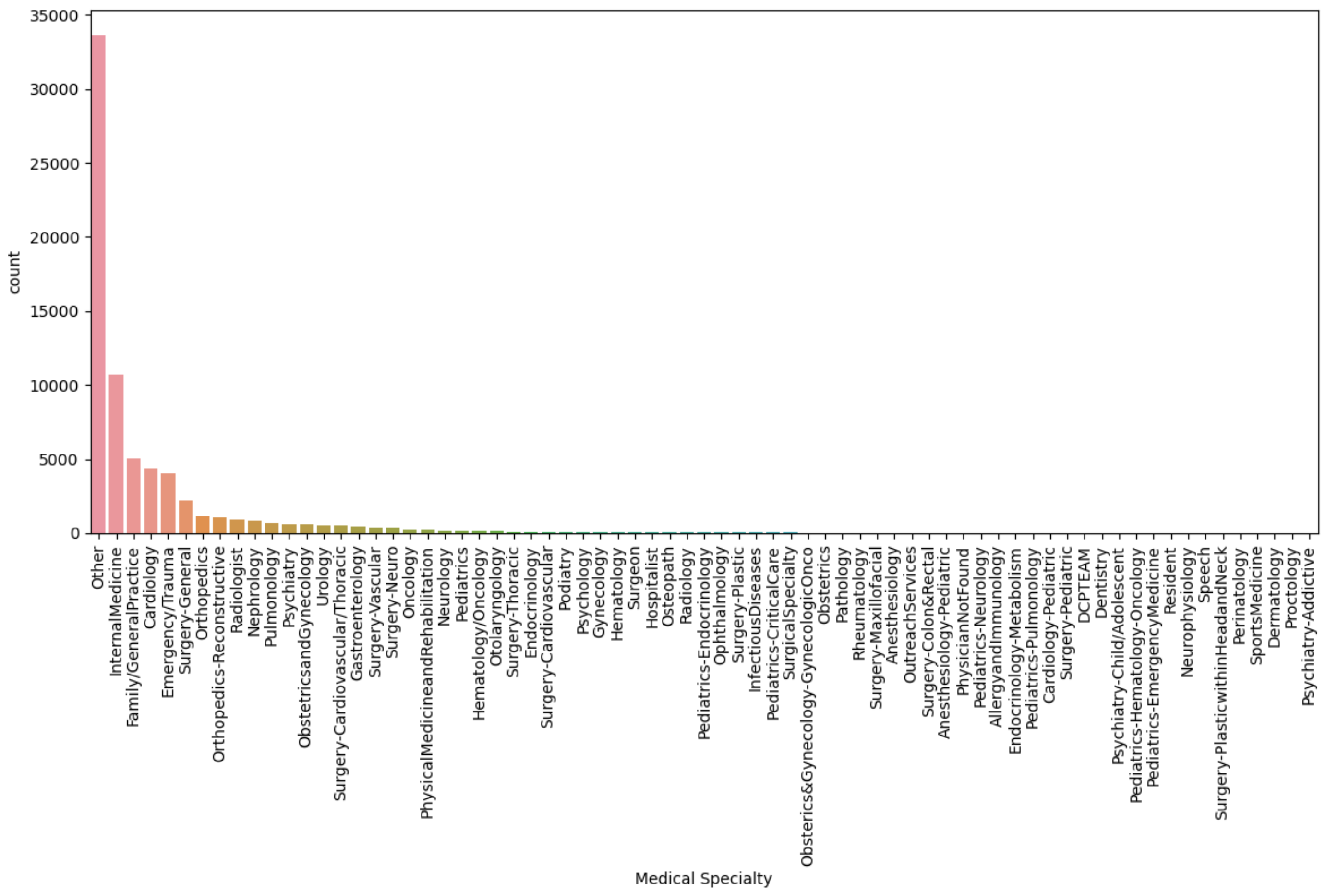


In [24]:
```
# filling NaN with "NAN" so that we can see in our data where these values are occuring.
# this gives us a means of representing the missing data which is useful in investigatio
```

```
#impute race missing values with mode
mode_race = df['race'].mode()[0]
df['race'].fillna(mode_race, inplace = True)

# drop the weight column
df.drop('weight', inplace = True, axis = 1)

# payer_code impute with it's own level to maintain information in data.
df['payer_code'].fillna('Not Specified', inplace = True)


df['medical_specialty'].fillna('Other', inplace = True)

# fill NaN with 'other'
df['admission_desc'].fillna('Other', inplace = True)
df['admission_source_desc'].fillna('Other', inplace = True)
df['discharge_desc'].fillna('Other', inplace = True)

# remove NaN from Diag 1-3
df.dropna(subset=['diag_1','diag_2','diag_3'], inplace = True)
```

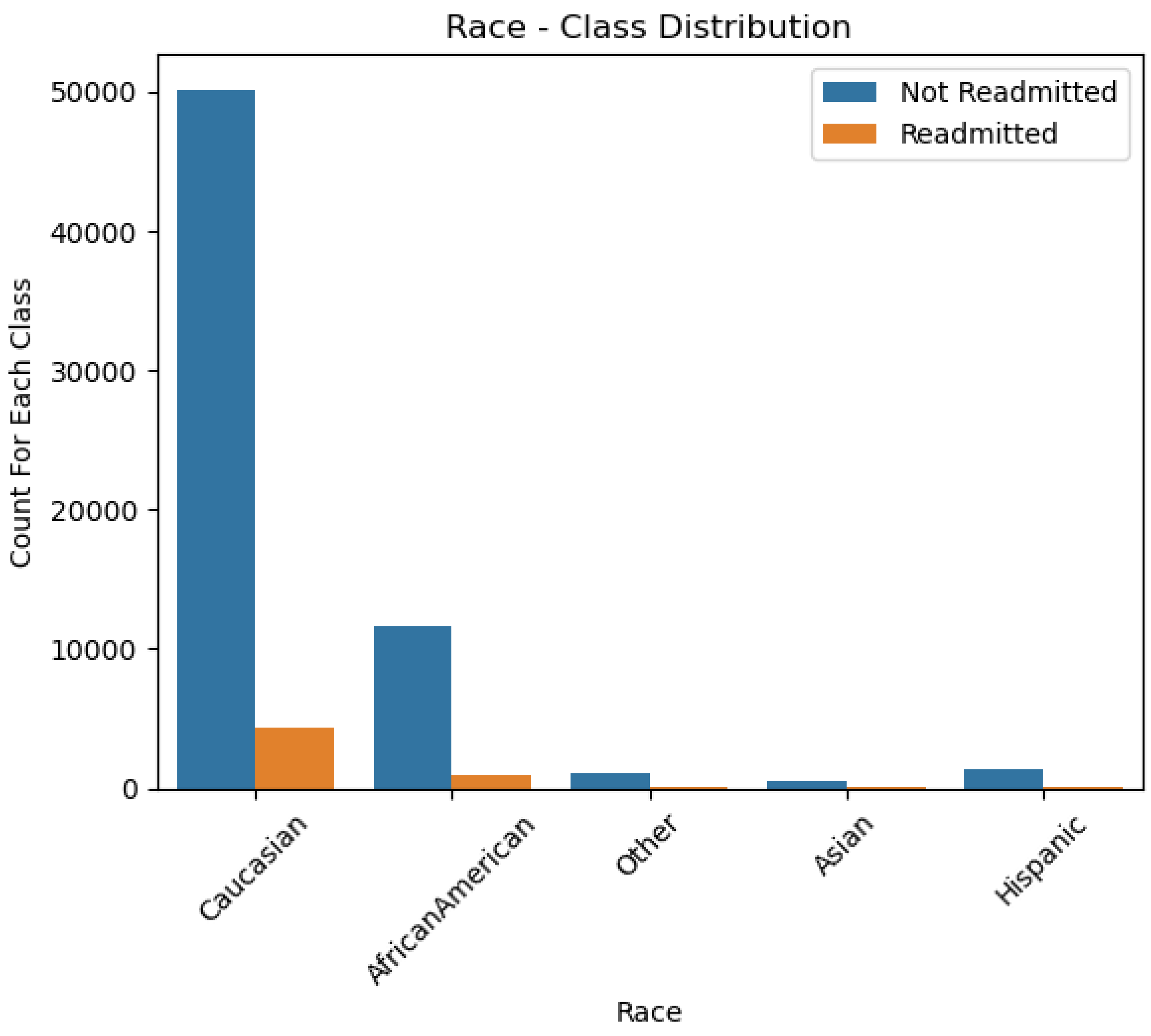
Medical Specialty - With use of "other" value to represent the NaN

In [25]:
```
plt.figure(figsize=(14,6))
sns.countplot(df, x = 'medical_specialty', order = df['medical_specialty'].value_counts(
plt.xticks(rotation = 90)
plt.xlabel('Medical Specialty')
plt.show()
```



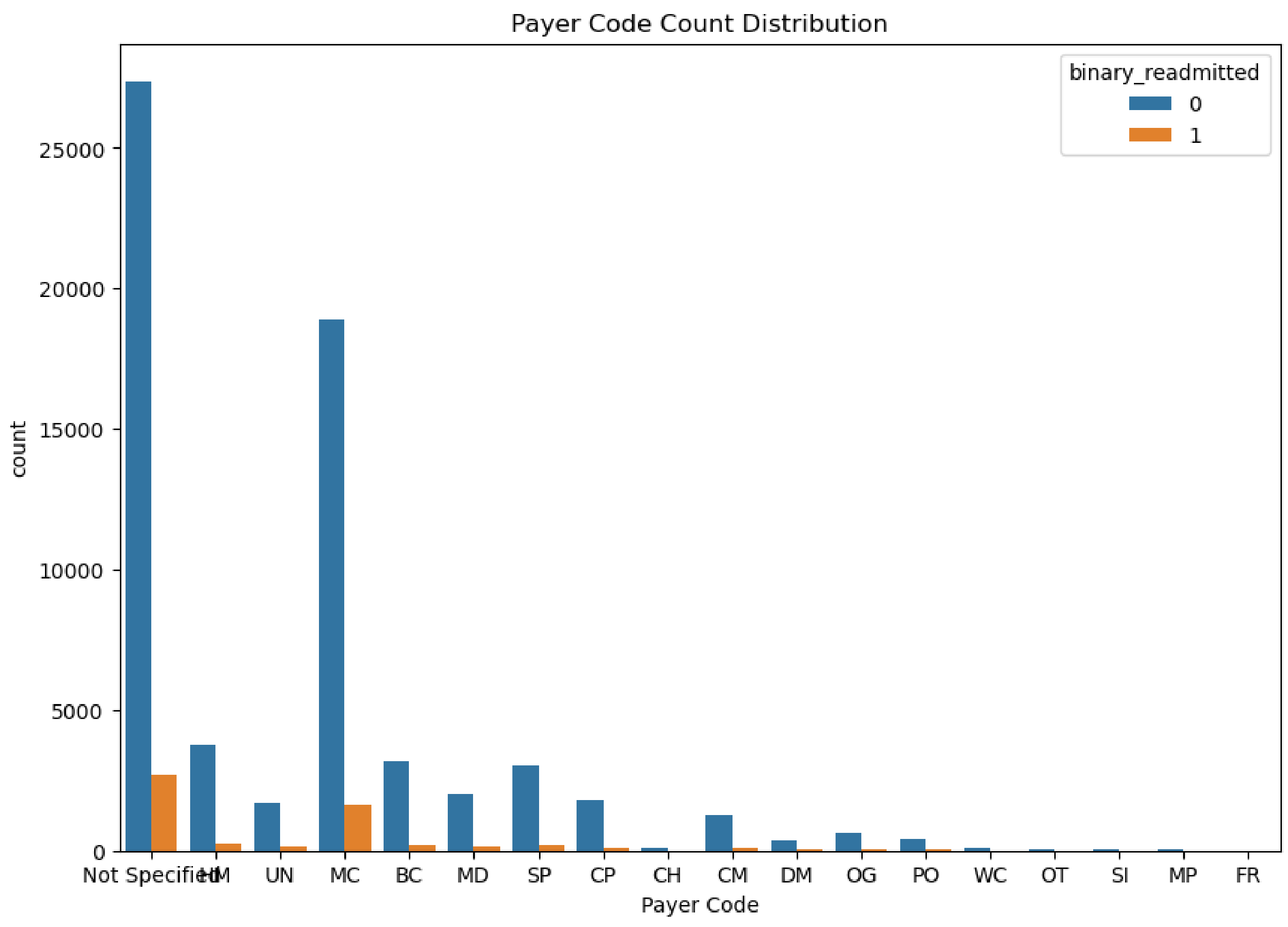
Race - With imputed Mode

```
In [26]: legend_labels = ['Not Readmitted', 'Readmitted']

         ax = sns.countplot(df, x = 'race', hue = 'binary_readmitted')
         ax.legend(legend_labels)
         plt.xticks(rotation = 45)
         plt.title('Race - Class Distribution')
         plt.xlabel('Race')
         plt.ylabel('Count For Each Class')
         plt.show()
```



Payer Code - Relabeld NaN as Not Specified

```
In [27]: plt.figure(figsize=(10,7))
         sns.countplot(df, x = 'payer_code', hue='binary_readmitted')
         plt.title('Payer Code Count Distribution')
         plt.xlabel('Payer Code')
         plt.show()
```



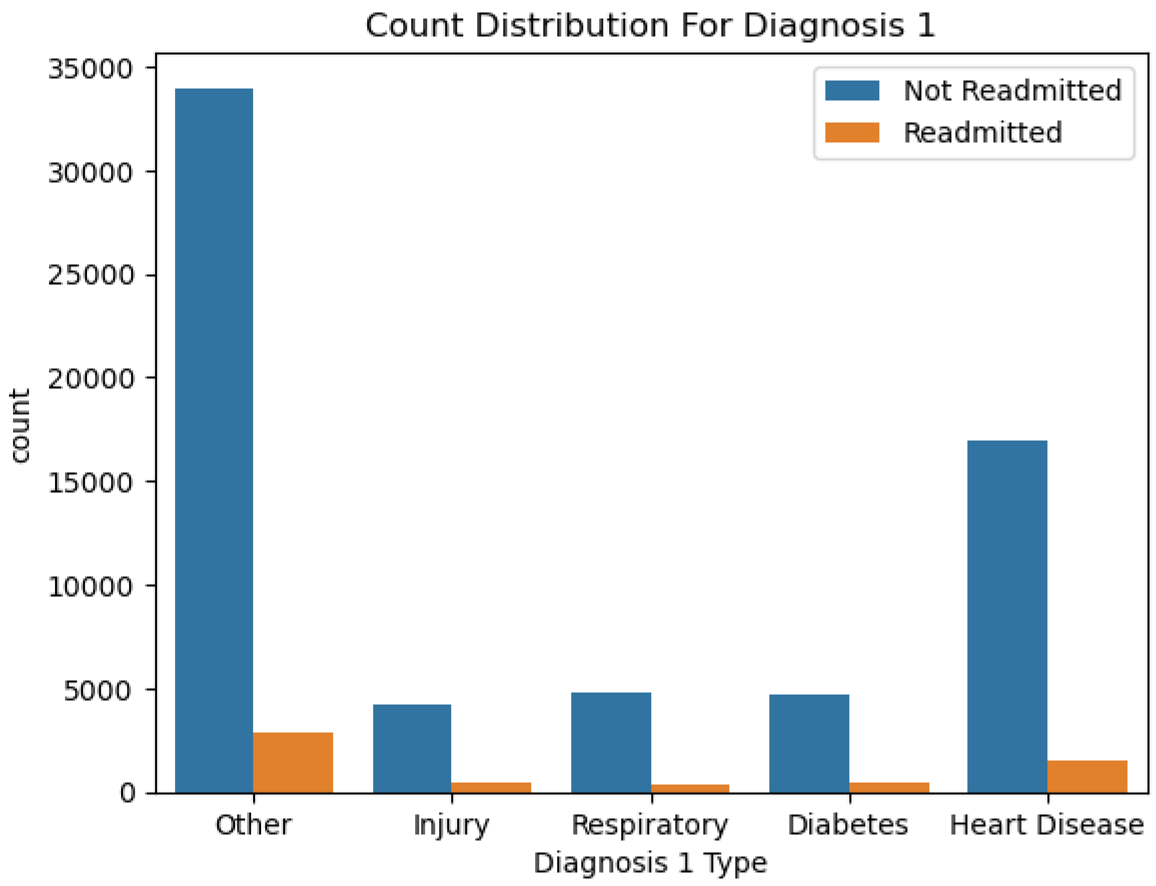
Click Here For information on Diagnosis Codes

In [28]:
```
# diag 1
custom_categories = {
    'Diabetes': ['250', '250.02', '250.03', '250.04', '250.1', '250.11', '250.12', '250.
    'Heart Disease': ['410', '411', '412', '413', '414', '415', '416', '417', '418', '41
    'Respiratory': ['460', '461', '462', '463', '464', '465', '466', '470', '471', '472'
    'Injury': ['800', '801', '802', '803', '804', '805', '806', '807', '808', '809', '81
    'Other': ['E909', 'V25', 'V26', 'V43', 'V45', 'V51', 'V53', 'V54', 'V55', 'V56', 'V5
}

# Create a new column 'diag_1_new' and set it to 'Other' initially
df['diag_1_new'] = 'Other'

# Map the values in 'diag_1' to custom categories
for category, codes in custom_categories.items():
    df.loc[df['diag_1'].isin(codes), 'diag_1_new'] = category

# Display the DataFrame with the new 'diag_1_new' feature
print(df[['diag_1', 'diag_1_new']])
```

```
        diag_1     diag_1_new
1          440          Other
2          997         Injury
3          486    Respiratory
4       250.03       Diabetes
5          414  Heart Disease
...        ...            ...
```

```
101752      998          Injury
101753      996          Injury
101754      V57           Other
101755      V57           Other
101762      V57           Other

[70256 rows x 2 columns]
```

In [29]:
```python
legend_label = ['Not Readmitted','Readmitted']
ax = sns.countplot(data = df, x = 'diag_1_new', hue = 'binary_readmitted')
ax.legend(legend_label)
plt.title('Count Distribution For Diagnosis 1')
plt.xlabel('Diagnosis 1 Type')
plt.show()
```

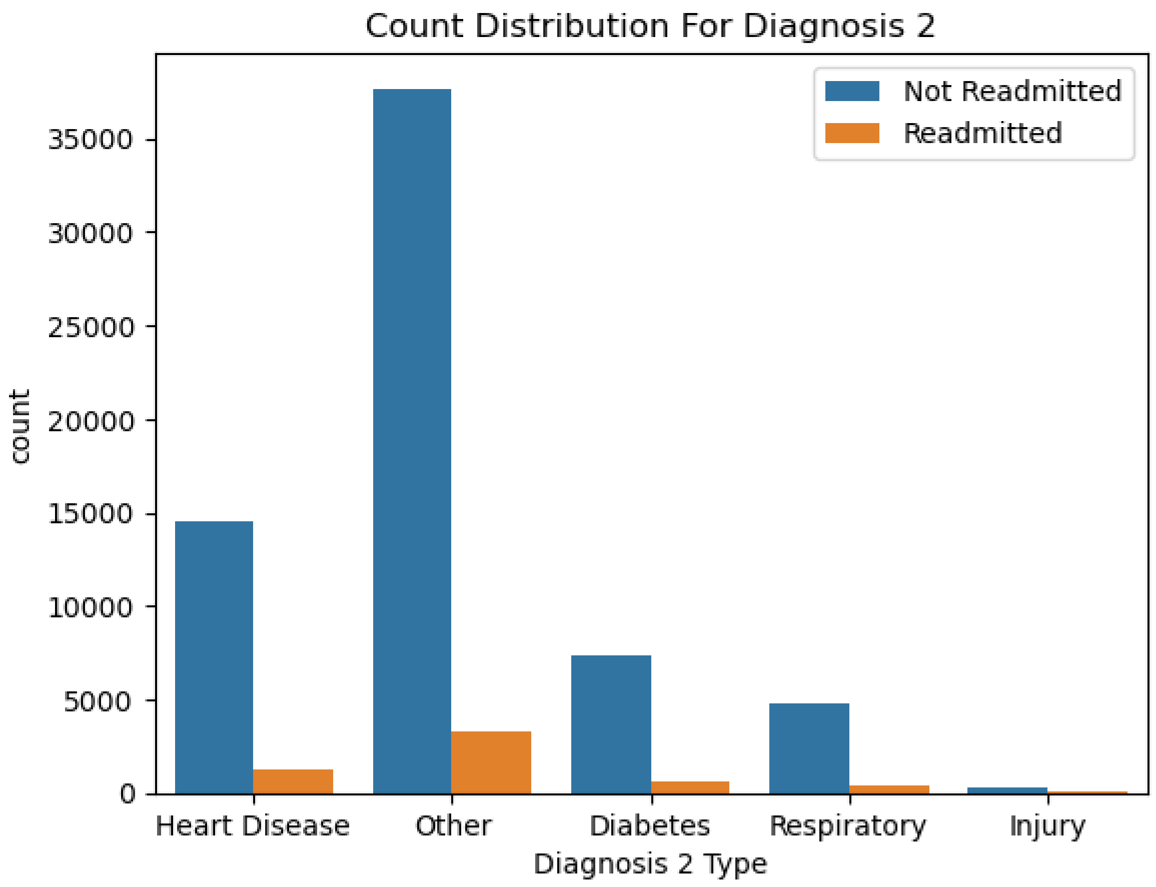


In [30]:
```python
# diag 2
custom_categories_diag_2 = {
    'Diabetes': ['250', '250.02', '250.03', '250.04', '250.1', '250.11', '250.12', '250.
    'Heart Disease': ['410', '411', '412', '413', '414', '415', '416', '417', '418', '41
    'Respiratory': ['460', '461', '462', '463', '464', '465', '466', '470', '471', '472'
    'Injury': ['800', '801', '802', '803', '804', '805', '806', '807', '808', '809', '81
    'Other': ['?', 'V', 'E']
}

# Create a new column 'diag_2_new' and set it to 'Other' initially
df['diag_2_new'] = 'Other'

# Map the values in 'diag_2' to custom categories
for category, codes in custom_categories_diag_2.items():
    df.loc[df['diag_2'].isin(codes), 'diag_2_new'] = category

# Display the DataFrame with the new 'diag_2_new' feature
print(df[['diag_2', 'diag_2_new']])
```

```
       diag_2     diag_2_new
1         413  Heart Disease
2           8          Other
```

```
3          250       Diabetes
4          401          Other
5          340          Other
...        ...            ...
101752     786          Other
101753    E878          Other
101754     788          Other
101755     348          Other
101762     599          Other

[70256 rows x 2 columns]
```

In [31]:
```
legend_label = ['Not Readmitted','Readmitted']
ax = sns.countplot(data = df, x = 'diag_2_new', hue = 'binary_readmitted')
ax.legend(legend_label)
plt.title('Count Distribution For Diagnosis 2')
plt.xlabel('Diagnosis 2 Type')
plt.show()
```

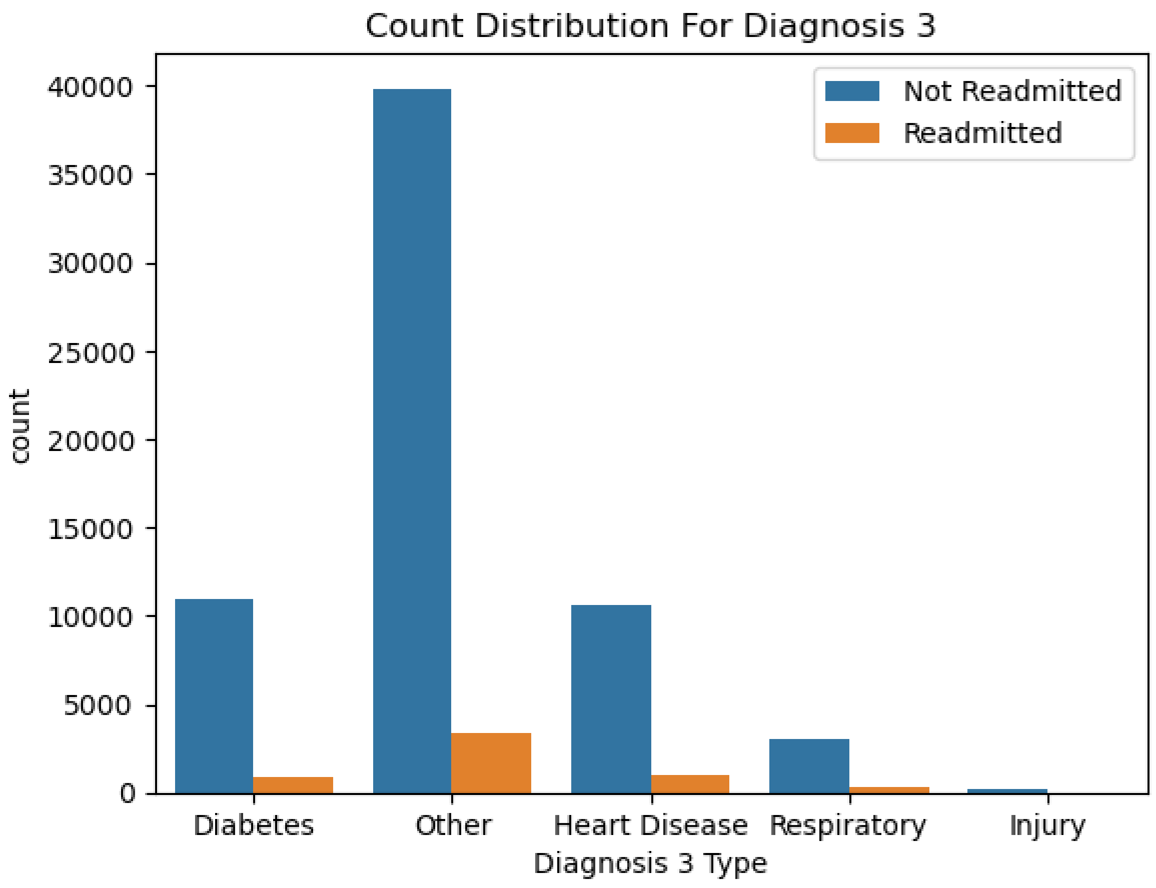


In [32]:
```
# diag 3
custom_categories_diag_3 = {
    'Diabetes': ['250', '250.02', '250.03', '250.04', '250.1', '250.11', '250.12', '250.
    'Heart Disease': ['410', '411', '412', '413', '414', '415', '416', '417', '418', '41
    'Respiratory': ['460', '461', '462', '463', '464', '465', '466', '470', '471', '472'
    'Injury': ['800', '801', '802', '803', '804', '805', '806', '807', '808', '809', '81
    'Other': ['NAN', 'E888', 'E878', 'V', 'V43', 'V70']
}

# Create a new column 'diag_3_new' and set it to 'Other' initially
df['diag_3_new'] = 'Other'

# Map the values in 'diag_3' to custom categories
for category, codes in custom_categories_diag_3.items():
    df.loc[df['diag_3'].isin(codes), 'diag_3_new'] = category
```

```
# Display the DataFrame with the new 'diag_3_new' feature
print(df[['diag_3', 'diag_3_new']])
```

```
        diag_3     diag_3_new
1       250.52       Diabetes
2          730          Other
3          427  Heart Disease
4          276          Other
5          401          Other
...        ...            ...
101752     584          Other
101753     401          Other
101754     349          Other
101755     V58          Other
101762     V43          Other

[70256 rows x 2 columns]
```

In [33]:
```
legend_label = ['Not Readmitted','Readmitted']
ax = sns.countplot(data = df, x = 'diag_3_new', hue = 'binary_readmitted')
ax.legend(legend_label)
plt.title('Count Distribution For Diagnosis 3')
plt.xlabel('Diagnosis 3 Type')
plt.show()
```



Drop Columns no longer needed, or which pose redundant information

In [34]:
```
df.drop(['encounter_id','patient_nbr','admission_type_id','discharge_disposition_id',
        'admission_source_id','diag_1','diag_2','diag_3'], inplace = True, axis = 1)
```

Cross Tabulation Loop

In [35]:
```python
column_features = ['race', 'gender', 'age', 'time_in_hospital', 'payer_code',
    'medical_specialty', 'num_lab_procedures', 'num_procedures',
    'num_medications', 'number_outpatient', 'number_emergency',
    'number_inpatient', 'number_diagnoses', 'max_glu_serum', 'A1Cresult',
    'metformin', 'repaglinide', 'nateglinide', 'chlorpropamide',
    'glimepiride', 'glipizide', 'glyburide', 'pioglitazone',
    'rosiglitazone', 'acarbose', 'miglitol', 'insulin',
    'glyburide-metformin', 'change', 'diabetesMed', 'discharge_desc',
    'admission_source_desc', 'admission_desc', 'binary_readmitted',
    'diag_1_new', 'diag_2_new', 'diag_3_new']

for feature in column_features:
    cross_tab = pd.crosstab(df['race'], df[feature])
    print(f"Cross-tabulation for race and {feature}:")
    print(cross_tab)
    print("\n")
```

```
Cross-tabulation for race and race:
race             AfricanAmerican  Asian  Caucasian  Hispanic  Other
race
AfricanAmerican            12625      0          0         0      0
Asian                          0    489          0         0      0
Caucasian                      0      0      54520         0      0
Hispanic                       0      0          0      1467      0
Other                          0      0          0         0   1155


Cross-tabulation for race and gender:
gender           Female   Male
race
AfricanAmerican    7685   4940
Asian               245    244
Caucasian         28086  26434
Hispanic            792    675
Other               583    572


Cross-tabulation for race and age:
age              [0-10)  [10-20)  [20-30)  [30-40)  [40-50)  [50-60)  [60-70)  \
race
AfricanAmerican       5      104      342      799     1813     2848     2820
Asian                 1        0        5       10       45       90      119
Caucasian            54      216      601     1561     4444     8751    12201
Hispanic              1       16       47      135      223      306      338
Other                 2        6       18       59      125      242      315

age              [70-80)  [80-90)  [90-100)
race
AfricanAmerican     2481     1172       241
Asian                140       69        10
Caucasian          14916    10115      1661
Hispanic             280      110        11
Other                255      115        18


Cross-tabulation for race and time_in_hospital:
time_in_hospital    1     2     3     4     5     6     7     8     9     10  \
race
AfricanAmerican   1805  2225  2128  1692  1255   937   719   551   347   272
Asian              103   102    78    62    39    33    16    22    10     6
Caucasian         8127  9254  9697  7361  5233  3971  3087  2247  1543  1189
Hispanic           230   305   266   181   138   104    63    55    41    29
Other              211   199   207   152    86    61    59    57    32    31

time_in_hospital   11    12    13    14
```

```
race
AfricanAmerican   250  177  141  126
Asian               5    0    9    4
Caucasian         950  729  617  515
Hispanic           14   17   10   14
Other              20   16   14   10


Cross-tabulation for race and payer_code:
payer_code          BC  CH   CM    CP   DM  FR    HM     MC    MD  MP  \
race
AfricanAmerican    616  22  333   265  124   0   598   2796   666   8
Asian               14   1    7     5    3   0    25     83    31   0
Caucasian         2653  93  975  1561  205   1  3272  17275  1266  33
Hispanic            58   1   21    48   18   0    66    206   133   0
Other               35   1   17    17   16   0    62    188    74   0

payer_code       Not Specified   OG  OT   PO  SI    SP    UN  WC
race
AfricanAmerican           6269  148  25   96   3   335   300  21
Asian                      229   15   1   10   0    39    26   0
Caucasian                22249  382  39  324  29  2652  1423  88
Hispanic                   742   41   1    9   1    85    32   5
Other                      560   55   0   12   0    83    35   0


Cross-tabulation for race and medical_specialty:
medical_specialty  AllergyandImmunology  Anesthesiology  \
race
AfricanAmerican                       1               1
Asian                                 0               0
Caucasian                             3               7
Hispanic                              2               0
Other                                 0               1

medical_specialty  Anesthesiology-Pediatric  Cardiology  Cardiology-Pediatric  \
race
AfricanAmerican                           3         627                     2
Asian                                     0          13                     0
Caucasian                                 6        3604                     3
Hispanic                                  0          48                     0
Other                                     0          81                     0

medical_specialty  DCPTEAM  Dentistry  Dermatology  Emergency/Trauma  \
race
AfricanAmerican          0          2            0               412
Asian                    0          0            0                87
Caucasian                4          2            1              3245
Hispanic                 1          0            0               144
Other                    0          0            0               161

medical_specialty  Endocrinology  Endocrinology-Metabolism  \
race
AfricanAmerican               22                         0
Asian                          0                         0
Caucasian                     70                         6
Hispanic                       1                         0
Other                          0                         0

medical_specialty  Family/GeneralPractice  Gastroenterology  Gynecology  \
race
AfricanAmerican                      1071                64           6
Asian                                  33                 1           1
Caucasian                            3699               369          38
Hispanic                              175                 4           2
Other                                  68                 9           3
```

```
medical_specialty  Hematology  Hematology/Oncology  Hospitalist  \
race
AfricanAmerican             3                   15            2
Asian                       1                    1            0
Caucasian                  39                  114           38
Hispanic                    0                    0            2
Other                       5                    0            0

medical_specialty  InfectiousDiseases  InternalMedicine  Nephrology  \
race
AfricanAmerican                     3              2837         332
Asian                               2               103           2
Caucasian                          24              7369         504
Hispanic                            1               252           8
Other                               0               141           7

medical_specialty  Neurology  Neurophysiology  \
race
AfricanAmerican           43                0
Asian                      1                0
Caucasian                126                1
Hispanic                   3                0
Other                      6                0

medical_specialty  Obsterics&Gynecology-GynecologicOnco  Obstetrics  \
race
AfricanAmerican                                       6           3
Asian                                                 0           0
Caucasian                                            12          12
Hispanic                                              0           1
Other                                                 1           1

medical_specialty  ObstetricsandGynecology  Oncology  Ophthalmology  \
race
AfricanAmerican                        156        61              6
Asian                                    5         0              0
Caucasian                              396       169             26
Hispanic                                33         4              0
Other                                    5         4              0

medical_specialty  Orthopedics  Orthopedics-Reconstructive  Osteopath  Other  \
race
AfricanAmerican            131                         157          1   5382
Asian                        5                           3          1    160
Caucasian                  980                         836         35  26963
Hispanic                    17                          17          0    633
Other                        8                           9          0    530

medical_specialty  Otolaryngology  OutreachServices  Pathology  Pediatrics  \
race
AfricanAmerican                26                 0          3          34
Asian                           1                 0          0           1
Caucasian                      71                 9          8         124
Hispanic                        4                 0          0           2
Other                           2                 0          1           3

medical_specialty  Pediatrics-CriticalCare  Pediatrics-EmergencyMedicine  \
race
AfricanAmerican                          7                             0
Asian                                    1                             0
Caucasian                               19                             2
Hispanic                                 0                             0
Other                                    2                             0

medical_specialty  Pediatrics-Endocrinology  Pediatrics-Hematology-Oncology  \
race
AfricanAmerican                         10                              1
Asian                                    0                              0
Caucasian                               23                              2
Hispanic                                 0                              0
Other                                    1                              0

medical_specialty  Pediatrics-Neurology  Pediatrics-Pulmonology  Perinatology  \
race
AfricanAmerican                       0                       1             1
Asian                                 0                       0             0
Caucasian                             6                       5             0
Hispanic                              0                       0             0
Other                                 0                       0             0

medical_specialty  PhysicalMedicineandRehabilitation  PhysicianNotFound  \
race
AfricanAmerican                                   41                  1
Asian                                              0                  2
Caucasian                                        161                  5
Hispanic                                           1                  0
Other                                              5                  0

medical_specialty  Podiatry  Proctology  Psychiatry  Psychiatry-Addictive  \
race
AfricanAmerican          11           0         152                     0
Asian                     1           0           1                     0
Caucasian                64           1         462                     1
Hispanic                  0           0          10                     0
Other                     1           0           6                     0

medical_specialty  Psychiatry-Child/Adolescent  Psychology  Pulmonology  \
race
AfricanAmerican                              2          13          123
Asian                                        0           0           20
Caucasian                                    2          55          505
Hispanic                                     0           2           21
Other                                        0           3            9

medical_specialty  Radiologist  Radiology  Resident  Rheumatology  Speech  \
race
AfricanAmerican             34         11         0             5       0
Asian                        5          0         0             1       0
Caucasian                  880         26         0             4       0
Hispanic                     5          0         0             0       1
Other                        7          0         1             0       0

medical_specialty  SportsMedicine  Surgeon  Surgery-Cardiovascular  \
race
AfricanAmerican                 0        1                      15
Asian                           0        1                       0
Caucasian                       1       36                      70
Hispanic                        0        2                       1
Other                           0        2                       1

medical_specialty  Surgery-Cardiovascular/Thoracic  Surgery-Colon&Rectal  \
race
AfricanAmerican                                114                     1
Asian                                            3                     1
Caucasian                                      395                     6
Hispanic                                         0                     0
Other                                            9                     1

medical_specialty  Surgery-General  Surgery-Maxillofacial  Surgery-Neuro  \
race
AfricanAmerican                322                      1             74
```

```
Asian                           28                     0              1
Caucasian                     1777                     9            307
Hispanic                        43                     0              5
Other                           41                     0              1

medical_specialty  Surgery-Pediatric  Surgery-Plastic  \
race
AfricanAmerican                    1                5
Asian                              0                0
Caucasian                          4               25
Hispanic                           0                0
Other                              0                1

medical_specialty  Surgery-PlasticwithinHeadandNeck  Surgery-Thoracic  \
race
AfricanAmerican                                   0                10
Asian                                             0                 2
Caucasian                                         1                78
Hispanic                                          0                 2
Other                                             0                 5

medical_specialty  Surgery-Vascular  SurgicalSpecialty  Urology
race
AfricanAmerican                 144                  6      106
Asian                             0                  0        1
Caucasian                       246                 18      411
Hispanic                          4                  1       15
Other                             2                  2        9


Cross-tabulation for race and num_lab_procedures:
num_lab_procedures   1    2    3    4    5    6    7    8    9   10   11   \
race
AfricanAmerican      311  118   62   33   23   20   33   32  122  103   74
Asian                 27    3    3    2    1    0    3    3    8    6    0
Caucasian           1787  618  415  241  178  169  213  202  537  451  374
Hispanic              69   13   13    6    7    6    3    5    7    6    7
Other                 52   11    4    8    5    1    1    5   10   18   10

num_lab_procedures  12   13   14   15   16   17   18   19   20   21   22   \
race
AfricanAmerican      43   42   36   50   55   54   62   59   80   85   65
Asian                 1    1    6    0    1    3    6    5    5    6    8
Caucasian           315  237  204  238  314  399  409  601  450  413  402
Hispanic              4    8    6    5    8    8    6   14   22   15    7
Other                 5    8    7    3    4    2    6   18   14   11   13

num_lab_procedures  23   24   25   26   27   28   29   30   31   32   33   \
race
AfricanAmerican      70   65   58  143  105  107  196  177  166  147  178
Asian                 7    8    4    7    9    4    8    5    8    6    5
Caucasian           491  436  614  620  468  598  739  727  731  745  712
Hispanic             19   14   18    9    9   21   11   16   18   17   20
Other                 7   13   12   12   11   20   10   20   21   17   11

num_lab_procedures  34   35   36    37    38    39    40    41    42    43   \
race
AfricanAmerican     237  294  278   287   372   322   328   291   272   496
Asian                 9   15   11     8     6     6    11     8     3    15
Caucasian           855  944  996  1091  1130  1079  1108  1073  1077  1295
Hispanic             27   27   20    34    27    27    56    29    27    32
Other                20   15   17    27    15    19    24    18    22    20

num_lab_procedures   44    45    46    47    48    49    50    51   52   53   \
race
AfricanAmerican      365   373   313   300   282   256   225   248  245  230
```

```
Asian                 14     9    11    12     7     8     7    10    5   11
Caucasian           1227  1175  1125  1077  1066  1036  1011  1008  940  941
Hispanic              27    20    23    28    33    35    26    19   23   26
Other                 22    21    31    21    15    28    24    15   19   14

num_lab_procedures   54   55    56   57   58   59   60   61   62   63   64    \
race
AfricanAmerican      208  229   195  197  189  184  190  205  184  167  128
Asian                 12    6     7    4    4   11    6    6    3    4    8
Caucasian           1005  966  1001  941  891  877  821  855  783  753  701
Hispanic              27   33    25   21   21   23   30   21   15   29   15
Other                  8   22    27   20   12   16   18   13   16   19   20

num_lab_procedures  65   66   67   68   69   70   71   72   73   74   75   \
race
AfricanAmerican     164  145  139   93  103   83   85   83   77   75   57
Asian                 7    9    4    4    4    1    4    4    4    3    2
Caucasian           668  657  607  616  565  485  459  417  368  340  300
Hispanic             22   19   19   18   13   15   11   12    6   11   10
Other                12    6    9   11    8   17   14   13   11    5    3

num_lab_procedures  76   77   78   79   80   81   82   83   84   85   86   \
race
AfricanAmerican      40   57   36   37   37   30   27   29   15   13   13
Asian                 3    1    1    1    1    1    0    1    2    0    1
Caucasian           276  231  209  178  158  162  115  134   92   93   70
Hispanic              8   12    3    3    6    3    4    3    1    2    4
Other                 7    8    7    7    8    8    3    2    1    5    4

num_lab_procedures  87   88   89   90   91   92   93   94   95   96   97   \
race
AfricanAmerican      17    7   11   10   12    9   10   11    5    3    6
Asian                 1    0    0    0    0    0    0    0    1    0    2
Caucasian            50   61   40   34   33   22   31   27   33   17   16
Hispanic              1    3    3    2    1    1    1    1    2    1    0
Other                 5    2    2    1    2    0    1    1    0    1    1

num_lab_procedures  98   99   100  101  102  103  104  105  106  107  108  \
race
AfricanAmerican       4    2    3    2    2    1    1    0    0    1    1
Asian                 0    0    0    0    0    0    0    0    0    0    0
Caucasian            15    5    7    7    4    4    0    6    5    0    2
Hispanic              1    0    0    0    0    1    0    0    0    0    0
Other                 0    0    1    0    0    0    0    0    0    0    1

num_lab_procedures  109  111  113  114  118  120  121  132
race
AfricanAmerican       0    1    2    0    0    0    1    1
Asian                 1    0    0    0    0    0    0    0
Caucasian             1    0    0    2    1    1    0    0
Hispanic              0    1    0    0    0    0    0    0
Other                 0    0    0    0    0    0    0    0


Cross-tabulation for race and num_procedures:
num_procedures       0      1     2     3     4     5     6
race
AfricanAmerican   5725   2628  1656  1288   505   416   407
Asian              211     98    75    60    20     7    18
Caucasian        23380  10847  7130  5454  2476  1897  3336
Hispanic           712    297   175   152    54    28    49
Other              470    216   152   128    52    47    90


Cross-tabulation for race and num_medications:
num_medications   1    2    3    4     5     6     7     8     9    10    11   \
race
AfricanAmerican   44   76  158  223   298   416   560   629   702   744   725
Asian              2    6   10   17    20    27    30    34    31    34    37
Caucasian        147  220  432  734  1088  1437  1852  2380  2687  2878  3119
Hispanic           4   19   23   38    36    66    67    86    83    81    91
Other              5   10   18   32    37    35    60    46    68    75    64

num_medications    12    13    14    15    16    17    18    19    20    21  \
race
AfricanAmerican   790   758   689   668   633   581   557   421   366   335
Asian              29    28    24    20    17    12    21    11    10     9
Caucasian        3275  3307  3070  3138  2946  2587  2321  2169  1905  1675
Hispanic           93    98    94    79    73    72    59    41    41    34
Other              64    66    52    72    54    46    44    34    49    30

num_medications    22    23    24   25   26   27   28   29   30   31   32  \
race
AfricanAmerican   351   252   213  215  179  152  138  105   95   80   68
Asian               7    12     5    3    5    8    4    0    2    3    1
Caucasian        1476  1303  1088  998  861  750  653  532  429  369  348
Hispanic           28    24    20   17   17   11    8   10    8    6    5
Other              25    15    18   12   22   10   14    6    7    7    6

num_medications   33   34   35   36   37   38   39   40  41  42  43  44  45  \
race
AfricanAmerican   54   52   40   35   30   24   23   16  18  15  10   6   5
Asian              1    0    0    3    1    0    1    0   1   0   0   0   0
Caucasian        282  221  220  162  173  145  121  122  92  87  90  65  63
Hispanic           7    4    6    2    1    4    0    1   2   1   1   0   0
Other              6    7    3    4    3    3    7    0   0   2   1   1   1

num_medications  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  \
race
AfricanAmerican   5   6   2   5   1   2   6   4   4   3   7   2   7   3   3
Asian             1   1   0   0   0   0   0   0   0   0   0   0   0   0   0
Caucasian        62  52  41  44  41  28  35  32  22  16  27  20  11  10  13
Hispanic          0   0   3   0   0   1   0   0   0   0   1   0   0   1   0
Other             2   0   1   3   3   0   4   0   0   0   0   0   0   0   0

num_medications  61  62  63  64  65  66  67  68  69  70  74  75  79  81
race
AfricanAmerican   3   0   2   2   5   1   1   0   1   1   0   0   0   0
Asian             0   1   0   0   0   0   0   0   0   0   0   0   0   0
Caucasian         6   9  11   4   6   3   1   2   1   1   1   2   1   1
Hispanic          0   0   0   0   0   0   0   0   0   0   0   0   0   0
Other             1   0   0   0   0   0   0   0   0   0   0   0   0   0


Cross-tabulation for race and number_outpatient:
number_outpatient     0     1     2     3    4    5    6   7   8   9   10  11  \
race
AfricanAmerican    11464   688   232   108   55   22   15  11  12   4    2   2
Asian                444    26    11     5    2    1    0   0   0   0    0   0
Caucasian          46301  4147  1824  1041  533  266  138  53  52  35   27  22
Hispanic            1302    91    23    28    7    6    7   1   1   0    0   0
Other               1034    68    29    11    5    2    0   5   0   0    0   0

number_outpatient  12  13  14  15  16  17  18  19  20  21  22  24  25  26  27  \
race
AfricanAmerican     2   4   0   2   0   0   0   0   1   0   0   0   0   0   0
Asian               0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Caucasian          13  11   9   9  10   4   2   1   4   3   3   2   1   1   3
Hispanic            0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
Other               1   0   0   0   0   0   0   0   0   0   0   0   0   0   0

number_outpatient  29  33  34  35  36  38
```

```
race
AfricanAmerican     0   0   0   0   0   1
Asian               0   0   0   0   0   0
Caucasian           1   1   1   1   1   0
Hispanic            0   0   0   0   0   0
Other               0   0   0   0   0   0


Cross-tabulation for race and number_emergency:
number_emergency      0     1    2    3   4   5   6   7   8   9  10  11  14  \
race
AfricanAmerican   11377   899  206   77  32  17   7   4   3   1   1   0   0
Asian               462    21    5    0   0   0   0   1   0   0   0   0   0
Caucasian         50442  3095  648  172  78  29  22   9   7   4   3   3   1
Hispanic           1343    89   18    8   3   3   0   1   0   1   1   0   0
Other              1063    72   13    5   2   0   0   0   0   0   0   0   0

number_emergency  15  16  19  20  25  37  42  76
race
AfricanAmerican    0   1   0   0   0   0   0   0
Asian              0   0   0   0   0   0   0   0
Caucasian          1   0   1   1   1   1   1   1
Hispanic           0   0   0   0   0   0   0   0
Other              0   0   0   0   0   0   0   0


Cross-tabulation for race and number_inpatient:
number_inpatient      0     1     2    3    4    5   6   7   8   9  10  11  \
race
AfricanAmerican   10537  1418   425  124   61   28  18   6   1   3   3   0
Asian               406    62    15    4    2    0   0   0   0   0   0   0
Caucasian         44629  6792  1894  664  288  125  71  19  17   7   7   4
Hispanic           1187   182    61   15    9    6   3   3   1   0   0   0
Other               993   121    23   13    3    2   0   0   0   0   0   0

number_inpatient  12  13  15
race
AfricanAmerican    0   1   0
Asian              0   0   0
Caucasian          2   0   1
Hispanic           0   0   0
Other              0   0   0


Cross-tabulation for race and number_diagnoses:
number_diagnoses     3     4     5     6     7     8      9  10  11  12  13  \
race
AfricanAmerican    561  1043  2013  1459  1394  1208   4942   1   0   0   1
Asian               19    53    62    66    42    38    209   0   0   0   0
Caucasian         1579  3057  6400  5759  5855  5946  25863   5   4   5   8
Hispanic            85   147   180   242   139   146    524   1   0   0   1
Other               69    79   136   145   133   121    472   0   0   0   0

number_diagnoses  14  15  16
race
AfricanAmerican    0   0   3
Asian              0   0   0
Caucasian          4   6  29
Hispanic           0   0   2
Other              0   0   0


Cross-tabulation for race and max_glu_serum:
max_glu_serum    >200  >300   None  Norm
race
AfricanAmerican    87    70  12323   145
```

```
Asian                7     8    467     7
Caucasian          803   598  51621  1498
Hispanic            41    30   1346    50
Other               15    11   1104    25


Cross-tabulation for race and A1Cresult:
A1Cresult           >7    >8   None  Norm
race
AfricanAmerican    394  1259  10186   786
Asian               28    49    387    25
Caucasian         2267  4199  45314  2740
Hispanic            71   195   1102    99
Other               59   114    910    72


Cross-tabulation for race and metformin:
metformin         Down     No  Steady   Up
race
AfricanAmerican     68  10134    2271  152
Asian                4    386      96    3
Caucasian          338  43110   10438  634
Hispanic            14   1099     332   22
Other                6    911     227   11


Cross-tabulation for race and repaglinide:
repaglinide       Down     No  Steady  Up
race
AfricanAmerican      3  12513     102   7
Asian                1    476      12   0
Caucasian           25  53759     682  54
Hispanic             0   1438      24   5
Other                0   1134      19   2


Cross-tabulation for race and nateglinide:
nateglinide       Down     No  Steady  Up
race
AfricanAmerican      1  12513     109   2
Asian                1    479       9   0
Caucasian            6  54170     331  13
Hispanic             1   1456      10   0
Other                0   1142      13   0


Cross-tabulation for race and chlorpropamide:
chlorpropamide    Down     No  Steady  Up
race
AfricanAmerican      0  12620       4   1
Asian                0    489       0   0
Caucasian            1  54460      56   3
Hispanic             0   1466       1   0
Other                0   1154       1   0


Cross-tabulation for race and glimepiride:
glimepiride       Down     No  Steady   Up
race
AfricanAmerican     25  12033     527   40
Asian                0    468      20    1
Caucasian          107  51545    2679  189
Hispanic             0   1410      54    3
Other                4   1090      58    3
```

```
Cross-tabulation for race and glipizide:
glipizide        Down     No  Steady   Up
race
AfricanAmerican    59  11153    1317   96
Asian               0    419      63    7
Caucasian         296  47491    6307  426
Hispanic            8   1279     160   20
Other               4    982     155   14


Cross-tabulation for race and glyburide:
glyburide        Down     No  Steady   Up
race
AfricanAmerican    40  11509     979   97
Asian               2    438      47    2
Caucasian         349  48287    5397  487
Hispanic            7   1329     119   12
Other               2   1042     105    6


Cross-tabulation for race and pioglitazone:
pioglitazone     Down     No  Steady   Up
race
AfricanAmerican    11  11847     735   32
Asian               1    451      36    1
Caucasian          67  50306    4005  142
Hispanic            3   1348     111    5
Other               1   1089      63    2


Cross-tabulation for race and rosiglitazone:
rosiglitazone    Down     No  Steady  Up
race
AfricanAmerican     9  11875     714  27
Asian               0    466      22   1
Caucasian          65  50839    3517  99
Hispanic            1   1379      84   3
Other               0   1095      55   5


Cross-tabulation for race and acarbose:
acarbose         Down     No  Steady  Up
race
AfricanAmerican     0  12607      17   1
Asian               0    487       2   0
Caucasian           1  54352     159   8
Hispanic            0   1465       2   0
Other               0   1153       2   0


Cross-tabulation for race and miglitol:
miglitol         Down     No  Steady
race
AfricanAmerican     0  12622       3
Asian               0    489       0
Caucasian           1  54504      15
Hispanic            0   1467       0
Other               0   1155       0


Cross-tabulation for race and insulin:
insulin          Down     No  Steady    Up
race
AfricanAmerican  1462   5528    4288  1347
Asian              55    276     113    45
Caucasian        5512  27365   16462  5181
```

```
Hispanic          178    695     441   153
Other             187    514     293   161


Cross-tabulation for race and glyburide-metformin:
glyburide-metformin  Down     No  Steady  Up
race
AfricanAmerican         1  12551      73   0
Asian                   0    480       9   0
Caucasian               3  54149     361   7
Hispanic                0   1458       9   0
Other                   0   1139      16   0


Cross-tabulation for race and change:
change              Ch     No
race
AfricanAmerican   5646   6979
Asian              201    288
Caucasian        24298  30222
Hispanic           698    769
Other              577    578


Cross-tabulation for race and diabetesMed:
diabetesMed         No    Yes
race
AfricanAmerican   3021   9604
Asian              132    357
Caucasian        13260  41260
Hispanic           371   1096
Other              253    902


Cross-tabulation for race and discharge_desc:
discharge_desc   Elective  Emergency  Newborn  Not Available  Not Mapped  \
race
AfricanAmerican      2195       7484        2            244          55
Asian                  89        218        1             10           0
Caucasian           12088      25158        3           2561         228
Hispanic              255        722        2             66           2
Other                 216        531        0             32           5

discharge_desc   Other  Trauma Center  Urgent
race
AfricanAmerican    538              2    2105
Asian               49              0     122
Caucasian         3612             18   10852
Hispanic           202              0     218
Other               99              0     272


Cross-tabulation for race and admission_source_desc:
admission_source_desc  Admitted as an inpatient to this hospital  \
race
AfricanAmerican                                                0
Asian                                                          0
Caucasian                                                      9
Hispanic                                                       0
Other                                                          0

admission_source_desc  Discharged to home  Discharged/transferred to ICF  \
race
AfricanAmerican                      8637                            108
Asian                                 359                              1
Caucasian                           33747                            361
```

```
Hispanic                            1098                                0
Other                                795                                7

admission_source_desc  Discharged/transferred to SNF  \
race
AfricanAmerican                                 1129
Asian                                             45
Caucasian                                       7579
Hispanic                                         109
Other                                             94

admission_source_desc  Discharged/transferred to a federal health care facility.  \
race
AfricanAmerican                                                     2
Asian                                                               0
Caucasian                                                           1
Hispanic                                                            0
Other                                                               0

admission_source_desc  Discharged/transferred to a long term care hospital.  \
race
AfricanAmerican                                                    76
Asian                                                               0
Caucasian                                                         156
Hispanic                                                            1
Other                                                               4

admission_source_desc  Discharged/transferred to a nursing facility certified under Medi
caid but not certified under Medicare.  \
race

AfricanAmerican                                                     5

Asian                                                               0

Caucasian                                                          19

Hispanic                                                            0

Other                                                               0


admission_source_desc  Discharged/transferred to another rehab fac including rehab units
of a hospital .  \
race

AfricanAmerican                                                   236

Asian                                                               7

Caucasian                                                         860

Hispanic                                                           11

Other                                                              19


admission_source_desc  Discharged/transferred to another short term hospital  \
race
AfricanAmerican                                                   122
Asian                                                              17
Caucasian                                                        1098
Hispanic                                                           24
Other                                                              27

admission_source_desc  Discharged/transferred to another type of inpatient care institut
```

```
ion  \
race

AfricanAmerican                                                     140

Asian                                                                 6

Caucasian                                                           618

Hispanic                                                             23

Other                                                                10


admission_source_desc  Discharged/transferred to home under care of Home IV provider  \
race
AfricanAmerican                                                       7
Asian                                                                 0
Caucasian                                                            50
Hispanic                                                              0
Other                                                                 2

admission_source_desc  Discharged/transferred to home with home health service  \
race
AfricanAmerican                                                    1178
Asian                                                                35
Caucasian                                                          6079
Hispanic                                                            103
Other                                                               107

admission_source_desc  Discharged/transferred within this institution to Medicare approv
ed swing bed  \
race

AfricanAmerican                                                       3

Asian                                                                 0

Caucasian                                                            22

Hispanic                                                              0

Other                                                                 0


admission_source_desc  Discharged/transferred/referred another institution for outpatien
t services  \
race

AfricanAmerican                                                       1

Asian                                                                 0

Caucasian                                                             1

Hispanic                                                              0

Other                                                                 0


admission_source_desc  Discharged/transferred/referred to a psychiatric hospital of psyc
hiatric distinct part unit of a hospital  \
race

AfricanAmerican                                                      16
```

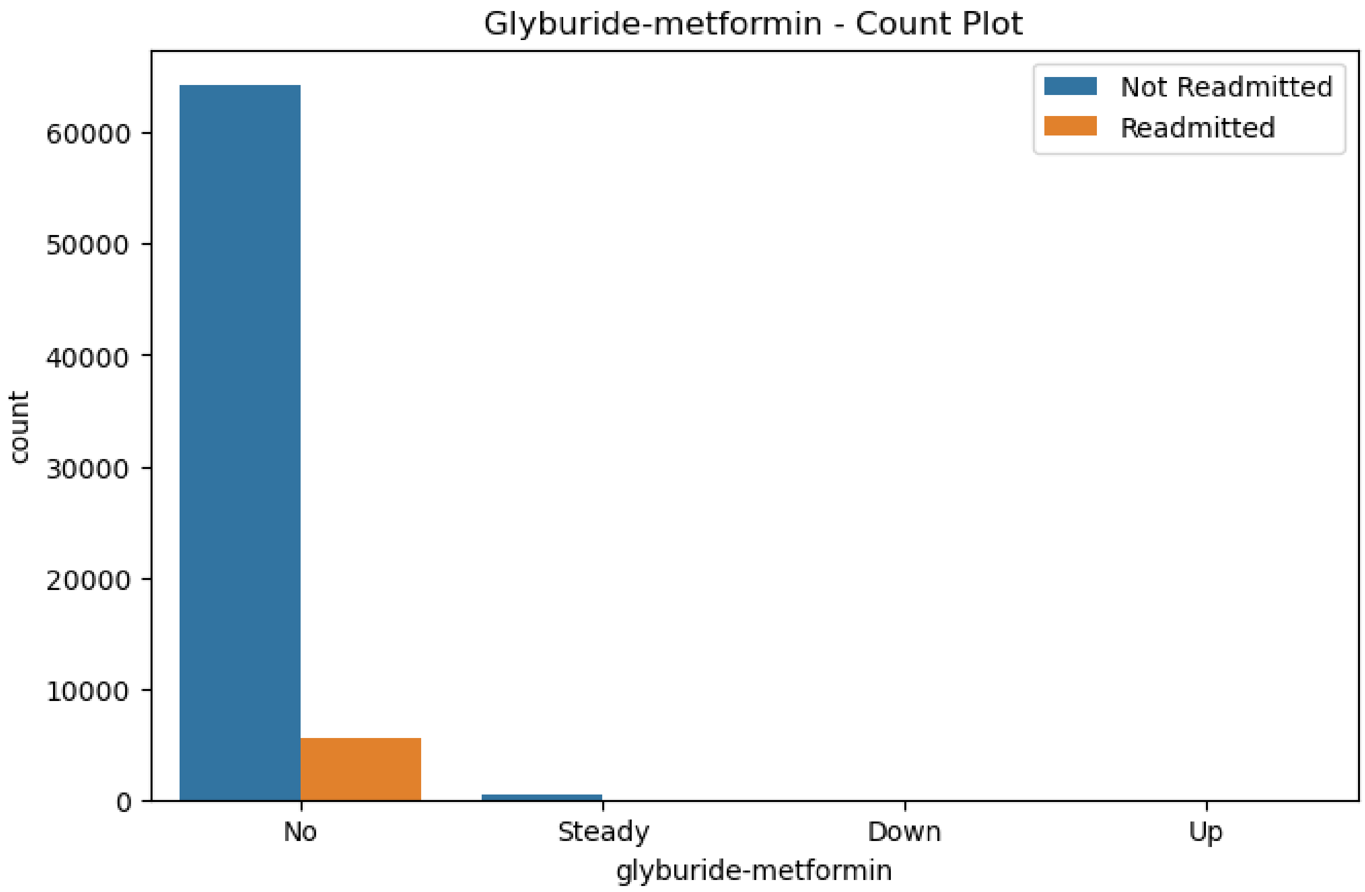
```
Asian                                                                  0

Caucasian                                                             44

Hispanic                                                               3

Other                                                                  0


admission_source_desc  Discharged/transferred/referred to this institution for outpatien
t services  \
race

AfricanAmerican                                                        2

Asian                                                                  0

Caucasian                                                              4

Hispanic                                                               0

Other                                                                  0


admission_source_desc  Expired  Expired at home. Medicaid only, hospice.  \
race
AfricanAmerican            211                                         0
Asian                        6                                         1
Caucasian                  892                                         5
Hispanic                    12                                         0
Other                       19                                         0

admission_source_desc  Expired in a medical facility. Medicaid only, hospice.  \
race
AfricanAmerican                                                        0
Asian                                                                  0
Caucasian                                                              1
Hispanic                                                               0
Other                                                                  0

admission_source_desc  Hospice / home  Hospice / medical facility  Left AMA  \
race
AfricanAmerican                    47                          25        69
Asian                               2                           0         0
Caucasian                         189                         200       231
Hispanic                            4                           2        14
Other                               7                           4        13

admission_source_desc  Neonate discharged to another hospital for neonatal aftercare  \
race
AfricanAmerican                                                        0
Asian                                                                  0
Caucasian                                                              5
Hispanic                                                               1
Other                                                                  0

admission_source_desc  Not Mapped  Other  \
race
AfricanAmerican               140    471
Asian                           3      7
Caucasian                     655   1692
Hispanic                        5     57
Other                           9     38

admission_source_desc  Still patient or expected to return for outpatient services
race
```

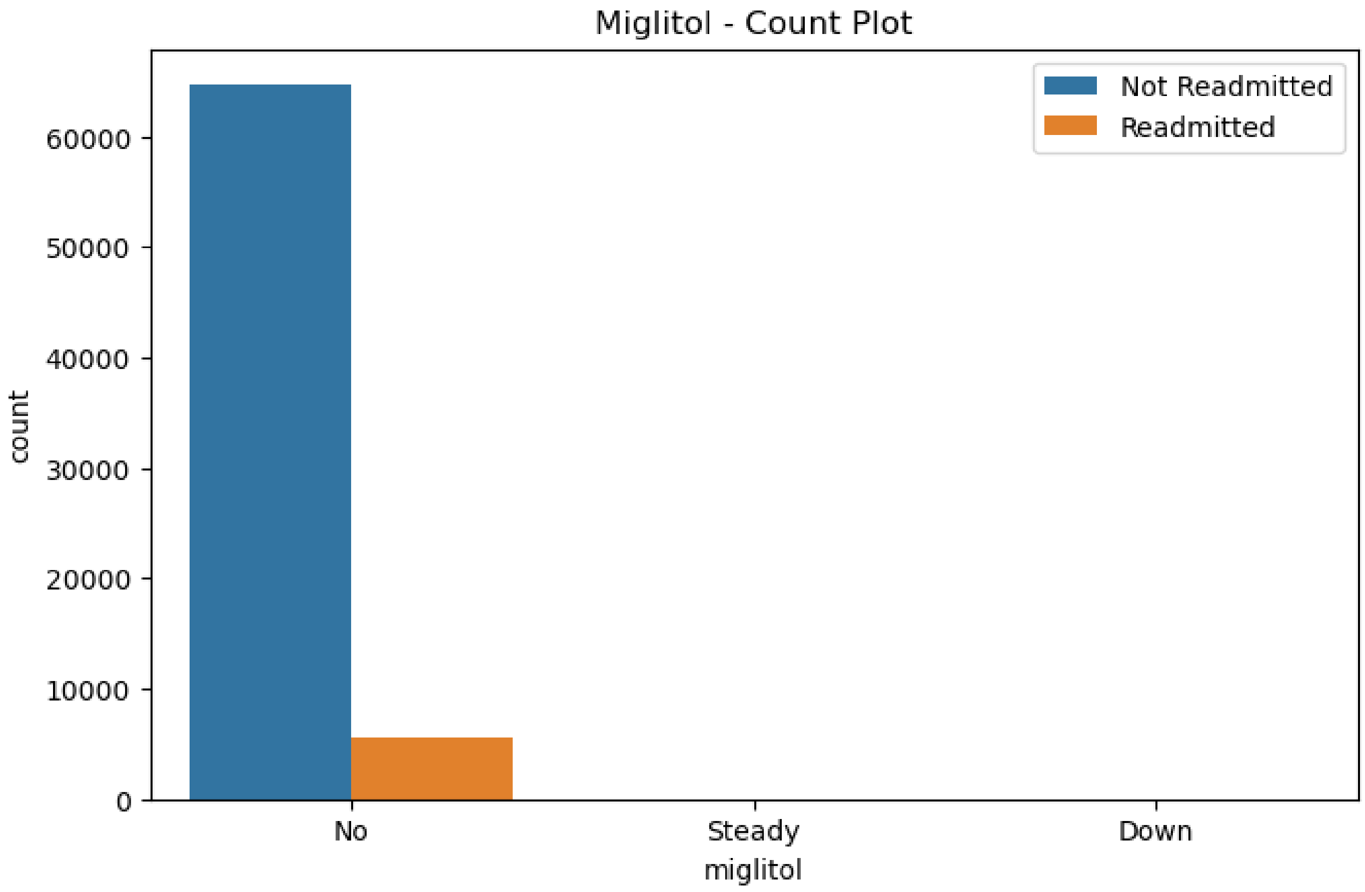
```
AfricanAmerican                                                           0
Asian                                                                     0
Caucasian                                                                 2
Hispanic                                                                  0
Other                                                                     0


Cross-tabulation for race and admission_desc:
admission_desc    Court/Law Enforcement   Emergency Room   Extramural Birth  \
race
AfricanAmerican                       3             7191                  2
Asian                                 0              305                  0
Caucasian                             4            26273                  0
Hispanic                              2              904                  0
Other                                 0              677                  0

admission_desc    Not Available   Not Mapped   Physician Referral   Sick Baby  \
race
AfricanAmerican              12            3                 3874           0
Asian                         0            1                  153           0
Caucasian                    63          132                19780           1
Hispanic                      0            0                  433           0
Other                         0            0                  337           0

admission_desc    Transfer from Ambulatory Surgery Center  \
race
AfricanAmerican                                         0
Asian                                                   0
Caucasian                                               2
Hispanic                                                0
Other                                                   0

admission_desc    Transfer from a Skilled Nursing Facility (SNF)  \
race
AfricanAmerican                                              119
Asian                                                          3
Caucasian                                                    355
Hispanic                                                       3
Other                                                          0

admission_desc    Transfer from another health care facility  \
race
AfricanAmerican                                          454
Asian                                                      3
Caucasian                                               1091
Hispanic                                                   6
Other                                                      3

admission_desc    Transfer from critial access hospital  \
race
AfricanAmerican                                        0
Asian                                                  0
Caucasian                                              6
Hispanic                                               0
Other                                                  0

admission_desc    Transfer from hospital inpt/same fac reslt in a sep claim  \
race
AfricanAmerican                                                   1
Asian                                                             0
Caucasian                                                         2
Hispanic                                                          0
Other                                                             0

admission_desc   Clinic Referral  HMO Referral  Normal Delivery  Other  \
race
```

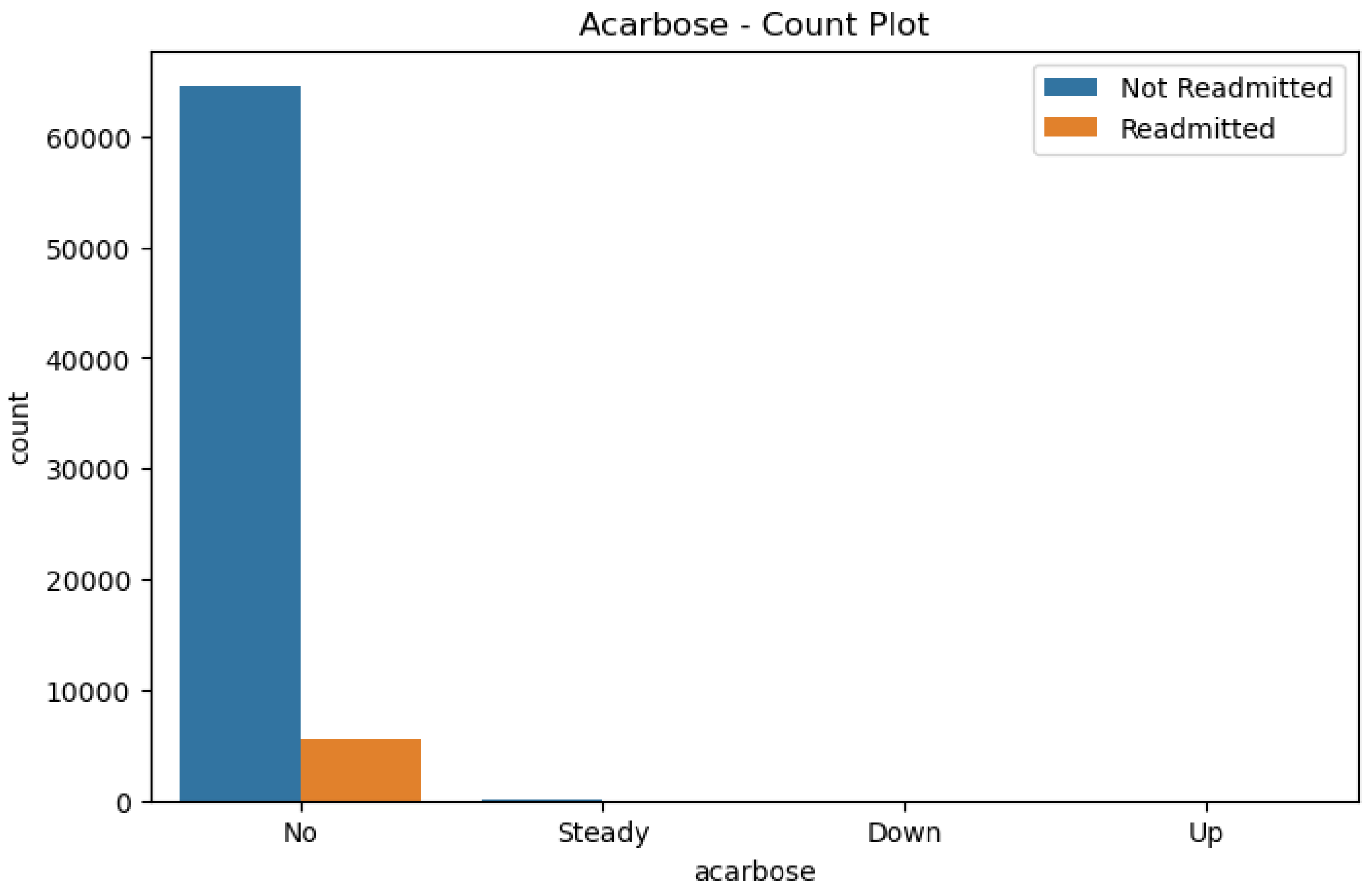
```
AfricanAmerican                231           8                  0    424
Asian                            1           0                  0     15
Caucasian                      558          81                  1   4203
Hispanic                         6           0                  0     94
Other                           23           0                  0     66

admission_desc   Transfer from a hospital
race
AfricanAmerican                       303
Asian                                   8
Caucasian                            1968
Hispanic                               19
Other                                  49


Cross-tabulation for race and binary_readmitted:
binary_readmitted      0     1
race
AfricanAmerican    11652   973
Asian                456    33
Caucasian          50149  4371
Hispanic            1356   111
Other               1086    69


Cross-tabulation for race and diag_1_new:
diag_1_new       Diabetes  Heart Disease  Injury  Other  Respiratory
race
AfricanAmerican      1498           2789     779   6787          772
Asian                  19            124      31    288           27
Caucasian            3465          14913    3690  28288         4164
Hispanic              135            300      82    838          112
Other                  83            311      88    589           84


Cross-tabulation for race and diag_2_new:
diag_2_new       Diabetes  Heart Disease  Injury  Other  Respiratory
race
AfricanAmerican      1648           2199      42   8098          638
Asian                  67             93       5    300           24
Caucasian            5894          13094     293  30883         4356
Hispanic              217            193      11    955           91
Other                 145            228       5    701           76


Cross-tabulation for race and diag_3_new:
diag_3_new       Diabetes  Heart Disease  Injury  Other  Respiratory
race
AfricanAmerican      2176           1618      24   8353          454
Asian                  85             67       1    326           10
Caucasian            9102           9586     173  32878         2781
Hispanic              302            168       8    936           53
Other                 219            143       5    746           42
```

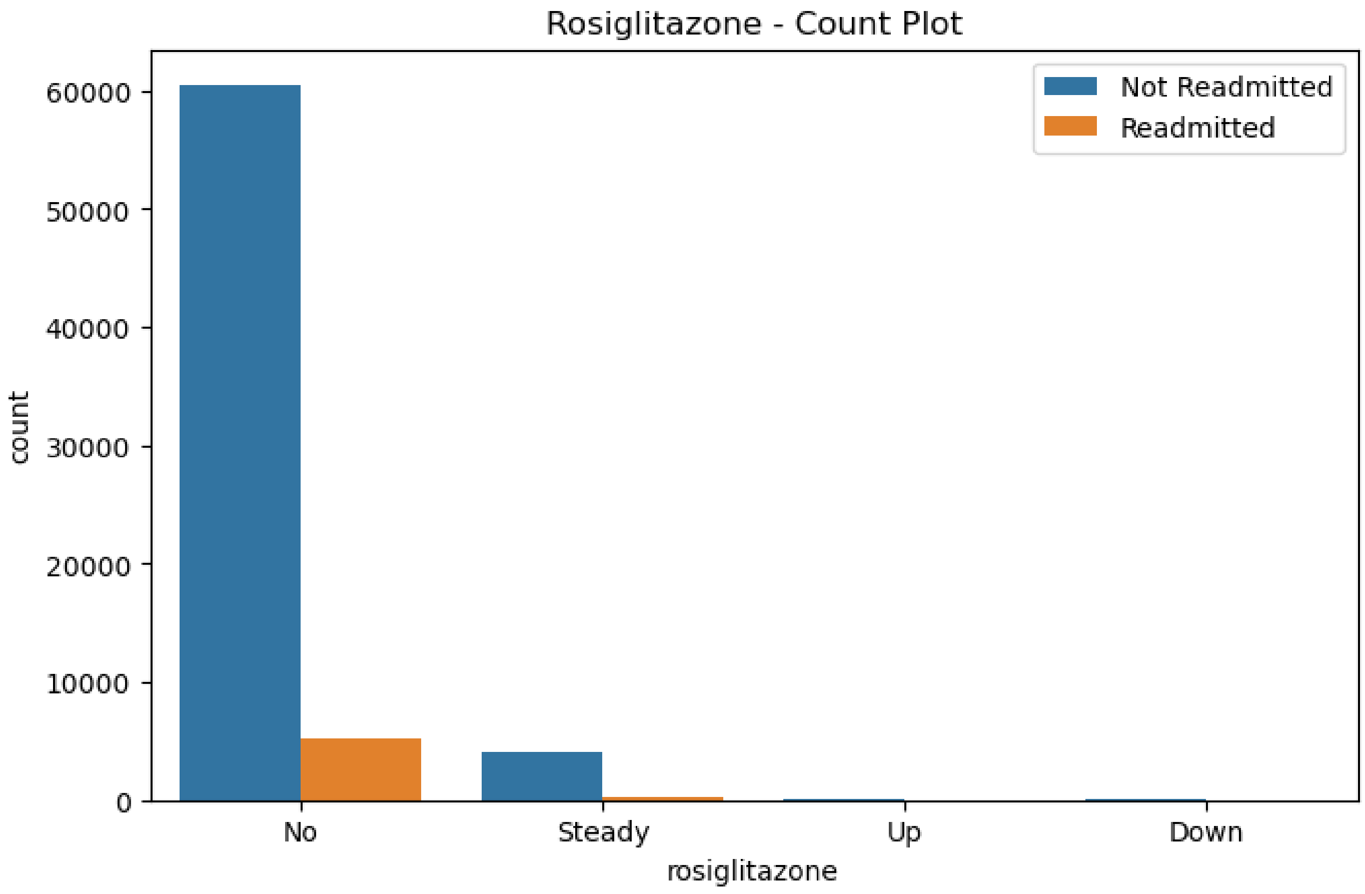
In [36]:
```python
honed_cat_var = df[['glyburide-metformin','miglitol','acarbose',
                'rosiglitazone','pioglitazone','glyburide','chlorpropamide',
                'nateglinide','repaglinide', 'binary_readmitted']]
```

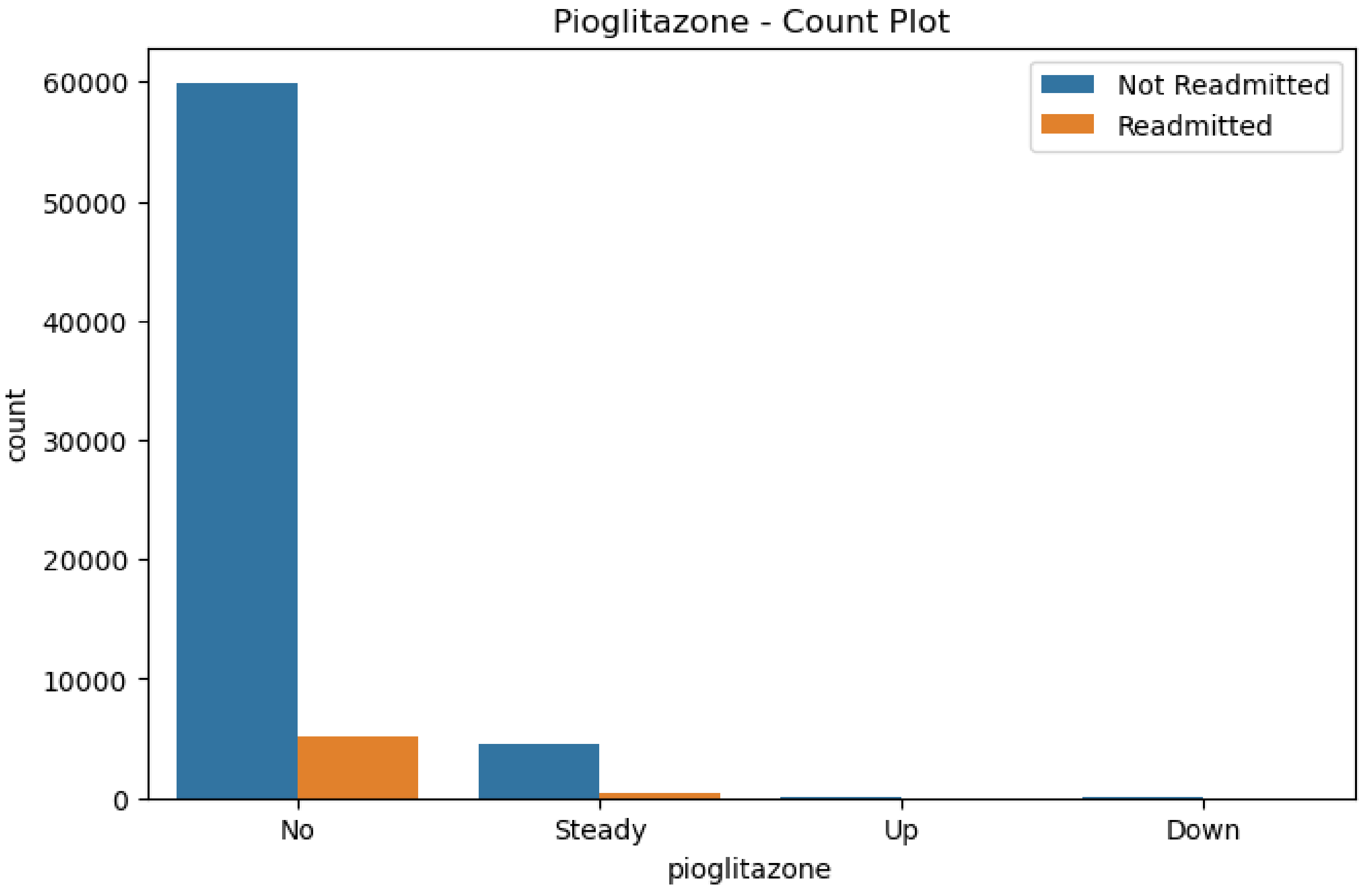
In [37]:
```python
legend_label = ['Not Readmitted','Readmitted']

for var in honed_cat_var:
    if var != 'binary_readmitted':
        plt.figure(figsize=(8,5))
        ax = sns.countplot(data = df, x = var, hue = 'binary_readmitted')
        ax.legend(legend_label)
        plt.title(f'{var.capitalize()} - Count Plot')
        plt.show()
```

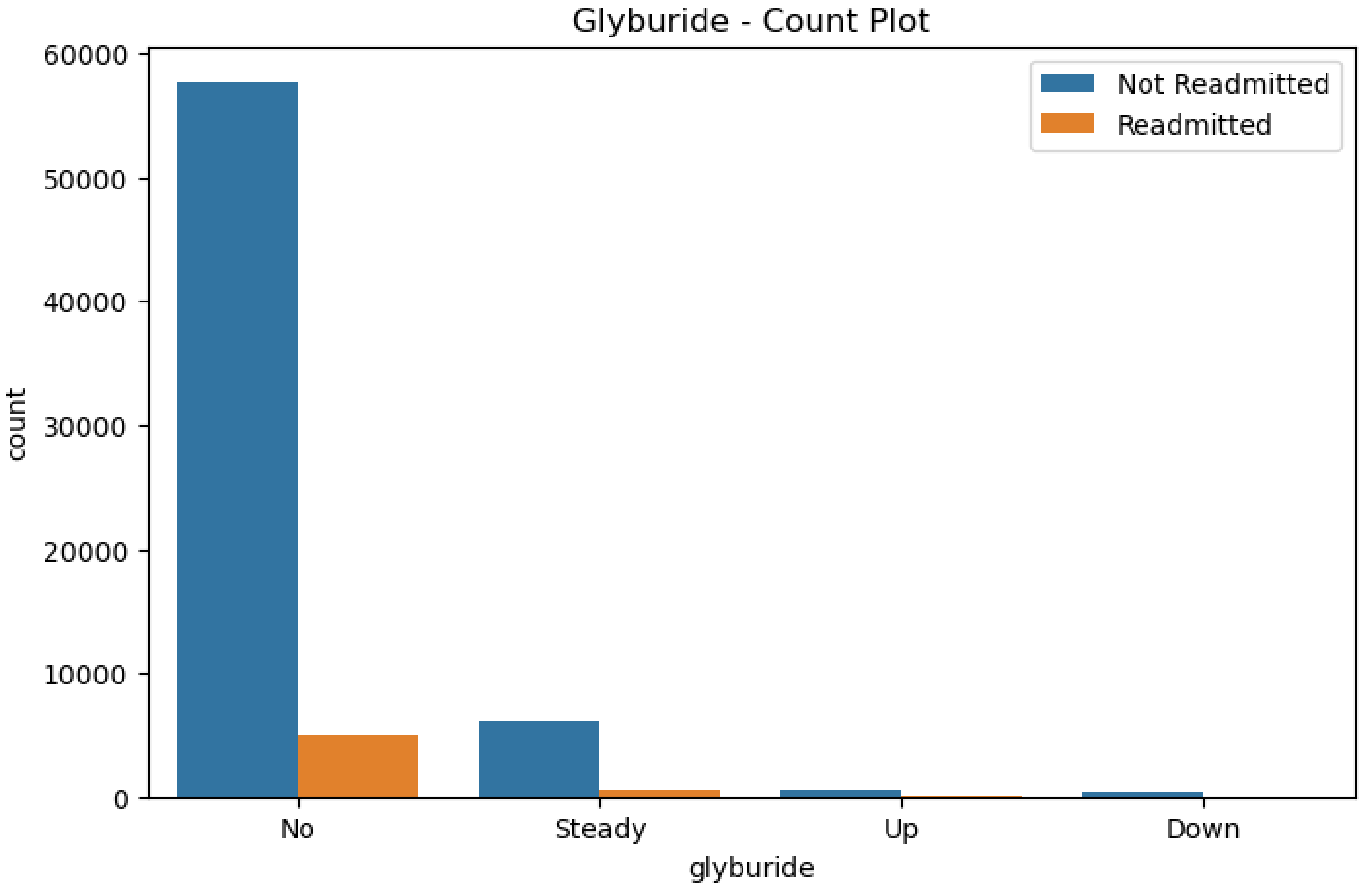


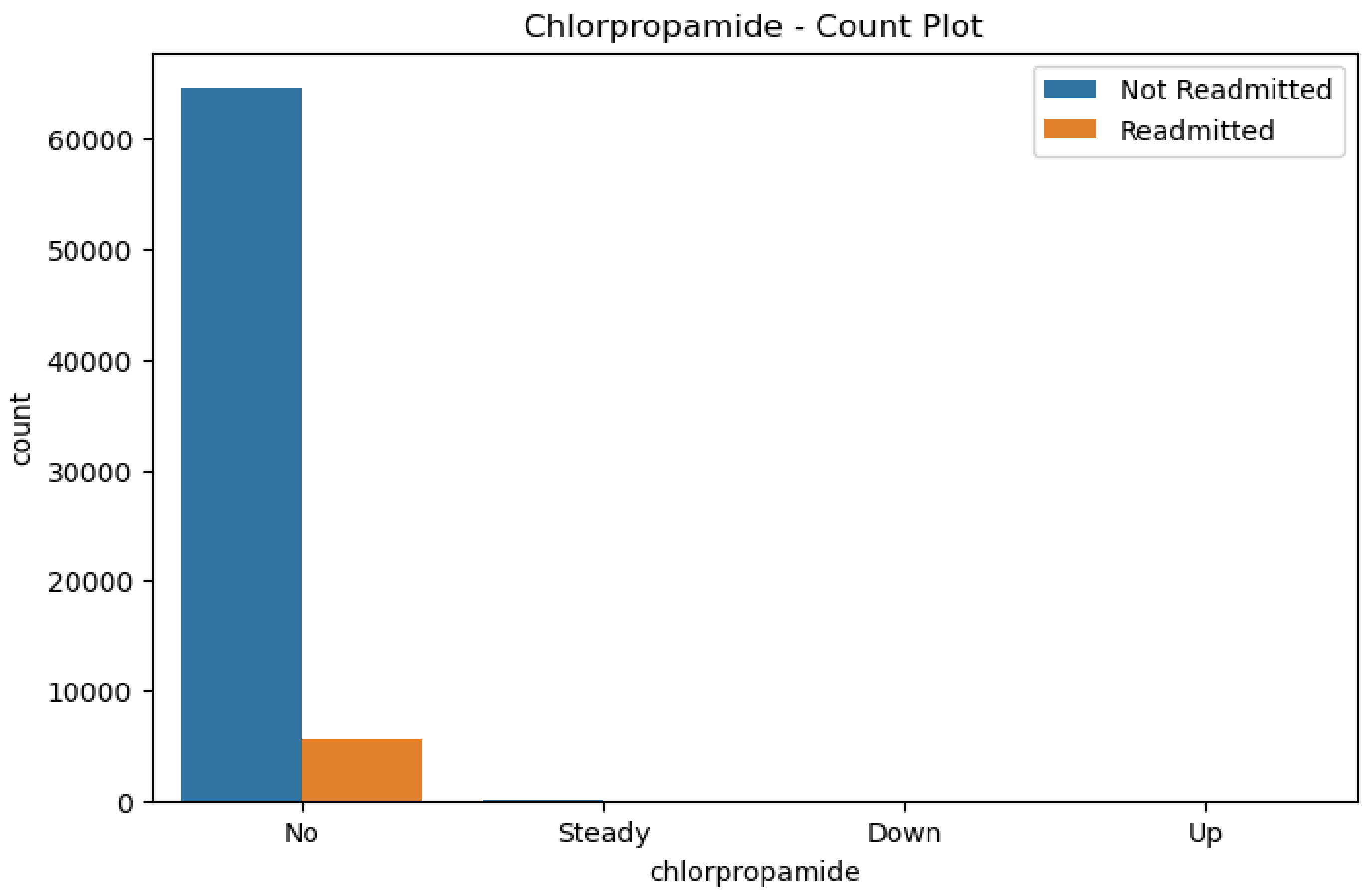


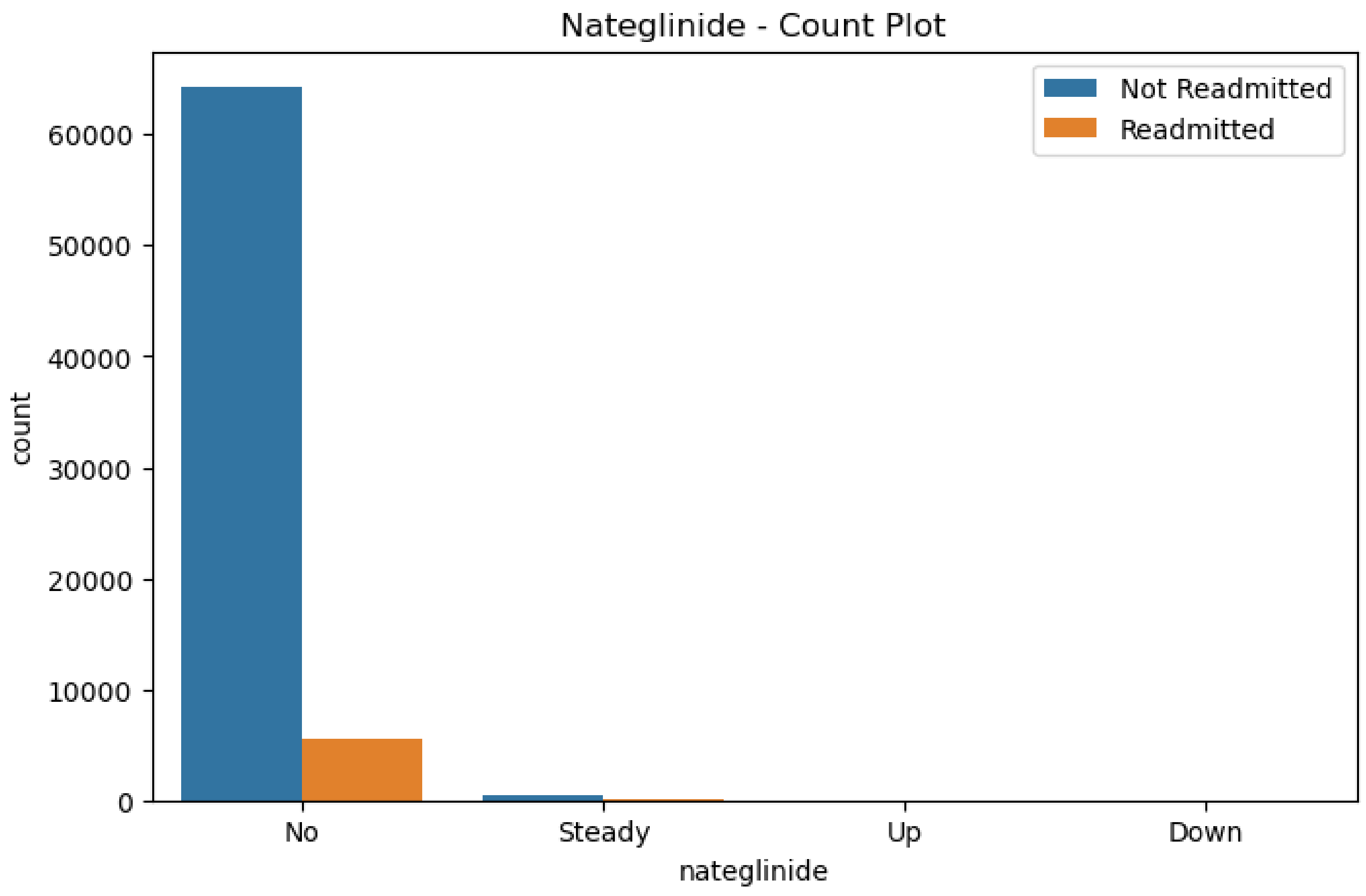

Acarbose - Count Plot
Not Readmitted
Readmitted
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
acarbose

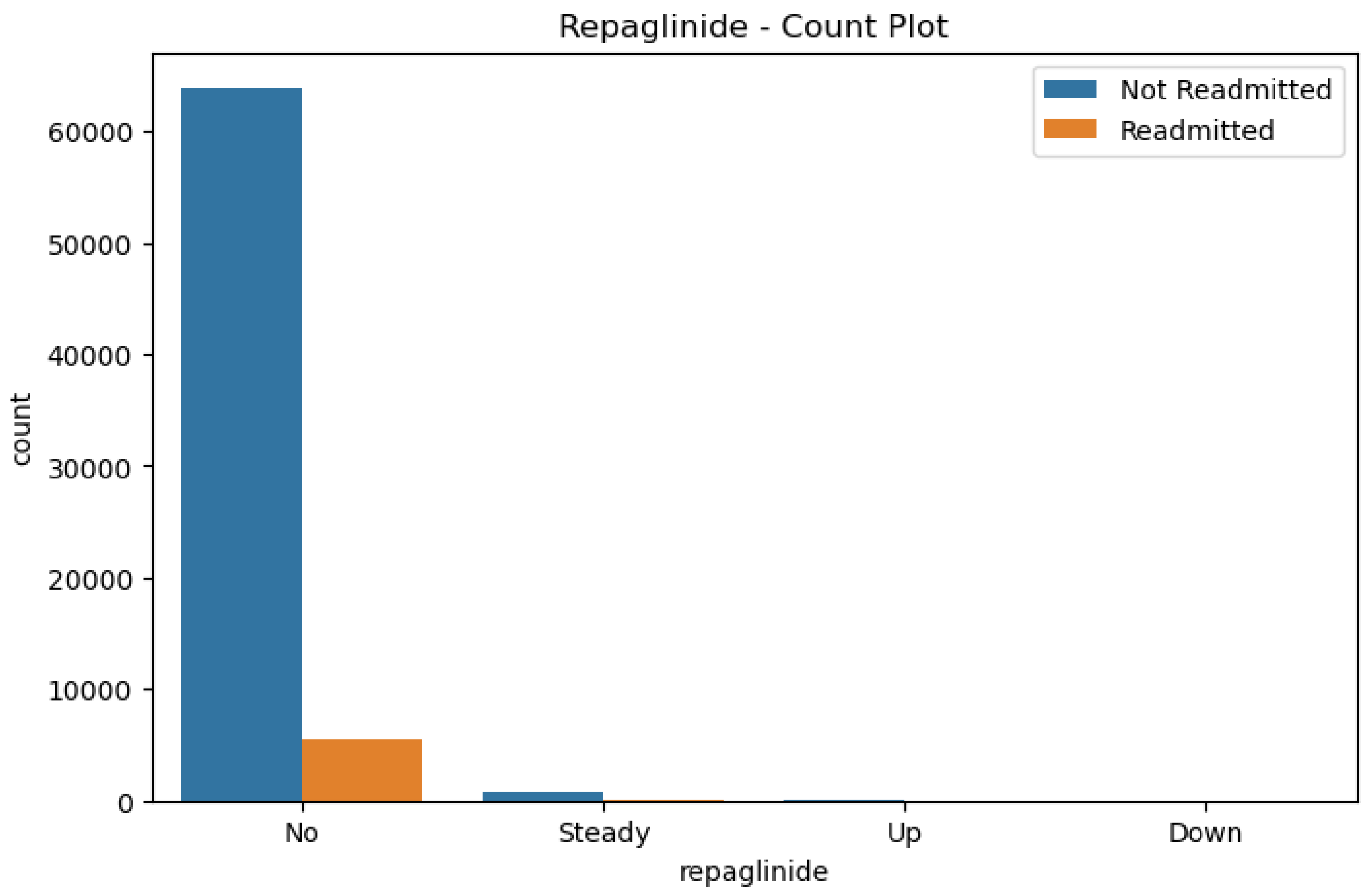

Rosiglitazone - Count Plot
Not Readmitted
Readmitted
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
rosiglitazone


Pioglitazone - Count Plot
Not Readmitted
Readmitted
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
pioglitazone


Glyburide - Count Plot
Not Readmitted
Readmitted
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
glyburide

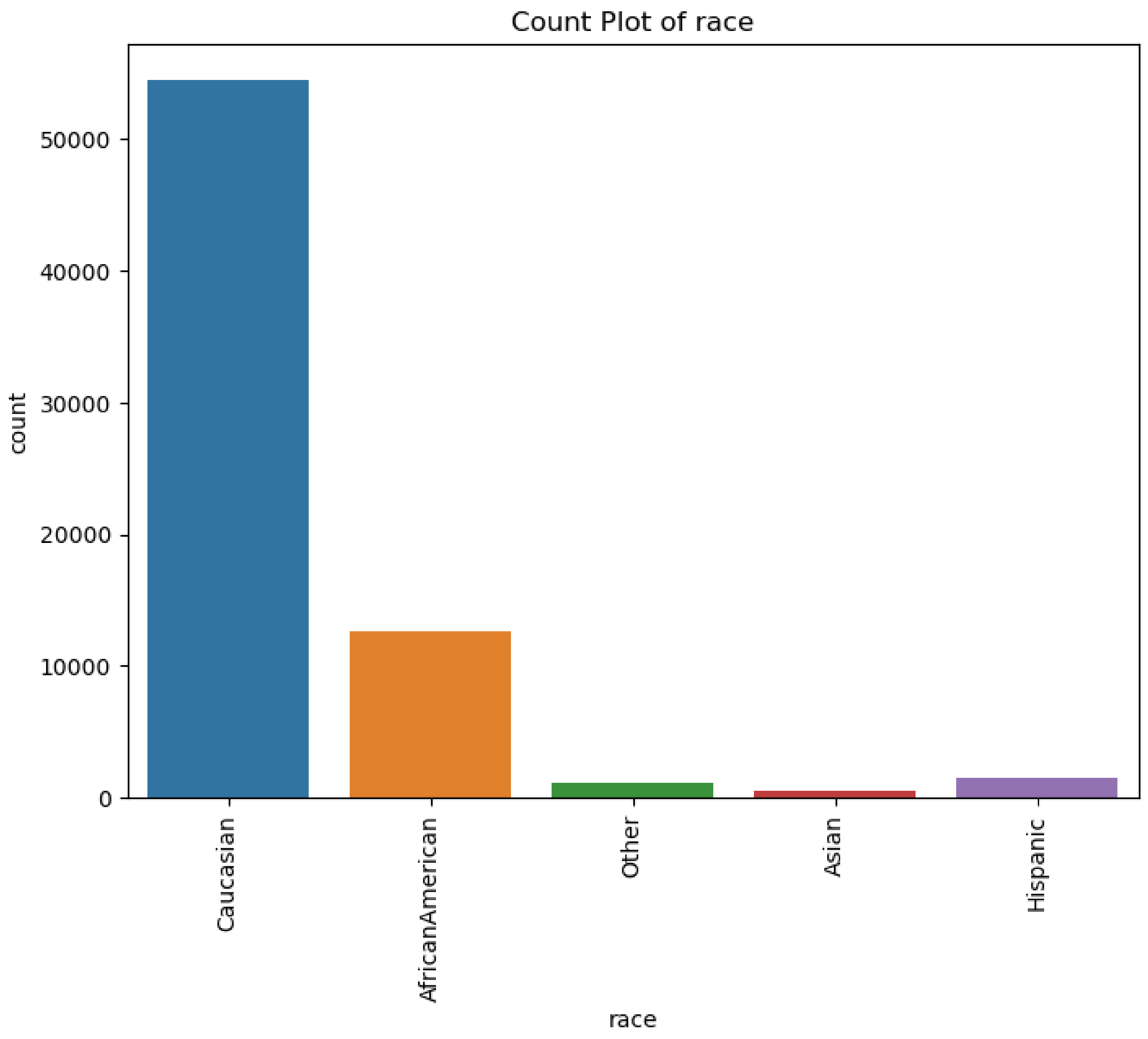

Chlorpropamide - Count Plot
Not Readmitted
Readmitted
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
chlorpropamide

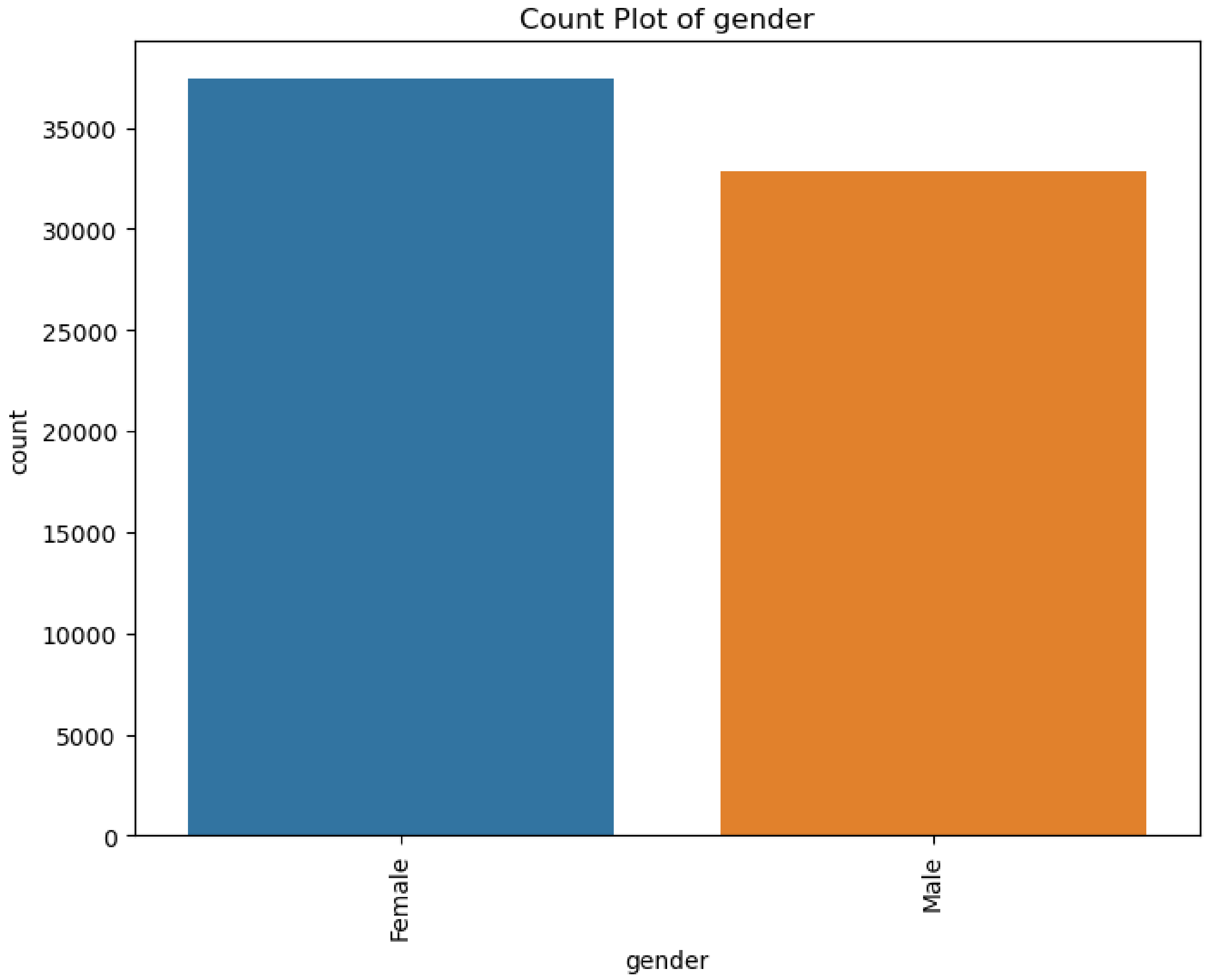

Nateglinide - Count Plot
Not Readmitted
Readmitted
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
nateglinide

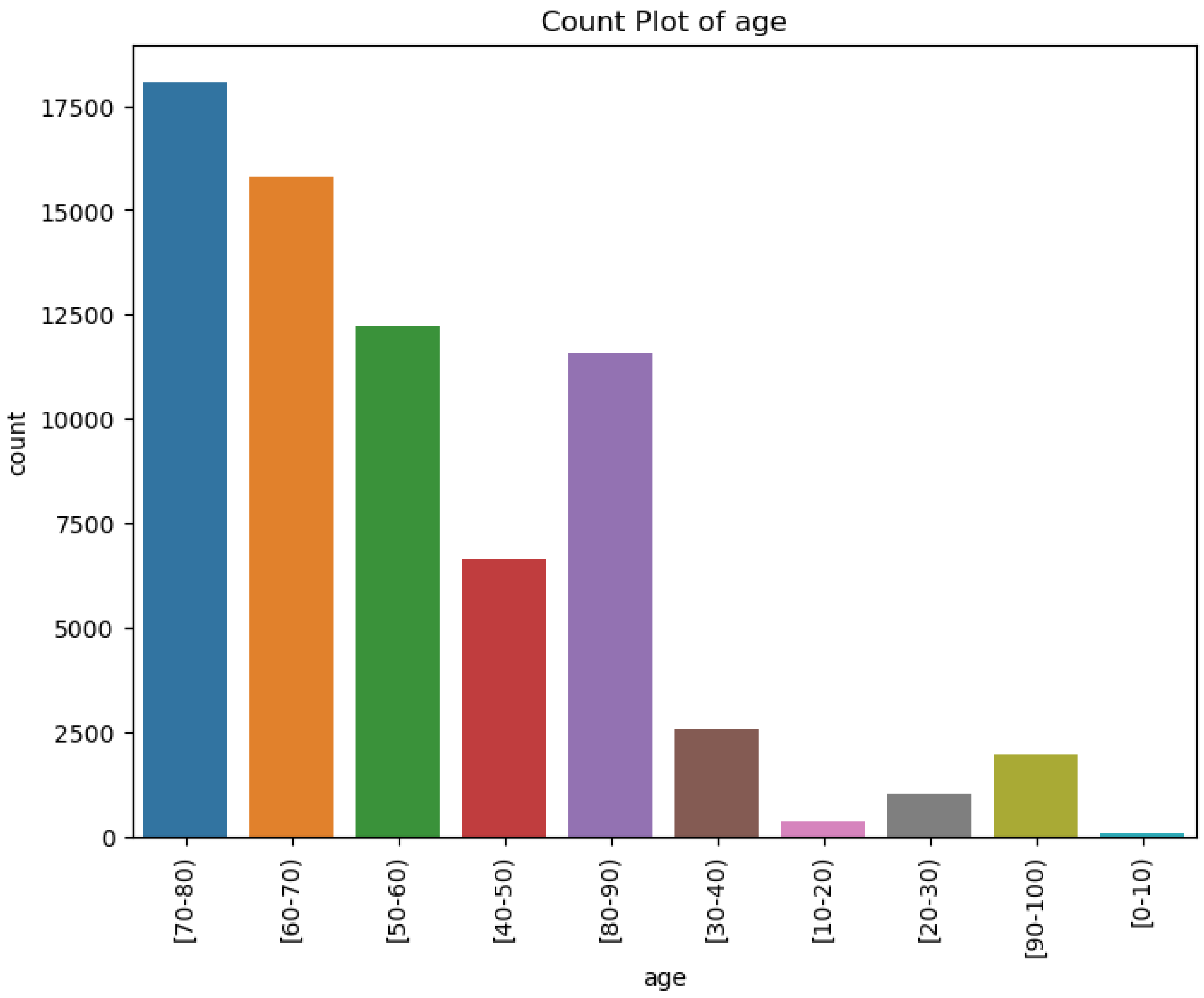


More Countplots for Categorical Variables

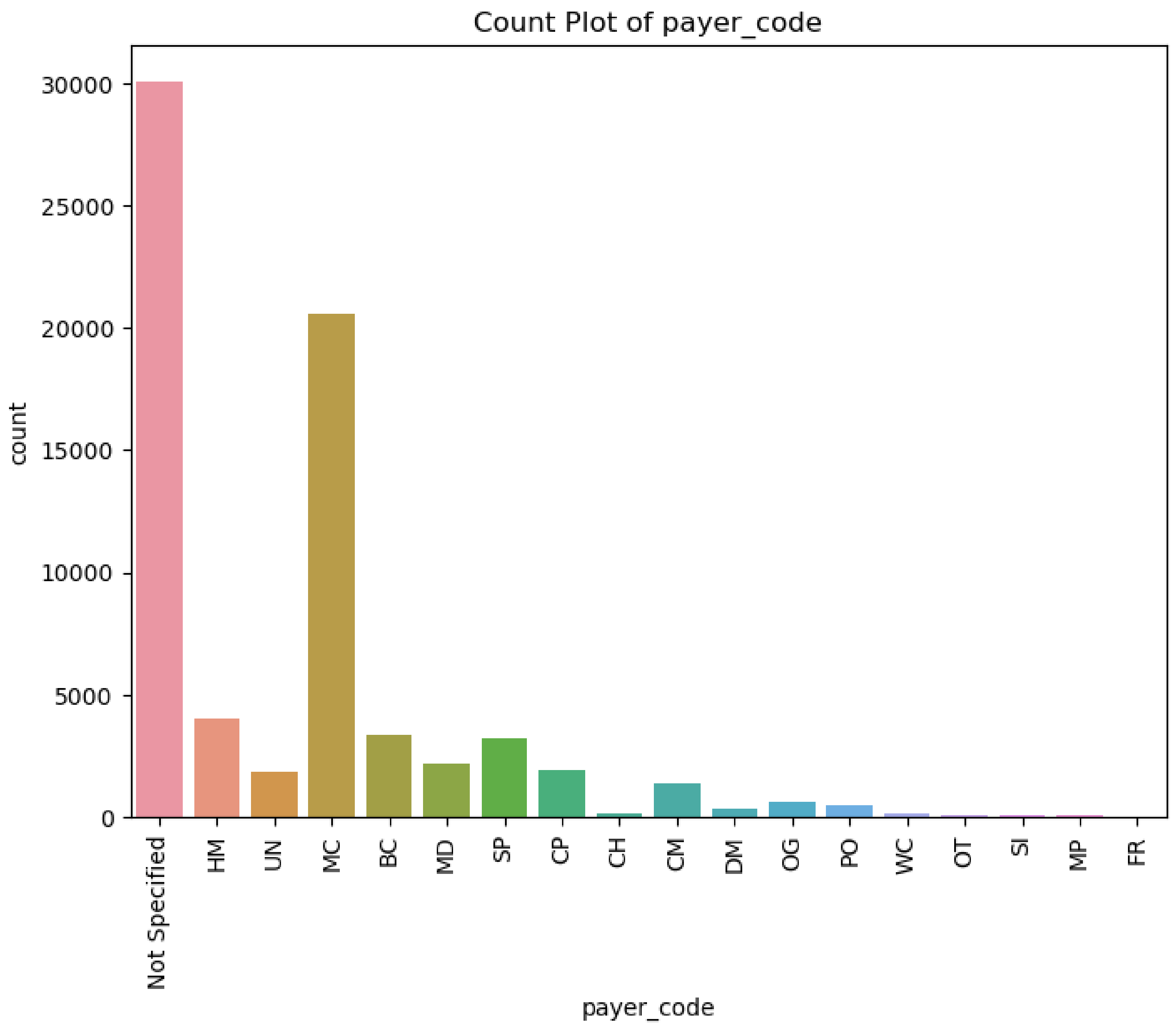
In [38]:
```
categorical_vars = df.select_dtypes(include=['object', 'category']).columns

# Create count plots for each categorical variable
for var in categorical_vars:
    plt.figure(figsize=(8, 6))  # Adjust the figure size as needed
    sns.countplot(x=var, data=df)
    plt.title(f'Count Plot of {var}')
    plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
    plt.show()
```

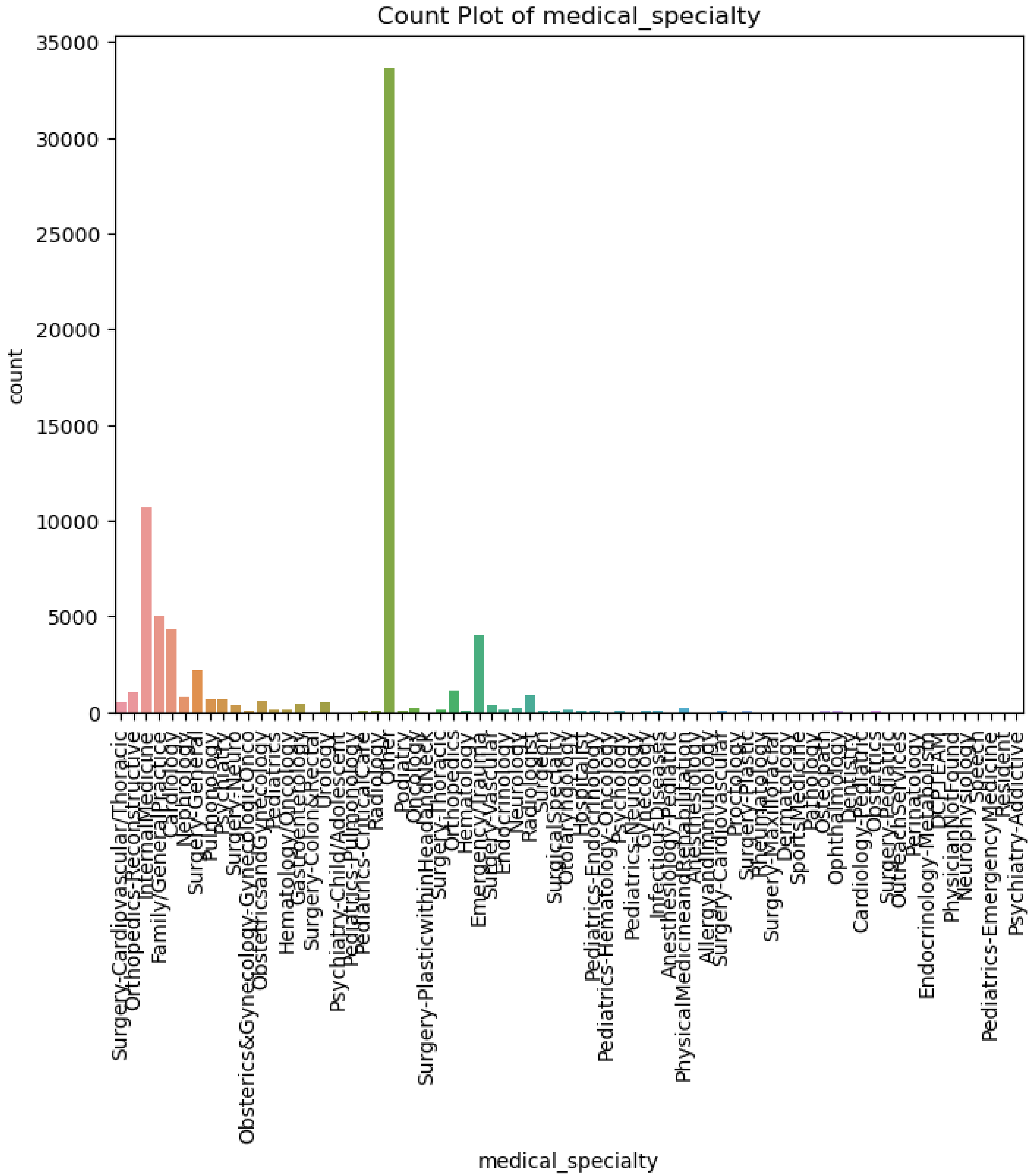

Count Plot of race
count
50000
40000
30000
20000
10000
0
Caucasian
AfricanAmerican
Other
Asian
Hispanic
race

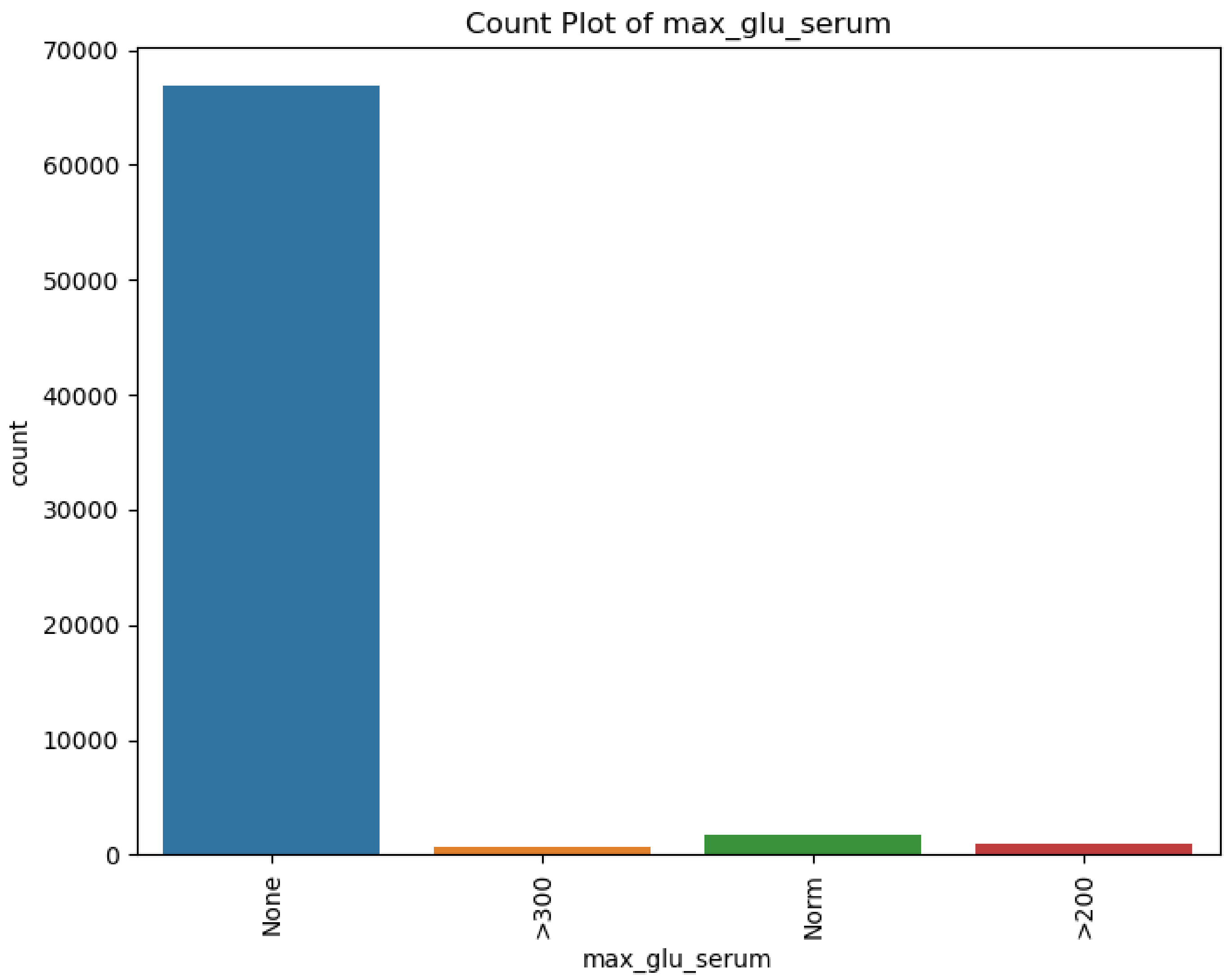

Count Plot of gender
count
35000
30000
25000
20000
15000
10000
5000
0
Female
Male
gender

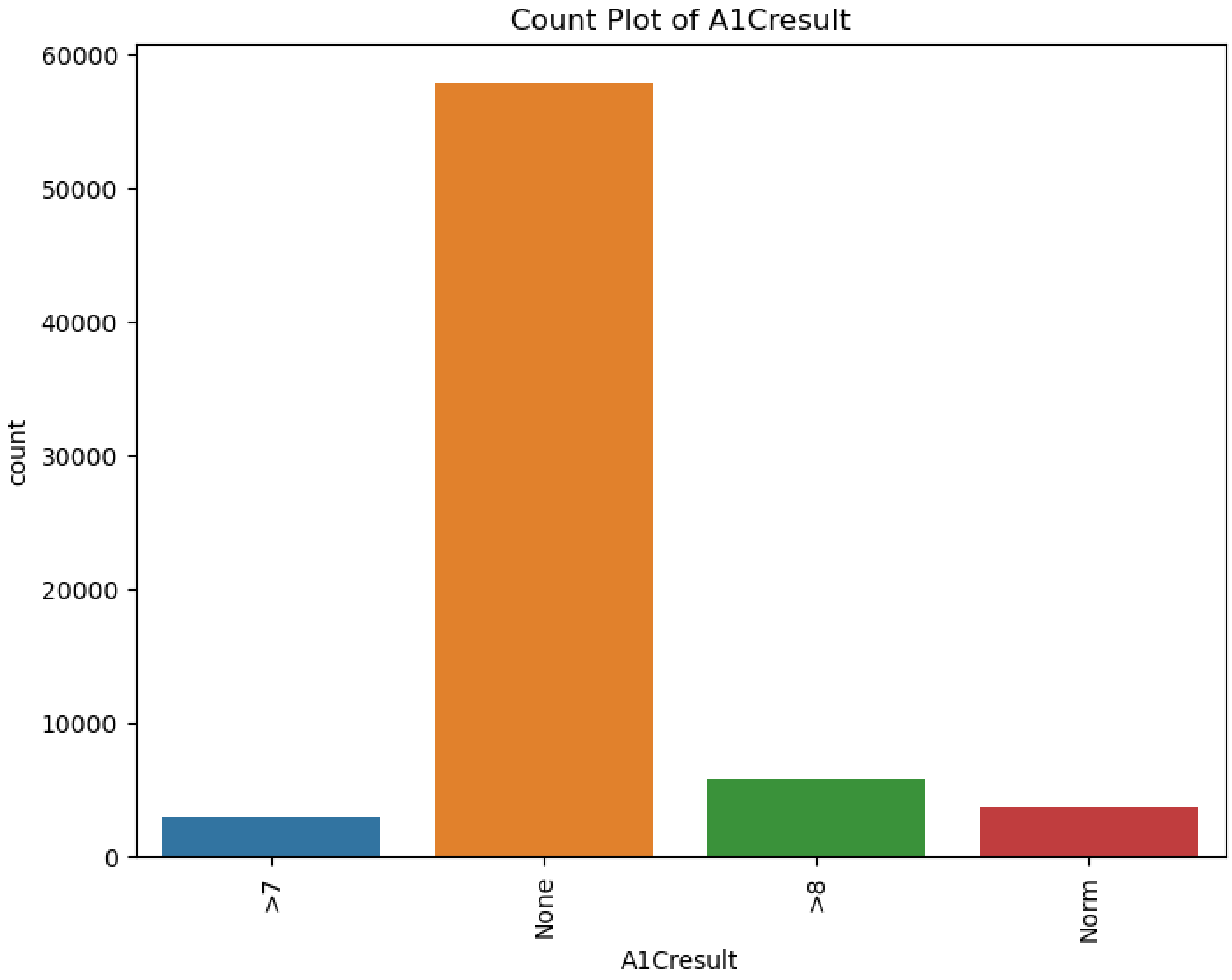


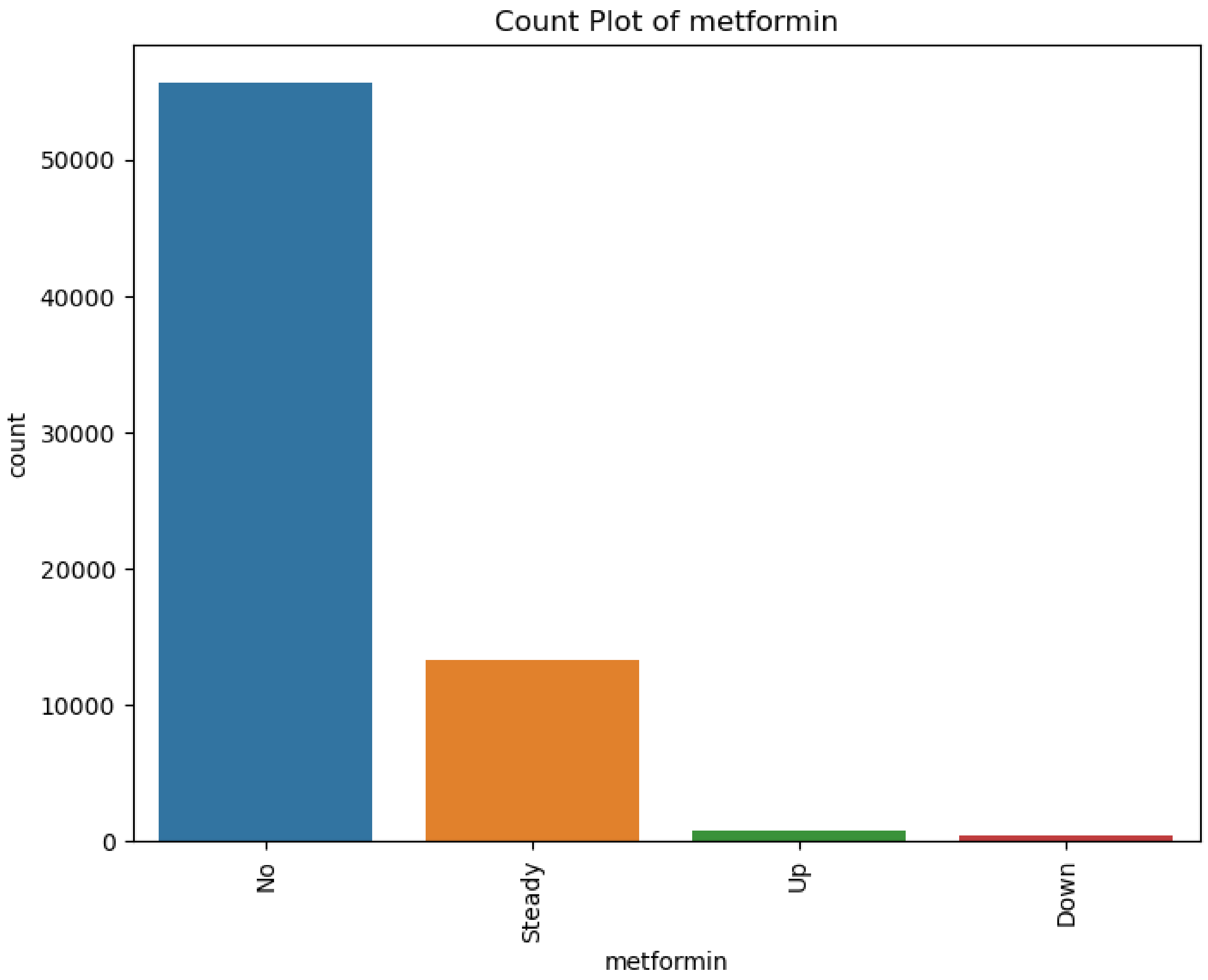
Count Plot of age
count
17500
15000
12500
10000
7500
5000
2500
0
[70-80)
[60-70)
[50-60)
[40-50)
[80-90)
[30-40)
[10-20)
[20-30)
[90-100)
[0-10)
age

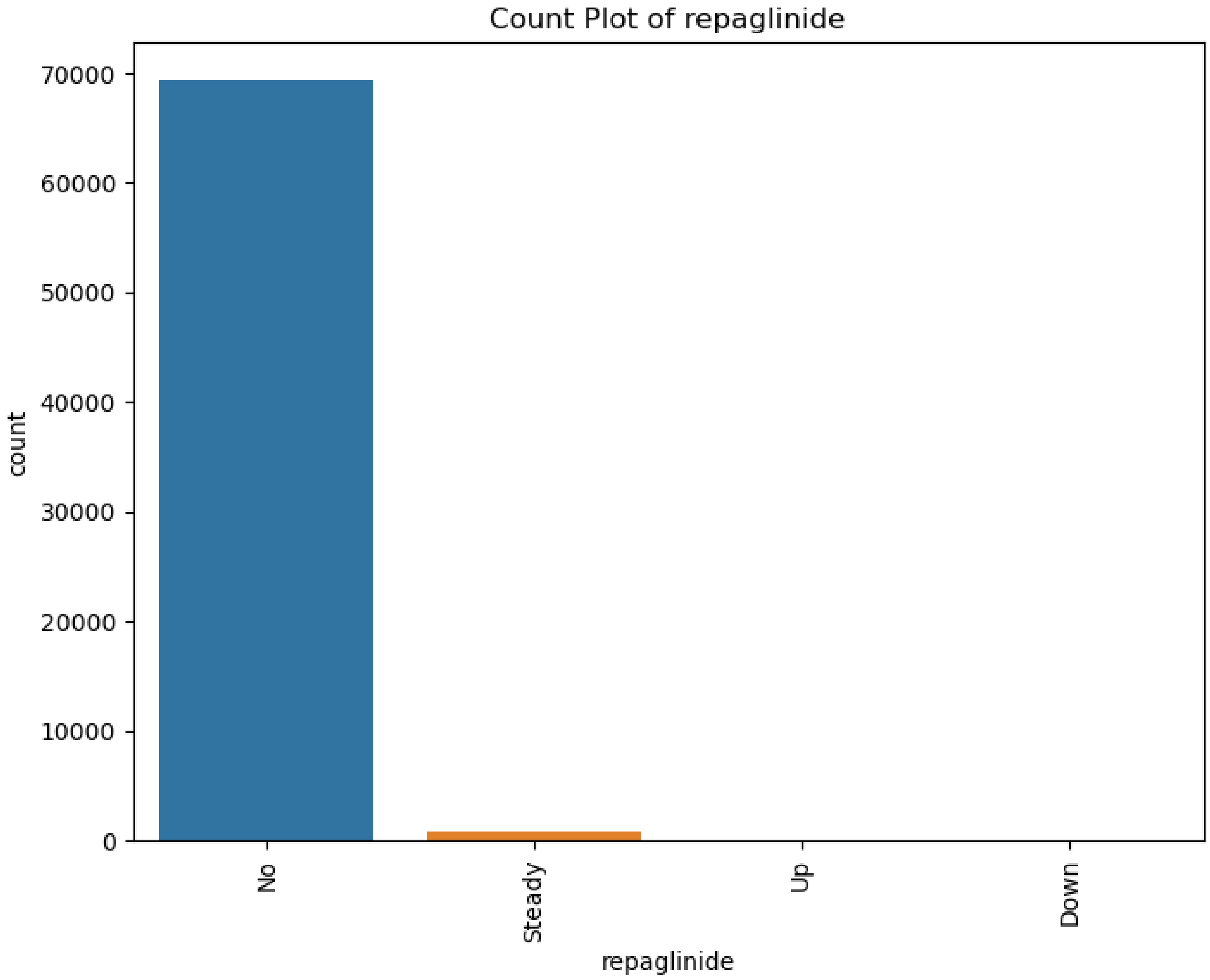

Count Plot of payer_code
count
30000
25000
20000
15000
10000
5000
0
Not Specified
HM
UN
MC
BC
MD
SP
CP
CH
CM
DM
OG
PO
WC
OT
SI
MP
FR
payer_code

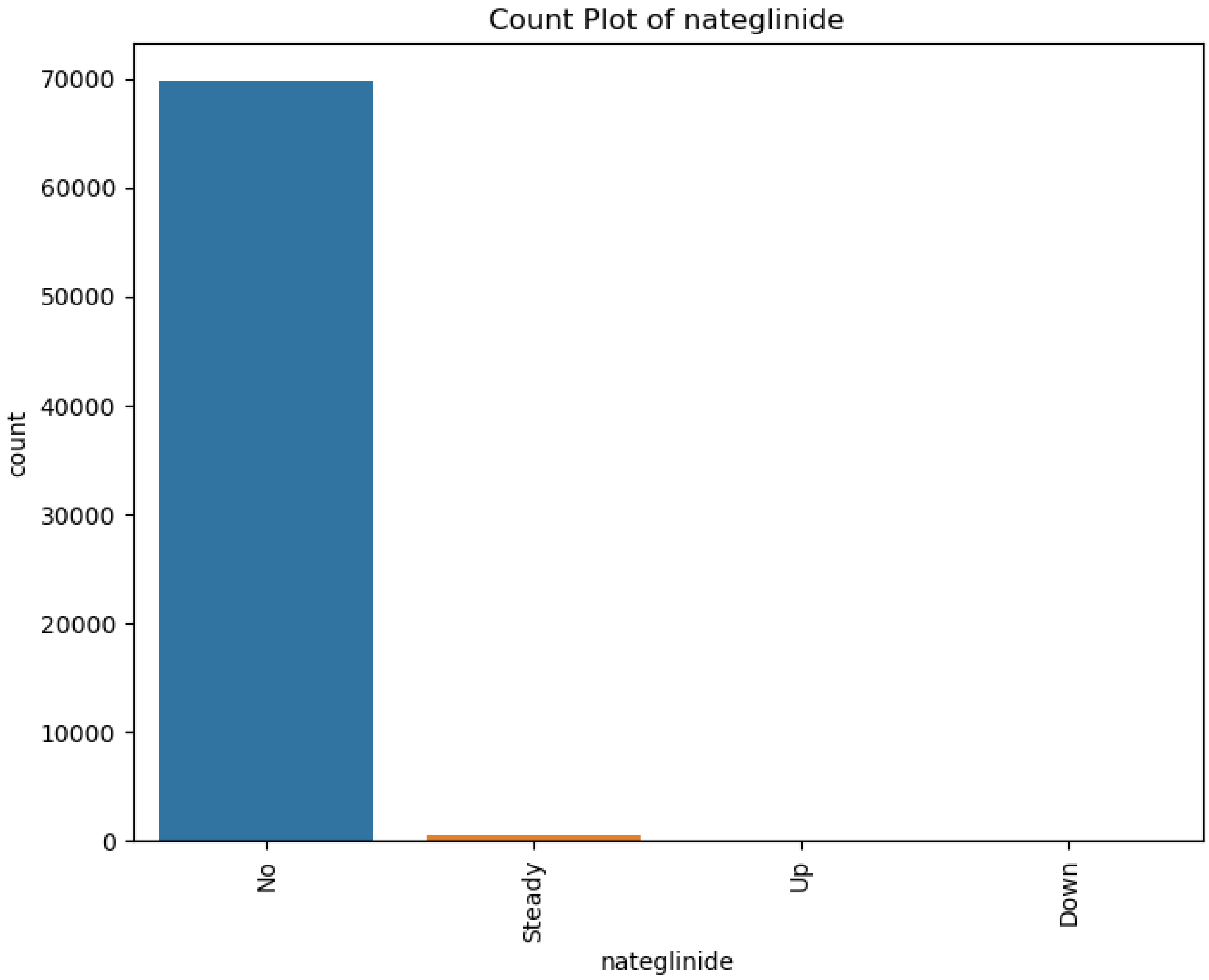

Count Plot of medical_specialty
count
medical_specialty
0
5000
10000
15000
20000
25000
30000
35000

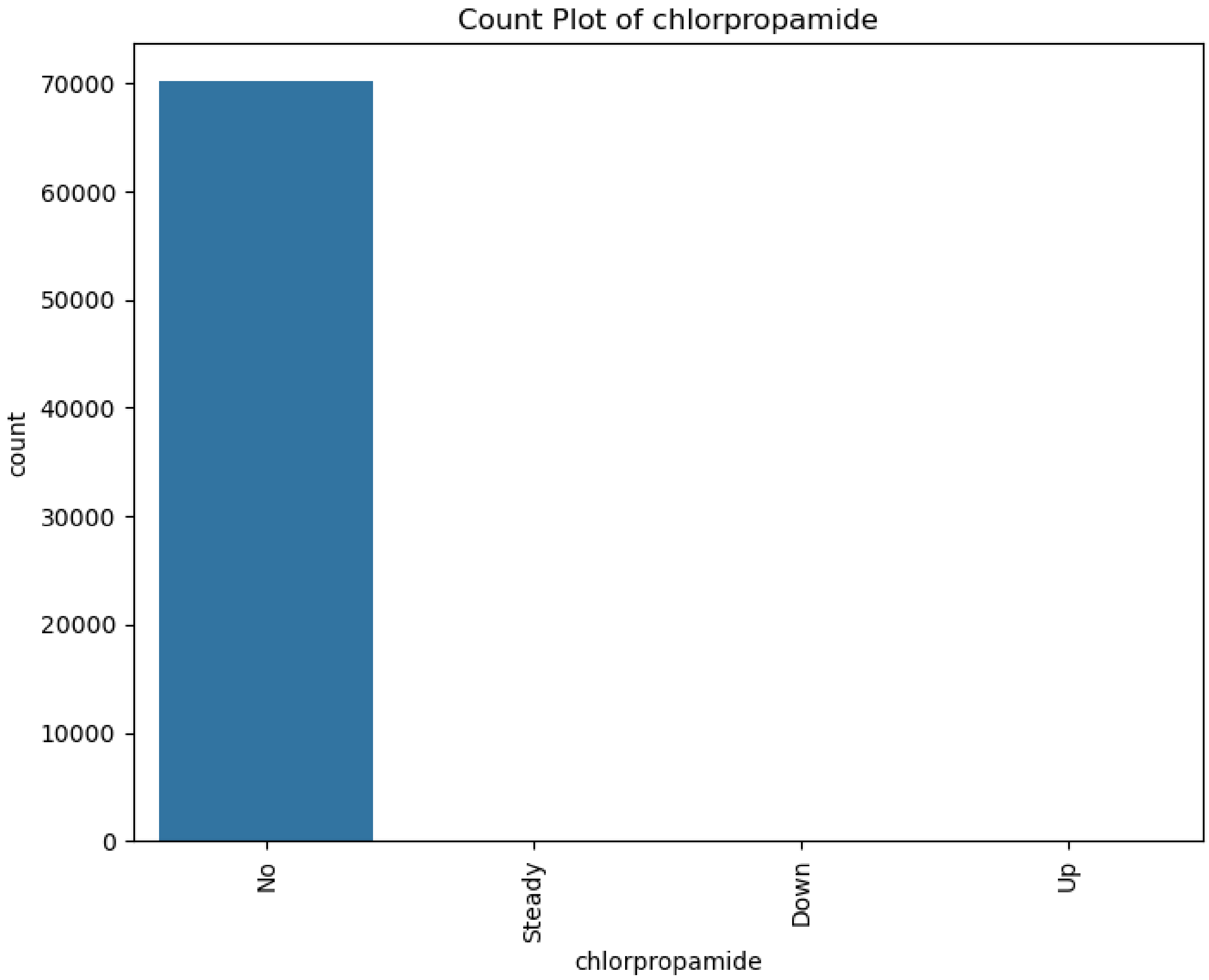

Count Plot of max_glu_serum
count
70000
60000
50000
40000
30000
20000
10000
0
None
>300
Norm
>200
max_glu_serum

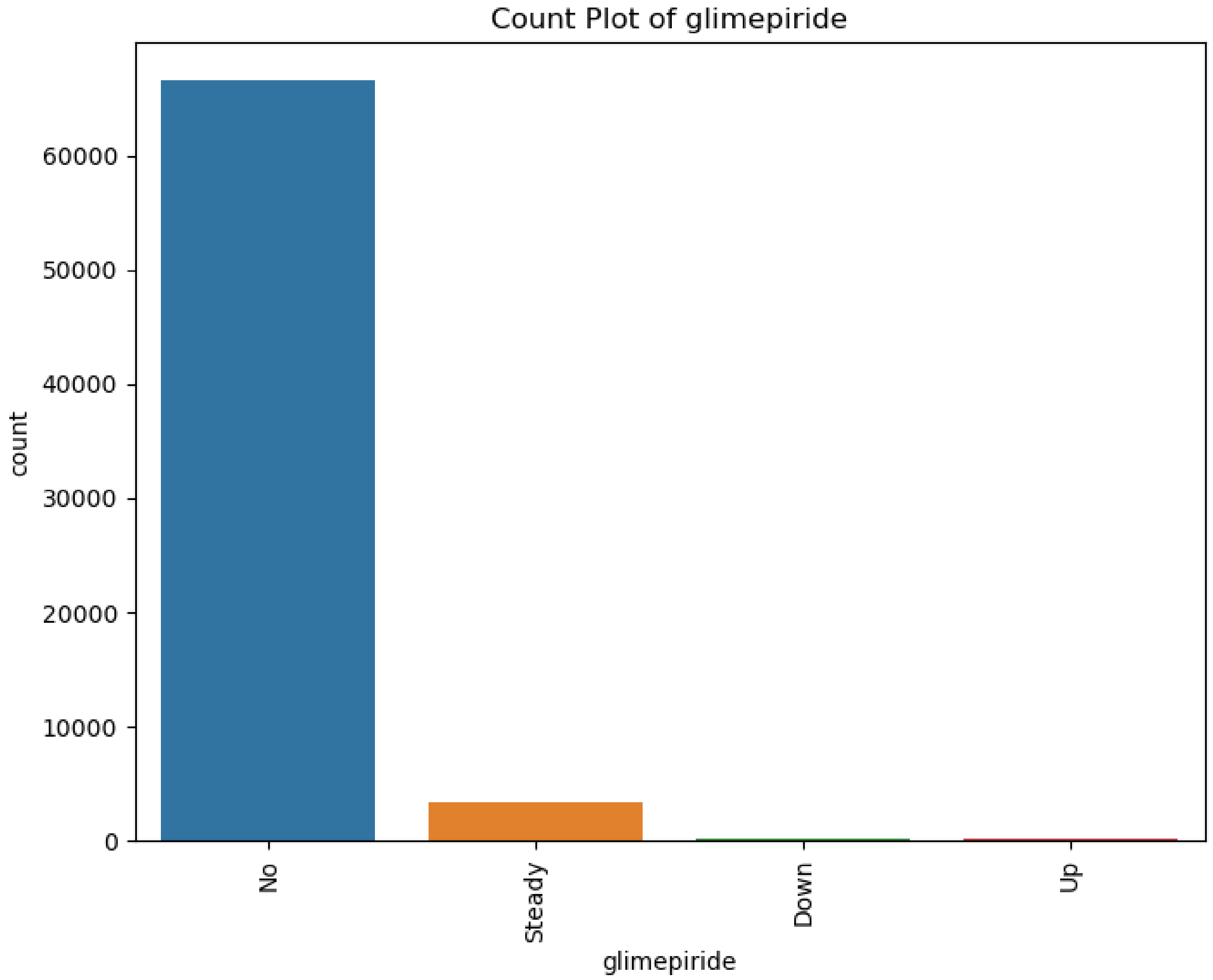

Count Plot of A1Cresult
count
60000
50000
40000
30000
20000
10000
0
>7
None
>8
Norm
A1Cresult

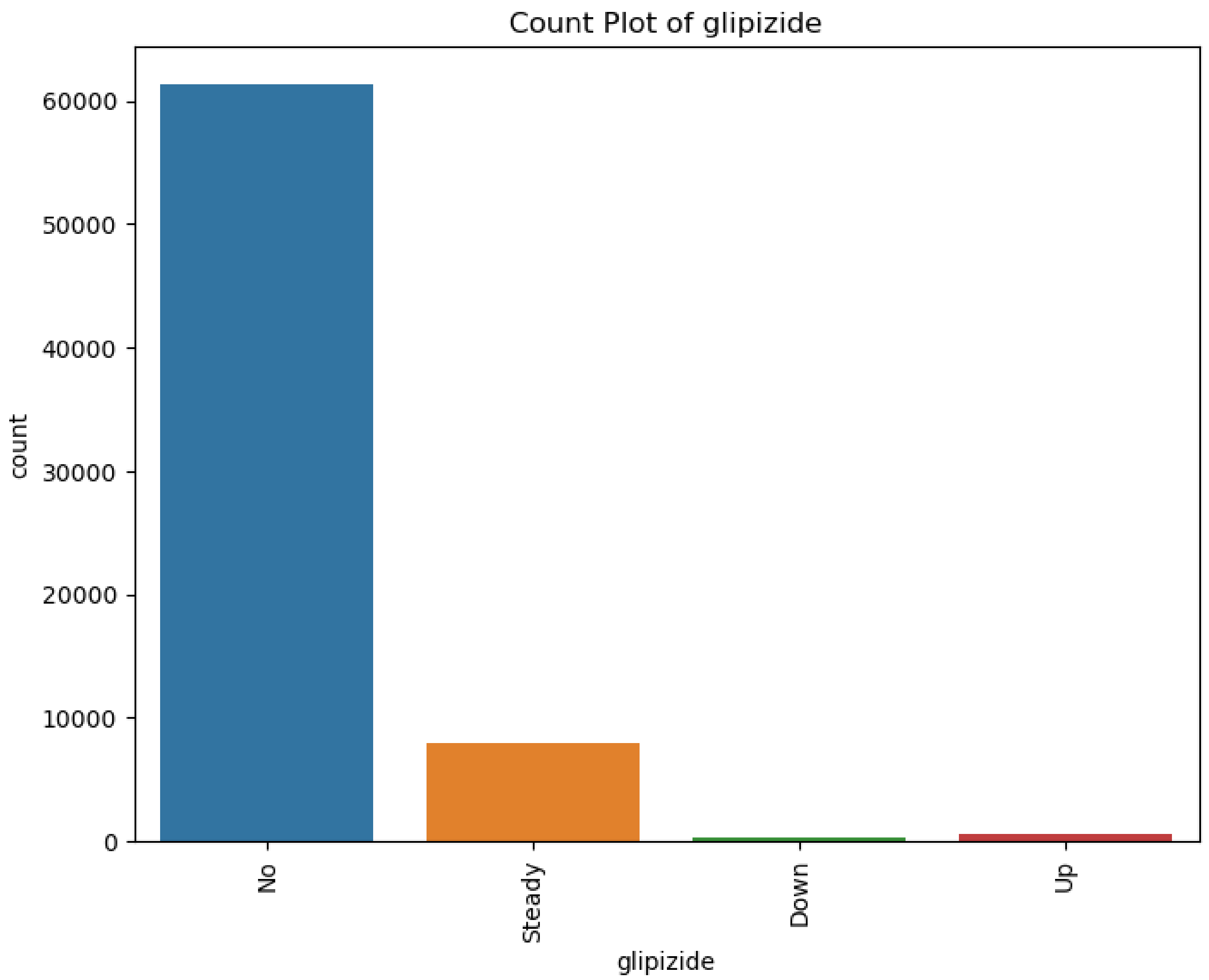

Count Plot of metformin
count
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
metformin


Count Plot of repaglinide
count
70000
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
repaglinide

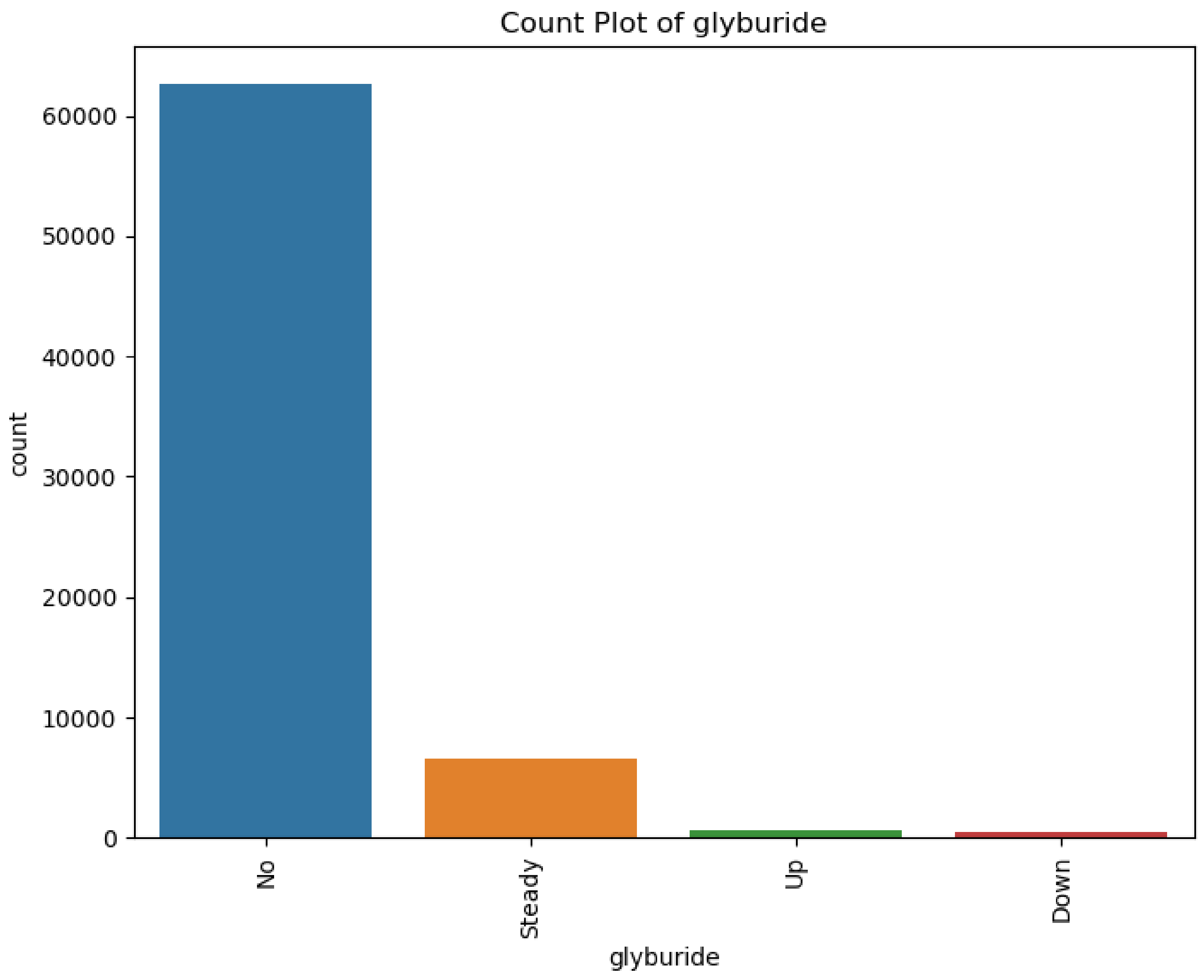

Count Plot of nateglinide
count
70000
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
nateglinide

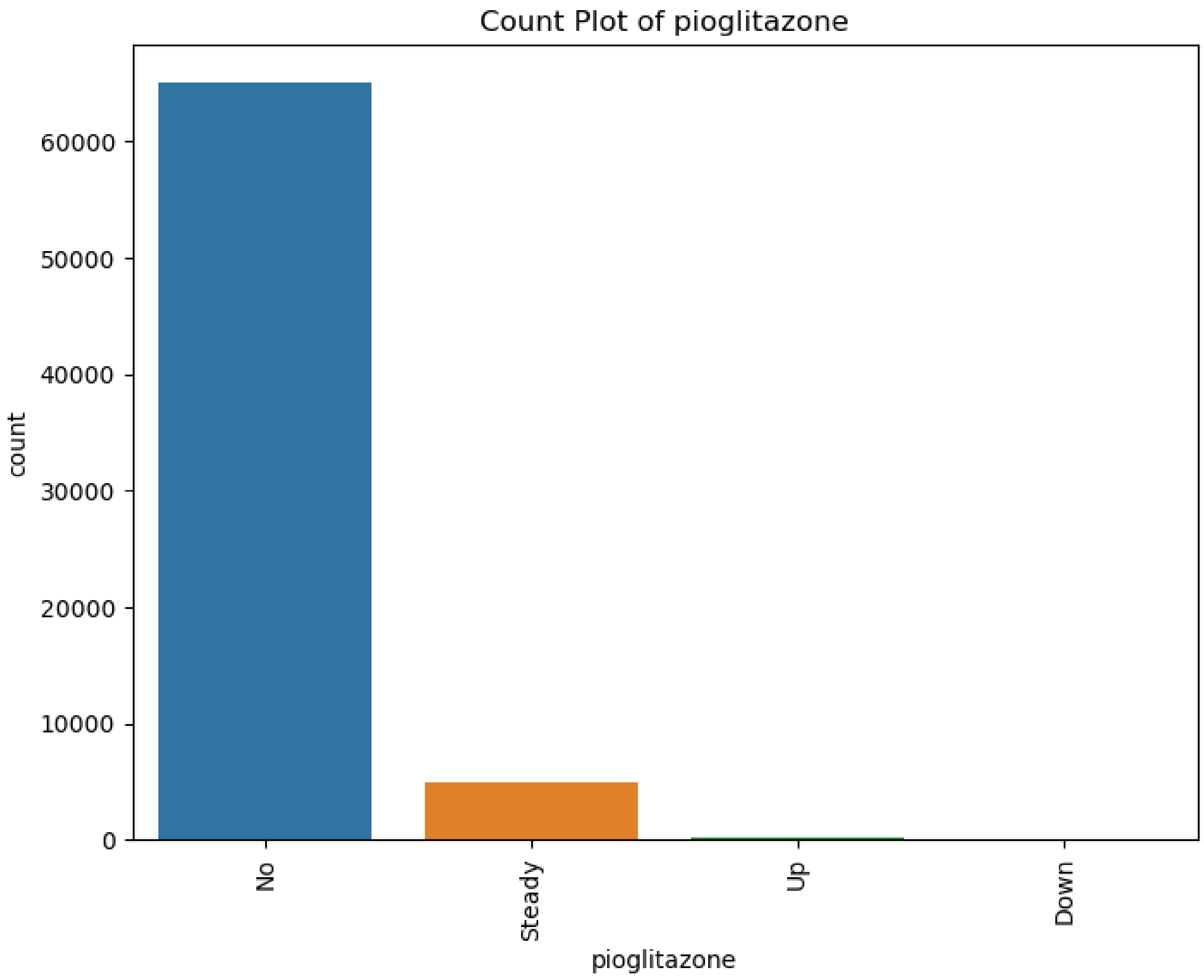

Count Plot of chlorpropamide
count
70000
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
chlorpropamide

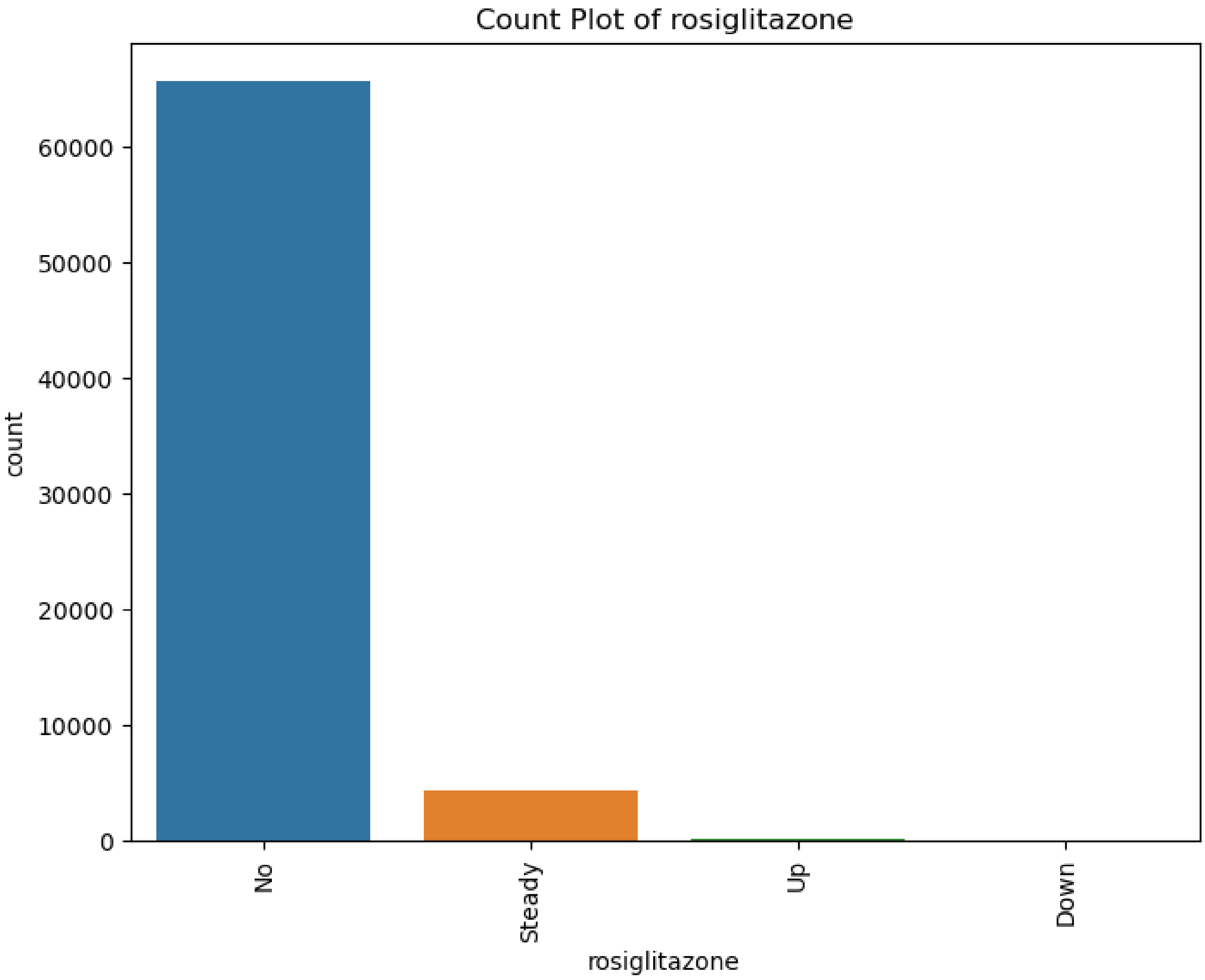

Count Plot of glimepiride
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
glimepiride

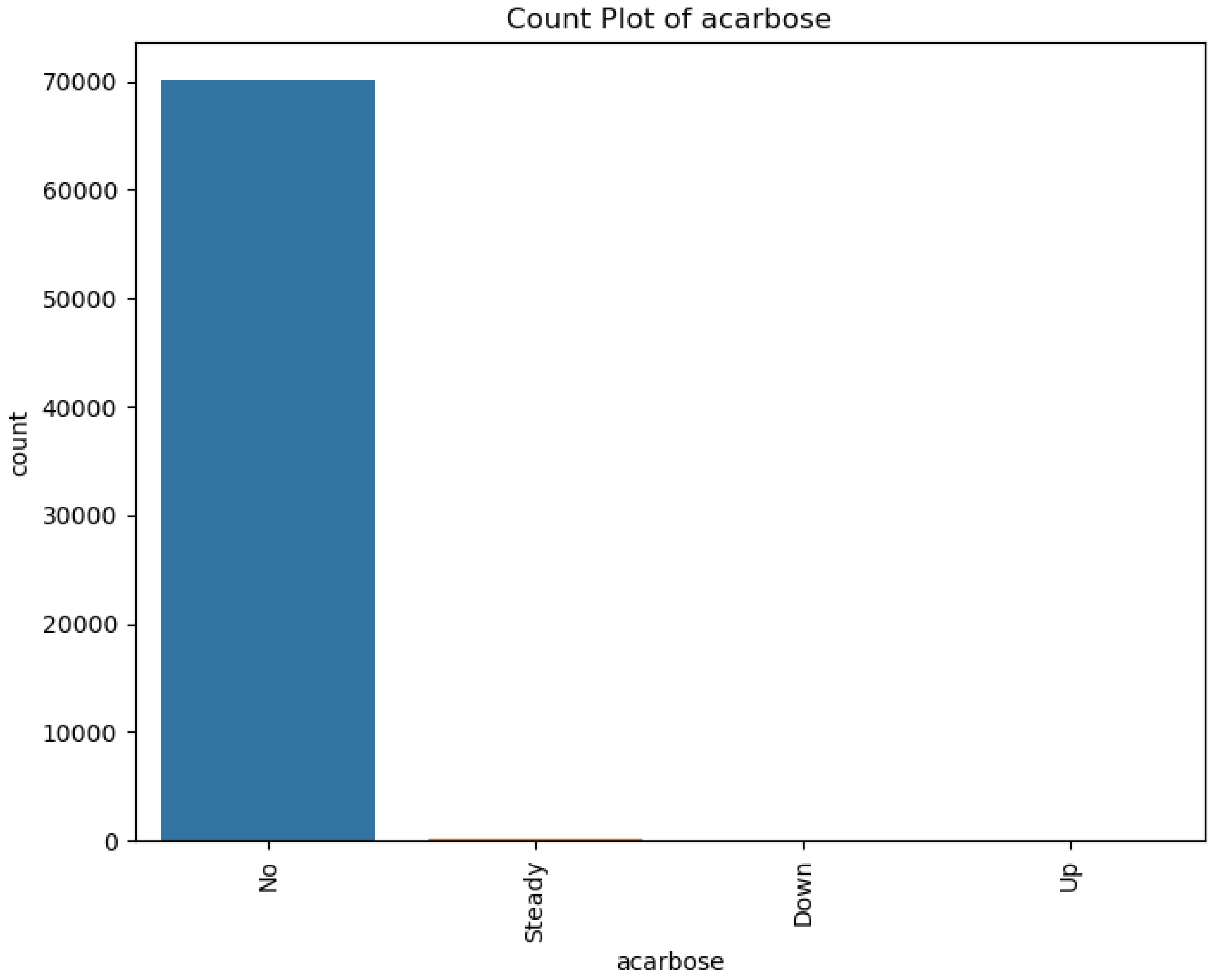


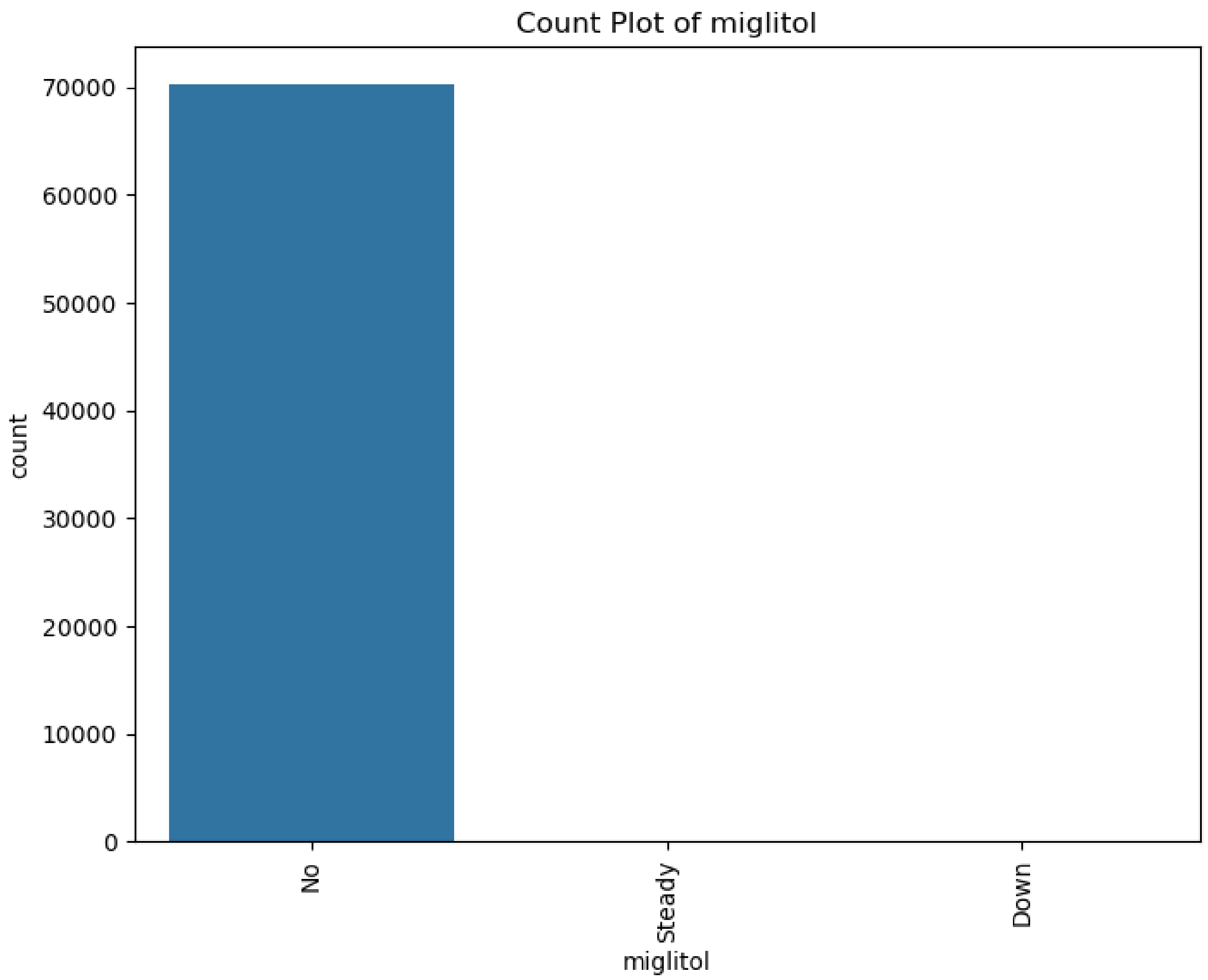
Count Plot of glipizide
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
glipizide

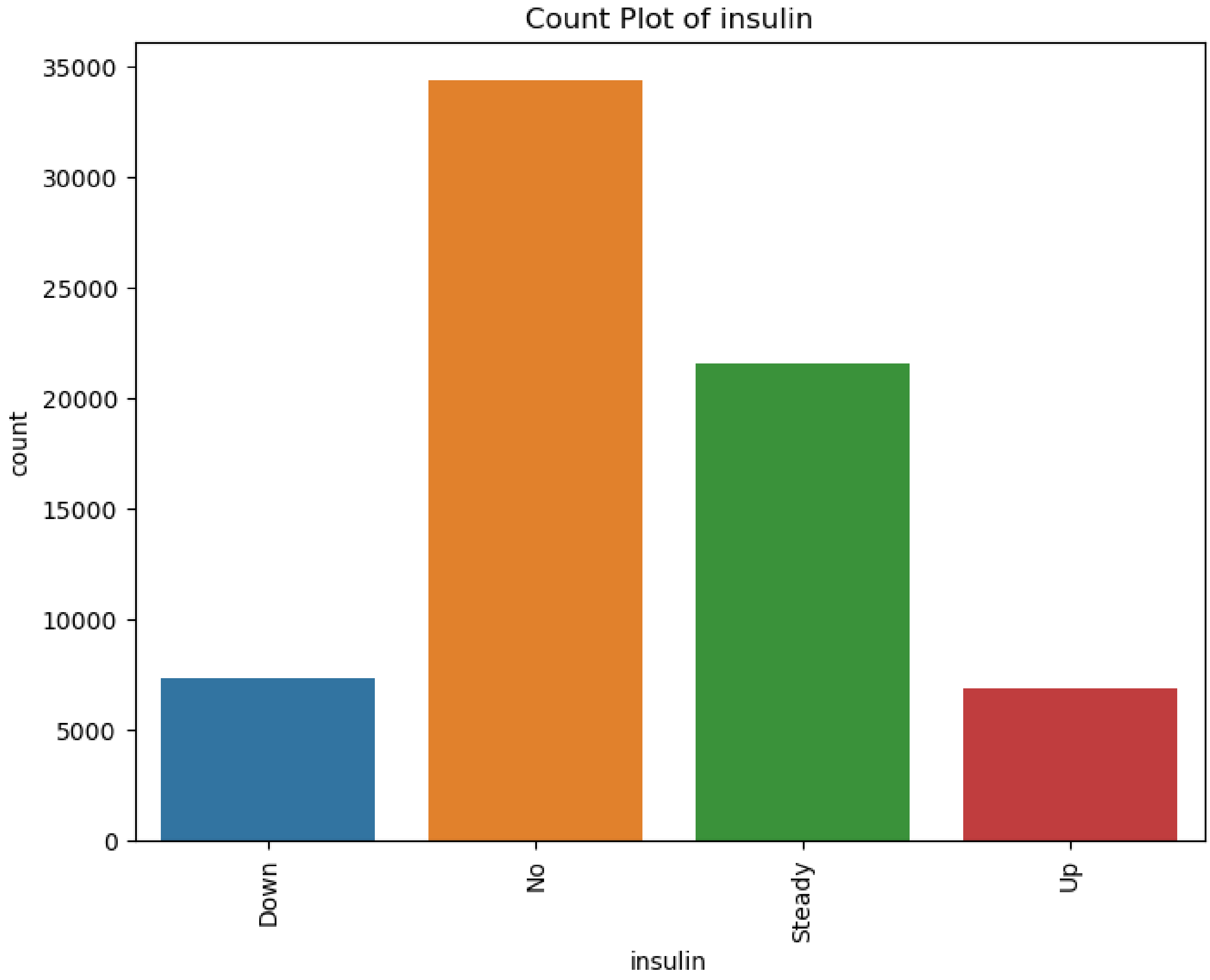

Count Plot of glyburide
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
glyburide

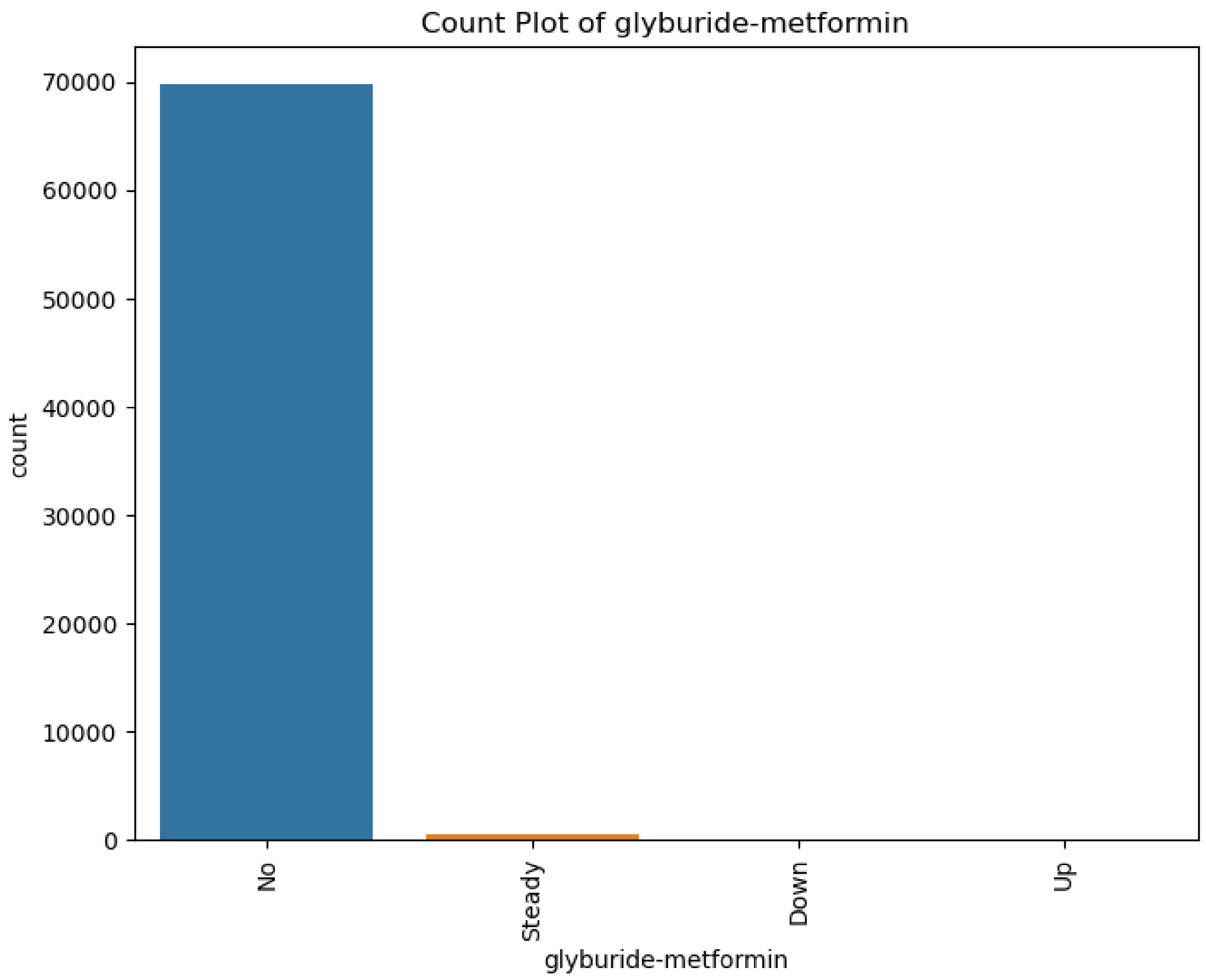

Count Plot of pioglitazone
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
pioglitazone

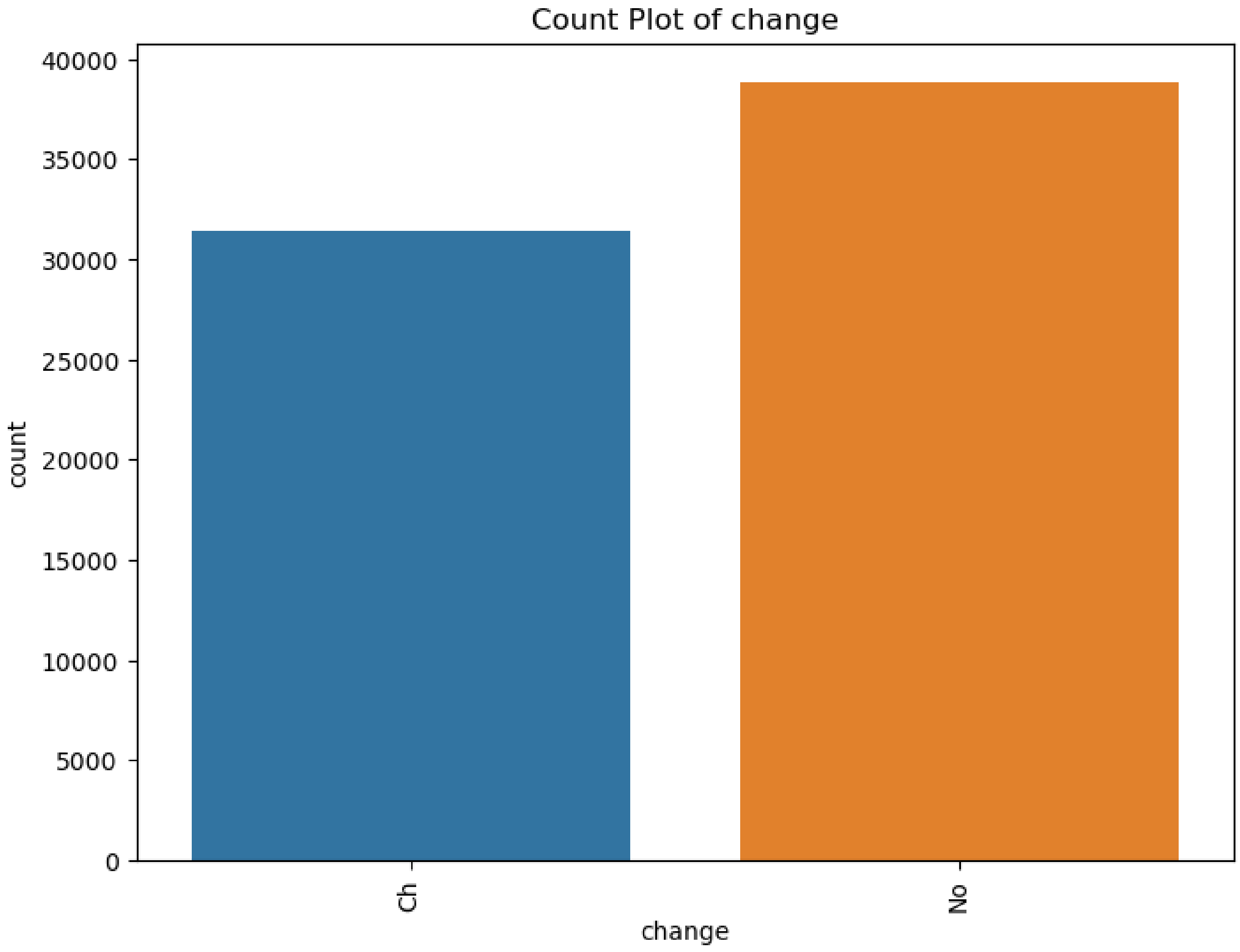

Count Plot of rosiglitazone
count
60000
50000
40000
30000
20000
10000
0
No
Steady
Up
Down
rosiglitazone


Count Plot of acarbose
count
70000
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
acarbose

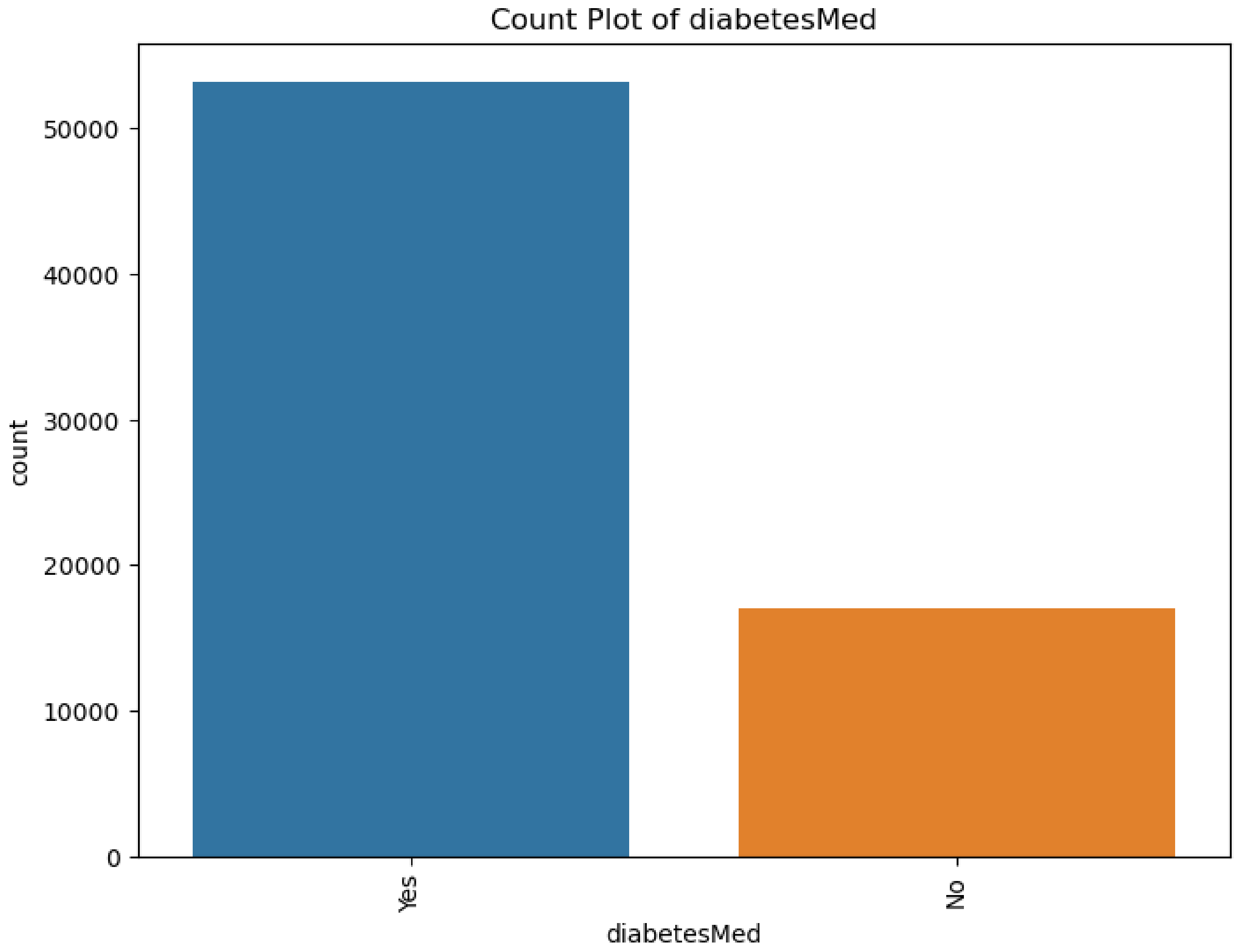

Count Plot of miglitol
count
70000
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
miglitol

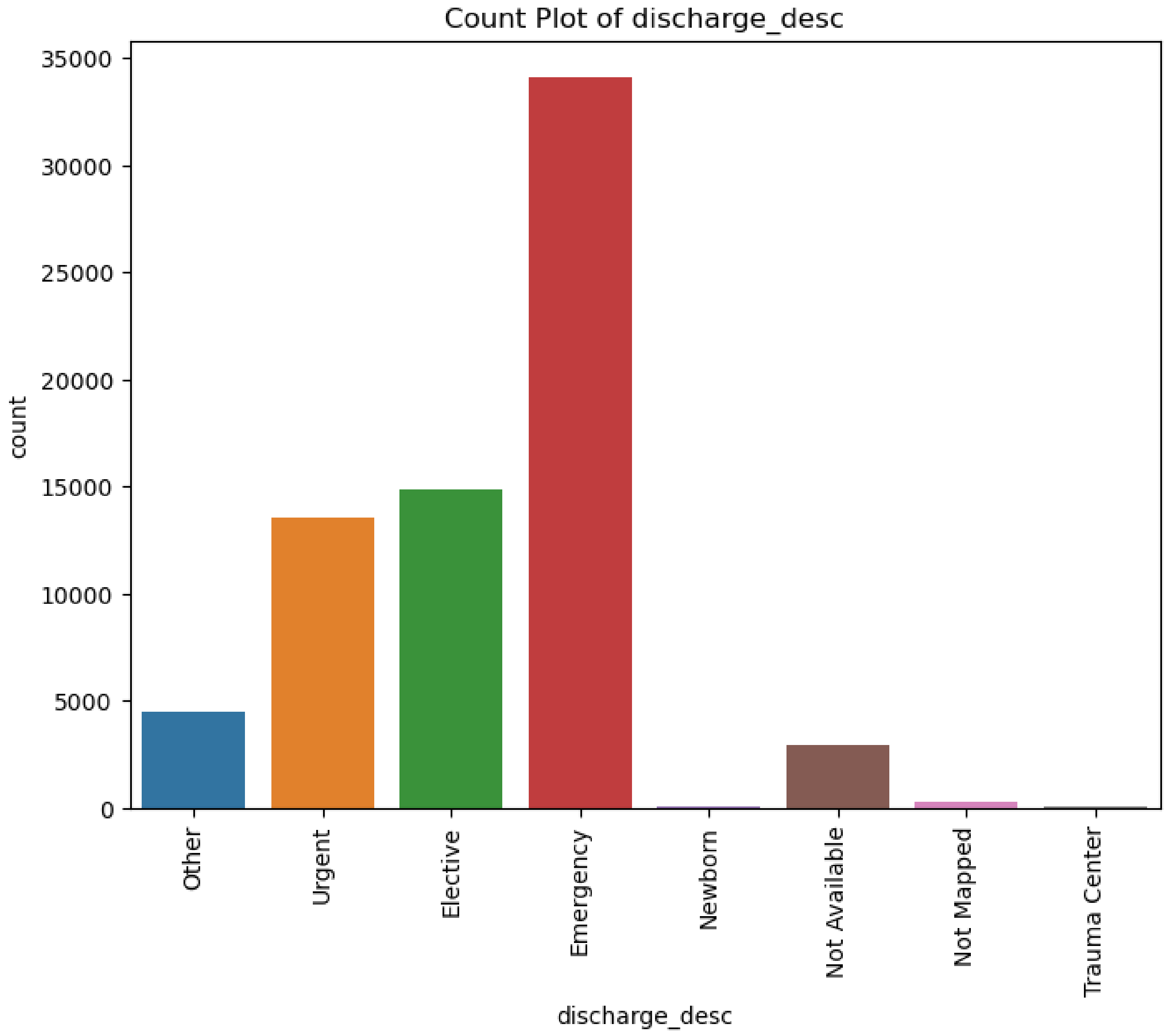

Count Plot of insulin
35000
30000
25000
20000
15000
10000
5000
0
count
Down
No
Steady
Up
insulin

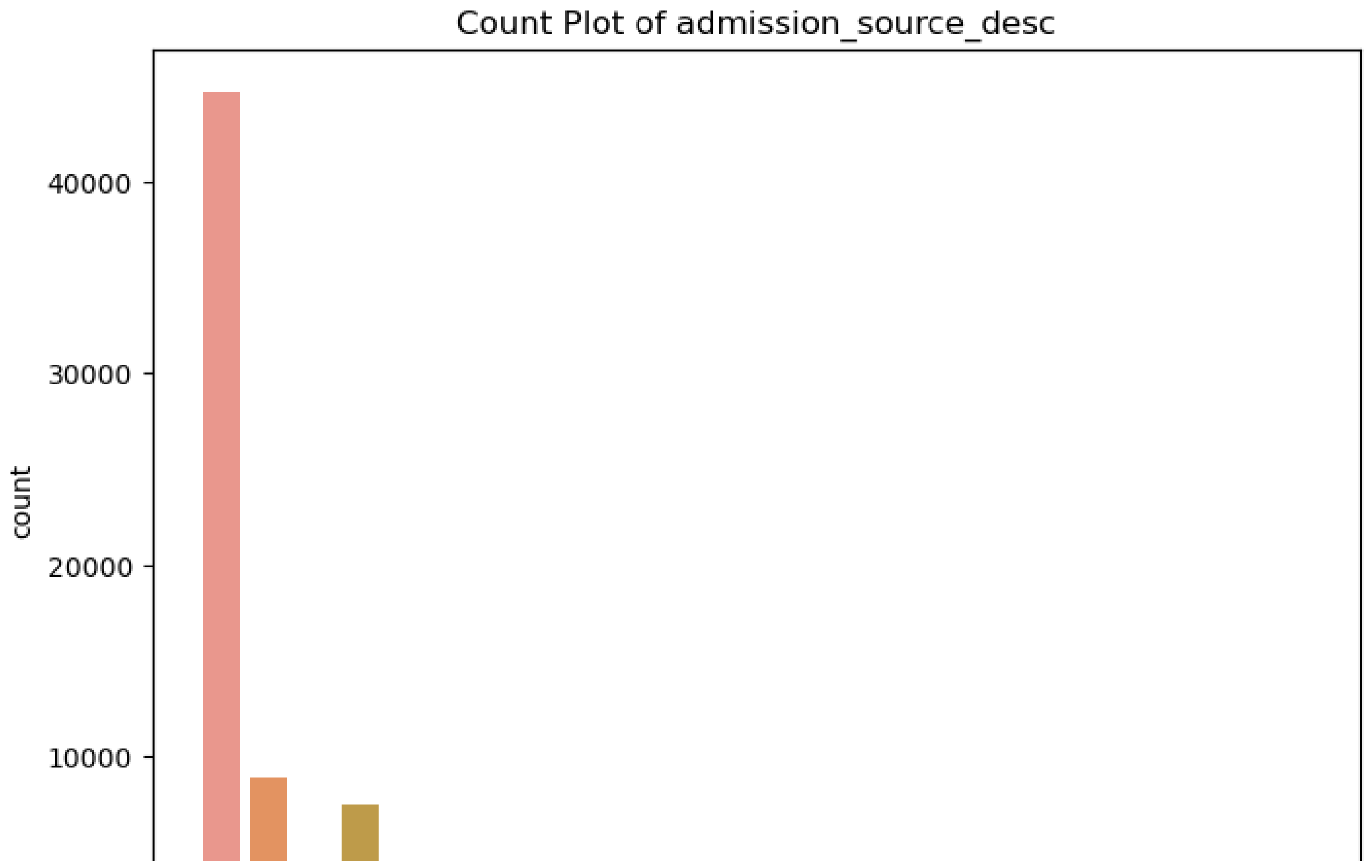

Count Plot of glyburide-metformin
count
70000
60000
50000
40000
30000
20000
10000
0
No
Steady
Down
Up
glyburide-metformin

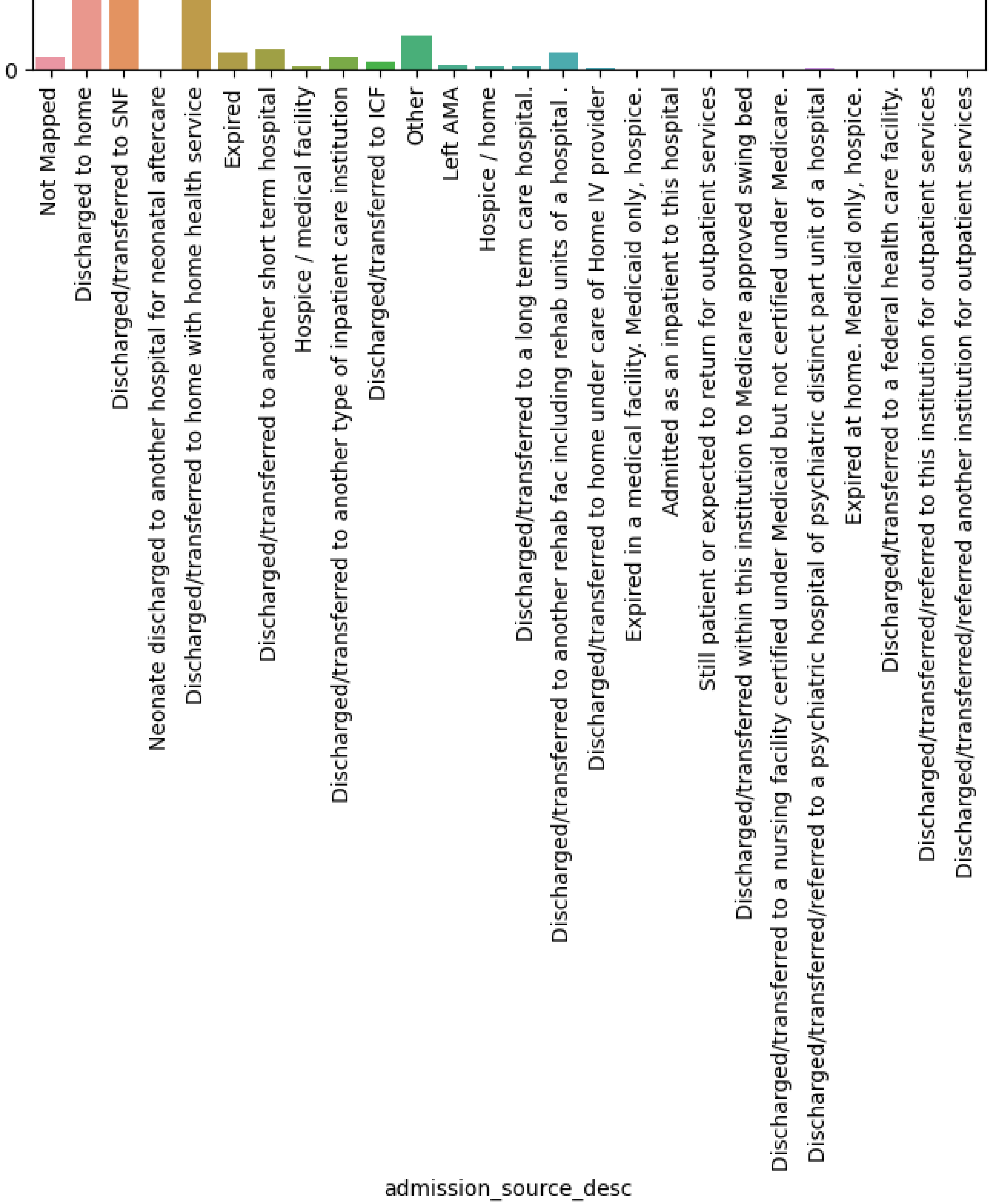


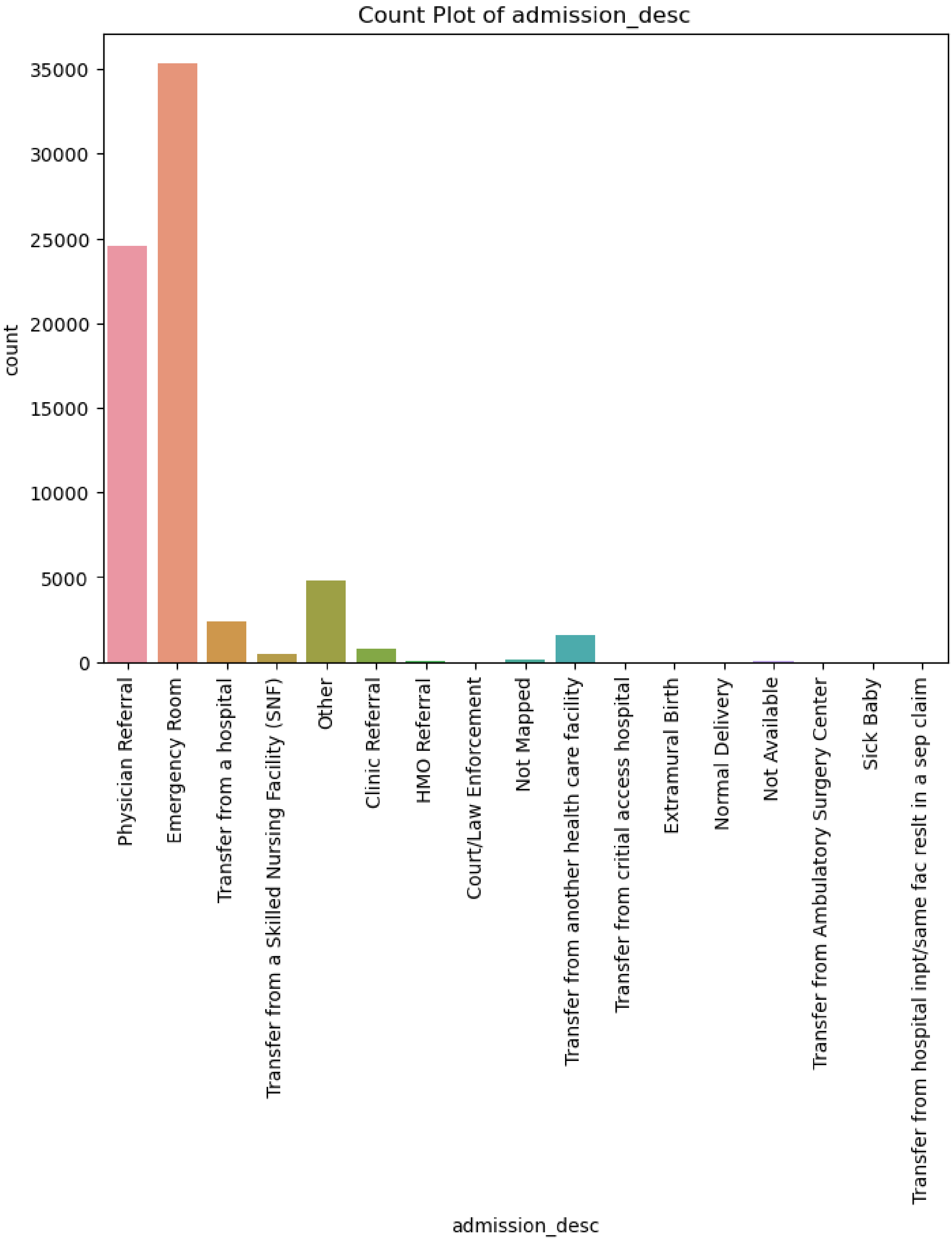
Count Plot of change
40000
35000
30000
25000
20000
15000
10000
5000
0
count
Ch
No
change

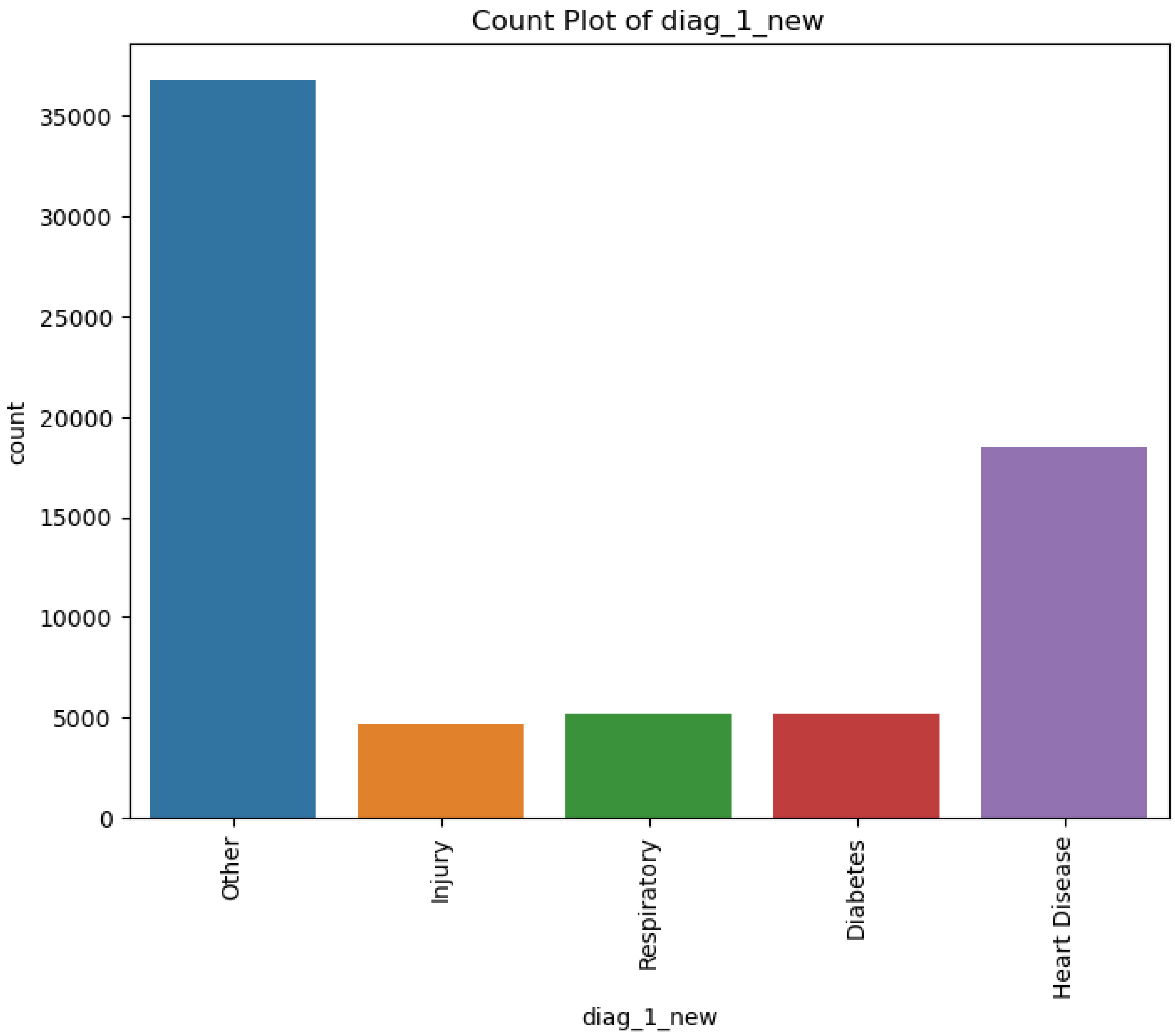

Count Plot of diabetesMed
count
50000
40000
30000
20000
10000
0
Yes
No
diabetesMed

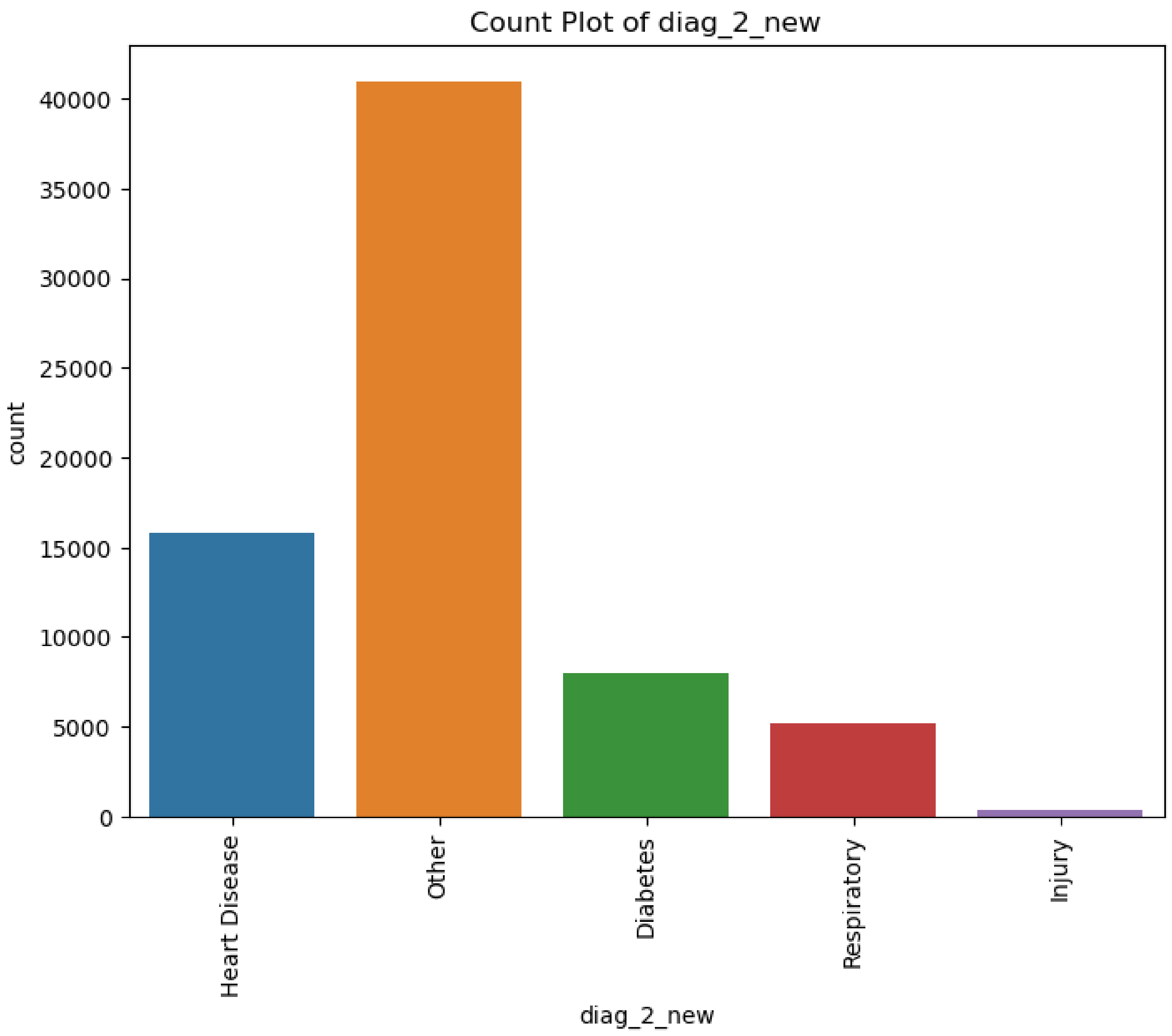

Count Plot of discharge_desc
count
35000
30000
25000
20000
15000
10000
5000
0
Other
Urgent
Elective
Emergency
Newborn
Not Available
Not Mapped
Trauma Center
discharge_desc

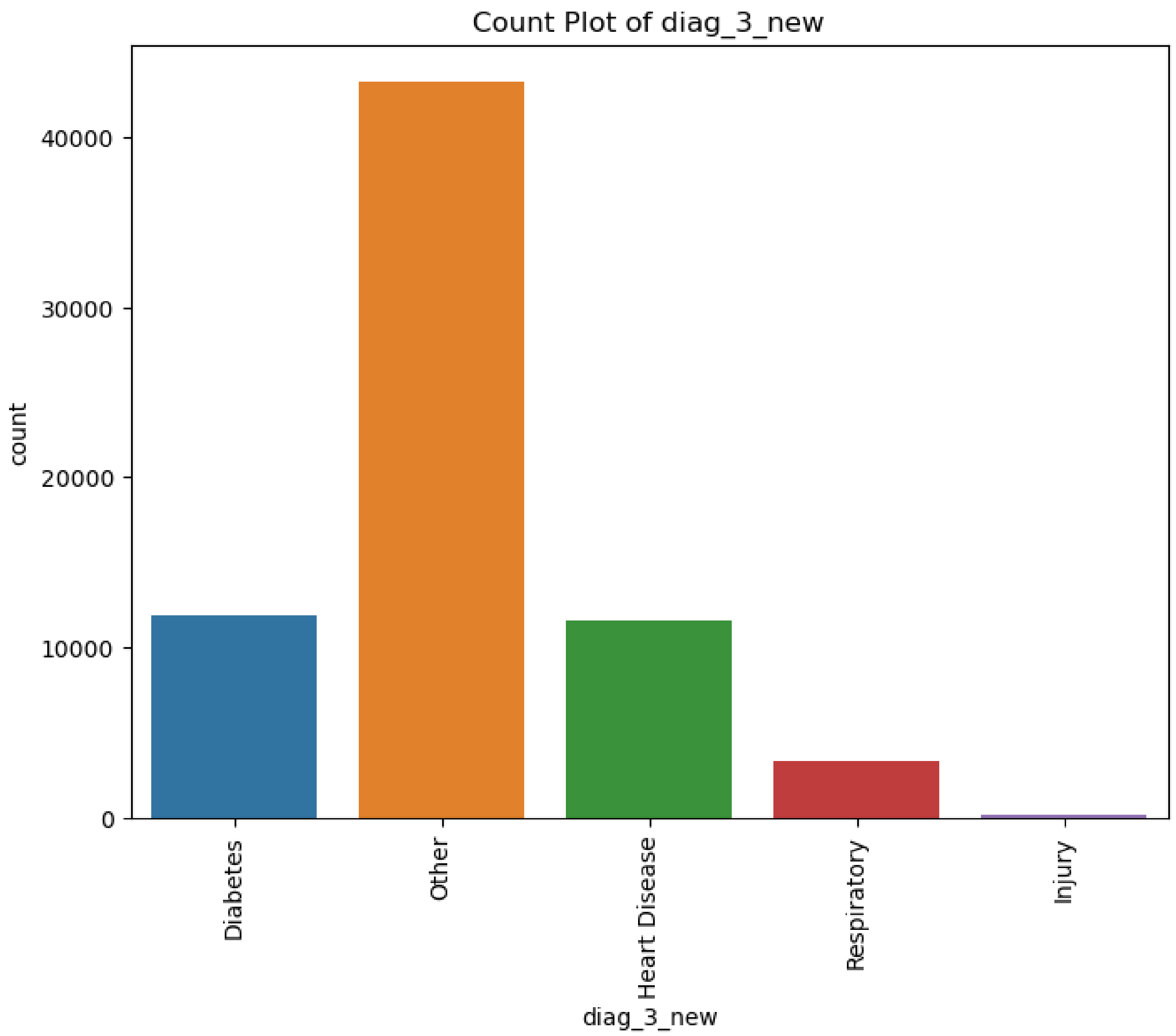

Count Plot of admission_source_desc
count
40000
30000
20000
10000


0
Not Mapped
Discharged to home
Discharged/transferred to SNF
Neonate discharged to another hospital for neonatal aftercare
Discharged/transferred to home with home health service
Expired
Discharged/transferred to another short term hospital
Hospice / medical facility
Discharged/transferred to another type of inpatient care institution
Discharged/transferred to ICF
Other
Left AMA
Hospice / home
Discharged/transferred to a long term care hospital.
Discharged/transferred to another rehab fac including rehab units of a hospital .
Discharged/transferred to home under care of Home IV provider
Expired in a medical facility. Medicaid only, hospice.
Admitted as an inpatient to this hospital
Still patient or expected to return for outpatient services
Discharged/transferred within this institution to Medicare approved swing bed
Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare.
Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital
Expired at home. Medicaid only, hospice.
Discharged/transferred to a federal health care facility.
Discharged/transferred/referred to this institution for outpatient services
Discharged/transferred/referred another institution for outpatient services
admission_source_desc


Count Plot of admission_desc
count
35000
30000
25000
20000
15000
10000
5000
0
Physician Referral
Emergency Room
Transfer from a hospital
Transfer from a Skilled Nursing Facility (SNF)
Other
Clinic Referral
HMO Referral
Court/Law Enforcement
Not Mapped
Transfer from another health care facility
Transfer from critial access hospital
Extramural Birth
Normal Delivery
Not Available
Transfer from Ambulatory Surgery Center
Sick Baby
Transfer from hospital inpt/same fac reslt in a sep claim
admission_desc


Count Plot of diag_1_new
count
35000
30000
25000
20000
15000
10000
5000
0
Other
Injury
Respiratory
Diabetes
Heart Disease
diag_1_new


Count Plot of diag_2_new
count
0
5000
10000
15000
20000
25000
30000
35000
40000
Heart Disease
Other
Diabetes
Respiratory
Injury
diag_2_new



Data Statistics

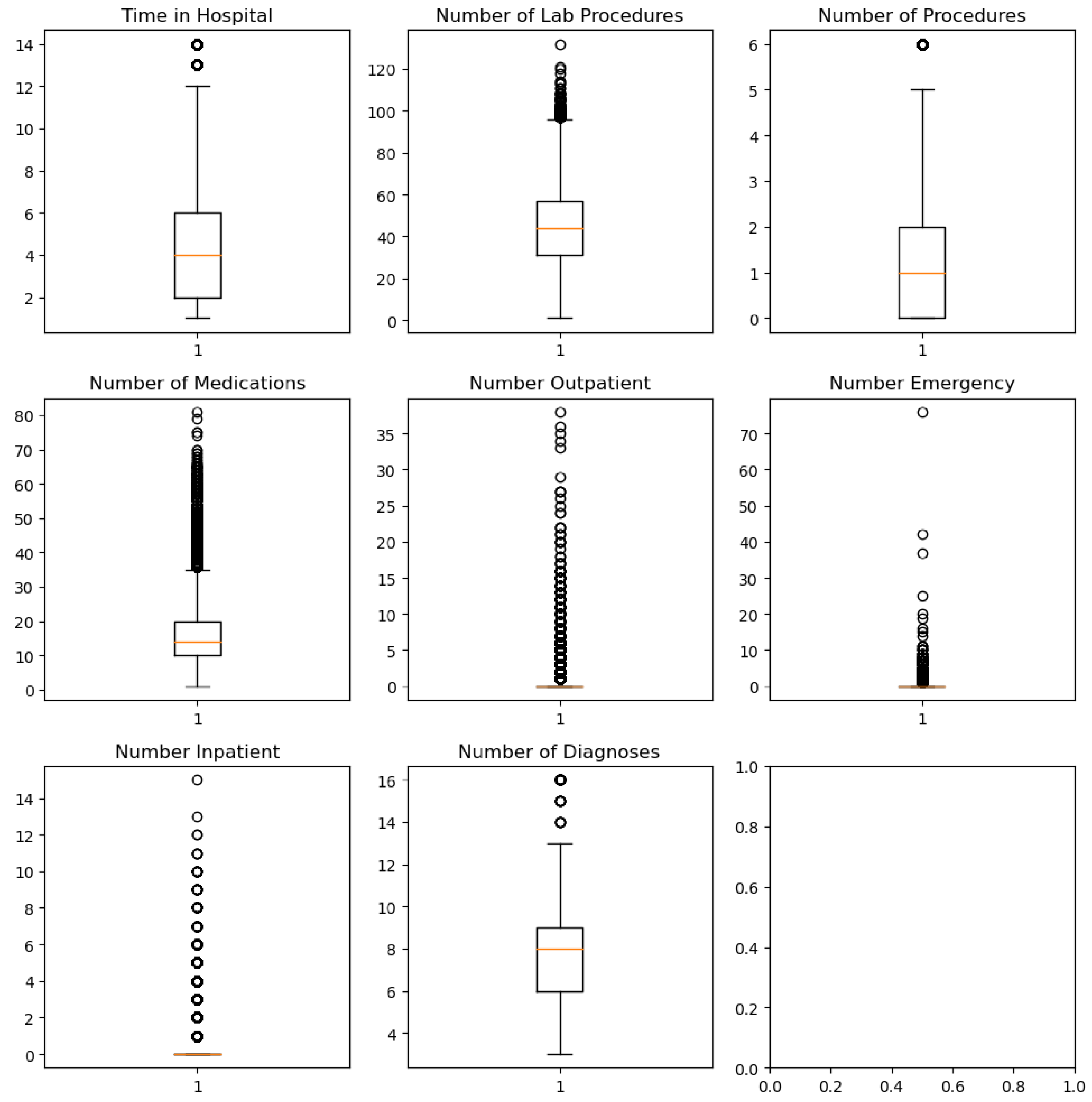
In [39]:
```
# des. stat. - numeric only
df.describe().T
```

Out[39]:

	count	mean	std	min	25%	50%	75%	max
time_in_hospital	70256.0	4.307020	2.956366	1.0	2.0	4.0	6.0	14.0
num_lab_procedures	70256.0	42.961085	19.914422	1.0	31.0	44.0	57.0	132.0
num_procedures	70256.0	1.444873	1.761094	0.0	0.0	1.0	2.0	6.0
num_medications	70256.0	15.850675	8.264550	1.0	10.0	14.0	20.0	81.0
number_outpatient	70256.0	0.301668	1.117115	0.0	0.0	0.0	0.0	38.0
number_emergency	70256.0	0.116816	0.619245	0.0	0.0	0.0	0.0	76.0
number_inpatient	70256.0	0.274909	0.745316	0.0	0.0	0.0	0.0	15.0
number_diagnoses	70256.0	7.346462	1.887392	3.0	6.0	8.0	9.0	16.0
binary_readmitted	70256.0	0.079096	0.269891	0.0	0.0	0.0	0.0	1.0

Boxplot of Int Features

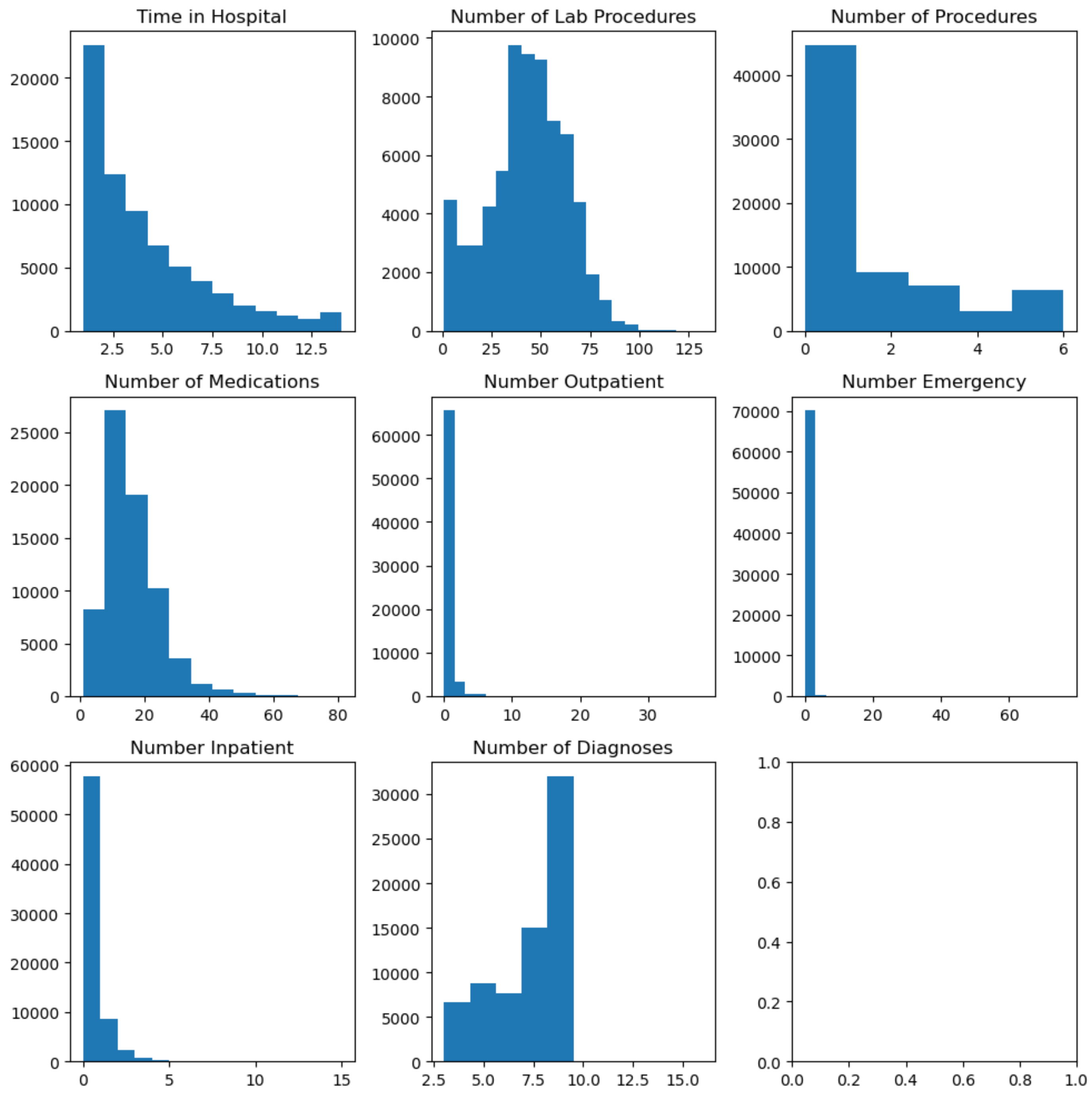
In [40]:
```
fig, axes = plt.subplots(nrows=3,ncols=3,figsize = (10,10))

axes[0,0].boxplot(df['time_in_hospital'])
axes[0,0].set_title('Time in Hospital')

axes[0,1].boxplot(df['num_lab_procedures'])
axes[0,1].set_title('Number of Lab Procedures')

axes[0,2].boxplot(df['num_procedures'])
axes[0,2].set_title('Number of Procedures')

axes[1,0].boxplot(df['num_medications'])
axes[1,0].set_title('Number of Medications')

axes[1,1].boxplot(df['number_outpatient'])
axes[1,1].set_title('Number Outpatient')

axes[1,2].boxplot(df['number_emergency'])
axes[1,2].set_title('Number Emergency')

axes[2,0].boxplot(df['number_inpatient'])
axes[2,0].set_title('Number Inpatient')

axes[2,1].boxplot(df['number_diagnoses'])
axes[2,1].set_title('Number of Diagnoses')
plt.tight_layout()
```



Histogram of Int Features

In [41]:
```
fig, axes = plt.subplots(nrows=3,ncols=3,figsize = (10,10))

axes[0,0].hist(df['time_in_hospital'], bins = 12)
axes[0,0].set_title('Time in Hospital')

axes[0,1].hist(df['num_lab_procedures'], bins = 20)
axes[0,1].set_title('Number of Lab Procedures')

axes[0,2].hist(df['num_procedures'], bins = 5)
axes[0,2].set_title('Number of Procedures')

axes[1,0].hist(df['num_medications'], bins = 12)
axes[1,0].set_title('Number of Medications')

axes[1,1].hist(df['number_outpatient'], bins = 25)
axes[1,1].set_title('Number Outpatient')
```

```
axes[1,2].hist(df['number_emergency'], bins = 25)
axes[1,2].set_title('Number Emergency')

axes[2,0].hist(df['number_inpatient'], bins = 15)
axes[2,0].set_title('Number Inpatient')

axes[2,1].hist(df['number_diagnoses'], bins = 10)
axes[2,1].set_title('Number of Diagnoses')

plt.tight_layout()
```



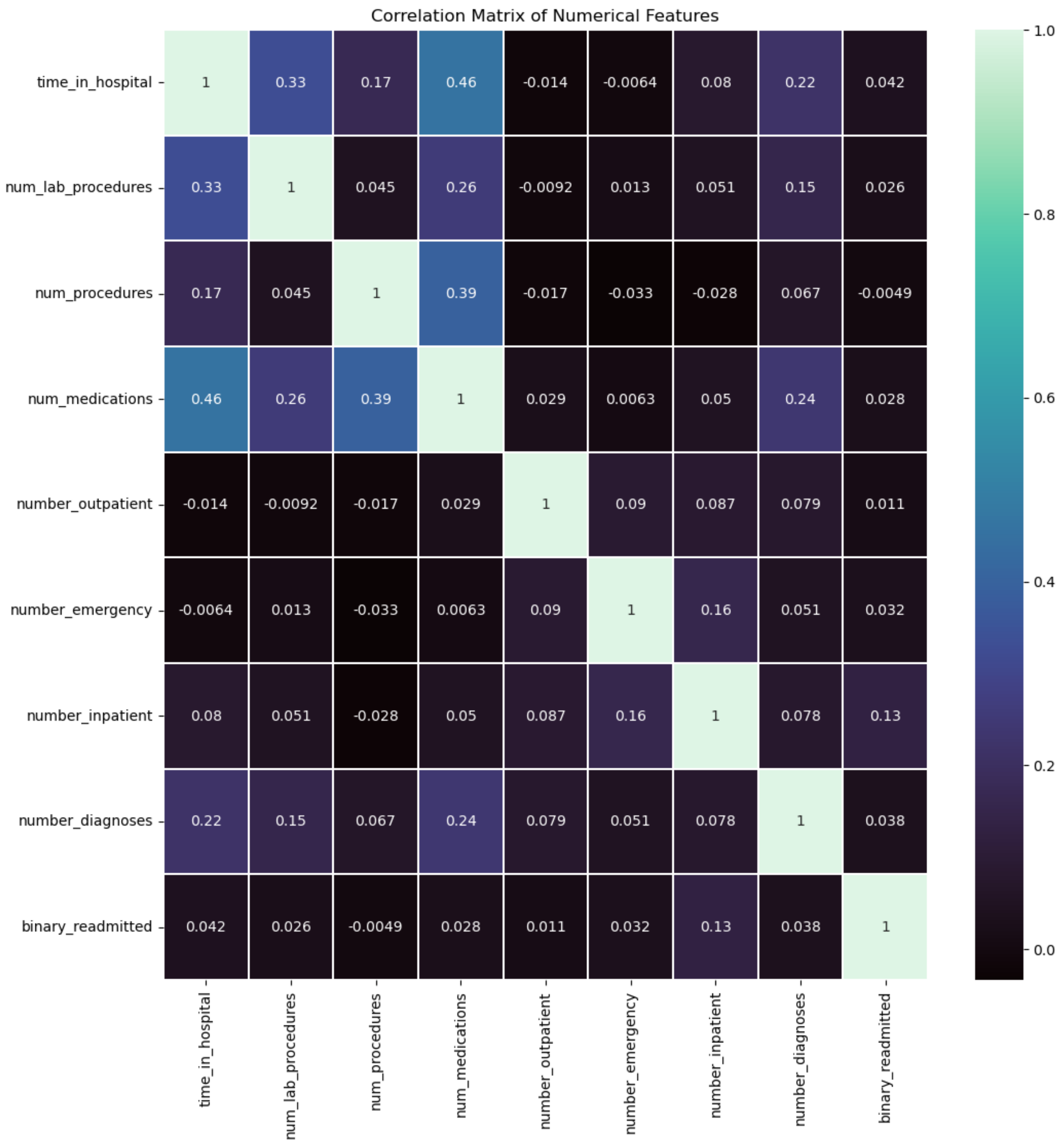
Dummy and Scaling

In [42]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
X = df.drop(['binary_readmitted'], axis = 1)
y = df['binary_readmitted']

X_train, X_test, y_train, y_test = train_test_split(X,y,train_size = .8, random_state=12

object_columns= X_train.select_dtypes(include = 'object')
X_train_encoded = pd.get_dummies(X_train, columns= object_columns.columns, drop_first =
X_test_encoded = pd.get_dummies(X_test, columns = object_columns.columns, drop_first = T

## Account for missing columns in X_test that are in X_train
missing_columns = set(X_train_encoded.columns) - set(X_test_encoded.columns)

# Add missing columns to X_test and fill them with zeros
for col in missing_columns:
    X_test_encoded[col] = 0

int_columns = X_train_encoded.select_dtypes(include = 'int')
scaler = StandardScaler()
X_train_encoded[int_columns.columns] = scaler.fit_transform(X_train_encoded[int_columns.
X_test_encoded[int_columns.columns] = scaler.transform(X_test_encoded[int_columns.column
```

In [43]:
```
correl = df.corr()
plt.figure(figsize = (12,12))
sns.heatmap(correl, cmap = 'mako',linewidth = .2, linecolor= 'white', annot=True)
plt.title('Correlation Matrix of Numerical Features')
plt.show()
```

```
C:\Users\Joey\AppData\Local\Temp\ipykernel_4540\1708422235.py:1: FutureWarning: The defa
ult value of numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only to sile
nce this warning.
  correl = df.corr()
```



Lasso / Logistic Regression + Grid Search

In [44]:
```
from sklearn.linear_model import LassoCV, LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc

# Define range of alpha values
alphas = np.logspace(-2, 2, 11)

# Create the LassoCV model with CV alpha selection
lasso_model = LassoCV(alphas=alphas, cv=5)

# Fit the LassoCV model to train
```

```
lasso_model.fit(X_train_encoded, y_train)

# Get the best alpha value
best_alpha = lasso_model.alpha_

# Apply the same one-hot encoding to the test data
object_columns_test = X_test.select_dtypes(include='object')
X_test_encoded = pd.get_dummies(X_test, columns=object_columns_test.columns, drop_first=

# Ensure the test data has the same columns as the training data
missing_columns = set(X_train_encoded.columns) - set(X_test_encoded.columns)
for column in missing_columns:
    X_test_encoded[column] = 0  # Add missing columns with all zeros

# Reorder columns in X_test_encoded to match X_train_encoded
X_test_encoded = X_test_encoded[X_train_encoded.columns]

# Create the Lasso logistic regression model with the best alpha value
lasso_logistic_best = LogisticRegression(penalty='l1', solver='saga', C=1 / best_alpha,

# Fit the model to the training data
lasso_logistic_best.fit(X_train_encoded, y_train)
```

Out[44]:

```
LogisticRegression
LogisticRegression(C=100.0, max_iter=10000, penalty='l1', random_state=12,
                   solver='saga')
```

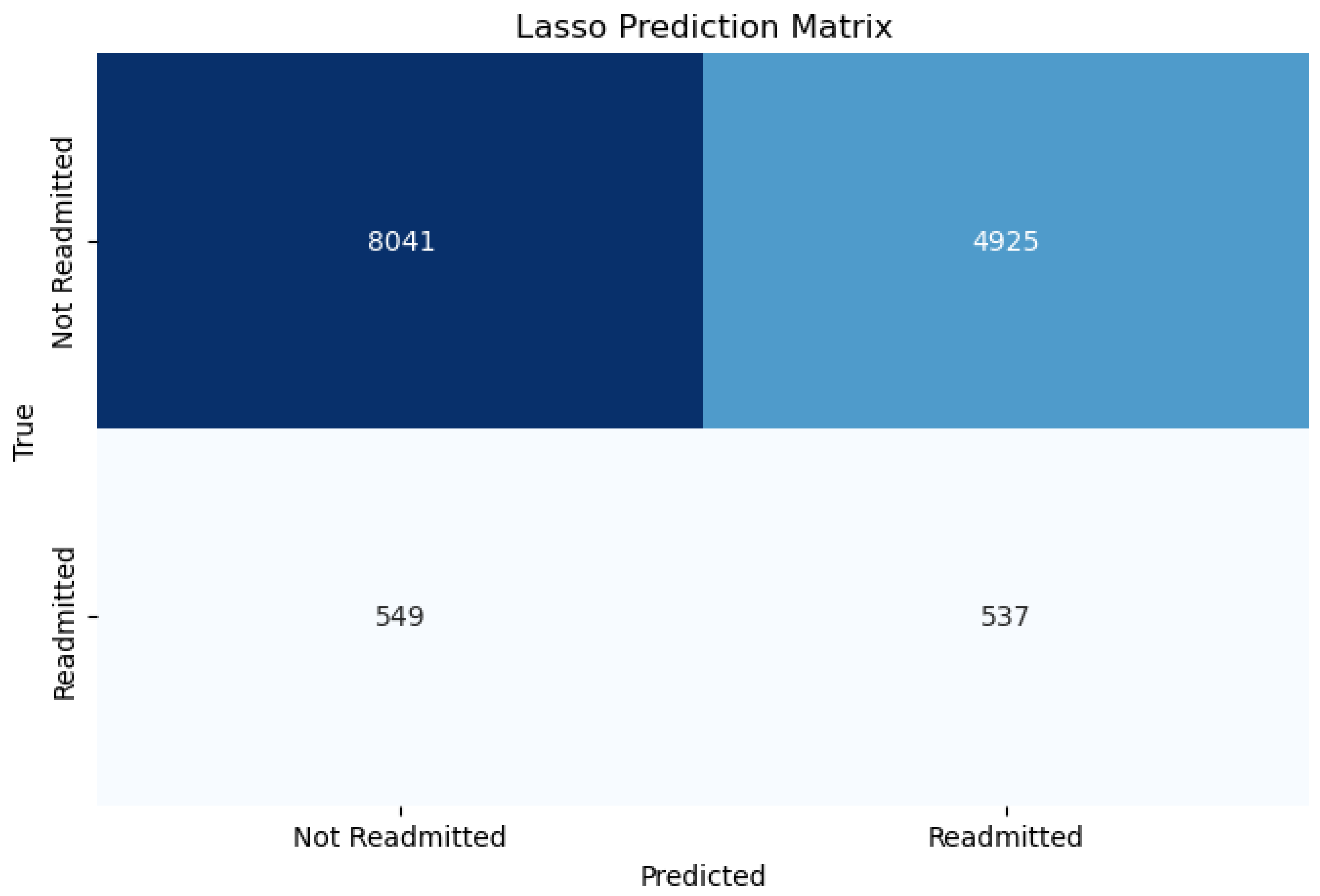
Predictions, Coefficients, Classification Report, ROC, and Confusion Matrix

In [45]:

```
# Make predictions on the test set
y_pred_prob = lasso_logistic_best.predict_proba(X_test_encoded)[:, 1]
y_pred = (y_pred_prob >= 0.65).astype(int)   # Change threshold ***

# Calculate accuracy on the test set
test_accuracy = accuracy_score(y_test, y_pred)

# Display features with non-zero coefficients
coefficients = lasso_logistic_best.coef_
feature_names = X_train_encoded.columns
coef_df = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients[0]})
non_zero_coef_df = coef_df[coef_df['Coefficient'] != 0]

print("Features with Non-Zero Coefficients:")
print(non_zero_coef_df)

print(f"Best Alpha (Lambda): {best_alpha:.4f}")
print(f"Test Set Accuracy: {test_accuracy:.4f}")

# threshold
user_defined_threshold = 0.65

# Calculate classification metrics with threshold
y_pred_custom_threshold = (y_pred_prob >= user_defined_threshold).astype(int)

classification_report_custom_threshold = classification_report(y_test, y_pred_custom_thr
confusion_matrix_custom_threshold = confusion_matrix(y_test, y_pred_custom_threshold)

print(f"User-Defined Threshold: {user_defined_threshold:.2f}")
print("Classification Report with Custom Threshold:")
print(classification_report_custom_threshold)
```

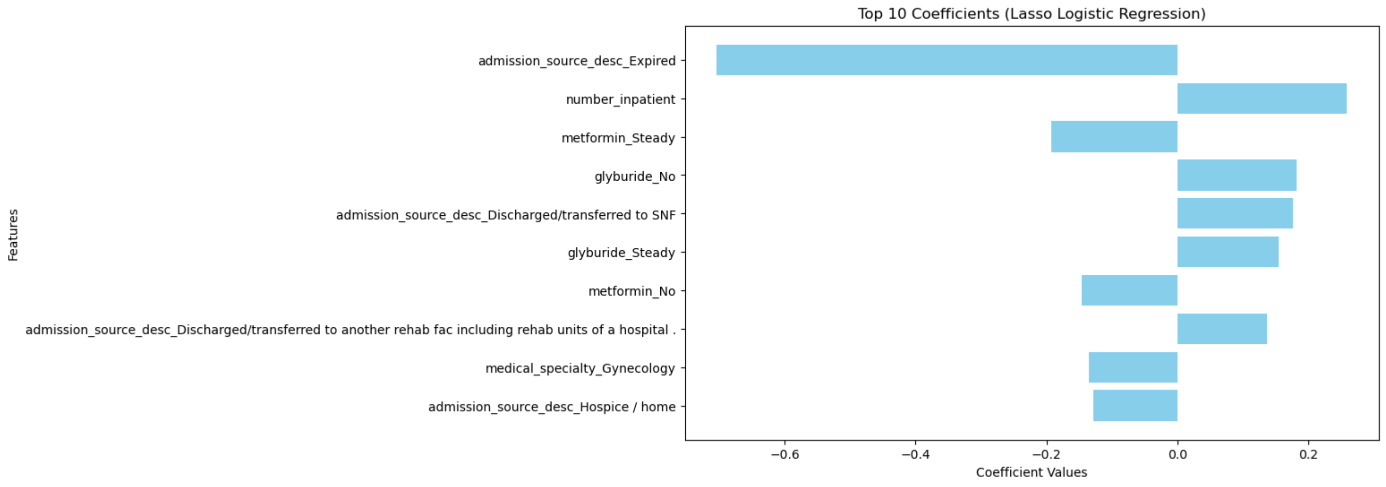
```
custom_label = ['Not Readmitted','Readmitted']
# Plot confusion matrix with threshold
plt.figure(figsize=(8, 5))
sns.heatmap(confusion_matrix_custom_threshold, annot=True, fmt="d", cmap="Blues", cbar=F
            xticklabels=custom_label, yticklabels=custom_label)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Lasso Prediction Matrix")
plt.show()
```

```
Features with Non-Zero Coefficients:
                      Feature  Coefficient
0            time_in_hospital     0.022256
1          num_lab_procedures     0.037550
2              num_procedures    -0.024361
3             num_medications     0.030043
4            number_outpatient    0.008690
..                        ...          ...
209    diag_2_new_Respiratory    -0.050334
210  diag_3_new_Heart Disease    -0.031457
211         diag_3_new_Injury    -0.004772
212          diag_3_new_Other    -0.034090
213    diag_3_new_Respiratory     0.005635

[214 rows x 2 columns]
Best Alpha (Lambda): 0.0100
Test Set Accuracy: 0.6104
User-Defined Threshold: 0.65
Classification Report with Custom Threshold:
              precision    recall  f1-score   support

           0       0.94      0.62      0.75     12966
           1       0.10      0.49      0.16      1086

    accuracy                           0.61     14052
   macro avg       0.52      0.56      0.46     14052
weighted avg       0.87      0.61      0.70     14052
```



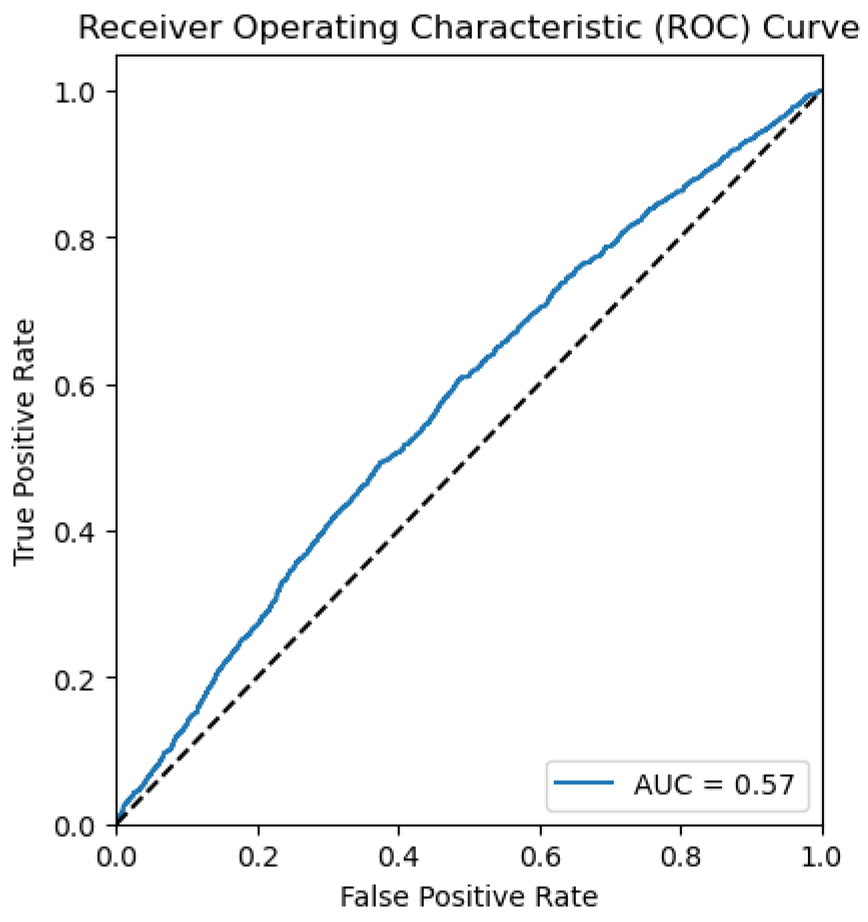
In [46]:
```
# Get the coefficients and corresponding feature names
coefficients = lasso_logistic_best.coef_
feature_names = X_train_encoded.columns.tolist()
coef_df = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients[0]})

# Sort the coefficients by absolute value in descending order
sorted_coef_df = coef_df.reindex(coef_df['Coefficient'].abs().sort_values(ascending=Fals

# Select the top 10 coefficients
top_10_coef = sorted_coef_df.head(10)
print(top_10_coef)
# Create a bar plot for the top 10 coefficients
plt.figure(figsize=(10, 6))
plt.barh(top_10_coef['Feature'], top_10_coef['Coefficient'], color='skyblue')
plt.xlabel('Coefficient Values')
plt.ylabel('Features')
plt.title('Top 10 Coefficients (Lasso Logistic Regression)')
plt.gca().invert_yaxis()
plt.show()
```

```
                                               Feature  Coefficient
176                       admission_source_desc_Expired    -0.704329
6                                     number_inpatient     0.258546
115                                    metformin_Steady    -0.192531
132                                        glyburide_No     0.180965
163  admission_source_desc_Discharged/transferred t...     0.175976
133                                    glyburide_Steady     0.154605
114                                        metformin_No    -0.146662
167  admission_source_desc_Discharged/transferred t...     0.136092
51                         medical_specialty_Gynecology    -0.134850
179                 admission_source_desc_Hospice / home    -0.128575
```



In [47]:
```
from sklearn.metrics import classification_report, roc_curve, roc_auc_score, confusion_m


# Generate classification report
classification_rep = classification_report(y_test, y_pred)

# Calculate ROC curve and AUC score
y_pred_proba = lasso_logistic_best.predict_proba(X_test_encoded)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
roc_auc = roc_auc_score(y_test, y_pred_proba)

# create confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# threshold here:
user_defined_threshold = 0.65

# Plot classification report
print("Classification Report:")
print(classification_rep)

# Plot ROC curve with threshold
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
plt.plot([0, 1], [0, 1], 'k--')
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) Curve')
plt.legend(loc='lower right')

# apply the created threshold
y_pred_threshold = (y_pred_proba >= user_defined_threshold).astype(int)
conf_matrix_threshold = confusion_matrix(y_test, y_pred_threshold)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.62      0.75     12966
           1       0.10      0.49      0.16      1086

    accuracy                           0.61     14052
   macro avg       0.52      0.56      0.46     14052
weighted avg       0.87      0.61      0.70     14052
```



Undersampling - Logistic Regression

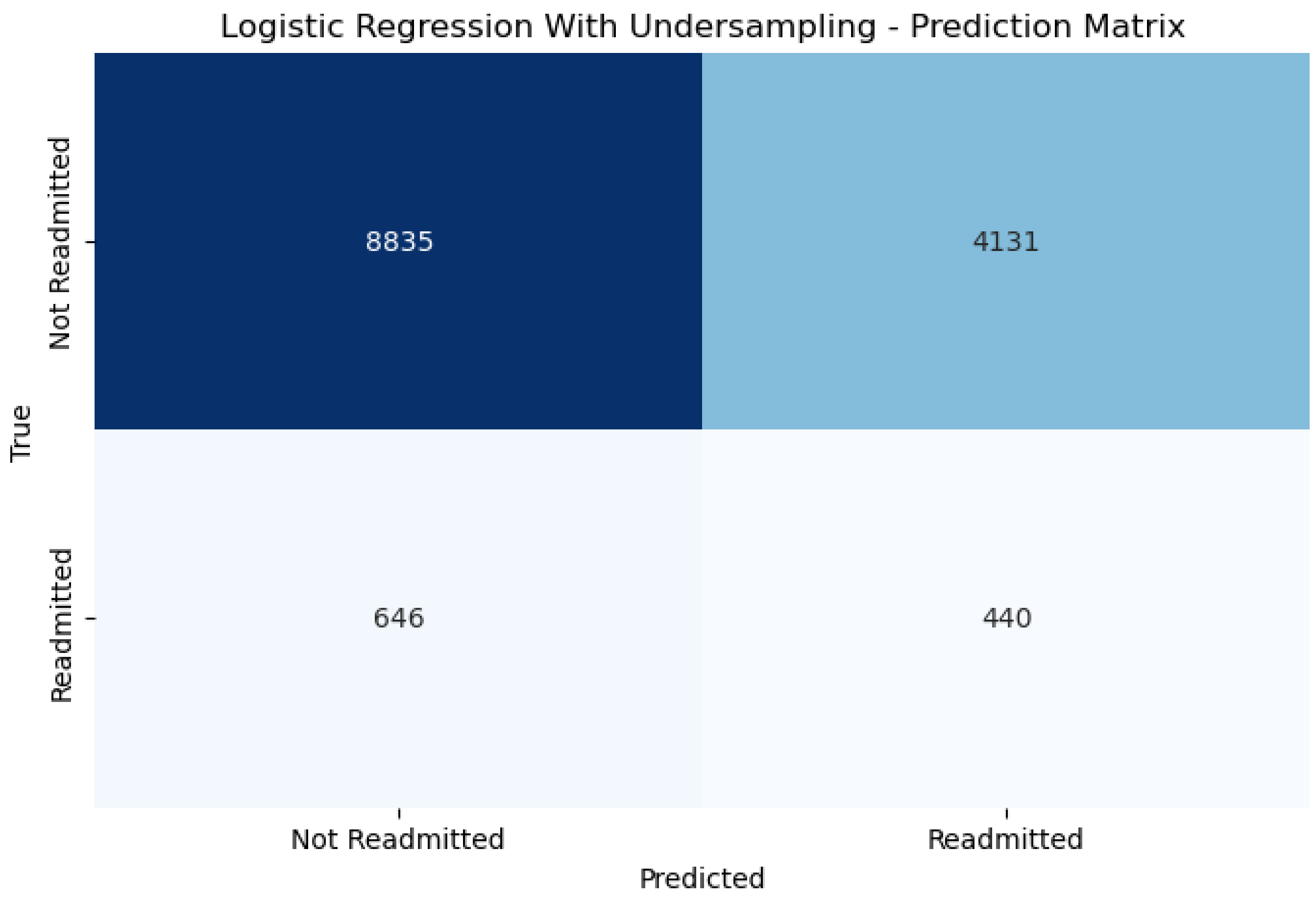
In [48]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from imblearn.under_sampling import RandomUnderSampler
from sklearn.linear_model import LassoCV, LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc

# Initialize RandomUnderSampler
undersampler = RandomUnderSampler(sampling_strategy='auto', random_state=12)

# Fit and apply undersampling to the training data
X_train_resampled, y_train_resampled = undersampler.fit_resample(X_train_encoded, y_trai
```

In [49]:
```
# Create the LassoCV model with CV alpha selection
alphas = np.logspace(-2, 2, 11)
lasso_model = LassoCV(alphas=alphas, cv=5)

# Fit the LassoCV model to the resampled training data
lasso_model.fit(X_train_resampled, y_train_resampled)

# Get the best alpha value
best_alpha = lasso_model.alpha_

# Apply the same one-hot encoding to the test data
X_test_encoded = pd.get_dummies(X_test, drop_first=True).astype(int)

# Ensure the test data has the same columns as the training data
missing_columns = set(X_train_encoded.columns) - set(X_test_encoded.columns)
```

```
for column in missing_columns:
    X_test_encoded[column] = 0  # Add missing columns with all zeros

# Reorder columns in X_test_encoded to match X_train_encoded
X_test_encoded = X_test_encoded[X_train_encoded.columns]

# Create the Lasso logistic regression model with the best alpha value
lasso_logistic_best = LogisticRegression(penalty='l1', solver='saga', C=1 / best_alpha,

# Fit the model to the resampled training data
lasso_logistic_best.fit(X_train_resampled, y_train_resampled)

# Make predictions on the test set
y_pred_prob = lasso_logistic_best.predict_proba(X_test_encoded)[:, 1]
y_pred = (y_pred_prob >= 0.982).astype(int)  # Change threshold ***

# Calculate accuracy on the test set
test_accuracy = accuracy_score(y_test, y_pred)

# Display features with non-zero coefficients
coefficients = lasso_logistic_best.coef_
feature_names = X_train_encoded.columns
coef_df = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients[0]})
non_zero_coef_df = coef_df[coef_df['Coefficient'] != 0]

print("Features with Non-Zero Coefficients:")
print(non_zero_coef_df)

print(f"Best Alpha (Lambda): {best_alpha:.4f}")
print(f"Test Set Accuracy: {test_accuracy:.4f}")

# threshold
user_defined_threshold = 0.982

# Calculate classification metrics with threshold
y_pred_custom_threshold = (y_pred_prob >= user_defined_threshold).astype(int)

classification_report_custom_threshold = classification_report(y_test, y_pred_custom_thr
confusion_matrix_custom_threshold = confusion_matrix(y_test, y_pred_custom_threshold)

print(f"User-Defined Threshold: {user_defined_threshold:.2f}")
print("Classification Report with Custom Threshold:")
print(classification_report_custom_threshold)

custom_labels = ['Not Readmitted','Readmitted']
# Plot confusion matrix with threshold
plt.figure(figsize=(8, 5))
sns.heatmap(confusion_matrix_custom_threshold, annot=True, fmt="d", cmap="Blues", cbar=F
            xticklabels=custom_labels, yticklabels=custom_labels)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Logistic Regression With Undersampling - Prediction Matrix")
plt.show()
```

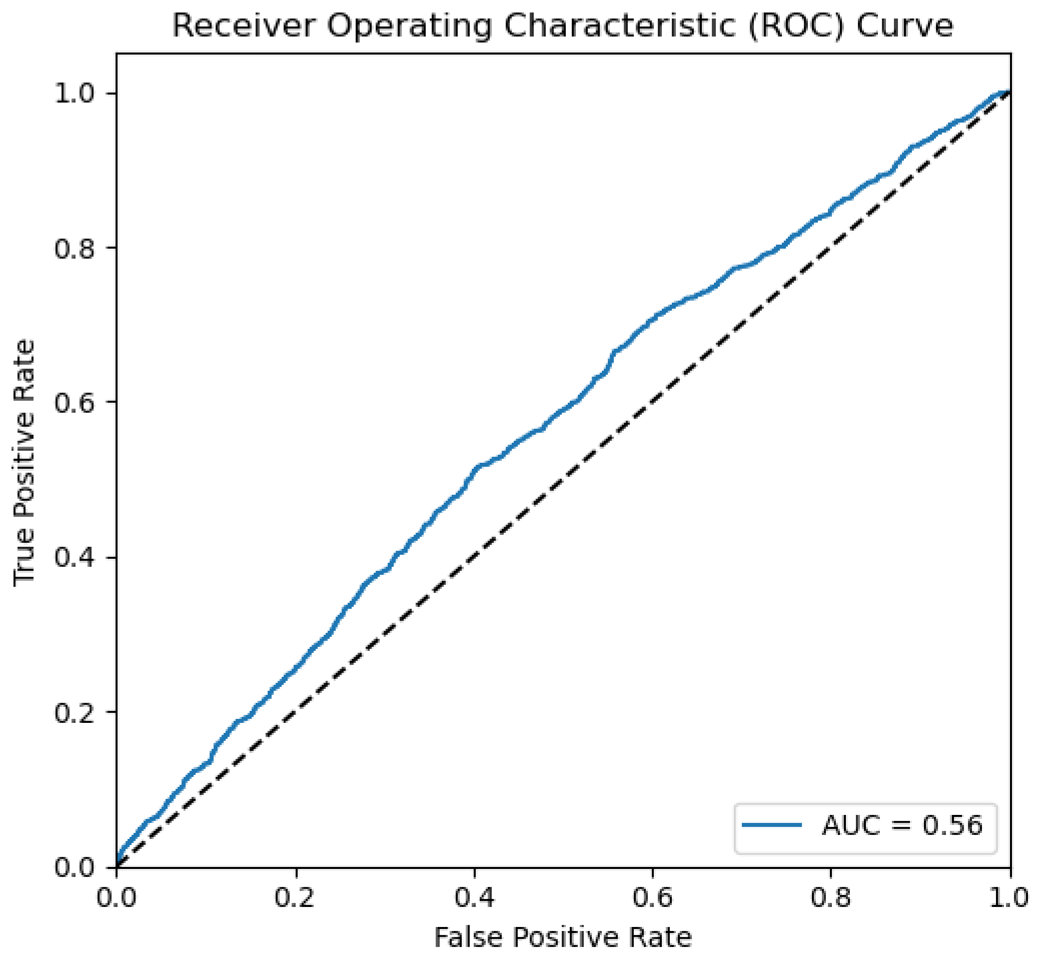
```
Features with Non-Zero Coefficients:
                      Feature  Coefficient
0            time_in_hospital     0.023936
1          num_lab_procedures     0.061299
2              num_procedures    -0.015450
3             num_medications    -0.013577
4           number_outpatient     0.012571
..                        ...          ...
209    diag_2_new_Respiratory    -0.055289
210  diag_3_new_Heart Disease    -0.038178
211         diag_3_new_Injury    -0.012873
212          diag_3_new_Other    -0.045747
```

```
213    diag_3_new_Respiratory     0.018719

[214 rows x 2 columns]
Best Alpha (Lambda): 0.0100
Test Set Accuracy: 0.6600
User-Defined Threshold: 0.98
Classification Report with Custom Threshold:
              precision    recall  f1-score   support

           0       0.93      0.68      0.79     12966
           1       0.10      0.41      0.16      1086

    accuracy                           0.66     14052
   macro avg       0.51      0.54      0.47     14052
weighted avg       0.87      0.66      0.74     14052
```



In [50]:
```python
from sklearn.metrics import classification_report, roc_curve, roc_auc_score, confusion_m


# Generate classification report
classification_rep = classification_report(y_test, y_pred)

# Calculate ROC curve and AUC score
y_pred_proba = lasso_logistic_best.predict_proba(X_test_encoded)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
roc_auc = roc_auc_score(y_test, y_pred_proba)

# create confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# threshold here:
user_defined_threshold = 0.982

# Plot classification report
print("Classification Report:")
print(classification_rep)

# Plot ROC curve with threshold
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.plot(fpr, tpr, label=f'AUC = {roc_auc:.2f}')
plt.plot([0, 1], [0, 1], 'k--')
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) Curve')
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.68      0.79     12966
           1       0.10      0.41      0.16      1086

    accuracy                           0.66     14052
   macro avg       0.51      0.54      0.47     14052
weighted avg       0.87      0.66      0.74     14052
```



Random Forest

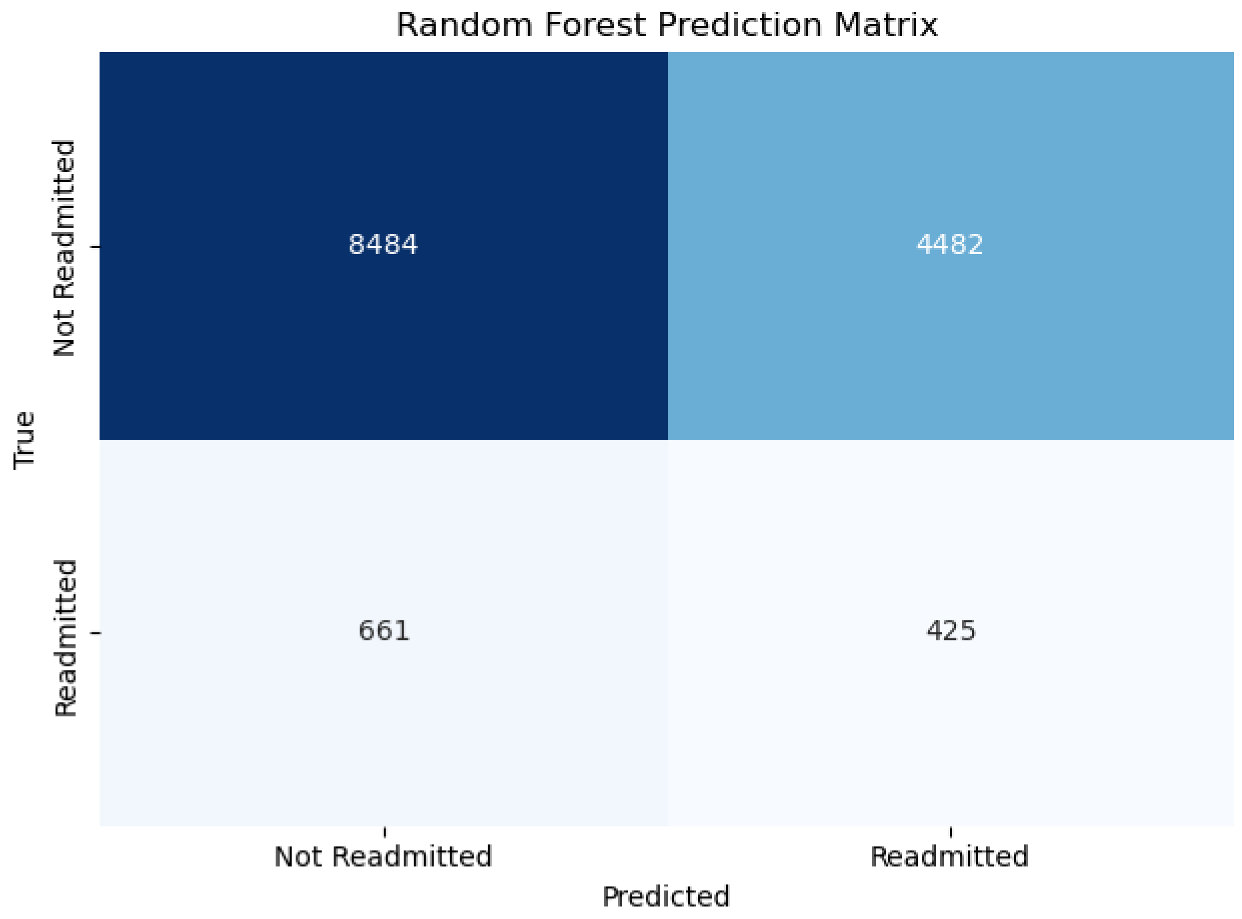
In [51]:
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

# Random Forest classifier
rf_classifier = RandomForestClassifier(random_state=12)

# GridSearchCV
param_grid = {
    'n_estimators': [100, 200, 300, 400, 500],
    'max_depth': [None, 10, 15, 20]
}

# GridSearchCV with 5-fold cross-validation
grid_search = GridSearchCV(estimator=rf_classifier, param_grid=param_grid, cv=5, n_jobs=

# Fit the grid search to the data
grid_search.fit(X_train_encoded, y_train)

# Get the best parameters and the best estimator
best_params = grid_search.best_params_
best_rf_classifier = grid_search.best_estimator_

# Print the best parameters
print("Best Parameters:")
print(best_params)
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
Best Parameters:
{'max_depth': None, 'n_estimators': 200}
```

In [52]:
```python
# Random Forest classifier with best parameters
best_rf_classifier.fit(X_train_encoded, y_train)

# Make probability predictions on the test set
y_prob_rf = best_rf_classifier.predict_proba(X_test_encoded)

# threshold
threshold = 0.313

# Apply the threshold
y_pred_rf = (y_prob_rf[:, 1] >= threshold).astype(int)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# accuracy
accuracy = accuracy_score(y_test, y_pred_rf)
print(f"Test Set Accuracy: {accuracy:.4f}")

# classification report
classification_rep_rf = classification_report(y_test, y_pred_rf)
print("Classification Report:")
print(classification_rep_rf)

# confusion matrix
conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)

sns.heatmap(conf_matrix_rf, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=custom_label,yticklabels=custom_label)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title(f"Random Forest Prediction Matrix")
plt.tight_layout()
plt.show()
```

```
Test Set Accuracy: 0.6340
Classification Report:
```

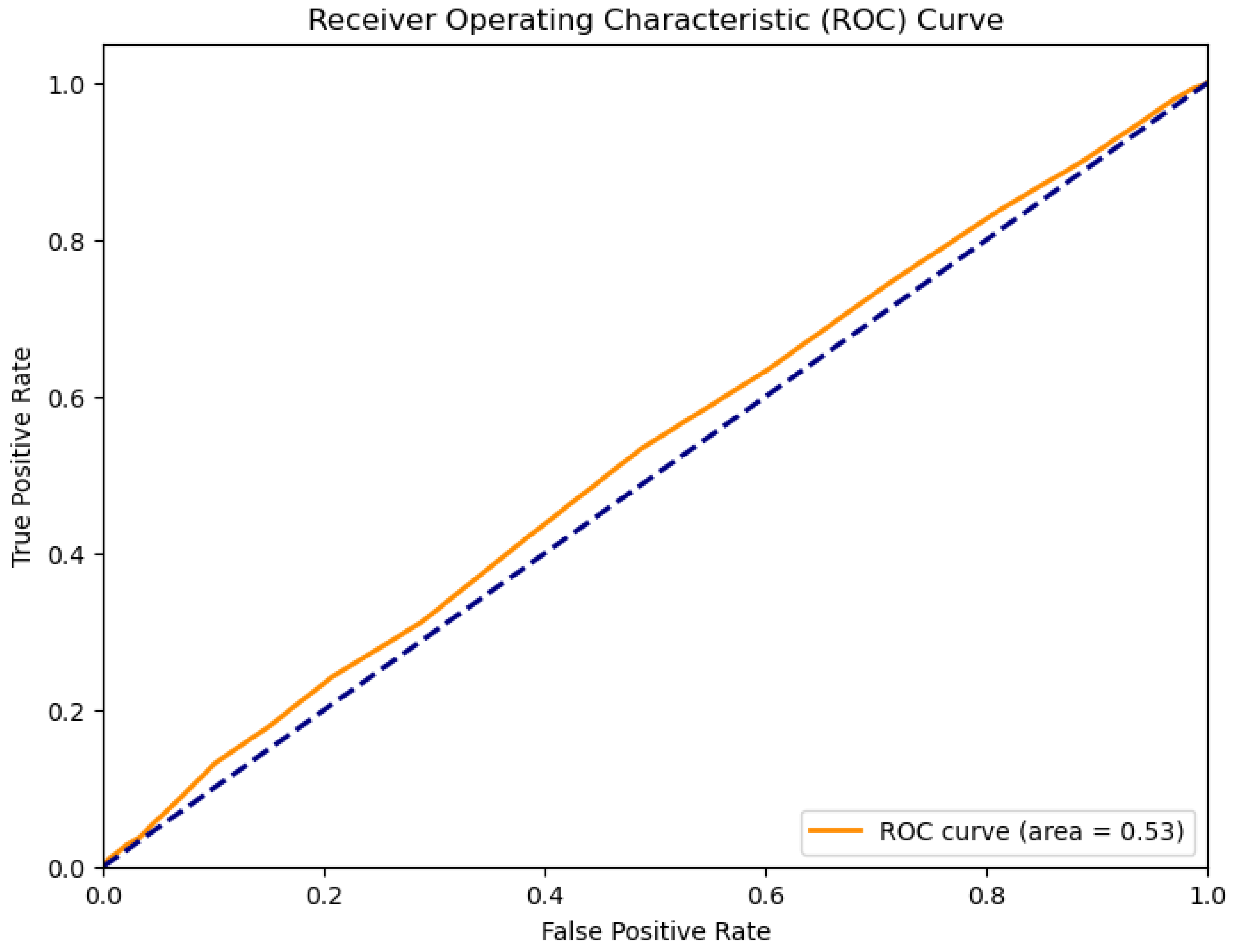
```
              precision    recall  f1-score   support

           0       0.93      0.65      0.77     12966
           1       0.09      0.39      0.14      1086

    accuracy                           0.63     14052
   macro avg       0.51      0.52      0.45     14052
weighted avg       0.86      0.63      0.72     14052
```



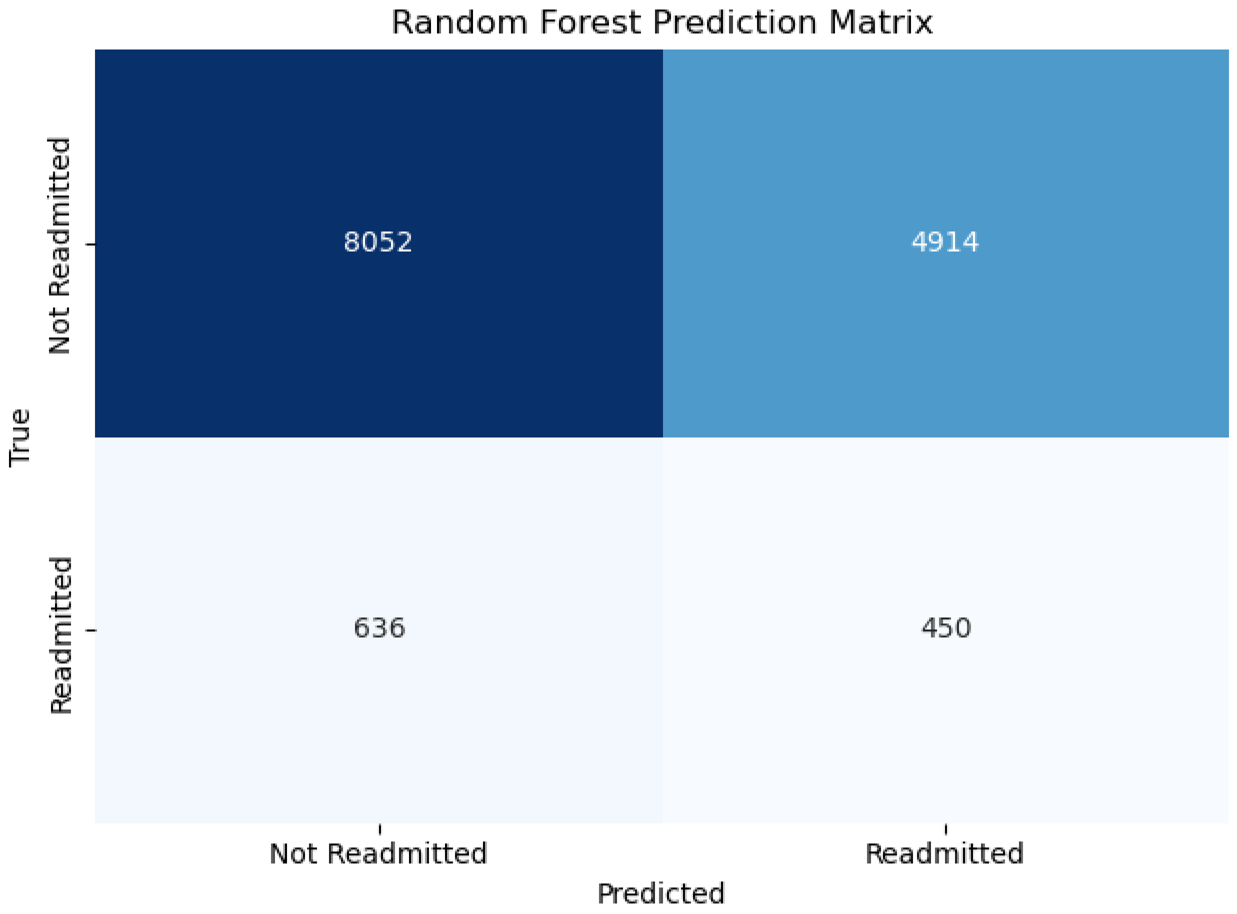
In [53]:
```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

# Random Forest classifier
rf_classifier = RandomForestClassifier(random_state=12, class_weight='balanced')

# GridSearchCV
param_grid = {
    'n_estimators': [100, 200, 300, 400, 500],
    'max_depth': [None, 10, 15, 20]
}

# GridSearchCV with 5-fold cross-validation
grid_search = GridSearchCV(estimator=rf_classifier, param_grid=param_grid, cv=5, n_jobs=

# Fit the grid search to the data
grid_search.fit(X_train_encoded, y_train)

# Get the best parameters and the best estimator
best_params = grid_search.best_params_
best_rf_classifier = grid_search.best_estimator_

# Print the best parameters
print("Best Parameters:")
print(best_params)
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
Best Parameters:
{'max_depth': None, 'n_estimators': 100}
```

In [54]:
```
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc

# Random Forest classifier with best parameters
best_rf_classifier.fit(X_train_encoded, y_train)

# Make probability predictions on the test set
y_prob_rf = best_rf_classifier.predict_proba(X_test_encoded)

# threshold
threshold = 0.12

# Apply the threshold
y_pred_rf = (y_prob_rf[:, 1] >= threshold).astype(int)

# Calculate ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_prob_rf[:, 1])

# Calculate AUC (Area Under the Curve)
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
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) Curve')
plt.legend(loc='lower right')
plt.show()

# accuracy
accuracy = accuracy_score(y_test, y_pred_rf)
print(f"Test Set Accuracy: {accuracy:.4f}")

# classification report
classification_rep_rf = classification_report(y_test, y_pred_rf)
print("Classification Report:")
print(classification_rep_rf)

# confusion matrix
conf_matrix_rf = confusion_matrix(y_test, y_pred_rf)

sns.heatmap(conf_matrix_rf, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=custom_label, yticklabels=custom_label)
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title(f"Random Forest Prediction Matrix")
plt.tight_layout()
plt.show()
```



```
Test Set Accuracy: 0.6050
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.62      0.74     12966
           1       0.08      0.41      0.14      1086

    accuracy                           0.61     14052
   macro avg       0.51      0.52      0.44     14052
weighted avg       0.86      0.61      0.70     14052
```


Random Forest Prediction Matrix
True
Not Readmitted
Readmitted
8052
4914
636
450
Not Readmitted
Readmitted
Predicted