**Project Summary**

|  |  |
| --- | --- |
| Batch details | Group No. 2 Batch: June 2024  Location: Bangalore |
| Team members | 1. Rahul Yadav  2. Aswin  3. Preeti Dhanuka  4. Bhavitha Sanivarapu  5. Nishith Ramesh  6. Lohith T S |
| Domain of Project | Health Care |
| Proposed project title | Prediction of Hospital readmission within 30 days |
| Group Number | 2 |
| Team Leader | Rahul Yadav |
| Mentor Name | Mr. P. V. Subramanian |

Date:



Signature of the Mentor Signature of the Team Leader

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl NO** | **Topic** | **Page No** |
| 1 | INTRODUCTION | 04 |
| 2 | OVERVIEW OF THE FINAL PROCESS | 06 |
| 3 | STEP BY STEP WALK-THROUGH OF THE SOLUTION | 13 |
| 4 | MODEL EVALUATION | 41 |
| 5 | COMPARISON TO BENCHMARK | 43 |
| 6 | VISUALIZATIONS | 43 |
| 7 | IMPLICATIONS | 45 |
| 8 | LIMITATIONS | 46 |
| 9 | CLOSING REFLECTIONS | 47 |
| 10 | APPENDIX | 47 |

**LIST OF TABLES AND FIGURES**

**List of Figures:**

Fig.2.1 - Model comparison chart 10

Fig.2.2 - Feature Importance Plot 12

Fig.3.1 - Boxplot for Time in Hospital Across Readmission Categories 17

Fig.3.2 - Bar plot for Readmission rate by race 18

Fig.3.3 - Bar Plot for Readmission by Gender 19

Fig.3.4 - Heatmap for Correlation Among Numerical Features 20

Fig.3.5 - Histogram of Age Distribution 22

Fig.3.6 - Histogram of numerical features 23

Fig.3.7. Countplot of categorical features 27

Fig.3.8. Distribution of Medical speciality 29

Fig.3.9. Distribution of diag\_1, diag\_2, diag\_3 31

Fig.3.10. Barplot readmission class imbalance 38

Fig.6.1. Boxplot comparison of recall measure across various models 44

Fig.6.2. Barplot of ‘Discharged to’ and the Target variables 44

Fig.6.3. Barplot between ‘number\_inpatient’ and the Target variables 45

Fig.7.1. Variable importance plot showing % Reduction in a criterion or

Gini Importance 45

Fig.10.1 - Model comparison using Recall before hyper parameter tuning 60

Fig.10.2 - Model comparison using Recall after hyper parameter tuning 61

Fig.10.3 - Recall of Random Forest model using Training Data 62

Fig.10.4 - Recall of Random Forest model using Test Data 63

**List of Tables:**

Tab.2.1 - Models used for this project 9

Tab.2.2 - Model comparison chart 10

Tab.2.3 - Random Forest Model Performance 11

Tab.2.4 - Feature Importance Table 12

Tab.3.1 - Count & Percentage of Missing values 14

Tab.3.2 - Treatment of missing values 15

Tab.3.3 - Count of Zero values 16

Tab.3.4 - Features and VIF 21

Tab.3.5 -Top 15 Important Features 40

Tab:4.1 - Model comparison chart before treating for data imbalance 42

Tab.4.2 - Decision Tree Model performance comparison chart after

treating for data imbalance 43

Tab.6.1. Crosstab between ‘Discharged to’ and the Target variables 44

Tab.6.2. Crosstab between ‘number\_inpatient’ and the Target variables 45

Tab.7.1. Variable importance showing % Reduction in a criterion or Gini Importance 45

Tab.10.1 - Data Dictionary 47

Tab.10.2 - Values for key ID variables 54

Tab.10.3 - Hyper-parameter tuning results and OS details 56

# **INTRODUCTION**

Hospital readmissions, defined as patient returns to the hospital within a specified timeframe post-discharge, represent a major challenge for healthcare systems worldwide. These readmissions not only burden healthcare providers but also reflect inefficiencies in patient care and care transitions. Addressing this issue is critical for improving patient outcomes, reducing costs, and enhancing the overall quality of care.

* 1. **OBJECTIVE**

The primary objective of this study is to leverage machine learning techniques to develop a robust and scalable framework for predicting hospital readmissions. The framework aims to:

1. Accurately identify patients at high risk of readmission at the point of discharge.
2. Analyze and prioritize key risk factors contributing to readmissions.
3. Support healthcare providers in implementing targeted interventions to minimize readmissions.
4. Demonstrate the feasibility and efficiency of integrating predictive analytics into routine clinical workflows.
   1. **CURRENT CHALLENGES**

Despite advancements in healthcare technologies, several challenges persist in addressing hospital readmissions:

1. **Cost Implications:** Unplanned readmissions contribute significantly to healthcare expenditure. For instance, Medicare penalizes hospitals with excessive 30-day readmission rates, further emphasizing the need for effective solutions.
2. **Complex Patient Profiles:** Many readmitted patients present with multimorbidities and social vulnerabilities, complicating risk assessments.
3. **Data Challenges:** Patient records often exhibit missing, incomplete, or noisy data, making the application of advanced analytics challenging.
4. **Insufficient Predictive Accuracy:** Traditional statistical models and scoring indices, such as the LACE index, often fail to capture nuanced patterns in diverse patient populations.
5. **Limited Integration:** Predictive tools are rarely integrated seamlessly into healthcare systems, hindering their practical use by clinicians.
6. **Regulatory Hurdles:** Ensuring patient data privacy while enabling robust analytics remains a pressing concern in healthcare.
   1. **PROPSED BUSINESS PROBLEM STATEMENT**

The project aims to develop a predictive model that employs machine learning to accurately forecast hospital readmissions. This model will address the limitations of traditional methods by providing real-time insights, actionable recommendations, and a scalable framework adaptable to various healthcare settings. By reducing preventable readmissions, this initiative seeks to improve patient care, optimize hospital resources, and lower healthcare costs.

* 1. **INDUSTRY REVIEW**
     1. **Current Practices**

1. **Traditional Methods:** Most hospitals rely on tools like the LACE index and clinical judgment to assess readmission risks. While widely used, these approaches lack precision and adaptability.
2. **Emerging Analytics: Recent initiatives involve piloting predictive models using electronic health record (EHR) data. These models often incorporate demographic, clinical, and social factors to provide a more comprehensive risk assessment.**
3. **Case Management Programs: Some healthcare systems employ dedicated teams to follow up with high-risk patient’s post-discharge, emphasizing patient education, medication adherence, and follow-up visits.**
   * 1. **Background Research**

Research highlights the potential of machine learning in revolutionizing readmission prediction. Key insights include:

1. **Feature Engineering: Integrating variables like medication adherence, social determinants, and post-discharge care has proven critical for predictive accuracy.**
2. **Model Performance: Studies consistently show that machine learning models outperform traditional methods, especially in handling large-scale and heterogeneous datasets.**
3. **Operational Challenges: Successful adoption requires user-friendly tools that integrate seamlessly into existing workflows, alongside clinician training to interpret and act on predictions.**
   * 1. **Literature Survey**
   1. **Title: "Prediction Models for Hospital Readmissions"**

**Authors:** Smith et al.

This study compares logistic regression, decision trees, and random forests for predicting hospital readmissions. Key findings include the influence of comorbidities, medication adherence, and patient demographics on prediction accuracy.

* 1. **Title: "Machine Learning Approaches to Predict Hospital Readmissions"**

**Authors: Gupta et al.**

The research evaluates Support Vector Machines, Gradient Boosting, and Neural Networks. Ensemble methods demonstrated the highest performance in predicting 30-day readmissions, with an emphasis on real-time applicability.

* 1. **Title: "Predictive Modeling for Readmission Reduction"**

**Authors: Johnson et al.**

This paper discusses integrating real-time EHR data into predictive models. Techniques like cross-validation and hyperparameter tuning were highlighted as essential for improving reliability.

* 1. **Title: "Meta-Analysis of Hospital Readmission Prediction Models"**

**Authors: Wang et al.**

A systematic review of prediction models evaluates metrics like c-statistics and calibration. The study underscores the importance of external validation to ensure real-world applicability.

* 1. **Title: "A Comprehensive Review of Readmission Risk Factors"**

**Authors: Patel et al.**

This review identifies key predictors such as socioeconomic status, caregiver support, and access to follow-up care. The authors advocate for combining clinical data with external datasets to enhance predictive capabilities.

# **OVERVIEW OF THE FINAL PROCESS**

We shall use the dataset from UCI Machine Learning Repository’s Diabetes 130-US Hospitals for years 1999–2008. This dataset includes over 50 features related to patient demographics, medical history, and hospital outcomes, making it ideal for analyzing readmission patterns. The instances represent hospitalized patient records diagnosed with diabetes.

URL: <https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008>

In this dataset, we have 101766 rows and 50 columns.

We have missing values in 7 columns denoted by ‘?’ as shown below:

1) race with 2.23% missing values

2) weight with 96.86% missing values

3) payer\_code with 39.56% missing values

4) medical\_specialty with 49.08% missing values

5) diag\_1 with 0.02% missing values

6) diag\_2 with 0.35% missing values

7) diag\_3 with 1.40% missing values

We dropped the column, weight with more than 80% missing values. We imputed other six columns having missing values in multiple features using iterative imputation, IterativeImputer in SKlearn.

Two ID variables, encounter\_id and patient\_nbr were removed from the dataset.

Columns with constant values do not contribute to the variability required for model building. Two such columns, examide and citoglipton, were identified and dropped.

Sparse datasets with high zero values can cause problems like over-fitting in the machine learning models and several other problems. We need to check if zero values are valid or not.

There are zero values in our dataset.

1. num\_procedures: – count of zero values: 46652
2. number\_outpatient: – count of zero values: 85027
3. number\_emergency: – count of zero values: 90383
4. number\_inpatient: – count of zero values: 67630

Zero values in the above columns are not invalid values as they are possible values:

* 1. num\_procedures (Number of procedures (other than lab tests) performed during the encounter)
  2. number\_outpatient (Number of outpatient visits of the patient in the year preceding the encounter)
  3. number\_emergency (Number of emergency visits of the patient in the year preceding the encounter)
  4. number\_inpatient (Number of inpatient visits of the patient in the year preceding the encounter)

A check for duplicate rows revealed that there were no duplicates in the dataset, ensuring its integrity.

6335 records of expired patients indicated by the column, discharge\_disposition\_id with values 11, 12, 18, 91, 20, 21, 25 and 26 are removed from our data. After removing unwanted columns and rows, we have the data with 95431 rows and 45 columns for further analysis.

We derived the target variable, 'Target' as follows:

Target value: 0 -- Readmitted NO and Readmitted > 30

Target value: 1 -- Readmitted < 30

The dataset is not balanced as our target variable, Exited is having 10804 records approximately, 11.32% of observations for the minority class.

**Treatment for data imbalance**

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don’t add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique or SMOTE for short.

Another method is under-sampling. Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. By keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modelling.

SMOTE method can generate noisy samples by interpolating new points between marginal outliers and in-liners. This issue can be resolved by cleaning the space resulting from over-sampling. We use the **SMOTEENN** method which combine over-and under-sampling using SMOTE and Edited Nearest Neighbours (ENN). Developed by Batista et al (2004), **SMOTEENN** method combines the SMOTE ability to generate synthetic examples for minority class and ENN ability to delete some observations from both classes that are identified as having different class between the observation's class and its K-nearest neighbour majority class. For more info, refer to [https://imbalanced-learn.org/stable/combine.html#combine](https://imbalanced-learn.org/stable/combine.html%23combine)

Detailed Exploratory Data Analysis was done to help us to look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Scaling continuous variables is crucial to ensure they have a meaningful impact on the model. However, scaling columns with binary values (0 or 1) is unnecessary as it doesn't improve the model's performance. Since binary variables, such as dummy variables, represent categorical information, scaling them would distort their intended influence. In our dataset, there are no continuous numerical variables, so there is no need for scaling. Without continuous variables, we don't have to worry about the issues that arise from scaling such features, and thus can focus on other preprocessing steps to prepare the data for model building.

Recursive Feature Elimination (RFE) is a powerful feature selection technique in machine learning that helps identify the most relevant features for model training. We selected top 15 features, namely,['gender', 'age', 'time\_in\_hospital', 'medical\_specialty', 'num\_lab\_procedures', 'num\_medications', 'number\_inpatient', 'diag\_1', 'diag\_2', 'diag\_3', 'metformin', 'insulin', 'change', 'diabetesMed', 'Discharged to' for building the model by using Recursive Feature Elimination (RFE) that works by searching for a subset of features by starting with all features in the dataset and successfully removing features until the desired number remains.

The most important assumption of absence of multi-collinearity was checked by calculating the Variance Inflation Factor (VIF). The following twenty (20) variables are collinear having VIF > 5:

|  |  |  |  |
| --- | --- | --- | --- |
| age | num\_lab\_procedures | num\_medications | diag\_1 |
| number\_diagnoses | max\_glu\_serum | A1Cresult | Metformin |
| repaglinide | nateglinide | chlorpropamide | Glimepiride |
| glipizide | glipizide | glyburidepioglitazone | rosiglitazone |
| Acarbose | miglitol | glyburide-metformin | diabetesMed |

We have built the following seven models:

|  |  |  |
| --- | --- | --- |
| **#** | **Model** | **Details** |
| 1 | Logistic Regression | Logistic regression predicts probabilities and assigns data points to binary classes (e.g., spam or not spam). |
| 2 | Naive Bayes | Naive Bayes model is based on Bayes’ theorem and assumes all features are independent of each other (hence “naive”) |
| 3 | K-Nearest Neighbors (KNN) | KNN is a simple algorithm that predicts the output for a new data point based on the similarity (distance) to its nearest neighbors in the training dataset, used for both classification and regression tasks. |
| 4 | Decision Tree (CART) | A decision tree splits data into branches based on feature values, creating a tree-like structure. |
| 5 | Random Forest | Random forest is an ensemble method that combines multiple decision trees. |
| 6 | XGBoost | Gradient Boosting algorithms such as XGBoost, LightGBM, CatBoost build models sequentially, meaning each new model corrects errors made by previous ones. Combines weak learners (like decision trees) to create a strong predictive model. |
| 7 | AdaBoost | AdaBoost techniques combine many weak machine-learning models to create a powerful classification model for the output. |

**Tab.2.1.** Models used for this project

We used K Fold cross validation to assess the performance of the above models using Default parameters. The top priority of this project is to identify if a customer will churn or won't. It's important that we don't predict churning as non-churning customers. That's why the model needs to be evaluated on the "Recall"- metric.

We shall tune the hyper parameters for the all the seven models and compare the performance of the models before and after tuning. Average of the 10 folds of Recall measure ± standard deviation, minimum and maximum are given for all the models in the Tab.2.1. Model comparison chart.

|  | | **With Default Parameters** | | **With tuned parameters** | |
| --- | --- | --- | --- | --- | --- |
| **#** | **Model** | **Recall score:**  **Mean ± Std. Deviation** | **Min & Max Recall score** | **Recall score:**  **Mean ± Std. Deviation** | **Min & Max Recall score** |
| 1 | LR | 0.02 ± 0.00 | 0.01, 0.02 | 0.02 ± 0.00 | 0.01, 0.02 |
| 2 | NB | 0.11 ± 0.01 | 0.10, 0.13 | 0.11 ± 0.01 | 0.11, 0.13 |
| 3 | KNN | 0.02 ± 0.01 | 0.01, 0.03 | 0.05 ± 0.01 | 0.04, 0.07 |
| 4 | **CART** | **0.18 ± 0.01** | **0.16, 0.20** | 0.55 ± 0.06 | 0.51, 0.66 |
| 5 | RF | 0.01 ± 0.00 | 0.01, 0.02 | **0.63 ± 0.01** | **0.60, 0.65** |
| 6 | XGBoost | 0.02 ± 0.00 | 0.02, 0.03 | 0.04 ± 0.01 | 0.03, 0.05 |
| 7 | AdaBoost | 0.01 ± 0.00 | 0.01, 0.02 | 0.00 ± 0.00 | 0.00, 0.01 |

**Tab.2.2.** Model comparison chart

|  |  |
| --- | --- |
| **Before tuning Comparison chart** | **After tuning Comparison chart** |
|  |  |

**Fig.2.1** Model comparison chart

Based on this measure, we observe that all models perform very poorly with the higher recall being approximately,18% before tuning. After tuning the hyper parameters of the models, we observe a significant improvement in the performance measure recall of Decision Tree model from 0.18 to 0.55 and Random Forest from 0.01 to 0.63.

Oversampling techniques like SMOTE (Synthetic Minority Over-sampling Technique) are used to balance the class distribution in a dataset. Cross-validation is a technique used to evaluate the performance of a machine learning model by dividing the data into train and test sets. ***Applying oversampling before cross-validation can lead to a bias in the evaluation of the model***. Since the same synthetic samples are used in each fold of the cross-validation, the model will be able to learn the characteristics of the synthetic samples, leading to overfitting and an overestimation of its performance. Hence, we split the dataset into training and test datasets in the ratio, 80:20 using stratified random sampling to ensure that the model is trained and evaluated on a representative sample of the data, and it can improve the model's overall performance.

We shall use the tuned parameters of the Random Forest model.

Given below the parameters after hyper parameter tuning for the model, Random Forest model and used for evaluating the model performance on both training dataset and test dataset:

warm\_start = False,

verbose = 0,

random\_state = None,

oob\_score = False,

n\_jobs = None,

n\_estimators = 50,

min\_weight\_fraction\_leaf = 0.0,

min\_samples\_split = 2,

min\_samples\_leaf = 1,

min\_impurity\_decrease = 0.0,

max\_samples = None,

max\_leaf\_nodes = None,

max\_features = 0.75,

max\_depth = 5,

criterion = 'gini',

class\_weight = 'balanced',

ccp\_alpha = 0.0,

bootstrap = True

For more details, refer to Appendix – Section 10.2.

**Evaluation of model performance with data treated for data imbalance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recall measure for the minority class of the Decision Tree model using** | **Before treating for data imbalance** | **After applying Random under-sampling** | **After applying SMOTE** | **After applying SMOTEEN** |
| Training dataset | 0.63 | 0.63 | 0.66 | 0.67 |
| Test dataset | 0.62 | 0.62 | 0.62 | 0.62 |

**Tab.2.3.** Random Forest Model Performance

We observe that the model performance measure, recall for test dataset has not improved from 0.62 by treating the data imbalance or by treating the data imbalance through SMOTE or Random under-sampling or SMOTEEN.

**Variable importance plot showing % of reduction in a criterion**

|  |  |
| --- | --- |
|  |  |
| **Fig.2.2.** Feature Importance Plot | **Tab.2.4.** Feature Importance Table |

The feature importance in Random Forest can be determined using a metric called Gini importance. It measures the total reduction of the Gini impurity of the dataset when a particular feature is used for splitting. The higher the Gini importance, the more important the feature is for the model. Based on the graph, Fig.2.2, we observe that top two (2) variables are ‘number\_inpatient’ and ‘discharge to’ impacting the target variable.

# **Step-by-Step Walkthrough of the Solution**

* 1. **Data Dictionary**

The dataset comprises clinical data for diabetic patients collected over 10 years from 130 U.S. hospitals. It includes patient records with attributes such as demographics, medical history, lab results, medications, and hospital visit details. A detailed data dictionary is provided in Section 10: Appendix.

* 1. **Variable categorization (count of numeric and categorical)**

**List of categorical variables:**

1. Race
2. Gender
3. Age
4. Weight
5. payer\_code
6. medical\_specialty
7. diag\_1
8. diag\_2
9. diag\_3
10. max\_glu\_serum
11. A1Cresult
12. Metformin
13. Repaglinide
14. Nateglinide
15. Chlorpropamide
16. Glimepiride
17. Acetohexamide
18. Glipizide
19. Glyburide
20. Tolbutamide
21. Pioglitazone
22. Rosiglitazone
23. Acarbose
24. Miglitol
25. Troglitazone
26. Tolazamide
27. Examide
28. Citoglipton
29. Insulin
30. glyburide-metformin
31. glipizide-metformin
32. glimepiride-pioglitazone
33. metformin-rosiglitazone
34. metformin-pioglitazone
35. change
36. diabetesMed
37. readmitted

**Count of Categorical Variables = 37**

These represent non-numeric attributes such as demographics, medication usage, and diagnoses.

They include:

* Demographics: gender, race, age (grouped in intervals)
* Administrative: admission\_type\_id, discharge\_disposition\_id, admission\_source\_id
* Medical Indicators: diag\_1, diag\_2, diag\_3, A1Cresult, max\_glu\_serum.
* Medication Variables: insulin, metformin, glipizide, glyburide, etc.

**List of numerical variables:**

1. encounter\_id
2. patient\_nbr
3. admission\_type\_id
4. discharge\_disposition\_id
5. admission\_source\_id
6. time\_in\_hospital
7. num\_lab\_procedures
8. num\_procedures
9. num\_medications
10. number\_outpatient
11. number\_emergency
12. number\_inpatient
13. number\_diagnoses

**Count of Categorical Variables = 13**

These represent integer values related to patient care, such as the duration of hospitalization, number of procedures, or diagnostic counts.

They include:

* time\_in\_hospital: Number of days spent in the hospital.
* num\_lab\_procedures: Number of lab tests performed during the encounter.
* num\_medications: Total medications administered.
* number\_diagnoses: Number of diagnoses recorded.
  1. **Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)**

**3.3.1. Count of Missing values**

We have missing values in 7 columns denoted by ‘?’ given below:

race

weight

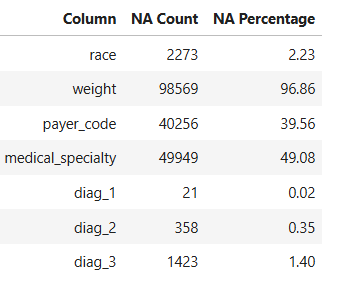
payer\_code

medical\_specialty

diag\_1

diag\_2

diag\_3



**Tab. 3.1.** Count & Percentage of Missing values

**We will drop the missing values, if the percentage of missing values exceed the threshold value of 80%.**

| **#** | **Feature** | **Missing Count** | **Missing Percentage** | **Action** |
| --- | --- | --- | --- | --- |
| 1 | **weight** | 98,569 | 96.86% | Drop (too many missing values) |
| 2 | **medical\_specialty** | 49,949 | 49.08% | Consider imputing |
| 3 | **payer\_code** | 40,256 | 39.56% | Consider imputing |
| 4 | **race** | 2,273 | 2.23% | Consider imputing |
| 5 | **diag\_1** | 21 | 0.02% | Consider imputing |
| 6 | **diag\_2** | 358 | 0.35% | Consider imputing |
| 7 | **diag\_3** | 1,423 | 1.40% | Consider imputing |

**Tab. 3.2**. Treatment of missing values

There are several methods available for imputing missing values. They include:

1. Remove Rows with Missing Values: Remove rows that contain missing values.
2. Impute Missing Values: Replace missing values with sensible values.
3. Impute Missing Values with KNN Imputer: Impute missing values using K nearest neighbors.
4. Impute Missing Values with Iterative Imputer: Impute missing values in multiple features using iterative imputation. IterativeImputer in SKlearn. It is a strategy for imputing missing values by modeling each feature with missing values as a function of other features in a round-robin fashion.

We opt for Impute Missing Values with Iterative Imputer since it is considered superior to other methods.

**IterativeImputer() from sklearn**

A more sophisticated approach is to use the IterativeImputer class, which models each feature with missing values as a function of other features, and uses that estimate for imputation. It does so in an iterated round-robin fashion: at each step, a feature column is designated as output y and the other feature columns are treated as inputs X. A regressor is fit on (X, y) for known y. Then, the regressor is used to predict the missing values of y. This is done for each feature in an iterative fashion, and then is repeated for max\_iter imputation rounds. The results of the final imputation round are returned.

**3.3.2. Unwanted columns or Redundant columns**

**1. Unwanted columns:**

Two ID variables, encounter\_id and patient\_nbr, were found to provide no analytical value and were removed from the dataset.

Delete the rows pertaining to those patients who expired indicated by the values 11, 12, 18, 19, 20,21,25, 26 in the column, discharge\_disposition\_id.

1. **Constant Value Columns**

Columns with constant values do not contribute to the variability required for model building. Two such columns, examide and citoglipton, were identified and dropped.

**3.3.3. Duplicate Rows & Missing Values**

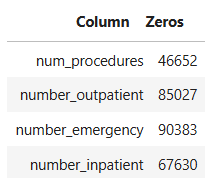
A check for duplicate rows revealed that there were no duplicates in the dataset, ensuring its integrity.

**3.3.4. Zero values**

Sparse datasets with high zero values can cause problems like over-fitting in the machine learning models and several other problems. We need to check if zero values are valid or not.

There are zero values in our dataset.

* + 1. num\_procedures
    2. number\_outpatient
    3. number\_emergency
    4. number\_inpatient



**Tab. 3.3.** Count of Zero values

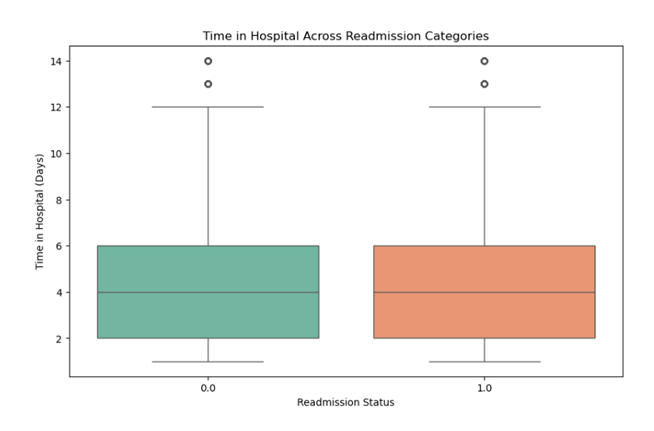
Zero values in the above columns are not invalid values as they are possible values:

1. num\_procedures (Number of procedures (other than lab tests) performed during the encounter)

1. number\_outpatient (Number of outpatient visits of the patient in the year preceding the encounter)
2. number\_emergency (Number of emergency visits of the patient in the year preceding the encounter)
3. number\_inpatient (Number of inpatient visits of the patient in the year preceding the encounter)

# **Distribution of Variables**

* + 1. **Bivariate Analysis**
* **Boxplot for Time in Hospital Across Readmission Categories**

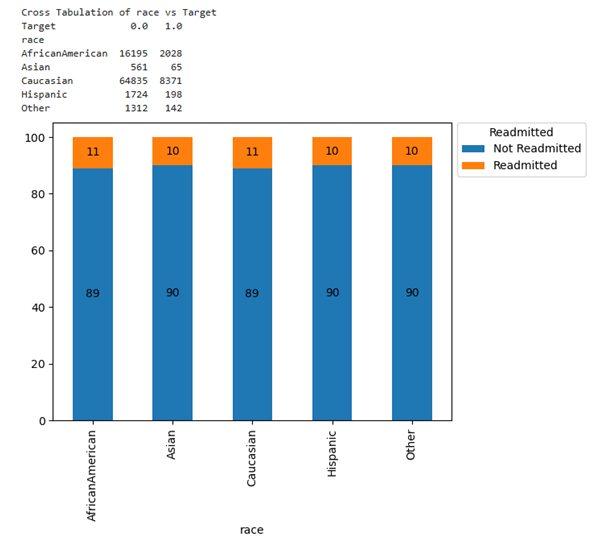
****

**Fig.3.1.** Boxplot for Time in Hospital Across Readmission Categories

**Inference:**

The boxplot for time in hospital across readmission categories shows no significant difference between patients who were readmitted and those who were not. This suggests that the length of hospital stay may not be a strong indicator for predicting readmission within 30 days. Other factors, such as underlying health conditions or post-discharge care, could play a more prominent role in readmission rates. Therefore, focusing on these aspects might provide more insights into preventing readmissions.

* **Bar Chart for Race vs. Readmission Rate**

****

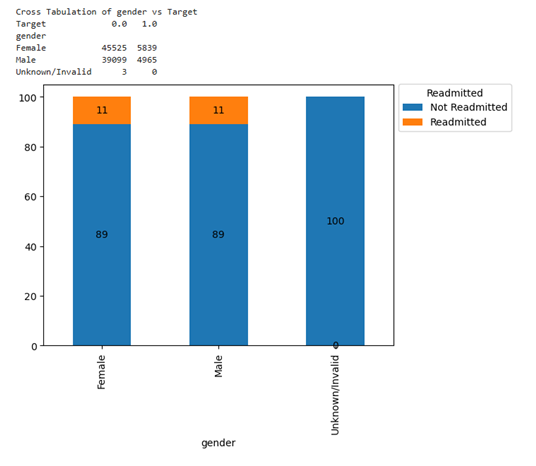
**Fig.3.2.** Bar plot for Readmission rate by race

**Inference:**

The bar chart for race vs. readmission rate does not reveal any pattern for hospital readmission. Further investigation into the healthcare access, cultural, and socioeconomic factors for these groups could provide valuable insights for reducing readmission rates.

It indicates that the readmission rate is consistently around 10% across all racial categories, with minimal variation. This suggests that race does not play a significant role in determining hospital readmission rates. Other factors such as age, underlying health conditions, access to healthcare, and socioeconomic conditions might have a more significant impact and require further investigation.

* **Bar Chart for Gender vs. Readmission Rate**

****

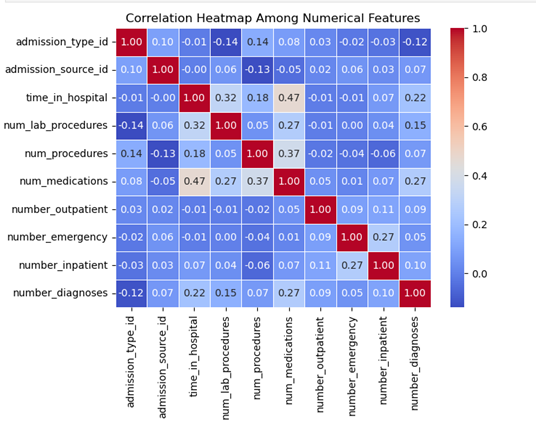
**Fig. 3.3.** Bar Plot for Readmission by Gender

**Inference:**

The count plot comparing readmission rates across genders reveals minimal differences between male and female patients. Both genders show nearly identical readmission trends, suggesting that gender is not a significant factor influencing readmission likelihood. This indicates that interventions aimed at reducing readmissions may not need to differentiate based on gender, as the risk appears uniformly distributed. However, further analysis involving other factors could provide additional insights into subtle gender-related trends.

### **3.4.2 Multivariate Analysis**

* **Heatmap for Correlation Among Numerical Features**

****

**Fig.3.4.** Heatmap for Correlation Among Numerical Features

**Inference:**

The heatmap shows weak correlations among the numerical features, with correlation values ranging from -0.14 to 0.47. Most of the variables exhibit low or negligible relationships, suggesting that these features are largely independent of one another. While there are mild positive correlations, such as between the number of medications and time in the hospital (0.47), no variable stands out as a strong predictor for outcomes like readmission. Therefore, these numerical features may not significantly influence readmission rates on their own. Further analysis, including interactions between variables or non-linear relationships, might be necessary to uncover stronger patterns.

# **Check for multi-collinearity**

****

**Tab. 3.4.** Features and VIF

**Inference:**

* Seven features (VIF) show severe multicollinearity:

|  |  |  |  |
| --- | --- | --- | --- |
| miglitol (1115.70) | chlorpropamide (911.70) | acarbose (293.59) | glyburide-metformin (132.72) |
| nateglinide (125.37) | repaglinide (55.82) | max\_glu\_serum (44.13) |  |

* Eight features (VIF) show moderate multicollinearity:

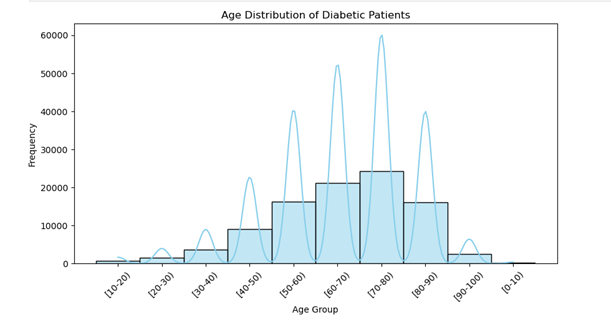
|  |  |  |  |
| --- | --- | --- | --- |
| glimepiride (20.38) | number\_diagnoses(20.06) | rosiglitazone (19.41) | age (18.07) |
| pioglitazone (17.19) | A1Cresult (14.67) | glyburide (12.47) | glipizide (11.22) |

* Five features (VIF) show slight multicollinearity:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| metformin (9.7) | num\_medications (8.35) | num\_lab\_procedures (6.95) | diabetesMed (6.839) | diag\_1 (5.5) |

# **Distribution of variables**

* 1. **Histogram of Age Distribution**

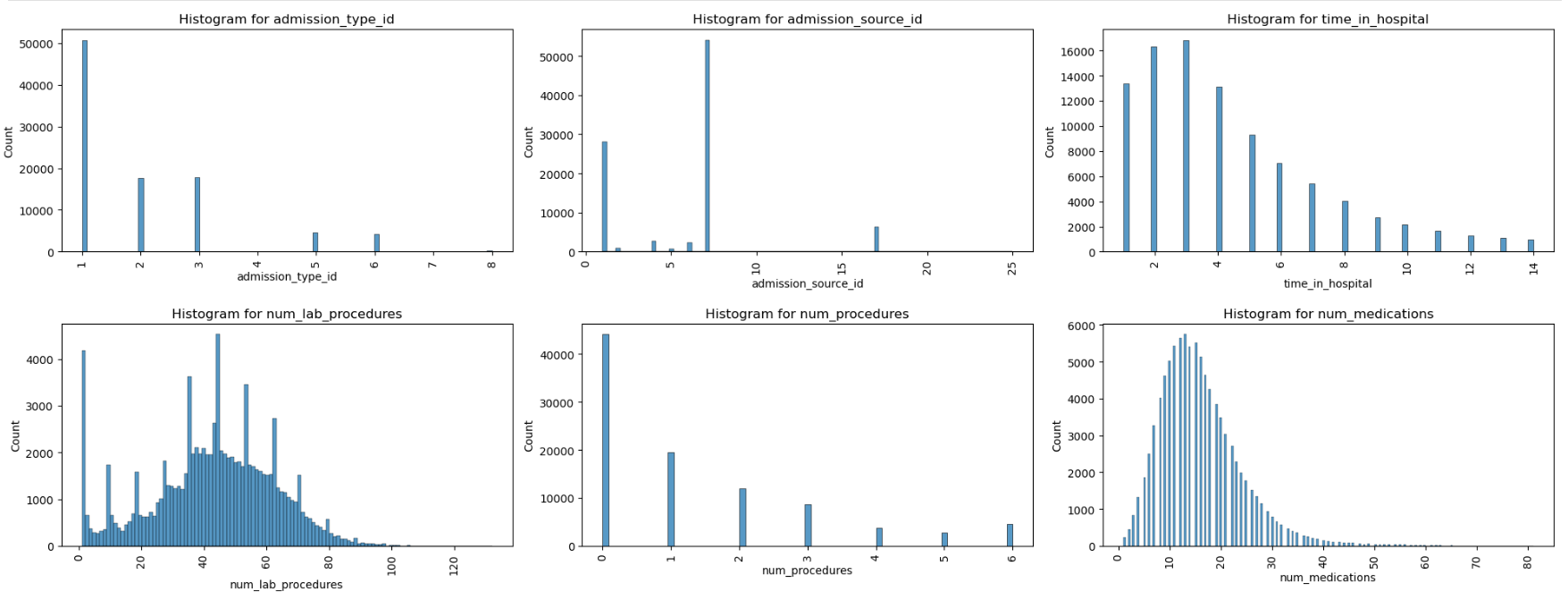
****

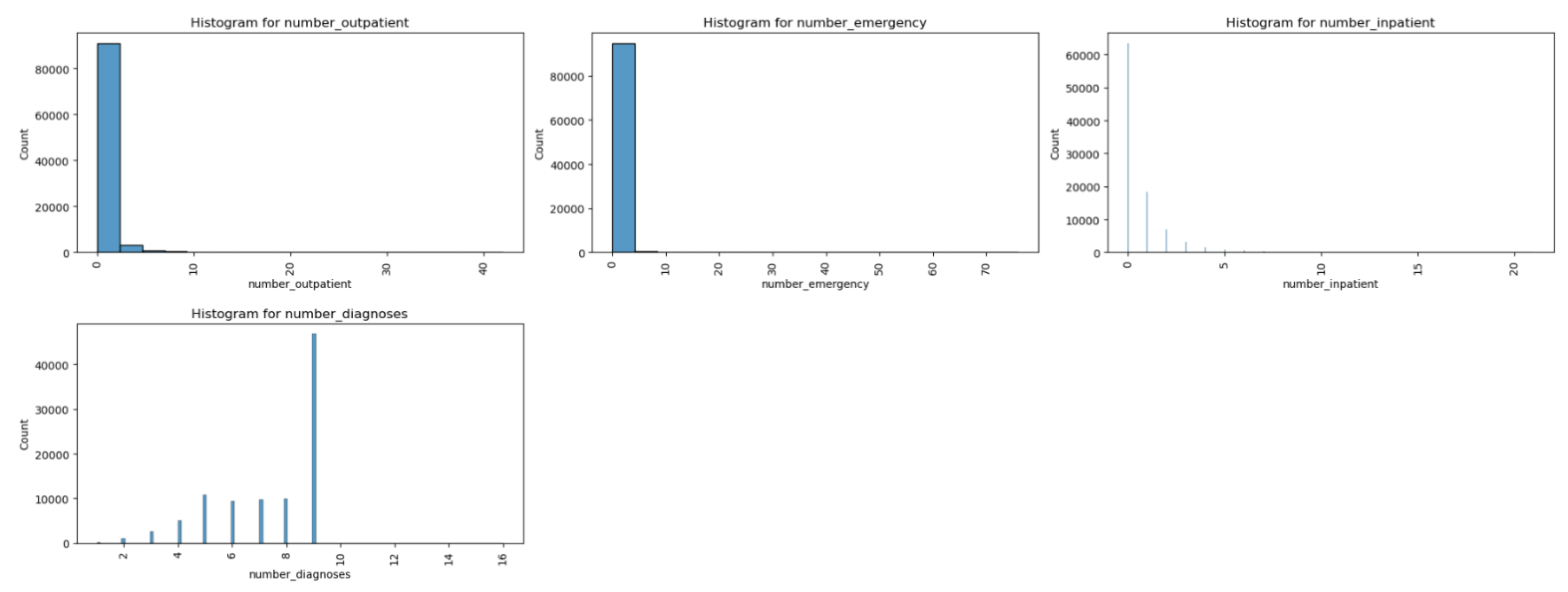
**Fig. 3.5.** Histogram of Age Distribution

**Inference:**

The histogram shows a significant peak in the 60-80 years age group, indicating that most diabetic patients are from this category. This suggests that older individuals are more prone to diabetes, likely due to age-related insulin resistance and comorbidities. The higher prevalence in this group may also correlate with increased readmission rates, highlighting the need for targeted healthcare interventions to manage their condition and reduce risks.

* 1. **For numerical discrete data**

****

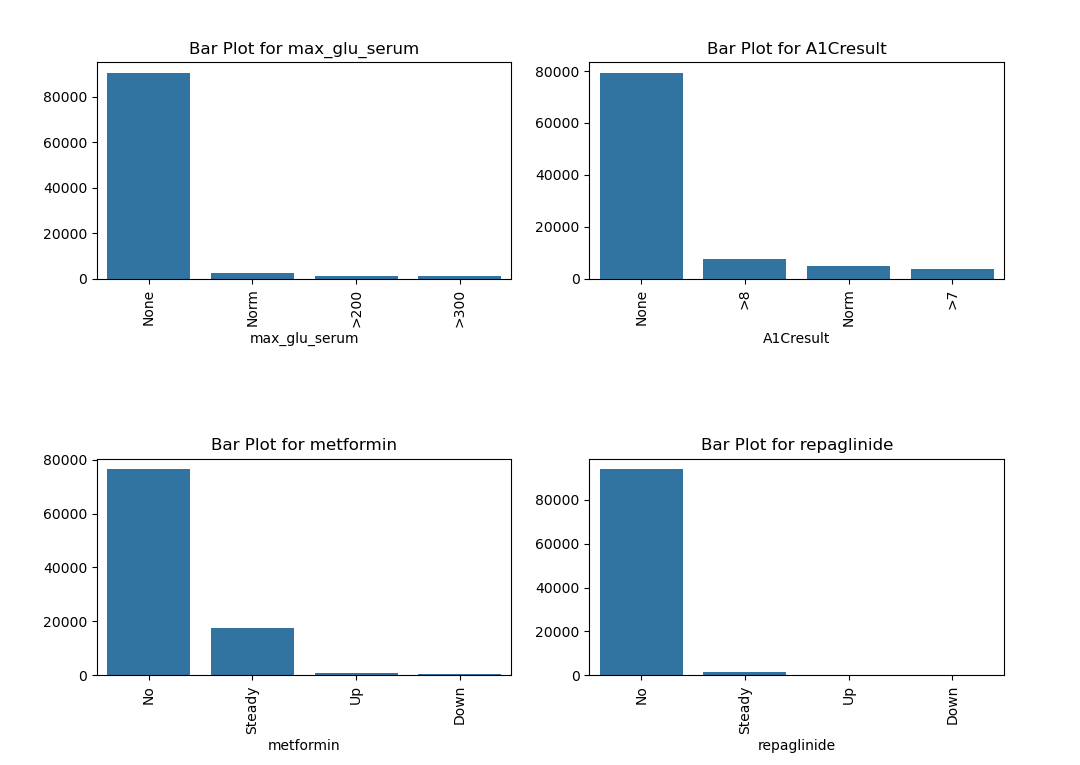
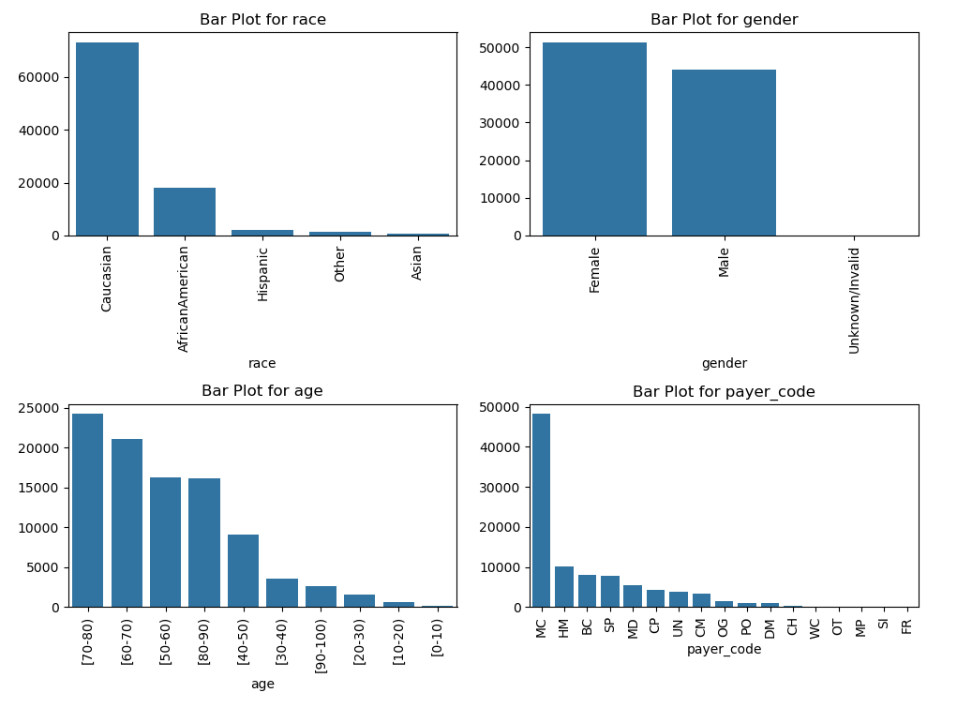
****

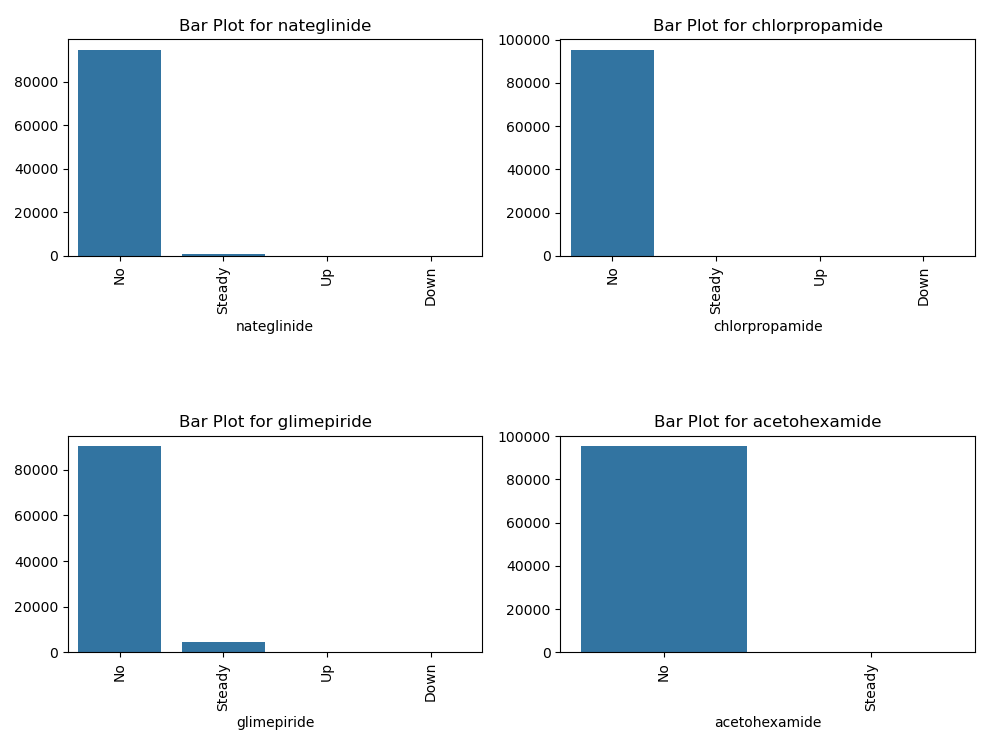
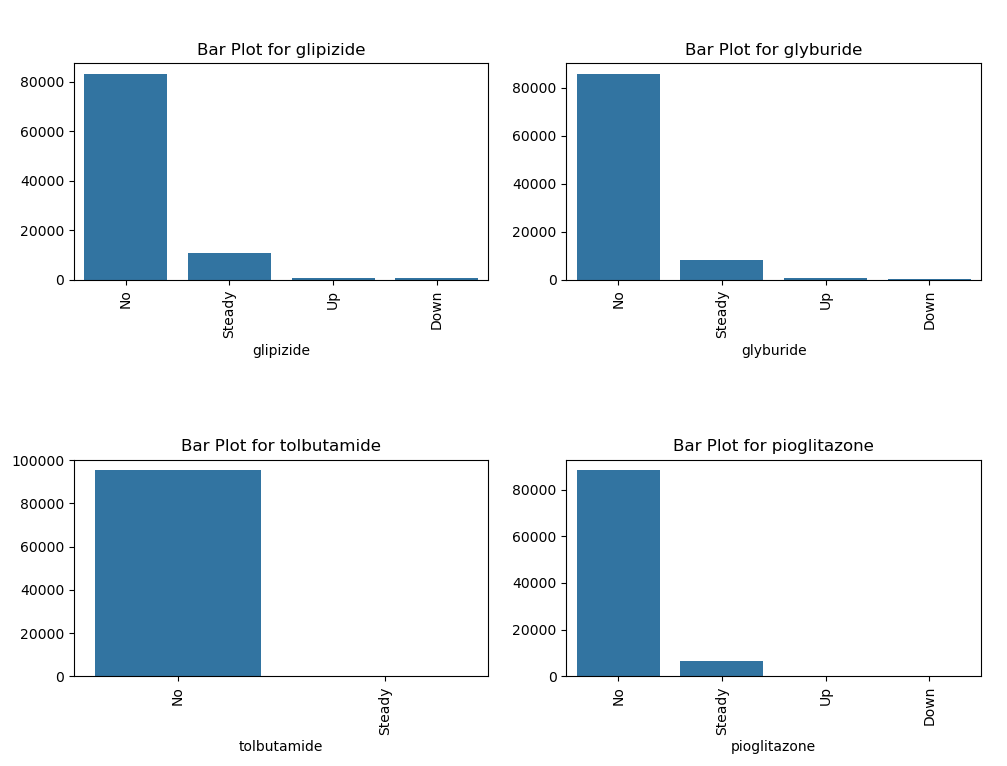
**Fig. 3.6.** Histogram of numerical features

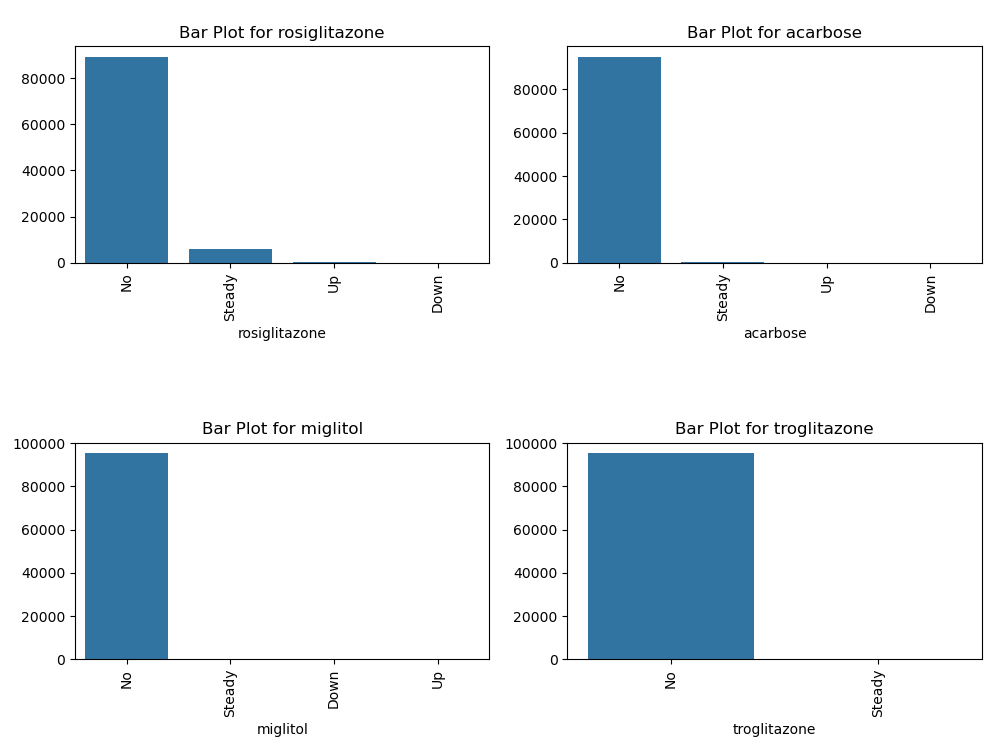
**Inference:**

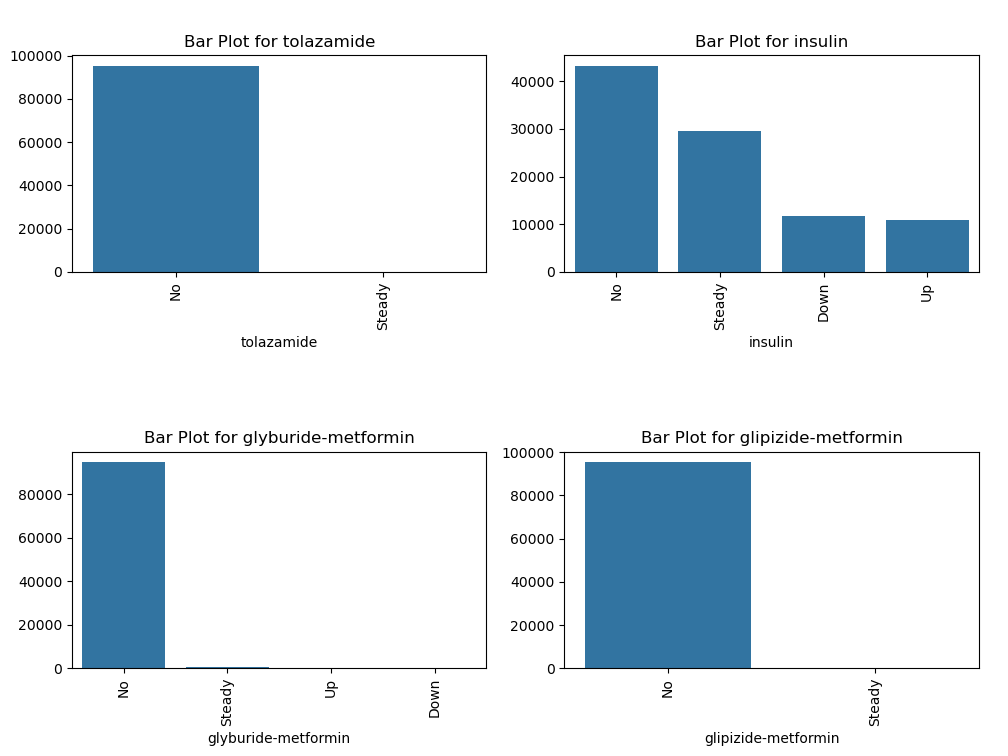
The histogram visualizes the frequency distribution of numerical features. Key observations:

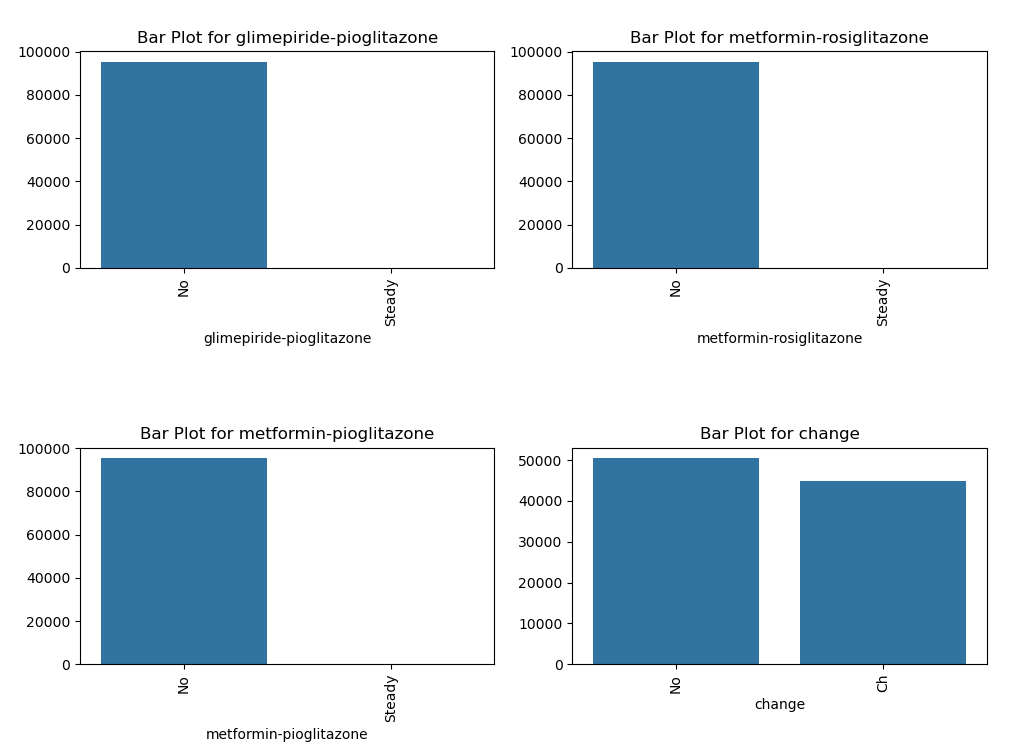
1. Peaks in the histogram represent intervals with the highest frequency of occurrences for the respective numerical data.
2. Skewness in the distribution (if any) can be identified:
   * A right-skewed distribution indicates more low values.
   * A left-skewed distribution shows more high values.
3. Gaps or isolated bins might indicate outliers or underrepresented values.
4. Helps identify data patterns like uniformity, normality, or bimodality.
   1. **For categorical variables**

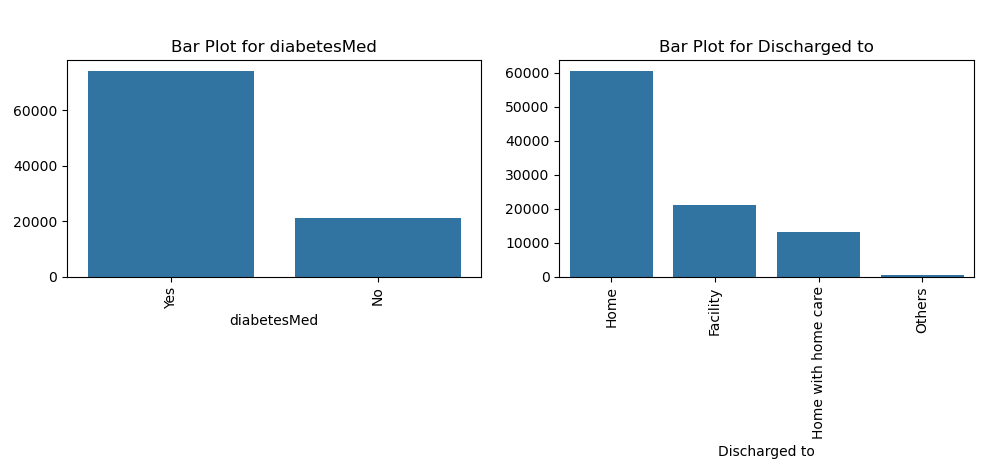
****

****

****

****

****

****

**Fig. 3.7.** Countplot of categorical features

**Inference:**

Barplot allows us to visualize the distribution of categorical data by showing the frequency or count of each category along the plot.

1. **Distribution of Categories**:

* The countplot effectively visualizes the frequency of each category, making it easy to identify dominant and less frequent categories.
* Taller bars indicate categories with higher occurrences, reflecting their prevalence or significance in the dataset.

1. **Imbalance Detection**:

* Any stark differences in bar heights suggest data imbalance. For example, one or two categories may dominate the dataset, which could bias analysis or predictive models.

1. **Insights into Trends**:

* The countplot highlights recurring patterns, such as which categories contribute most frequently to an outcome (e.g., readmissions).
* It may reveal associations between specific categories and the target variable (if segmented).

1. **Rare Categories**:

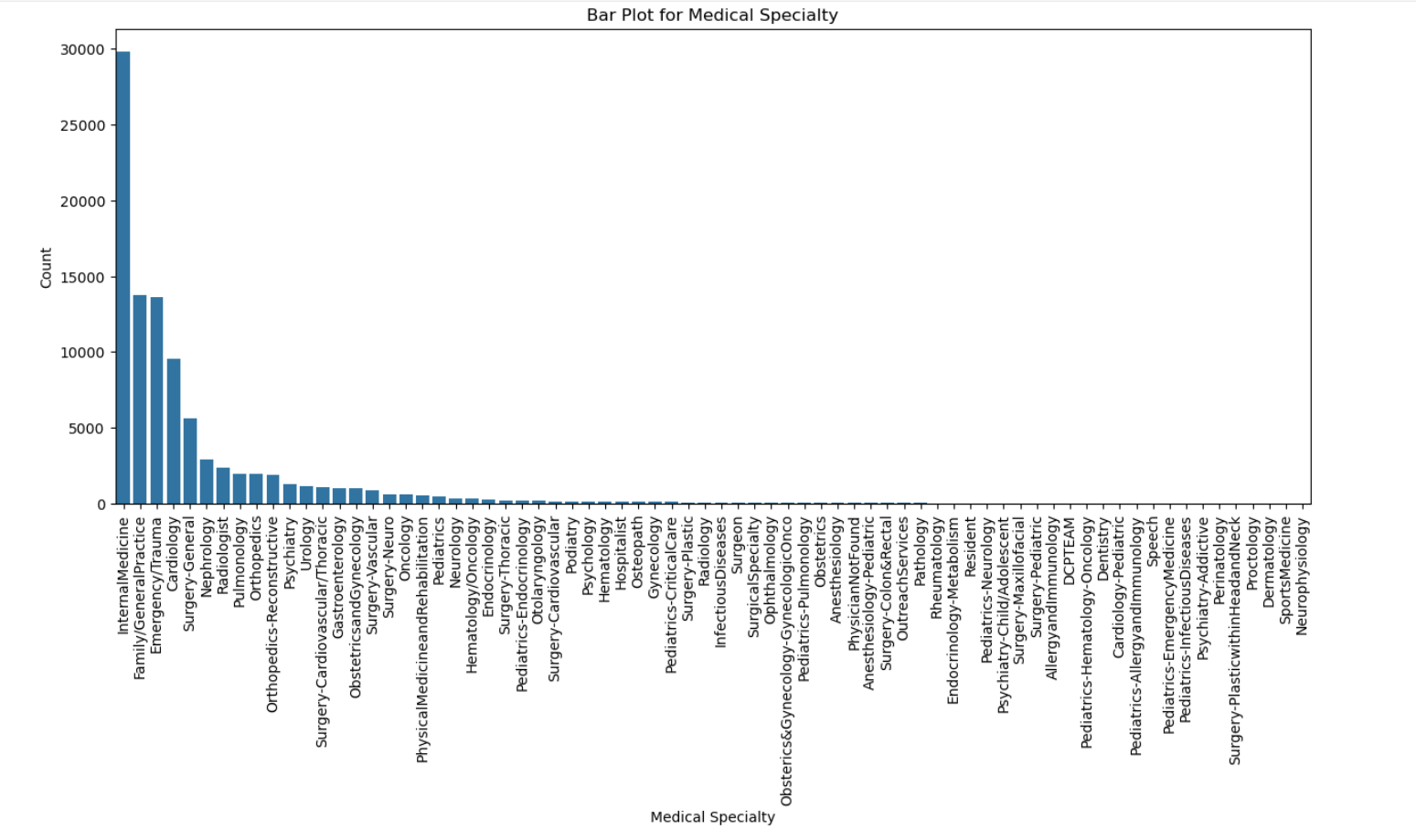
* Shorter bars indicate rare categories, which might represent outliers, unique cases, or underrepresented groups that could require special attention in analysis.

1. **Data Cleaning Indicators**:

* Extremely short bars might indicate data entry errors, uncommon values, or need for grouping into "Others" to avoid fragmentation.

1. **Category Comparisons**:

* Allows comparison across categories, making it easier to identify which groups are over- or under-represented.
  1. **Distribution of Medical speciality**

****

**Fig. 3.8.** Distribution of Medical speciality

**Inference:**

**For Medical Speciality variable:**

1. The tallest bar represents the most cases of readmission, and from the bar graph, we can see that the tallest bar corresponds to the Internal Medicine category in Medical Speciality. Hence, Internal Medicine category appears in most of the readmission cases.
2. The least number of readmission cases in the bar graph is given by the shortest bar, which is the bar corresponding to the Neurophysiology in Medical Speciality. Hence, Neurophysiology category appears in least of the readmission cases.
3. Specialties like Family Practice and Cardiology show moderate readmissions, reflecting their balanced case complexity. Chronic illness-focused specialties, such as Endocrinology, tend to have higher readmissions compared to acute care fields like ENT. The distribution also suggests a potential link between patient volume and readmission rates, emphasizing the importance of preventive care and effective follow-ups to reduce hospitalizations.
   1. **Distribution of diag\_1, diag\_2, diag\_3**

|  |
| --- |
|  |
|  |
|  |

**Fig. 3.9.** Distribution of diag\_1, diag\_2, diag\_3

**Inference:**

**For diag\_1:**

1. **Circulatory category:**

* The tallest bar indicates this is the most common diagnostic category for readmissions.
* Likely includes conditions such as hypertension, heart failure, or myocardial infarction.

1. **Neoplasms category:**

* The shortest bar represents the least number of readmissions.
* Could suggest fewer acute episodes requiring frequent hospitalizations.

**For diag\_2:**

* The **circulatory category** remains prominent, highlighting its significance in secondary diagnoses related to readmissions.
* **Diabetes-related diagnoses** are also frequently observed, indicating a high prevalence of diabetes as a comorbidity in readmitted patients.
* Categories like **injuries or accidents** are less frequent, as indicated by shorter bars, showing that these are less common secondary contributors.

**For diag\_3:**

* Similar trends are observed, with **circulatory and diabetes-related categories** remaining significant contributors to readmission cases.
* **Respiratory conditions** appear more prominently in diag\_3, indicating their role as tertiary contributors to readmissions.
* The shortest bars in diag\_3 represent rarer conditions, such as certain infectious diseases or neurological disorders, which occur less frequently in readmission cases.

# **Presence of outliers and its treatment**

**Summary on Outlier Treatment:**

Outliers can significantly affect the mean and standard deviation, leading to increased error variance and reduced power in statistical tests. Identifying outliers is essential for improving model accuracy, especially in continuous numerical variables, where they distort central tendency measures. Outlier detection can be done using visual methods like box plots or statistical techniques such as z-scores. While discrete variables are less affected, outliers should still be addressed if they are physically possible. The primary goal of outlier treatment is to ensure more reliable analysis and robust machine learning models.

**Unfortunately, we don't have any continuous numerical variables in our dataset.**

# **Statistical significance of variables**

#### **For Numerical variables:**

**t test for discrete data**

The one sample t-test requires the sample data to be numeric and continuous, as it is based on the normal distribution. Hence, t test is not -appropriate for our numerical data which are discrete.

**Kruskal–Wallis test**

The Kruskal–Wallis test is a statistical test used to compare two or more groups for a continuous or discrete variable. It is a non-parametric test, meaning that it assumes no particular distribution of your data and is analogous to the one-way analysis of variance (ANOVA).

**Inference Based on Kruskal-Wallis Test Results**

The Kruskal-Wallis test was conducted to evaluate whether there are statistically significant differences in the mean values of numerical variables between the two groups of the variable readmitted (0 = not readmitted, 1 = readmitted). Below are the inferences for each variable analyzed:

1. **admission\_type\_id**:
   * There is a statistically significant difference between the mean values of the two groups.
   * This indicates that the type of admission significantly impacts whether a patient is readmitted.
2. **admission\_source\_id:**
   * A statistically significant difference exists between the two groups.
   * The source of admission has an influence on patient readmission.
3. **time\_in\_hospital:**
   * A statistically significant difference was found between the groups.
   * Patients' length of stay in the hospital is associated with their likelihood of readmission.
4. **num\_lab\_procedures**:
   * There is a statistically significant difference between the mean values of the two groups.
   * The number of laboratory procedures impacts the probability of readmission.
5. **num\_procedures:**
   * No statistically significant difference was found between the groups.
   * This indicates that the number of procedures performed is not a strong predictor of readmission.
6. **num\_medications**:
   * A statistically significant difference was observed.
   * The number of medications prescribed significantly affects the likelihood of readmission.
7. **number\_outpatient:**
   * A statistically significant difference exists between the two groups.
   * The number of outpatient visits is related to readmission rates.
8. **number\_emergency:**
   * A statistically significant difference was found.
   * The frequency of emergency visits is strongly associated with patient readmission.
9. **number\_inpatient:**
   * A statistically significant difference was observed.
   * The number of inpatient visits significantly influences the likelihood of readmission.
10. **number\_diagnoses:**
    * There is a statistically significant difference between the two groups.
    * The number of diagnoses recorded impacts patient readmission rates.

**Key Insights**

* Most of the variables analyzed show a statistically significant difference between the groups, indicating they are important factors in understanding patient readmission patterns.
* However, **num\_procedures** does not show a significant difference, suggesting it is less relevant in predicting readmission.
* Variables like **time\_in\_hospital**, **num\_medications**, and **number\_inpatient** appear to have a strong association with readmission, making them potential focus areas for intervention.

#### **For Categorical variables:**

**The χ2 - (Chi Square) test of independence analysis utilizes a cross tabulation table between the variables of interest r rows and c columns.**

Based on the cell counts, it is possible to test if there is a relationship, dependence, between the variables and to estimate the strength of the relationship.

**Assumptions**

* The two samples are independent
* No expected cell count is = 0
* No more than 20% of the cells have and expected cell count < 5

**Hypothesis**

Null hypothesis H0: Variables are independent

Alternative hypothesis H1: Variables are NOT independent

From the output, the function chk\_chisq checks whether each categorical variable is dependent on the target variable readmitted using the Chi-Square test of independence. Here's an overview of the results and interpretations for some variables:

1. **Variable: race**
   * **p-value**: 0.1051 (greater than 0.05)
   * **Conclusion**: race and readmitted are independent.
   * **Expected Counts**: No cells with counts < 5.
2. **Variable: gender**
   * **p-value**: 0.7334 (greater than 0.05)
   * **Conclusion**: gender and readmitted are independent.
   * **Expected Counts**: 33.33% of cells with counts < 5 (violates assumption).
3. **Variable: age**
   * **p-value**: 8.68 e-23 (very small, lesser than 0.05)
   * **Conclusion**: age and readmitted are dependent.
   * **Expected Counts**: No cells with counts < 5.
4. **Variable: payer\_code**
   * **p-value**: 2.13e-12 (very small, lesser than 0.05)
   * **Conclusion**: payer\_code and readmitted are dependent.
   * **Expected Counts**: 5.88% cells with counts < 5 (violates assumption).
5. **Variable: medical\_specialty**
   * **p-value**: 6.39e-36 (very small, less than 0.05)
   * **Conclusion**: medical\_specialty and readmitted are dependent.
   * **Expected Counts**: 29.66% of cells with counts < 5 (violates assumption).
6. **Variable: diag\_1**
   * **p-value**: 1.78e-89 (very small, less than 0.05)
   * **Conclusion**: diag\_1 and readmitted are dependent.
   * **Expected Counts**: 52.04% of cells with counts < 5 (violates assumption).
7. **Variable: diag\_2**
   * **p-value**: 1.13e-37 (very small, less than 0.05)
   * **Conclusion**: diag\_2 and readmitted are dependent.
   * **Expected Counts**: 56.47% of cells with counts < 5 (violates assumption).
8. **Variable: diag\_3**
   * **p-value**: 3.65e\_52 (very small, less than 0.05)
   * **Conclusion**: diag\_3 and readmitted are dependent.
   * **Expected Counts**: 56.45% of cells with counts < 5 (violates assumption).
9. **Variable: max\_glu\_serum**
   * **p-value**: 0.00135 (less than 0.05)
   * **Conclusion**: max\_glu\_serum and readmitted are dependent.
   * **Expected Counts**: No cells with counts < 5
10. **Variable: A1Cresult**
    * **p-value:** 1.57e-09 (very small, less than 0.05)
    * **Conclusion:** A1Cresult and readmitted are dependent.
    * **Expected Counts:** 50% of cells with counts < 5 (violates assumption).
11. **Variable: metformin**
    * **p-value:** 1.61e-13 (very small, less than 0.05)
    * **Conclusion:** metformin and readmitted are dependent.
    * **Expected Counts:** No cells with counts < 5
12. **Variable: repaglinide**

o **p-value:** 0.01449 (less than 0.05)

* + **Conclusion:** repaglinide and readmitted are dependent.
  + **Expected Counts:** 12.50% of cells with counts < 5 (violates assumption).

1. **Variable: nateglinide**
   * **p-value:** 0.7118 (greater than 0.05)
   * **Conclusion:** nateglinide and readmitted are independent.
   * **Expected Counts:** 25.00% of cells with counts < 5 (violates assumption).
2. **Variable: chlorpropamide**
   * **p-value:** 0.4362 (greater than 0.05)
   * **Conclusion:** chlorpropamide and readmitted are independent.
   * **Expected Counts:** 25.00% of cells with counts < 5 (violates assumption).
3. **Variable: glimepiride**
   * **p-value:** 0.0281 (lesser than 0.05)
   * **Conclusion:** glimepiride and readmitted are dependent.
   * **Expected Counts:** No cells with counts < 5
4. **Variable: acetohexamide**
   * **p-value:** 1 (greater than 0.05)
   * **Conclusion:** acetohexamideand readmitted are dependent.
   * **Expected Counts:** 50.00% of cells with counts < 5 (violates assumption).
5. **Variable: glipizide**
   * **p-value:** 0.0156 (smaller than 0.05)
   * **Conclusion:** glipizide and readmitted are dependent.
   * **Expected Counts:** No cells with counts < 5
6. **Variable: glyburide**
   * **p-value:** 0.1141 (smaller than 0.05)
   * **Conclusion:** glyburide and readmitted are independent.
   * **Expected Counts:** No cells with counts < 5
7. **Variable: tolbutamide**
   * **p-value** 0.5456(greater than 0.05)
   * **Conclusion:** tolbutamide and readmitted are independent.
   * **Expected Counts:** 25.00% of cells with counts < 5
8. **Variable: pioglitazone**
   * **p-value** 0.5456(greater than 0.05)
   * **Conclusion:** pioglitazone and readmitted are independent.
   * **Expected Counts:** No cells with counts < 5
9. **Variable: rosiglitazone**
   * **p-value** 0.08 (greater than 0.05)
   * **Conclusion:** rosiglitazone and readmitted are independent.
   * **Expected Counts:** No cells with counts < 5
10. **Variable: acarbose**
    * **p-value** 0.4417 (greater than 0.05)
    * **Conclusion:** acarbose and readmitted are independent.
    * **Expected Counts:** 37.50% of cells with counts < 5 (violates assumption)
11. **Variable: miglitol**
    * **p-value** 0.1049 (greater than 0.05)
    * **Conclusion:** miglitol and readmitted are independent.
    * **Expected Counts:** 62.50% of cells with counts < 5 (violates assumption)
12. **Variable: troglitazone**
    * **p-value** 1 (greater than 0.05)
    * **Conclusion:** troglitazone and readmitted are independent.
    * **Expected Counts:** 50.00% of cells with counts < 5 (violates assumption)
13. **Variable: tolazamide**
    * **p-value** 0.6462 (greater than 0.05)
    * **Conclusion:** tolazamide and readmitted are independent.
    * **Expected Counts:** 25.00% of cells with counts < 5 (violates assumption)
14. **Variable: insulin**
    * **p-value** 4.35e-41 (very small, less than 0.05)
    * **Conclusion:** insulin and readmitted are dependent.
    * **Expected Counts:** No cells with counts < 5
15. **Variable: glyburide-metformin**
    * **p-value** 0.7542 (greater than 0.05)
    * **Conclusion:** glyburide-metformin and readmitted are dependent.
    * **Expected Counts:** 25.00% of cells with counts < 5
16. **Variable: glipizide-metformin**
    * **p-value** 1 (greater than 0.05)
    * **Conclusion:** glipizide-metformin and readmitted are independent.
    * **Expected Counts:** 25.00% of cells with counts < 5 (violates assumption)
17. **Variable: glimepiride-pioglitazone**
    * **p-value** 1 (greater than 0.05)
    * **Conclusion** glimepiride-pioglitazoneand readmitted are independent.
    * **Expected Counts:** 50.00% of cells with counts < 5 (violates assumption)
18. **Variable: metformin-rosiglitazone**
    * **p-value** 1 (greater than 0.05)
    * **Conclusion** metformin-rosiglitazoneand readmitted are independent.
    * **Expected Counts:** 50.00% of cells with counts < 5 (violates assumption)
19. **Variable: metformin-pioglitazone**
    * **p-value** 1 (greater than 0.05)
    * **Conclusion** metformin-pioglitazone and readmitted are independent.
    * **Expected Counts:** 50.00% of cells with counts < 5 (violates assumption)
20. **Variable: change**
    * **p-value** 1 (greater than 0.05)
    * **Conclusion** change and readmitted are independent.
    * **Expected Counts:** No cells with counts < 5
21. **Variable: diabetesMed**
    * **p-value** 2.2e-15 (very small, less than 0.05)
    * **Conclusion** diabetesMed and readmitted are independent.
    * **Expected Counts:** No cells with counts < 5
22. **Variable: Discharged to**
    * **p-value** 1.38e-172 (very small, less than 0.05)
    * **Conclusion** Discharged to and readmitted are dependent.
    * **Expected Counts:** No cells with counts < 5

**Key Points:**

1. **Assumptions**:
   * The Chi-Square test requires that no more than 20% of cells have expected counts < 5. This is violated for the following variables:

*1) payer\_code 2) medical\_specialty 3) diag\_1*

*4) diag\_2 5) diag\_3 6) repaglinide*

*7) nateglinide 8) chlorpropamide 9) acetohexamide 10) tolbutamide 11) acarbose 12) miglitol*

*13) troglitazone 14) tolazamide 15) glyburide-metformin*

*16) glipizide-metformin 17) glimepiride-pioglitazone 18) metformin-rosiglitazone*

*19) metformin-pioglitazone*

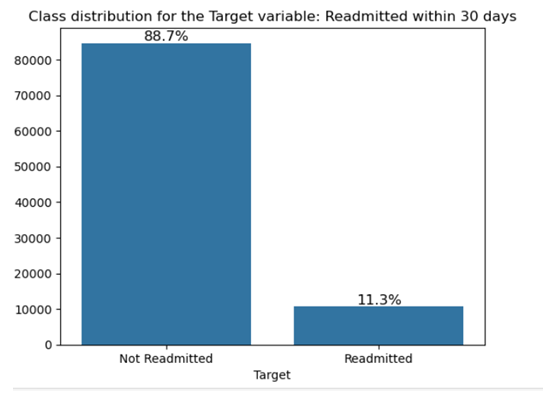
* + Results for these variables should be treated with caution or supplemented with Fisher’s Exact Test if possible.

1. **Significance**:
   * The following variables with a p-value less than 0.05 are dependent on readmitted:

*age, payer\_code, medical\_specialty, diag\_1, diag\_2, diag\_3,max\_glu\_serum, A1Cresult, metformin, repaglinide, glimepiride,glipizide, pioglitazone, insulin, change, diabetesMed, Discharged to.*

1. **Next Steps**:
   * Address assumption violations by collapsing categories or using alternative methods.
   * Focus on significant variables while ensuring assumptions are met.

# **Class imbalance and its treatment**



**Fig. 3.10.** Barplot readmission class imbalance

**Observations**

As visible, our data is highly imbalanced. Imbalanced datasets can lead to a bias towards the majority class constituting 88.7% of the total, as the model is trained on a majority of samples from the majority class. This can result in poor performance in the minority class. Hence, we need to treat data imbalance.

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don’t add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique or SMOTE for short.

**Another method is under-sampling.** Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. By keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modelling.

# **Feature Engineering**

## **Whether any transformations required**

* Data transformation is essential for preparing the dataset for model building. In our case, label encoding was applied to the dataset to make it suitable for the model. Additionally, the target variable, 'Target' was created based on the following conditions:
* 0: If df['readmitted'] is '>30' or 'NO'
* 1: If df['readmitted'] is '<30’
* From the variable, 'discharge\_disposition\_id' indicating where the patient went after discharge, we created a variable, 'Discharged to' as follows:

Home = [1, 13] # values in the column, discharge\_disposition\_id

Facility = [2, 3, 4, 5, 6, 8, 9, 10, 14, 15, 16, 17, 22, 23, 24, 30, 27, 28, 29] Home\_w\_homecare = [6, 8]

Others = [7]

## **Scaling the data**

**Scaling:**

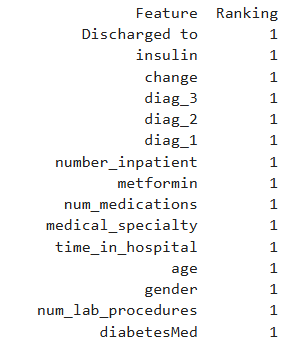
Scaling continuous variables is crucial to ensure they have a meaningful impact on the model. However, scaling columns with binary values (0 or 1) is unnecessary as it doesn't improve the model's performance. Since binary variables, such as dummy variables, represent categorical information, scaling them would distort their intended influence. **In our dataset, there are no continuous numerical variables, so there is no need for scaling.** Without continuous variables, we don't have to worry about the issues that arise from scaling such features, and thus can focus on other preprocessing steps to prepare the data for model building.

# **Feature selection**

1. **Recursive Feature Elimination:**

Recursive Feature Elimination (RFE) is a powerful feature selection technique in machine learning that helps identify the most relevant features for model training. By recursively removing the least important features, RFE aims to improve the model’s performance by reducing overfitting and enhancing interpretability. It is especially useful when working with datasets that contain many features, allowing you to narrow down the dataset to only the most significant ones.

1. **Steps to Use RFE:**
2. **Choose an Estimator:** RFE requires an estimator (like DecisionTreeClassifier, LogisticRegression, etc.) that can calculate feature importance.
3. **Specify Number of Features:** Define the number of features you want to keep using the n\_features\_to\_select parameter.
4. **Fit the Model:** Fit the RFE model to the training data using fit(). It evaluates the features and eliminates the least important ones.
5. **Evaluate Feature Selection:** Use the support\_ attribute to see which features are selected (True for selected). The ranking\_ attribute shows the rank of all features (1 for most important).
6. **Transform the Data:** Use transform() to apply the feature selection to the dataset, keeping only the most important features.



**Tab. 3.5.** Top 15 Important Features

**Top 15 features selected by RFE**

gender, age, time\_in\_hospital, medical\_specialty, num\_lab\_procedures, num\_medications, number\_inpatient, diag\_1, diag\_2, diag\_3, metformin, insulin, change, diabetesMed, 'Discharged to’

**Basic Patient Information**

* gender**:** Whether the patient is male or female.
* age: The patient's age.

**Hospital Stay Details**

* ‘Discharged to’: The type of discharge (e.g., home, transferred to another facility).
* time\_in\_hospital: The length of the hospital stay.

**Medical History and Treatments**

* medical\_specialty: The primary medical specialty of the patient.
* num\_lab\_procedures: The number of lab tests conducted.
* num\_medications: The number of medications prescribed.
* number\_inpatient: The number of inpatient admissions in the past year.
* diag\_1, diag\_2, diag\_3: The primary, secondary, and tertiary diagnoses.
* number\_diagnoses: The total number of diagnoses.

**Diabetes-Related Factors**

* metformin: Whether the patient is taking metformin.
* Insulin: Whether the patient is taking insulin.
* change: Whether there was a change in medication dosage.
* diabetesMed: Whether the patient is taking any diabetes medication.

**Why are these features important?**

These features likely influence readmission because they provide insights into the patient's health condition, the complexity of their treatment, and the potential for complications. For example:

* Older patients and those with multiple chronic conditions may be more prone to readmission.
* Longer hospital stays and complex medical histories can increase the risk of complications.
* Changes in medication or new diagnoses may require additional care or monitoring.
* Diabetes-related factors are crucial because diabetes can lead to various complications that may necessitate readmission.

By focusing on these key features, your model can make more accurate predictions about patient readmission, which can help healthcare providers identify patients at risk and take proactive measures to prevent readmission.

# **Dimensionality reduction**

By employing a feature selection technique, we were able to significantly reduce the dimensionality of our dataset. This process involved identifying and selecting the 15 most relevant features from the original dataset. By focusing on these key features, we aim to improve the accuracy and efficiency of our predictive model while maintaining its interpretability. So, dimensionality reduction was not necessary in this case, as feature selection itself effectively addressed the issue of high-dimensional data.

# **Model evaluation**

Model evaluation was central to the study, focusing on understanding the recall metric due to its importance in identifying patients at risk of readmission. Recall measures the proportion of actual positive cases (patients readmitted within 30 days) correctly identified by the model. Misclassifying these patients as non-readmitted has significant clinical and operational implications.

**Models Assessed:**

The study evaluated seven machine learning models:

1. **CART (Decision Tree)**
2. **Naïve Bayes (NB)**
3. **K-Nearest Neighbors (KNN)**
4. **XGBoost**
5. **Logistic Regression (LR)**
6. **AdaBoost**
7. **Random Forest (RF)**

**Evaluation Methodology:**

* **K-Fold Cross-Validation**: A 10-fold cross-validation method was used where appropriate to ensure model robustness and prevent overfitting. When SMOTE or SMOTEEN was applied, we tested with training data and test data after splitting the dataset in the ratio 80:20.
* **Default vs. Tuned Parameters**: Models were first assessed with default parameters and later with hyperparameter tuning to optimize performance. Tuning was conducted systematically to balance recall improvement with computational efficiency.

**Findings:**

The following table summarizes the recall scores of the models before and after tuning:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Recall (Default)** | **Recall (Tuned)** | **Improvement (%)** |
| CART | 0.18 ± 0.01 | 0.55 ± 0.01 | **+205.56%** |
| Random Forest | 0.01 ± 0.00 | 0.63 ± 0.01 | **+6200%** |
| Naïve Bayes | 0.11 ± 0.01 | 0.11 ± 0.01 | No Change |
| KNN | 0.02 ± 0.01 | 0.05 ± 0.01 | +150% |
| XGBoost | 0.02 ± 0.00 | 0.04 ± 0.01 | +100% |
| Logistic Reg. | 0.02 ± 0.00 | 0.02 ± 0.00 | No Change |
| AdaBoost | 0.01 ± 0.00 | 0.00 ± 0.00 | -100% |

**Tab.4.1.** Model comparison chart before treating for data imbalance

**Insights:**

* Decision Tree (CART) and Random Forest models showed the highest recall improvements after hyperparameter tuning. Their ability to model complex relationships while maintaining interpretability made them the most suitable choices for predicting readmissions.
* Naïve Bayes and Logistic Regression demonstrated limited performance, likely due to the dataset’s complexity and non-linear patterns.
* Ensemble methods like AdaBoost and XGBoost showed incremental improvements but fell short compared to CART and Random Forest in terms of recall.

The parameters after hyper parameter tuning are given below and used for evaluating the model performance on both training dataset and test dataset. For more details, refer to Appendix – Section 10.2.

{'warm\_start': False, 'verbose': 0, 'random\_state': None, 'oob\_score': False, 'n\_jobs': None, 'n\_estimators': 50, 'min\_weight\_fraction\_leaf': 0.0, 'min\_samples\_split': 2, 'min\_samples\_leaf': 1, 'min\_impurity\_decrease': 0.0, 'max\_samples': None, 'max\_leaf\_nodes': None, 'max\_features': 0.75, 'max\_depth': 5, 'criterion': 'gini', 'class\_weight': 'balanced', 'ccp\_alpha': 0.0, 'bootstrap': True}

**Evaluation of model performance with data treated for data imbalance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Recall measure for the minority class of the Decision Tree model using** | **Before treating for data imbalance** | **After applying Random under-sampling** | **After applying SMOTE** | **After applying SMOTEEN** |
| Training dataset | 0.63 | 0.63 | 0.66 | 0.67 |
| Test dataset | 0.62 | 0.62 | 0.62 | 0.62 |

**Tab.4.2.** Decision Tree Model performance comparison chart after treating for data imbalance

We observe that the model performance measure, recall has did not improve from 0.62 when we treated the data imbalance through Random under-sampling, SMOTE or SMOTEEN.

# **Comparison with Benchmark**

The performance of the models was benchmarked against previous studies that used similar datasets and predictive modeling techniques for hospital readmissions. Typically, recall values for such models ranged from **50% to 60%** for the minority class.

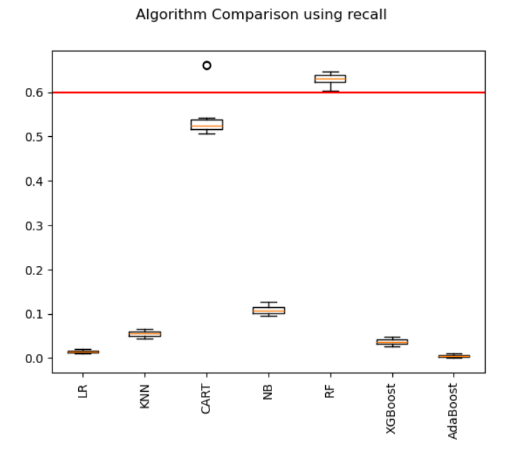
**Our Findings:**

* The tuned Random Forest model achieved a recall of **62%**, outperforming conventional benchmarks before treating for data imbalance. We observe that the model performance measure, recall did not improve from 0.62 when we treated the data imbalance through Random under-sampling, SMOTE or SMOTEEN.
* These results underscore the effectiveness of hyperparameter tuning and enhancing model performance.

# **Visualizations**

These visualizations offer actionable insights for hospital administrators and healthcare providers.

**Model Comparison Chart:** Random Forest with tuned parameters showed the most significant improvements, validating their suitability for this problem.



**Fig.6.1.** Boxplot comparison of recall measure across various models

**Feature Importance Plot:**

* From Fig.7.1. in section 7, we infer, the Random Forest feature importance analysis highlighted ‘number\_inpatient’ and ‘Discharged to’ as the two most influential predictors.
* These variables directly relate to hospital policies and patient care, reinforcing their practical importance.

**Barplot Plots:**

* A barplot is a type of plot that displays the numerical values for different categorical variables.

|  |  |
| --- | --- |
|  |  |
| **Tab.6.1.** Crosstab between ‘Discharged to’ and the Target variables | **Fig.6.2.** Barplot of ‘Discharged to’ and the Target variables |

**Inference:**

Instances of readmission is less when discharged to Home.

|  |  |
| --- | --- |
|  |  |
| **Tab.6.2.** Crosstab between ‘number\_inpatient’ and the Target variables | **Fig.6.3.** Barplot between ‘number\_inpatient’ and the Target variables |

**Inference:**

From the bar plot showing the Number of outpatient visits of the patient in the year preceding the encounter vs readmission, we infer that readmission cases are less when number of patient visits are least.

# **Implications**

|  |  |
| --- | --- |
|  |  |
| **Tab.7.1.** Variable importance showing % Reduction in a criterion or Gini Importance | **Fig.7.1.** Variable importance plot showing % Reduction in a criterion or Gini Importance |

We observed that the features, ‘number\_inpatient’ and ‘discharged to’ indicating the type of discharge (e.g., home, transferred to another facility) are the top most influencing factors for readmission of patients within 30 days.

We infer instances of readmission is less when discharged to Home when compared to discharged to home healthcare referral. This result raises the question of why home health care services did not produce evidence of lower post-discharge return to hospital rates. Focused attention by home health care programs on strategies to reduce readmissions is needed. <https://www.sciencedirect.com/science/article/abs/pii/S0020748921000894>

Similarly, number of inpatient visits of the patient in the year preceding the encounter is directly connected with hospital readmission.

This strengthens our finding that the features, ‘number\_inpatient’ and ‘discharge to’ are the two top most important factors influencing the target variable, hospital readmission.

[https://blog.rehabselect.net/5-top-reasons-hospital-readmissions#:~:text=Disengagement%20and%20Non%2DCompliance,top%20causes%20of%20preventable%20readmissions.](https://blog.rehabselect.net/5-top-reasons-hospital-readmissions%23:~:text=Disengagement%20and%20Non%2DCompliance,top%20causes%20of%20preventable%20readmissions.)

The study’s findings have significant implications for healthcare practice and policy:

1. **Clinical Applications:**
   * **Patient Monitoring:** Hospitals can prioritize patients with higher inpatient visits and specific discharge dispositions for follow-up care.
   * **Resource Allocation:** High-risk patients identified by the model can be provided additional resources, such as home visits or telehealth consultations, to prevent readmissions.
2. **Policy Recommendations:**
   * **Customized Care Plans:** Hospitals should design care plans tailored to patient-specific risk factors, such as frequent inpatient visits.
   * **Discharge Planning:** Improved discharge planning can mitigate the risks associated with non-compliance and disengagement.
3. **Technological Advancements:**
   * Integrating predictive models into hospital management systems can automate risk stratification and guide interventions.

# **Limitations**

Despite its strengths, the study had some limitations:

1. **Dataset Temporal Scope:**
   * The dataset spans 1999–2008, potentially limiting its relevance to contemporary healthcare practices.
2. **Missing Data Imputation Bias:**
   * Iterative imputation, while robust, may introduce bias in predictions if the missing data patterns differ significantly from reality.
3. **Model Generalizability:**
   * The models’ performance may vary across different datasets, particularly those with different patient populations or healthcare systems.
4. **Limited Feature Scope:**
   * While the selected features were impactful, the exclusion of other potentially important variables (e.g., socioeconomic factors) may constrain the model’s predictive power.

# **9. Closing Reflections**

This study highlights the transformative potential of machine learning in healthcare, specifically for addressing hospital readmissions. By combining robust preprocessing, feature selection, and advanced modeling techniques, the study achieved superior recall scores, demonstrating practical applicability.

**Future Directions:**

1. Incorporate modern datasets with updated medical practices to validate and refine the models.
2. Explore ensemble and deep learning methods to enhance performance further.
3. Collaborate with healthcare institutions to pilot the models in real-world settings, gathering feedback for iterative improvement.

# **10.APPENDIX**

## **DATA DICTIONARY**

| **#** | **Variable Name** | **Role** | **Type** | **Description** | **Missing Values** |
| --- | --- | --- | --- | --- | --- |
| 1 | encounter\_id | ID |  | Unique identifier of an encounter | no |
| 2 | patient\_nbr | ID |  | Unique identifier of a patient | no |
| 3 | Race | Feature | Categorical | Values: Caucasian, Asian, African American, Hispanic, and other.  **Missing values are denoted by ?** | yes |
| 4 | Gender | Feature | Categorical | Values: male, female, and unknown/invalid | no |
| 5 | Age | Feature | Categorical | Grouped in 10-year intervals: [0, 10), [10, 20),..., [90, 100) | no |
| 6 | Weight | Feature | Categorical | Weight in pounds.  **Missing values are denoted by ?** | yes |
| 7 | admission\_type\_id | Feature | Categorical | Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available | no |
| 8 | discharge\_disposition\_id | Feature | Categorical | Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available | no |
| 9 | admission\_source\_id | Feature | Categorical | Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital | no |
| 10 | time\_in\_hospital | Feature | Integer | Integer number of days between admission and discharge | no |
| 11 | payer\_code | Feature | Categorical | Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay.  **Missing values are denoted by ?** | yes |
| 12 | medical\_specialty | Feature | Categorical | Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon.  **Missing values are denoted by ?** | yes |
| 13 | num\_lab\_procedures | Feature | Integer | Number of lab tests performed during the encounter | no |
| 14 | num\_procedures | Feature | Integer | Number of procedures (other than lab tests) performed during the encounter | no |
| 15 | num\_medications | Feature | Integer | Number of distinct generic names administered during the encounter | no |
| 16 | number\_outpatient | Feature | Integer | Number of outpatient visits of the patient in the year preceding the encounter | no |
| 17 | number\_emergency | Feature | Integer | Number of emergency visits of the patient in the year preceding the encounter | no |
| 18 | number\_inpatient | Feature | Integer | Number of inpatient visits of the patient in the year preceding the encounter | no |
| 19 | diag\_1 | Feature | Categorical | The primary diagnosis (coded as first three digits of ICD9); 848 distinct values.  **Missing values are denoted by ?** | yes |
| 20 | diag\_2 | Feature | Categorical | Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values.  **Missing values are denoted by ?** | yes |
| 21 | diag\_3 | Feature | Categorical | Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values.  **Missing values are denoted by ?** | yes |
| 22 | number\_diagnoses | Feature | Integer | Number of diagnoses entered to the system | no |
| 23 | max\_glu\_serum | Feature | Categorical | Indicates the range of the result or if the test was not taken. Values: >200, >300, normal, and none if not measured | no |
| 24 | A1Cresult | Feature | Categorical | Indicates the range of the result or if the test was not taken. Values: >8 if the result was greater than 8%, >7 if the result was greater than 7% but less than 8%, normal if the result was less than 7%, and none if not measured. | no |
| 25 | metformin | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 26 | repaglinide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 27 | nateglinide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 28 | chlorpropamide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 29 | glimepiride | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 30 | acetohexamide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 31 | glipizide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 32 | glyburide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 33 | tolbutamide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 34 | pioglitazone | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 35 | rosiglitazone | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 36 | acarbose | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 37 | Miglitol | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 38 | troglitazone | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 39 | tolazamide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 40 | examide | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 41 | citoglipton | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 42 | Insulin | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 43 | glyburide-metformin | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 44 | glipizide-metformin | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 45 | glimepiride-pioglitazone | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 46 | metformin-rosiglitazone | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 47 | metformin-pioglitazone | Feature | Categorical | The feature indicates whether the drug was prescribed or there was a change in the dosage. Values: up if the dosage was increased during the encounter, down if the dosage was decreased, steady if the dosage did not change, and no if the drug was not prescribed | no |
| 48 | Change | Feature | Categorical | Indicates if there was a change in diabetic medications (either dosage or generic name). Values: change and no change | no |
| 49 | iabetesMed | Feature | Categorical | Indicates if there was any diabetic medication prescribed. Values: yes and no | no |
| 50 | readmitted | Target | Categorical | Days to inpatient readmission. Values: <30 if the patient was readmitted in less than 30 days, >30 if the patient was readmitted in more than 30 days, and No for no record of readmission. | no |

**Tab.10.1.** Data Dictionary

**Values for Key id variables**

|  |  |
| --- | --- |
| admission\_type\_id | description |
| 1 | Emergency |
| 2 | Urgent |
| 3 | Elective |
| 4 | Newborn |
| 5 | Not Available |
| 6 | NULL |
| 7 | Trauma Center |
| 8 | Not Mapped |
|  |  |
| discharge\_disposition\_id | description |
| 1 | Discharged to home |
| 2 | Discharged/transferred to another short term hospital |
| 3 | Discharged/transferred to SNF |
| 4 | Discharged/transferred to ICF |
| 5 | Discharged/transferred to another type of inpatient care institution |
| 6 | Discharged/transferred to home with home health service |
| 7 | Left AMA |
| 8 | Discharged/transferred to home under care of Home IV provider |
| 9 | Admitted as an inpatient to this hospital |
| 10 | Neonate discharged to another hospital for neonatal aftercare |
| 11 | Expired |
| 12 | Still patient or expected to return for outpatient services |
| 13 | Hospice / home |
| 14 | Hospice / medical facility |
| 15 | Discharged/transferred within this institution to Medicare approved swing bed |
| 16 | Discharged/transferred/referred another institution for outpatient services |
| 17 | Discharged/transferred/referred to this institution for outpatient services |
| 18 | NULL |
| 19 | Expired at home. Medicaid only, hospice. |
| 20 | Expired in a medical facility. Medicaid only, hospice. |
| 21 | Expired, place unknown. Medicaid only, hospice. |
| 22 | Discharged/transferred to another rehab fac including rehab units of a hospital . |
| 23 | Discharged/transferred to a long term care hospital. |
| 24 | Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare. |
| 25 | Not Mapped |
| 26 | Unknown/Invalid |
| 30 | Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere |
| 27 | Discharged/transferred to a federal health care facility. |
| 28 | Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital |
| 29 | Discharged/transferred to a Critical Access Hospital (CAH). |
|  |  |
| admission\_source\_id | description |
| 1 | Physician Referral |
| 2 | Clinic Referral |
| 3 | HMO Referral |
| 4 | Transfer from a hospital |
| 5 | Transfer from a Skilled Nursing Facility (SNF) |
| 6 | Transfer from another health care facility |
| 7 | Emergency Room |
| 8 | Court/Law Enforcement |
| 9 | Not Available |
| 10 | Transfer from critial access hospital |
| 11 | Normal Delivery |
| 12 | Premature Delivery |
| 13 | Sick Baby |
| 14 | Extramural Birth |
| 15 | Not Available |
| 17 | NULL |
| 18 | Transfer From Another Home Health Agency |
| 19 | Readmission to Same Home Health Agency |
| 20 | Not Mapped |
| 21 | Unknown/Invalid |
| 22 | Transfer from hospital inpt/same fac reslt in a sep claim |
| 23 | Born inside this hospital |
| 24 | Born outside this hospital |
| 25 | Transfer from Ambulatory Surgery Center |
| 26 | Transfer from Hospice |

**Tab.10.2.** Values for key ID variables

## **Hyper parameter tuning of the model log**

**Log file:** Date & Time: 2024-12-30 14-40-00

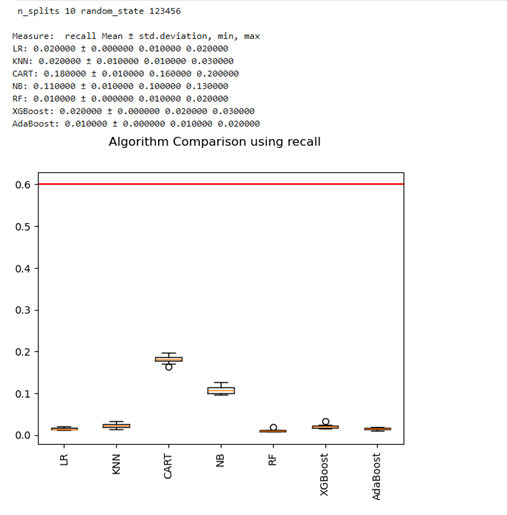
| **S. No.** | **Model** | **Parameter Space** | **Best Parameter** | **Best Recall - Binary** |
| --- | --- | --- | --- | --- |
| 1 | LogisticRegression() | {'C': array([1.00000000e-04, 2.63665090e-04, 6.95192796e-04, 1.83298071e-03,  4.83293024e-03, 1.27427499e-02, 3.35981829e-02, 8.85866790e-02, 2.33572147e-01, 6.15848211e-01, 1.62377674e+00, 4.28133240e+00,  1.12883789e+01, 2.97635144e+01, 7.84759970e+01, 2.06913808e+02, 5.45559478e+02, 1.43844989e+03, 3.79269019e+03, 1.00000000e+04]),  'class\_weight': [None], 'dual': [False], 'fit\_intercept': [True], 'intercept\_scaling': [1, 2, 3], 'l1\_ratio': [None],  'max\_iter': [100, 1000, 2500, 5000], 'multi\_class': ['auto'], 'n\_jobs': [None], 'penalty': ['l1', 'l2', 'elasticnet', 'none'],  'random\_state': [None], 'solver': ['lbfgs', 'newton-cg', 'liblinear', 'sag', 'saga'], 'tol': [0.0001, 0.001], 'verbose': [0], 'warm\_start': [False]} | {'warm\_start': False, 'verbose': 0, 'tol': 0.001, 'solver': 'sag', 'random\_state': None, 'penalty': 'l2', 'n\_jobs': None, 'multi\_class': 'auto', 'max\_iter': 1000, 'l1\_ratio': None, 'intercept\_scaling': 3, 'fit\_intercept': True, 'dual': False, 'class\_weight': None, 'C': 11.288378916846883} | 0.0172 |
| 2 | KNeighborsClassifier() | {'algorithm': ['auto', 'ball\_tree', 'kd\_tree', 'brute'], 'leaf\_size': [20, 30], 'metric': ['minkowski'], 'metric\_params': [None], 'n\_jobs': [None], 'n\_neighbors': [2, 3, 5, 11, 15, 21], 'p': [1, 2], 'weights': ['uniform', 'distance']} | {'weights': 'distance', 'p': 2, 'n\_neighbors': 3, 'n\_jobs': None, 'metric\_params': None, 'metric': 'minkowski', 'leaf\_size': 30, 'algorithm': 'auto'} | 0.0516 |
| 3 | DecisionTreeClassifier() | 'ccp\_alpha': [0.0], 'class\_weight': ['balanced'], 'criterion': ['gini', 'entropy'], 'max\_depth': [None, 2, 3, 5, 10, 20], 'max\_features': [None], 'max\_leaf\_nodes': [None], 'min\_impurity\_decrease': [0.0], 'min\_samples\_leaf': [1, 5, 10, 20, 50, 100], 'min\_samples\_split': [2, 5], 'min\_weight\_fraction\_leaf': [0.0], 'random\_state': [None], 'splitter': ['best']} | {'splitter': 'best', 'random\_state': None, 'min\_weight\_fraction\_leaf': 0.0, 'min\_samples\_split': 5, 'min\_samples\_leaf': 10, 'min\_impurity\_decrease': 0.0, 'max\_leaf\_nodes': None, 'max\_features': None, 'max\_depth': 3, 'criterion': 'entropy', 'class\_weight': 'balanced', 'ccp\_alpha': 0.0} | 0.699 |
| 4 | GaussianNB() | {'var\_smoothing': [0.001, 1e-09]} | {'var\_smoothing': 1e-09} | 0.160 |
| 5 | RandomForestClassifier() | {'bootstrap': [True], 'ccp\_alpha': [0.0], 'class\_weight': ['balanced'], 'criterion': ['gini', 'entropy', 'log\_loss'], 'max\_depth': [None, 4, 5, 6, 7, 8], 'max\_features': ['sqrt', 0.25, 0.5, 0.75, 1.0], 'max\_leaf\_nodes': [None], 'max\_samples': [None], 'min\_impurity\_decrease': [0.0], 'min\_samples\_leaf': [1], 'min\_samples\_split': [2], 'min\_weight\_fraction\_leaf': [0.0], 'n\_estimators': [10, 30, 50, 100], 'n\_jobs': [None], 'oob\_score': [False], 'random\_state': [None], 'verbose': [0], 'warm\_start': [False]} | {'warm\_start': False, 'verbose': 0, 'random\_state': None, 'oob\_score': False, 'n\_jobs': None, 'n\_estimators': 50, 'min\_weight\_fraction\_leaf': 0.0, 'min\_samples\_split': 2, 'min\_samples\_leaf': 1, 'min\_impurity\_decrease': 0.0, 'max\_samples': None, 'max\_leaf\_nodes': None, 'max\_features': 0.75, 'max\_depth': 5, 'criterion': 'gini', 'class\_weight': 'balanced', 'ccp\_alpha': 0.0, 'bootstrap': True} | 0.621 |
| 6 | XGBClassifier | {'objective': ['binary:logistic'], 'base\_score': [None], 'booster': [None], 'callbacks': [None], 'colsample\_bylevel': [None, array([0.5, 0.6, 0.7, 0.8, 0.9])], 'colsample\_bynode': [None], 'colsample\_bytree': [None, array([0.5, 0.6, 0.7, 0.8, 0.9])], 'device': [None], 'early\_stopping\_rounds': [None], 'enable\_categorical': [False], 'eval\_metric': [None], 'feature\_types': [None], 'gamma': [None], 'grow\_policy': [None], 'importance\_type': [None], 'interaction\_constraints': [None], 'learning\_rate': [None, 0.01, 0.1, 0.2, 0.3, 0.4], 'max\_bin': [None], 'max\_cat\_threshold': [None], 'max\_cat\_to\_onehot': [None], 'max\_delta\_step': [None], 'max\_depth': [None, 3, 6, 10, 15], 'max\_leaves': [None], 'min\_child\_weight': [None], 'missing': [nan], 'monotone\_constraints': [None], 'multi\_strategy': [None], 'n\_estimators': [None, 100, 250, 500, 750], 'n\_jobs': [None], 'num\_parallel\_tree': [None], 'random\_state': [None], 'reg\_alpha': [None], 'reg\_lambda': [None], 'sampling\_method': [None], 'scale\_pos\_weight': [None], 'subsample': [None, array([0.5, 0.6, 0.7, 0.8, 0.9])], 'tree\_method': [None], 'validate\_parameters': [None], 'verbosity': [None]} | {'verbosity': None, 'validate\_parameters': None, 'tree\_method': None, 'subsample': None, 'scale\_pos\_weight': None, 'sampling\_method': None, 'reg\_lambda': None, 'reg\_alpha': None, 'random\_state': None, 'objective': 'binary:logistic', 'num\_parallel\_tree': None, 'n\_jobs': None, 'n\_estimators': None, 'multi\_strategy': None, 'monotone\_constraints': None, 'missing': nan, 'min\_child\_weight': None, 'max\_leaves': None, 'max\_depth': 10, 'max\_delta\_step': None, 'max\_cat\_to\_onehot': None, 'max\_cat\_threshold': None, 'max\_bin': None, 'learning\_rate': None, 'interaction\_constraints': None, 'importance\_type': None, 'grow\_policy': None, 'gamma': None, 'feature\_types': None, 'eval\_metric': None, 'enable\_categorical': False, 'early\_stopping\_rounds': None, 'device': None, 'colsample\_bytree': None, 'colsample\_bynode': None, 'colsample\_bylevel': None, 'callbacks': None, 'booster': None, 'base\_score': None} | 0.0413 |
| 7 | AdaBoostClassifier() | {'algorithm': ['SAMME', 'SAMME.R'], 'estimator': [None], 'learning\_rate': [0.01, 0.05, 0.1, 1], 'n\_estimators': [10, 100, 200, 300], 'random\_state': [None, 1234]} | {'random\_state': None, 'n\_estimators': 10, 'learning\_rate': 1, 'estimator': None, 'algorithm': 'SAMME.R'} | 0.0033 |

**--- > This application was processed using the PC with the below configuration ---->**

| **Parameter** | **Value** |
| --- | --- |
| os.name | Nt |
| sys.platform | win32 |
| platform.system() | Windows |
| sysconfig.get\_platform() | win-amd64 |
| platform.machine() | AMD64 |
| platform.architecture() | Tuple: ('64bit', 'WindowsPE') |
| Total RAM | 8.0 GB |
| RAM memory % used | 83.5 |
| RAM Used (GB) | 6.46 |
|  | |

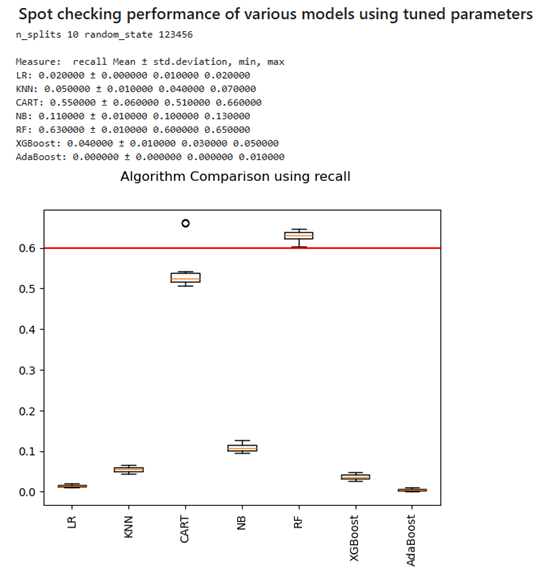
**Tab.10.3.** Hyper-parameter tuning results and OS details

## **Spot checking of model performance before tuning on data not treated for data imbalance**



**Fig.10.1.** Model comparison using Recall before hyper parameter tuning

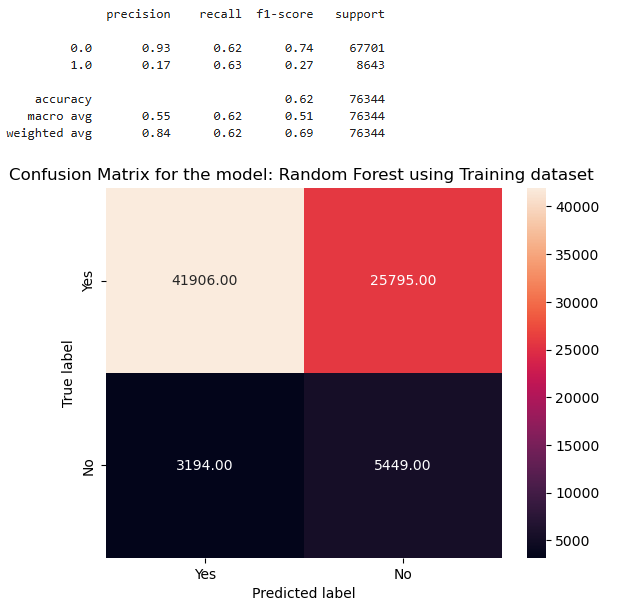
## **Spot checking of model performance after tuning on data not treated for data imbalance**

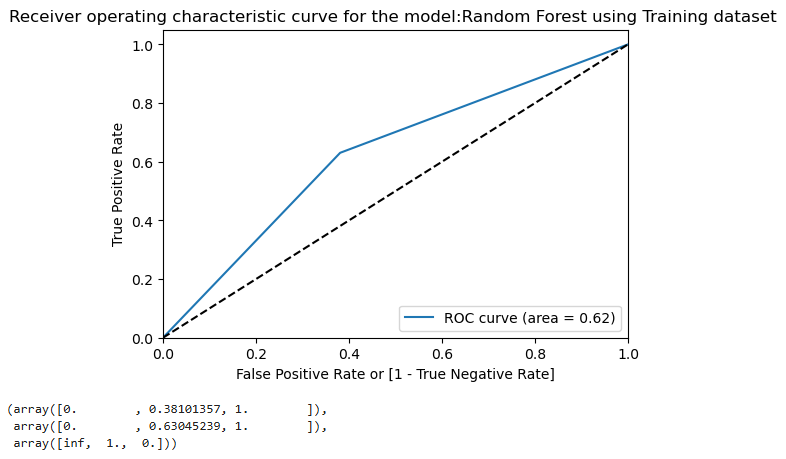


**Fig.10.2.** Model comparison using Recall after hyper parameter tuning

## **Random Forest model performance on training data after tuning after treating for data imbalance**

## **For Training Data:**

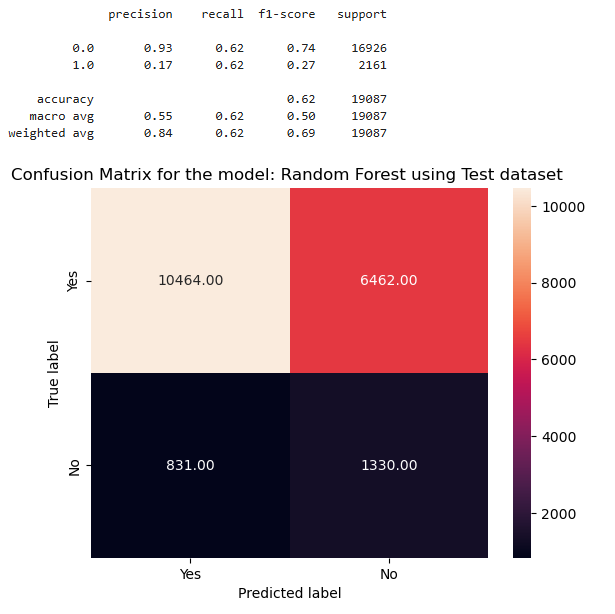


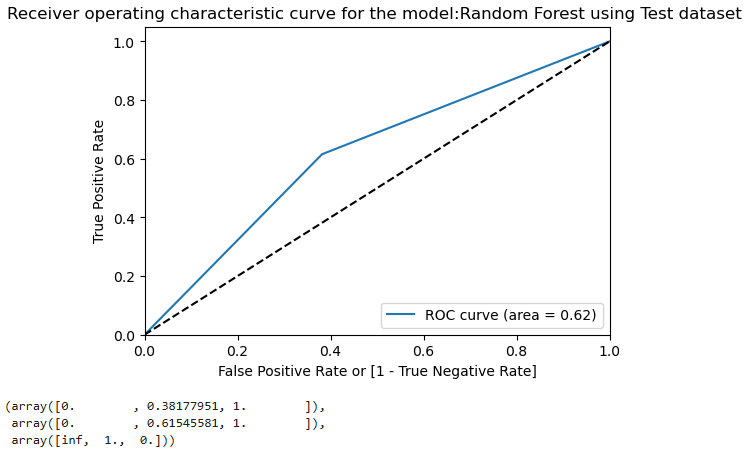


**Fig.10.3.** Recall of Random Forest model using Training Data

## **Random Forest model performance on test data after tuning after treating for data imbalance**

## **For Test Data:**





**Fig.10.4.** Recall of Random Forest model using Test Data