**Interim Project report on Prediction of Hospital readmission within 30 days**

**Submitted By**

**Group No. 2 Batch: June 2024** **Location: Bangalore**

**Group Members**

1. Rahul Yadav
2. Aswin
3. Preeti Dhanuka
4. Bhavitha Sanivarapu
5. Nishith Ramesh
6. Lohith T S

**Research Supervisor**

**P. V. Subramanian**

Date:



Signature of the Mentor Signature of the Team Leader

[LIST OF TABLES AND FIGURES 3](#_Toc185163830)

[1) Industry Review 4](#_Toc185163831)

[**1.1.** **Industry Review - Current Practices, Background Research** 4](#_Toc185163832)

[**1.2.** **Literature Survey - Publications, Application, past and undergoing research** 4](#_Toc185163833)

[2) Dataset and Domain 5](#_Toc185163834)

[**a)** **Data Dictionary** 5](#_Toc185163837)

[**b)** **Variable categorization (count of numeric and categorical)** 5](#_Toc185163838)

[**c)** **Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)** 7](#_Toc185163839)

[**d)** **Alternate sources of data that can supplement the core dataset (at least 2-3 columns)** 10](#_Toc185163840)

[**e)** **Project Justification - Project Statement, Complexity involved, Project Outcome** 10](#_Toc185163841)

[**f)** **Commercial, Academic or Social value** 11](#_Toc185163842)

[3) Data Exploration (EDA) 12](#_Toc185163843)

[**a)** **Relationship between variables** 12](#_Toc185163845)

[**3.3** **Checking for distribution of variables** 19](#_Toc185163846)

[**3.4** **Checking for presence of outliers and its treatment** 26](#_Toc185163847)

[**3.5** **Checking for statistical significance of variables** 26](#_Toc185163848)

[**3.6** **Checking for class imbalance and its treatment** 30](#_Toc185163849)

[4) Feature Engineering 32](#_Toc185163854)

[**4.1.** **Transformations required** 32](#_Toc185163855)

[**4.2.** **Scaling the data** 32](#_Toc185163856)

[**4.3.** **Feature selection** 32](#_Toc185163857)

[**4.4.** **Dimensionality Reduction** 35](#_Toc185163858)

[5) Assumptions 35](#_Toc185163859)

[6) Building and evaluation of Base model 41](#_Toc185163860)

[7) References and Bibliography 49](#_Toc185163861)

[**A.** **References** 49](#_Toc185163862)

[**B.** **Bibliography** 50](#_Toc185163863)

[8) Appendix 50](#_Toc185163864)

[**8.1.** **Data Dictionary** 50](#_Toc185163873)

LIST OF TABLES AND FIGURES

**List of Figures**

Fig:2.1: Count plot for classes in the Target variable 10

Fig 3.1.1: Heatmap for Correlation Among Numerical Features 12

Fig 3.1.2: Boxplot for Time in Hospital Across Readmission Categories 13

Fig 3.1.3: Barplot for Readmission rate by race 14

Fig 3.1.4: Count Plot for Readmission by Gender 15

Fig 3.1.5: Readmission rate by drug doses 17

Fig 3.3.1: Histogram of Age Distribution 19

Fig 3.3.2: Histogram of numerical features 20

Fig3.3.3: Countplot of categorical features 23

Fig 3.3.4: Distribution of Medical speciality 23

Fig3.3.5: Distribution of diag\_1, diag\_2, diag\_3 25

Fig 3.6.1: Barplot readmission class imbalance 30

Fig 5.1: Log odds linear plot for numerical variables 40

Fig 6.1: Comparison of Model Performance without SMOTE using Recall 45

Fig 6.2: Comparison of Model Performance with SMOTE using Recall 46

Fig 6.3: Comparison of Model Performance with Random Undersampling 48

**List of Tables:**

Tab:2.1 – Count & Percentage of Missing values 7

Tab.2.2 – Treatment of missing values 8

Tab:2.3 – Count of Zero values 9

Tab:2.4 – Count & Percentage of Classes in the Target variable 10

Tab:3.2 – Features and VIF 18

Tab 4.3 -- Top 15 Important Features 34

Tab 5.1 -- VIF for Features 37

Tab 5.2 -- VIF for Features that are non-collinear 38

Tab:8.1 – Data Dictionary 50

# Industry Review

* 1. **Industry Review - Current Practices, Background Research**

In the healthcare industry, predicting patient readmission within 30 days is a critical task. Hospitals and healthcare providers aim to minimize readmissions to improve patient outcomes and reduce costs. Readmissions are often linked to chronic conditions such as diabetes, heart disease, and respiratory illnesses.

Current practices include:

* Electronic Health Records (EHR): Utilizing patient data such as demographics, lab results, and medical history to track patient care and predict potential readmissions.
* Risk Stratification Models: Healthcare systems employ predictive models like LACE Index and HOSPITAL score to identify high-risk patients.
* Care Coordination Programs: Programs focusing on post-discharge follow-up, medication adherence, and patient education are widely implemented.
  1. **Literature Survey - Publications, Application, past and undergoing research**

Several studies and research publications focus on readmission prediction using machine learning and statistical models:

* Applications: Predictive analytics for early identification of high-risk patients, improving discharge planning, and personalized care plans.
* Past Research: Studies highlight the use of logistic regression, decision trees, and ensemble models to predict readmissions.
* Ongoing Research: Incorporation of deep learning models, integration of social determinants of health, and use of natural language processing (NLP) for unstructured clinical notes.

**Publications**

1. Research paper with title, "Harnessing Artificial Neural Networks in Neuroscientific Research to Predict Hospital Readmissions and Enhance Patient Care" by authors: Fatima Muskan, Mustafa Numan

Artificial Neural Networks (ANNs) hold immense promise for revolutionizing healthcare by improving predictions, personalizing treatment, and advancing neuroscientific research. By addressing the associated challenges and ethical considerations, healthcare providers and researchers can harness the full potential of ANNs to enhance patient care, support personalized medicine, and advance medical knowledge. As AI technology continues to evolve, the responsible and effective use of ANNs will play a crucial role in shaping the future of healthcare and improving patient outcomes worldwide.

**Reference:**

<https://www.researchgate.net/profile/Mustafa-Numan-2/publication/383921585_Harnessing_Artificial_Neural_Networks_in_Neuroscientific_Research_to_Predict_Hospital_Readmissions_and_Enhance_Patient_Care/links/66e13fcff84dd1716ce319e7/Harnessing-Artificial-Neural-Networks-in-Neuroscientific-Research-to-Predict-Hospital-Readmissions-and-Enhance-Patient-Care.pdf>

1. Research paper with title, "An intervention to improve care and reduce costs for high-risk patients with frequent hospital readmissions: A pilot study" by Maria, CRaven, Kelly M Doran, Shannon Kostrowski, Colleen C Gillespie & Brian D Elbel

A small percentage of high-risk patients account for a large proportion of Medicaid spending in the United States, which has become an urgent policy issue. The objective of the study was to pilot a novel patient-centered intervention for high-risk patients with frequent hospital admissions to determine its potential to improve care and reduce costs.

Nineteen patients were enrolled; all were male. 18/19 were substance user and 17/19 were homeless. Patients had a total of 64 inpatient admissions in the 12 months before the intervention, versus 40 in the following 12 months, a 37.5% reduction. Most patients (73.3%) had fewer inpatient admissions in the year after the intervention compared to the prior year.

**Reference:**

<https://link.springer.com/article/10.1186/1472-6963-11-270>

# Dataset and Domain

2. 1. **Data Dictionary**

This dataset is the clinical data for diabetic patients collected from 130 US hospitals over a period of 10 years. The dataset includes patient records with attributes such as age, gender, race, medical history, lab results, medications, and the number of prior hospital visits. Detailed data dictionary is given in the Section 8. 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.)**

1. **Missing values**

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

race

weight

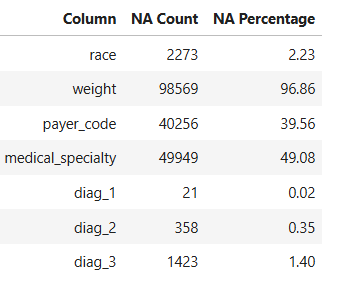
payer\_code

medical\_specialty

diag\_1

diag\_2

diag\_3



Tab:2.1 – 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. 2.2 - 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.

1. **Unwanted columns**

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

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.

1. **Duplicate Rows**

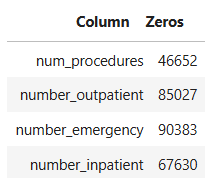
A check for duplicate rows revealed that there were no duplicates in the dataset, ensuring its integrity.

1. **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:2.3 – Count of Zero values

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

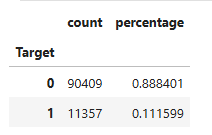
* 1. num\_procedures (Number of procedures (other than lab tests) performed during the encounter)
  2. number\_outpatient (Number of outpatient visits of the patient in the year preceding the encounter)
  3. number\_emergency (Number of emergency visits of the patient in the year preceding the encounter)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

1. **Target variable**

We shall derive the target variable, 'Target' as follows:

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

Target value: 1 -- Readmitted < 30



Tab:2.4 – Count & Percentage of Classes in the Target variable

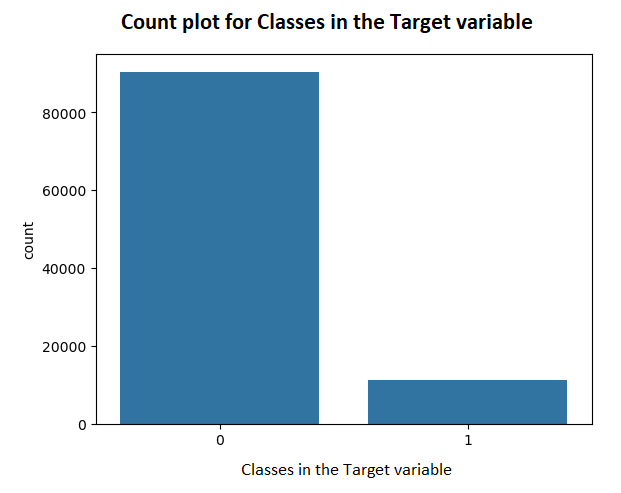


Fig:2.1– Count plot for classes in the Target variable

* 1. **Alternate sources of data that can supplement the core dataset (at least 2-3 columns)**

Lifestyle Factors such as Smoking Status or alcohol consumption: Whether the patient smokes or consumes alcohol can affect diabetes management and readmission risk.

Post-Discharge Support can affect diabetes management and readmission risk.

* 1. **Project Justification - Project Statement, Complexity involved, Project Outcome**

**Project Statement:** This project aims to predict whether a diabetic patient will be readmitted within 30 days post-discharge. Early prediction can help healthcare providers intervene proactively to reduce readmission rates, enhancing patient outcomes while reducing hospital operational costs.

**Complexity involved:** Medium

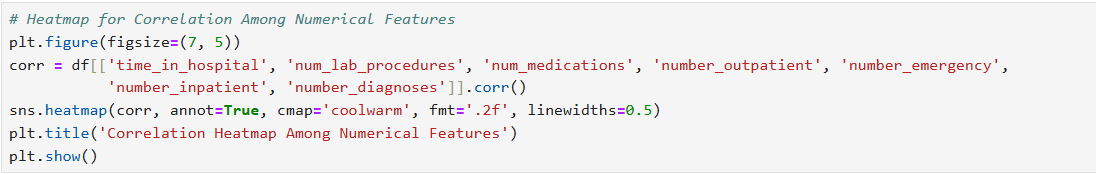
**Project Outcome:** Helps both hospital management and patients to eliminate the factors contributing to early readmission thus enabling patients to get the best treatment.

* 1. **Commercial, Academic or Social value**
* **Commercial Value:**
* Reduces hospital penalties due to excessive readmission rates, as per regulatory guidelines.
* Enables cost-effective resource allocation through targeted interventions for high-risk patients.
* Supports predictive analytics integration into hospital management systems, enhancing operational efficiency.
* **Academic Value:**
* Provides a real-world healthcare dataset for applying and evaluating predictive modeling techniques.
* Contributes to research on improving outcomes for diabetic patients, offering insights into effective disease management strategies.
* Offers a framework for addressing class imbalance and multicollinearity challenges in healthcare datasets.
* **Social Value:**
* Improves patient quality of life by minimizing complications through timely and personalized care interventions.
* Supports evidence-based decision-making for enhanced patient management strategies.
* Reduces the societal burden of diabetes-related complications, fostering healthier communities.
* Promotes equitable healthcare delivery by identifying underserved populations at risk of readmission.

# Data Exploration (EDA)

1. 1. **Relationship between variables**

* **Heatmap for Correlation Among Numerical Features**

****

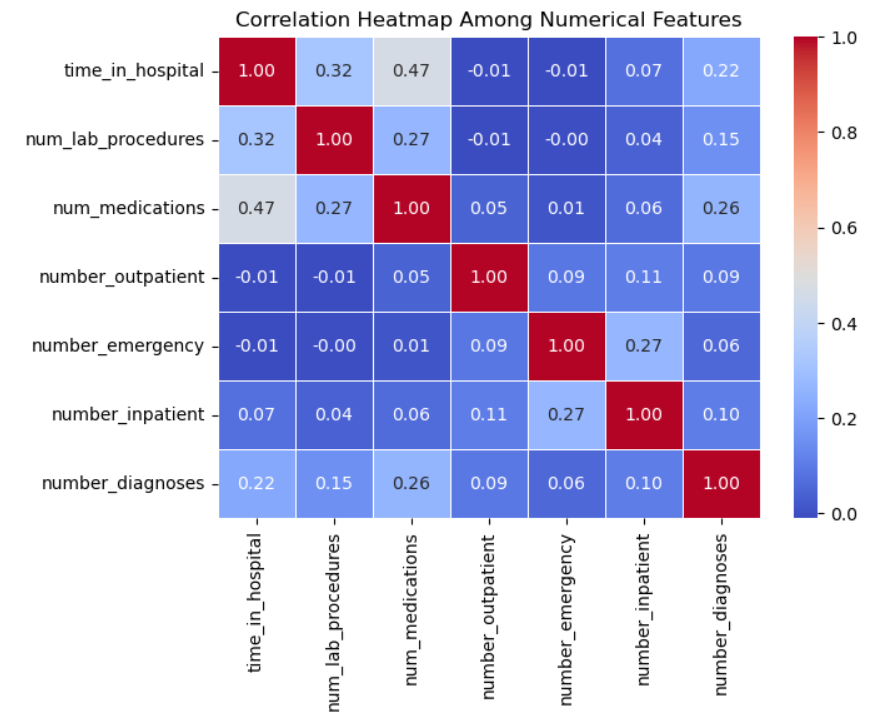
****

Fig 3.1.1: Heatmap for Correlation Among Numerical Features

**Inference:**

The correlation analysis shows a weak relationship among the numerical features, with a maximum value of 0.47 and a minimum of -0.01. This suggests only mild positive correlations and no significant negative correlation. The weak correlations imply that numerical variables like time in hospital, number of lab procedures, and medications are largely independent, and no single feature strongly predicts readmission rates. Therefore, these variables may not be major contributors to readmission outcomes based on the current correlation analysis.

* **Boxplot for Time in Hospital Across Readmission Categories**

****

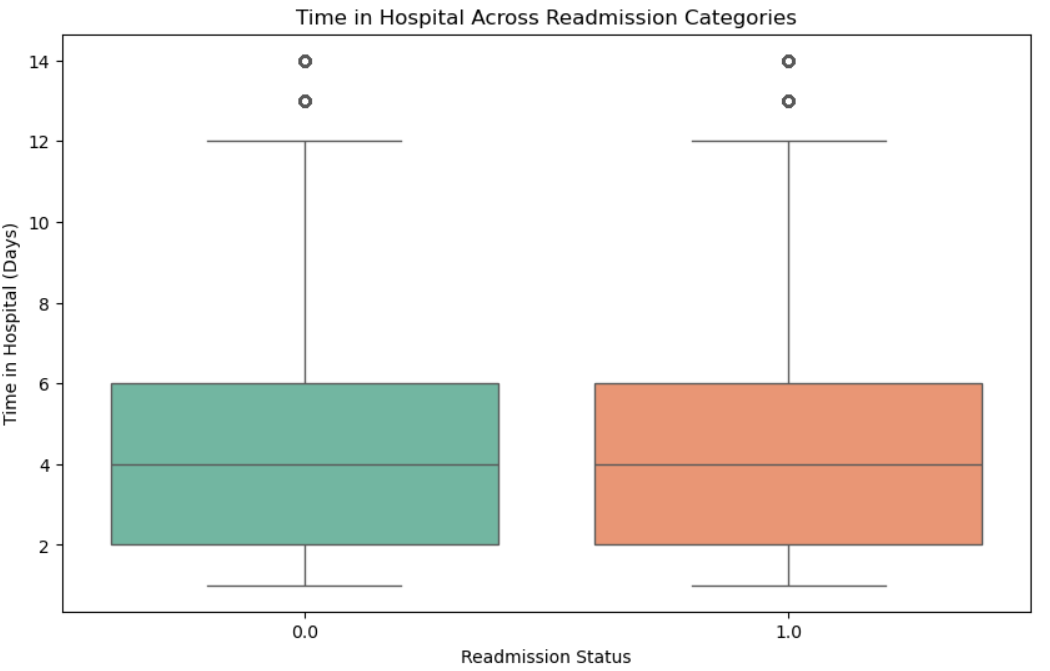
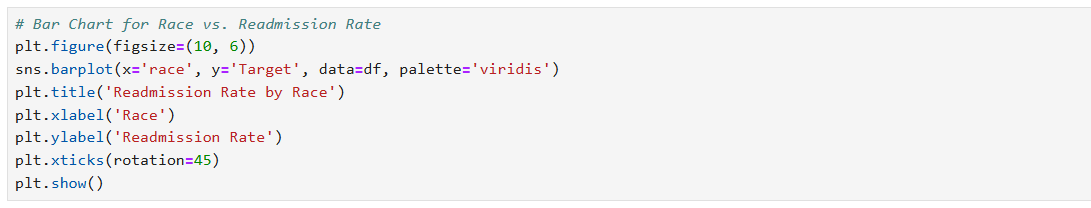
****

Fig 3.1.2: 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**

****

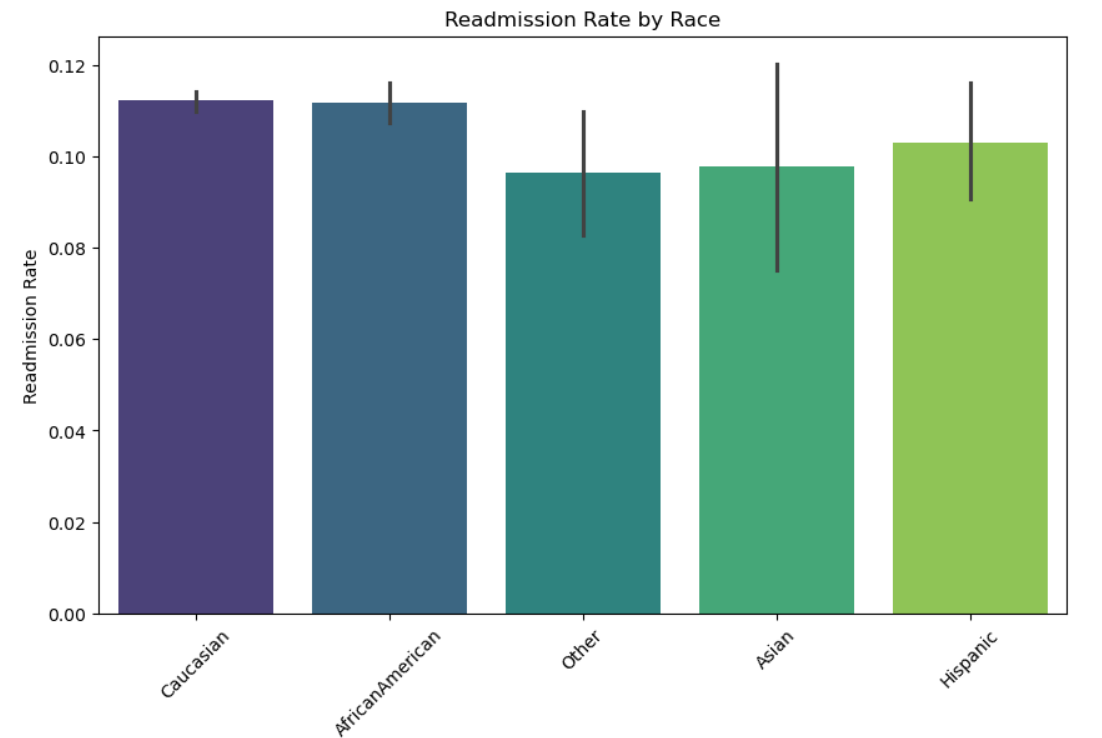
****

Fig 3.1.3: Barplot for Readmission rate by race

**Inference:**

The bar chart for race vs. readmission rate reveals disparities in readmission rates among different racial groups. Certain groups may exhibit higher readmission rates, indicating potential underlying factors such as access to healthcare, socioeconomic status, or genetic predispositions. These disparities highlight the importance of targeting interventions to address the specific needs of higher-risk racial groups. Further investigation into the healthcare access, cultural, and socioeconomic factors for these groups could provide valuable insights for reducing readmission rates.

Based on the readmission rates across racial categories, it is observed that Caucasian, African American, Asian, and Hispanic groups have similar readmission rates, around 0.1, while the "Other" category exhibits a slightly lower rate of 0.09. This suggests that there are no significant disparities in readmission rates across these racial groups. It indicates that factors other than race, such as age, comorbidities, or access to healthcare, might be more influential in determining readmission rates. Further analysis could help identify other key factors contributing to these patterns.

* **Count Plot for Readmission by Gender**

****

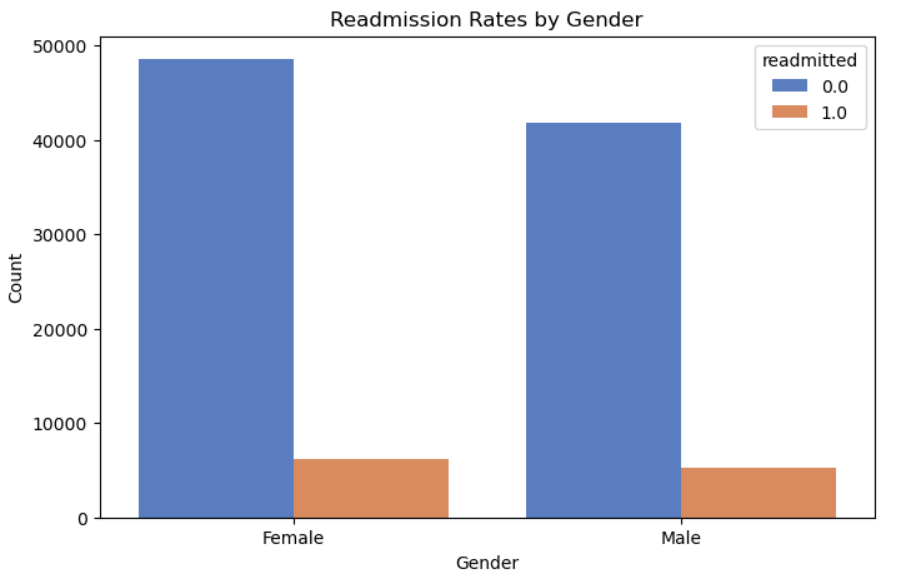
****

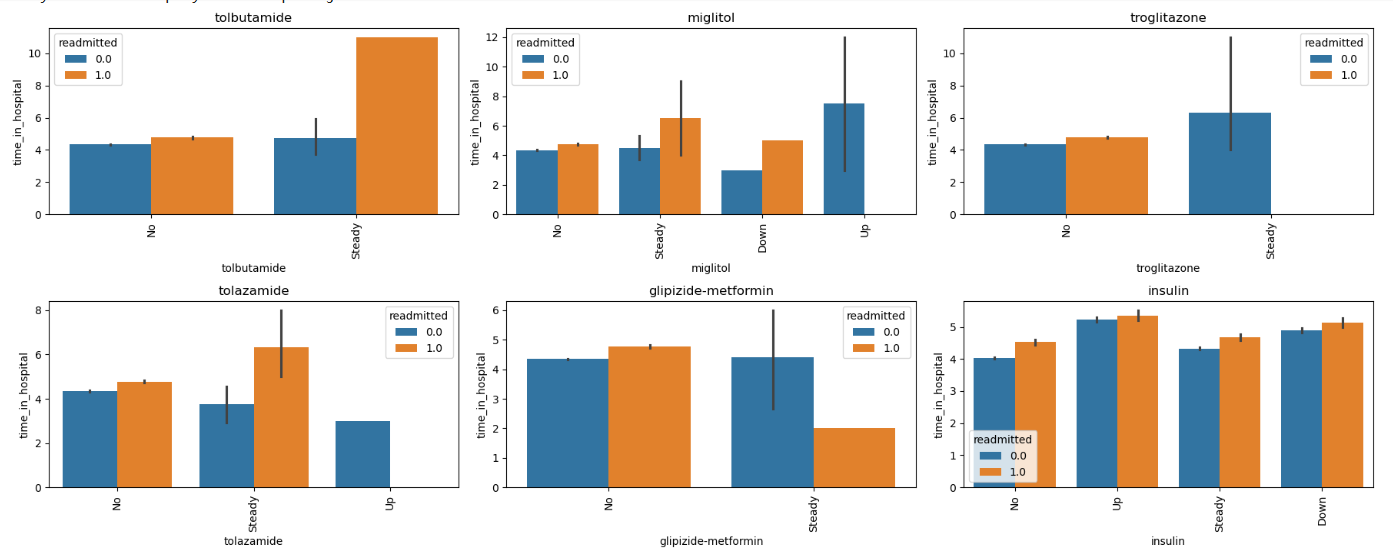
Fig 3.1.4: Count Plot for Readmission by Gender

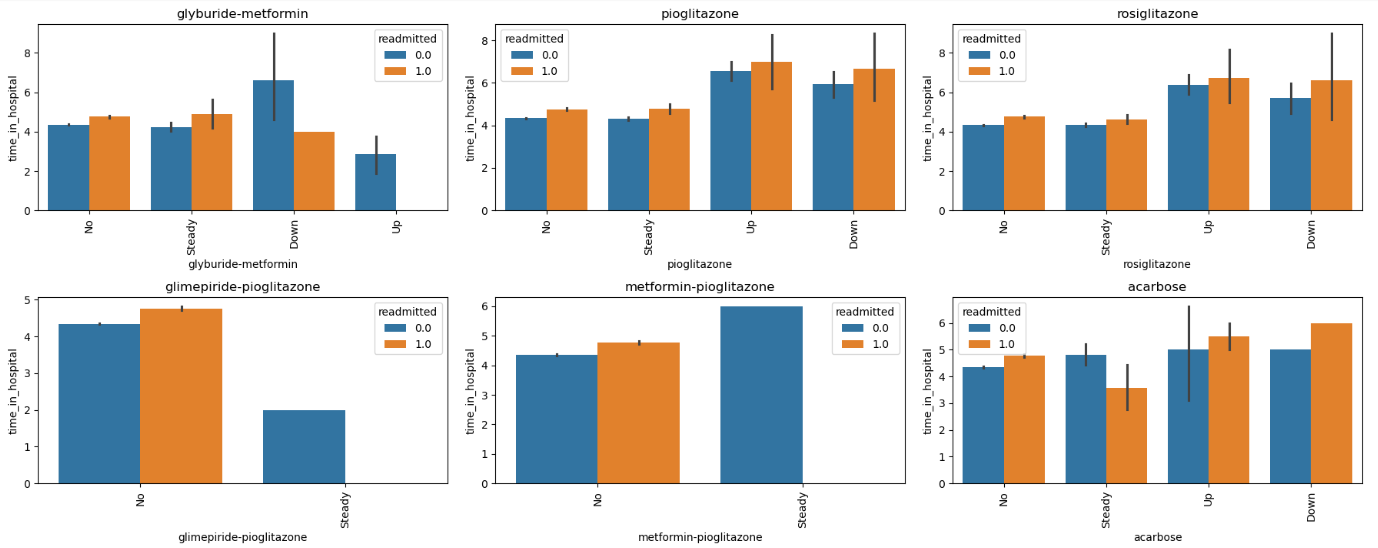
**Inference:**

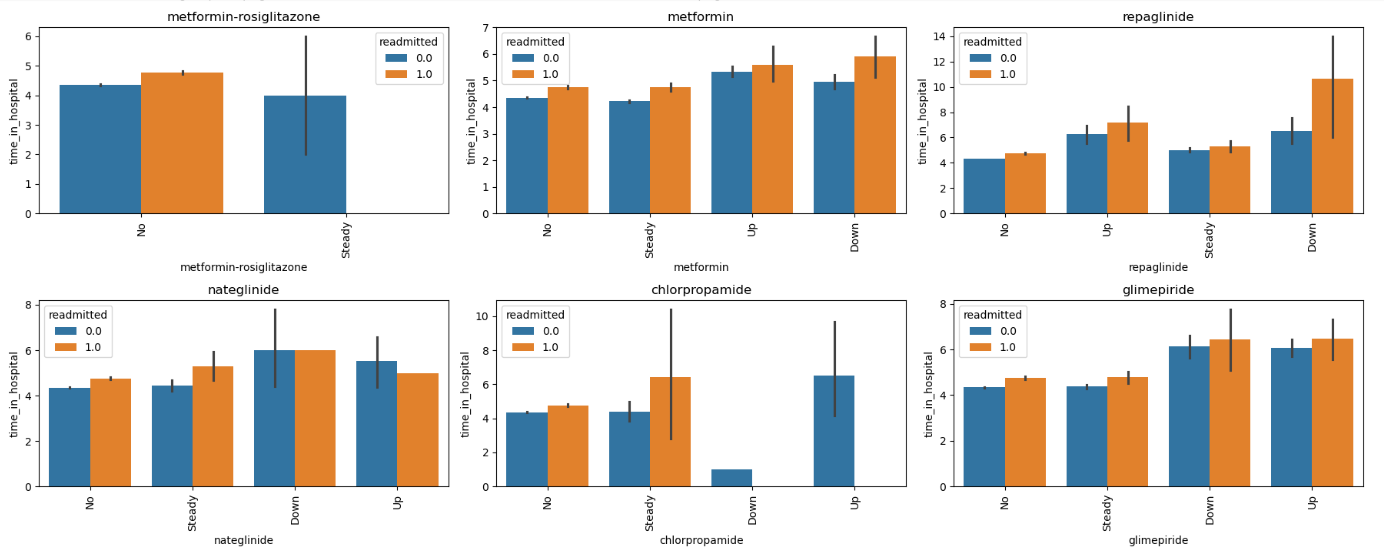
The count plot comparing readmission rates across genders reveals minimal differences between male and female patients. Both genders show nearly identical readmission trends, suggesting that gender is not a significant factor influencing readmission likelihood. This indicates that interventions aimed 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.

* **Relationship Between Drug Dosage and Time Spent in Hospital**

****

****

****

****

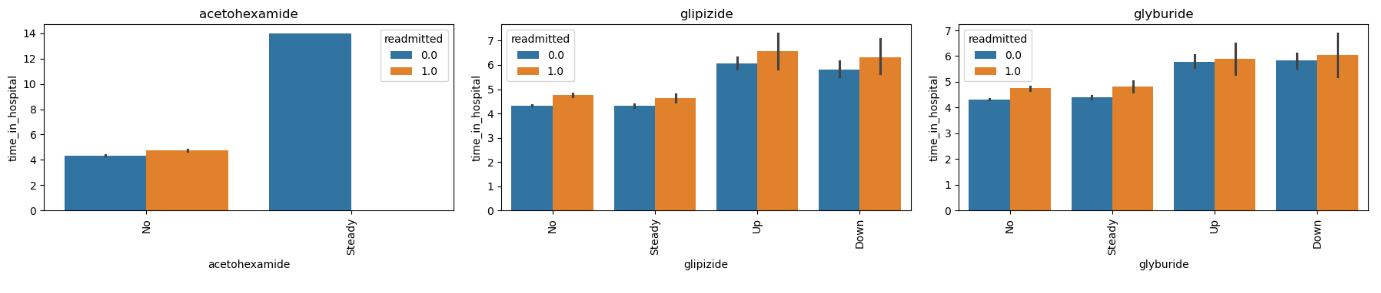
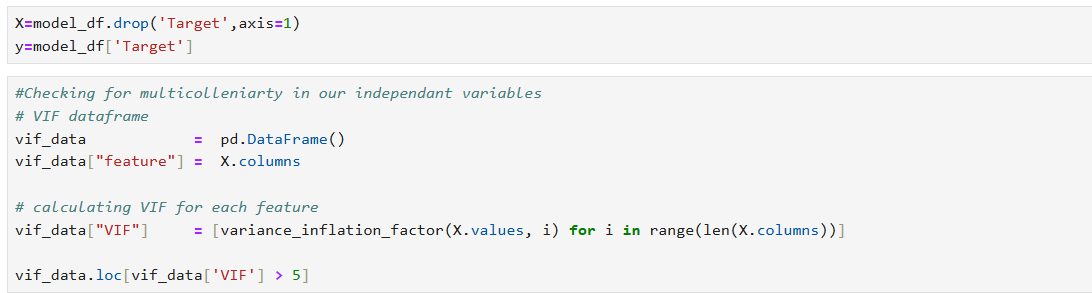
****

Fig 3.1.5: Readmission rate by drug doses

**Inference:**

The barplot highlights the relationship between drug dosage and the average time spent in the hospital. It shows that patients receiving higher dosages tend to have longer hospital stays, suggesting a correlation between dosage levels and the severity of medical conditions. Conversely, lower dosages are associated with shorter hospital durations, reflecting effective treatment for less severe cases. The barplot provides a clear visual comparison of how average hospitalization time varies across different dosage levels. This insight can help refine treatment protocols, optimize drug administration, and improve resource allocation in healthcare settings.

**3.2 Checking for multi-collinearity**

****

****

Tab 3.2: Features and VIF

**Inference:**

High Multicollinearity: Features like miglitol (1042.9), chlorpropamide (746.6), and repaglinide (57.9) show severe multicollinearity.

Moderate Multicollinearity: Features such as A1Cresult (14.7), metformin (9.8), and glimepiride (20.8) indicate moderate multicollinearity.

Acceptable Multicollinearity: Features like num\_lab\_procedures (7.0) and diabetesMed (6.7) show acceptable multicollinearity levels.

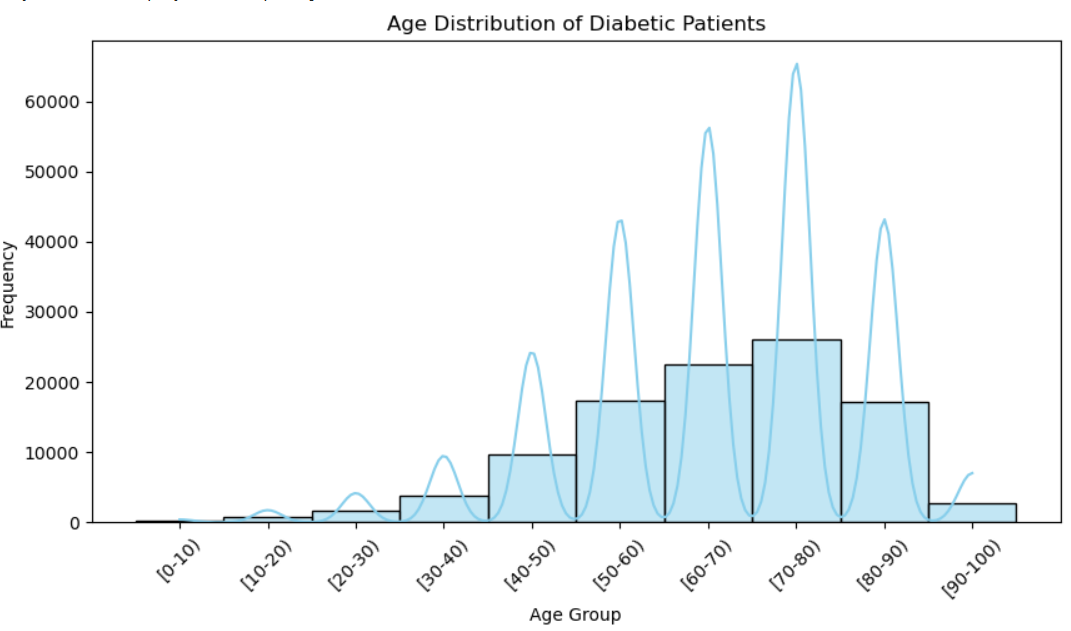
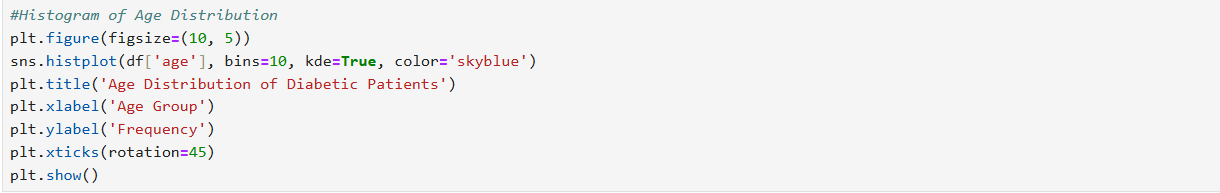
* 1. **Checking for distribution of variables**
* **Histogram of Age Distribution**

Fig 3.3.1: 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.

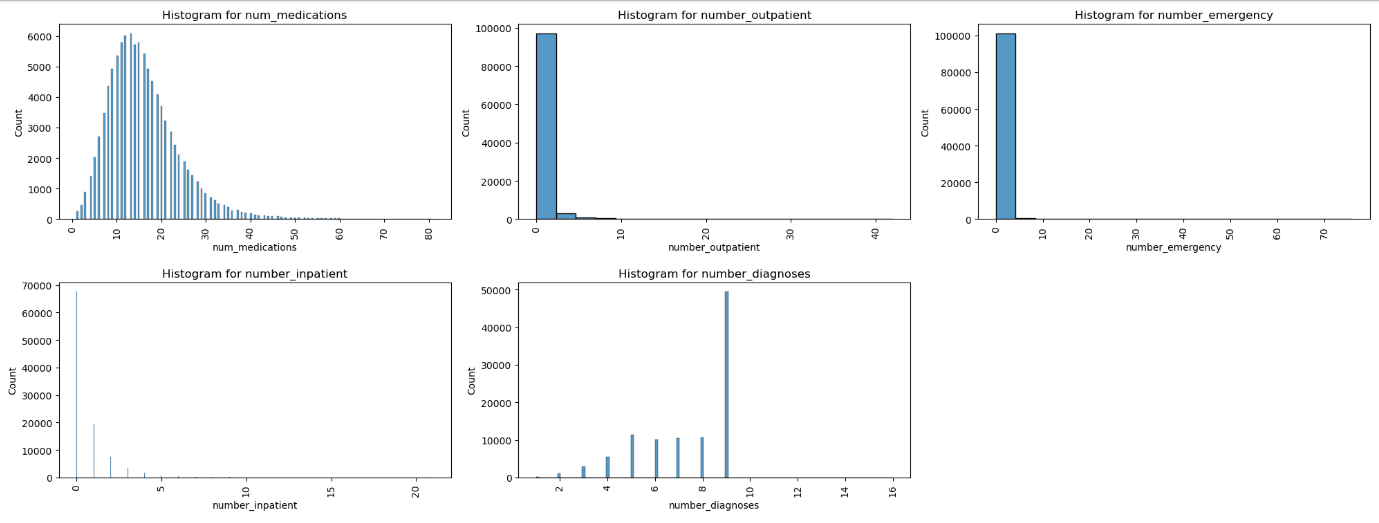
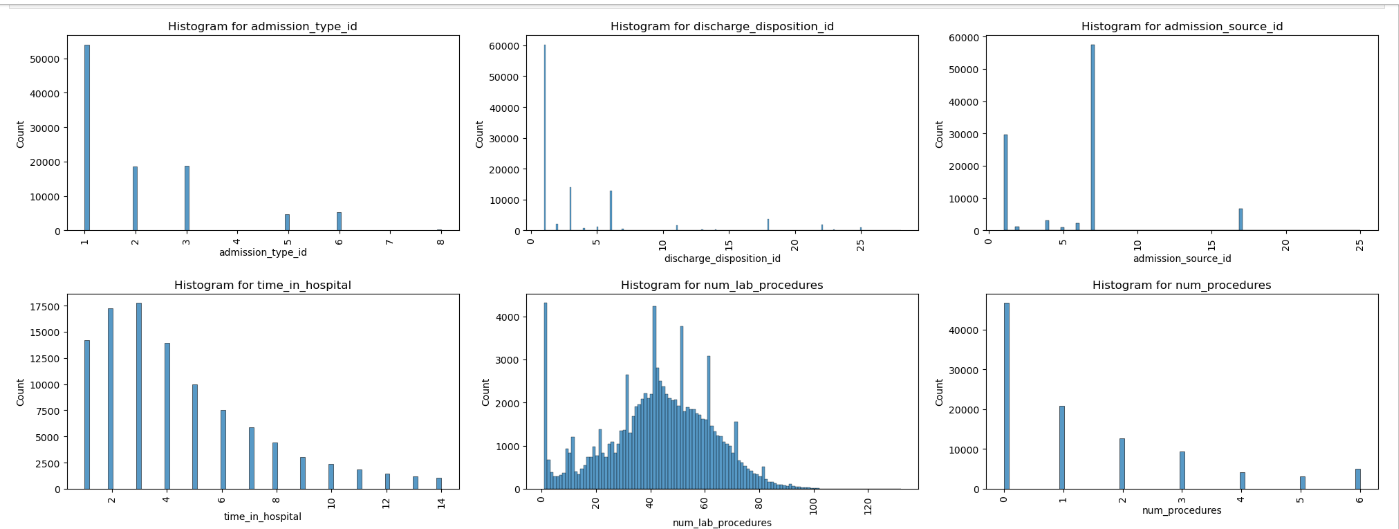
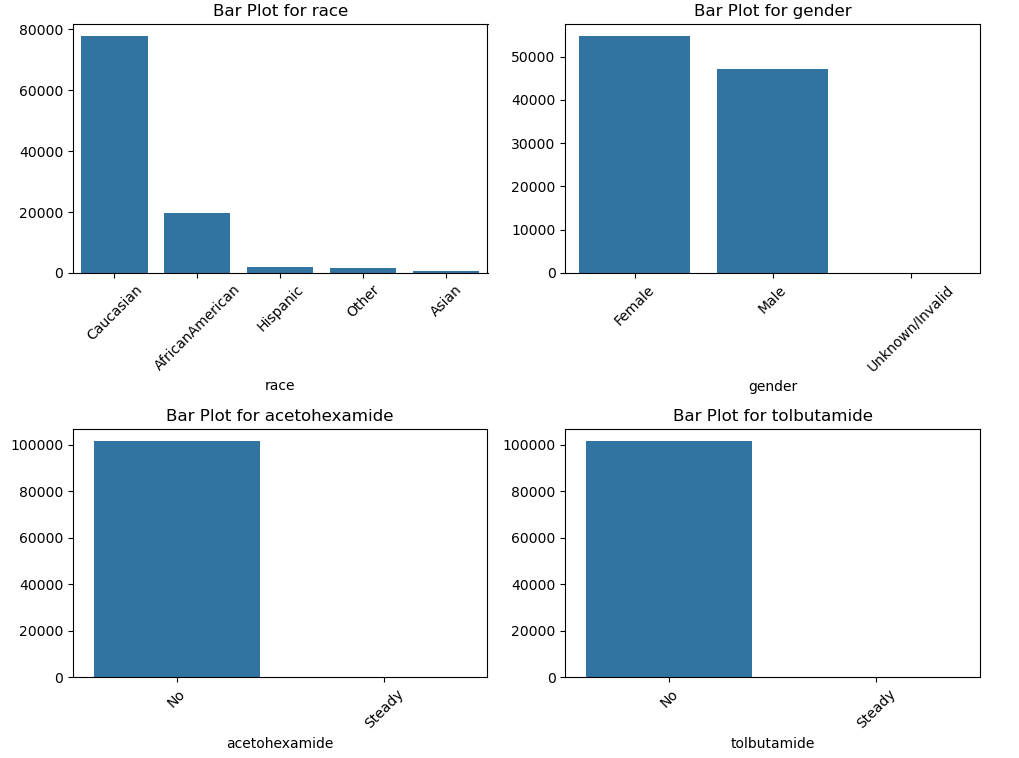
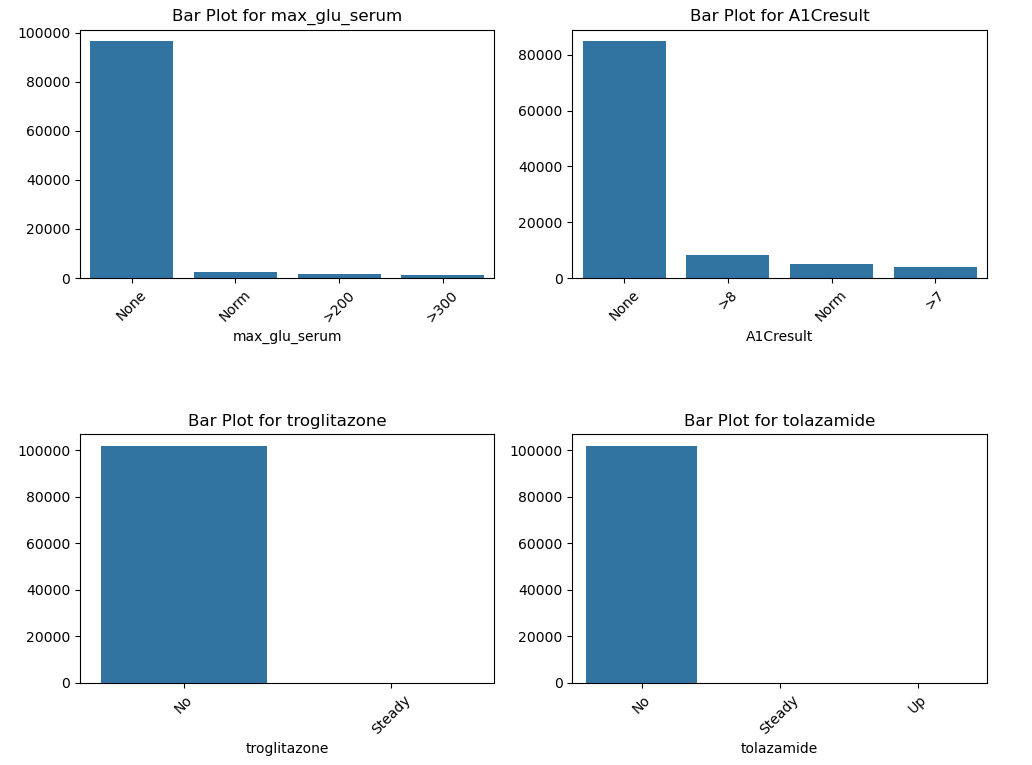
* **For numerical discrete data**

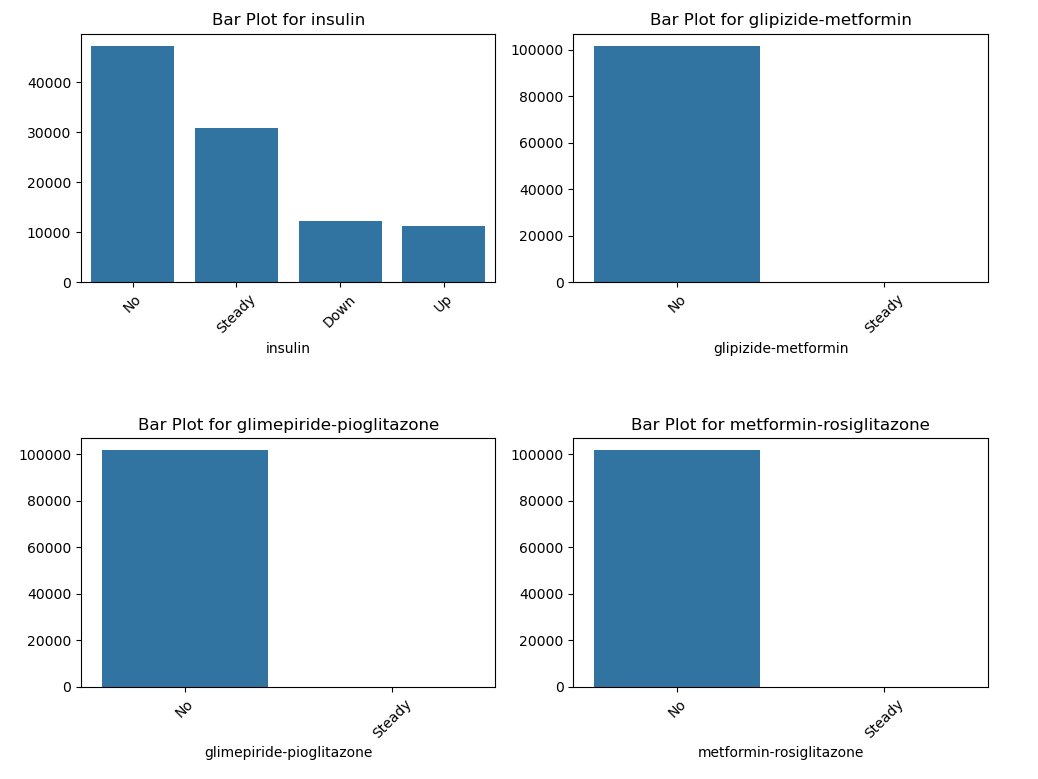
Fig 3.3.2: Histogram of numerical features

* **For categorical variables**

****

****

****

****

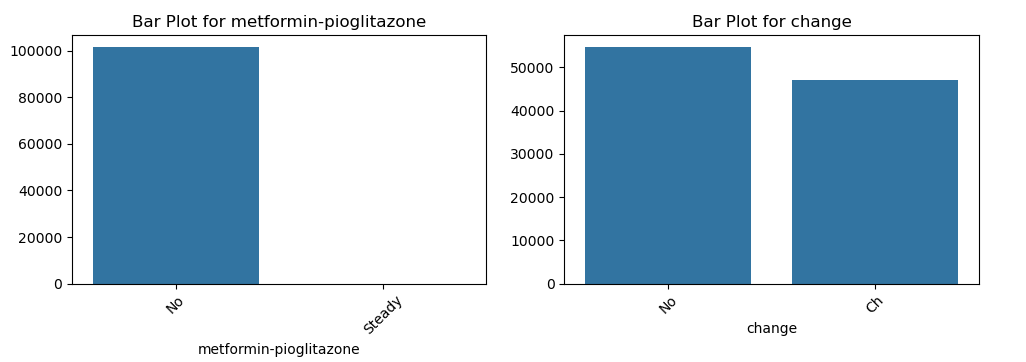
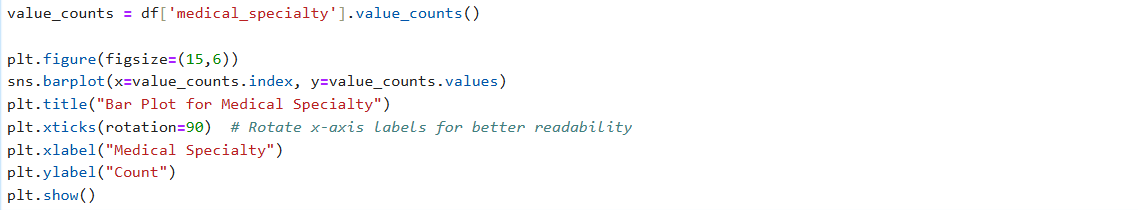
****

Fig3.3.3: Countplot of categorical features

* **Distribution of Medical speciality**

****

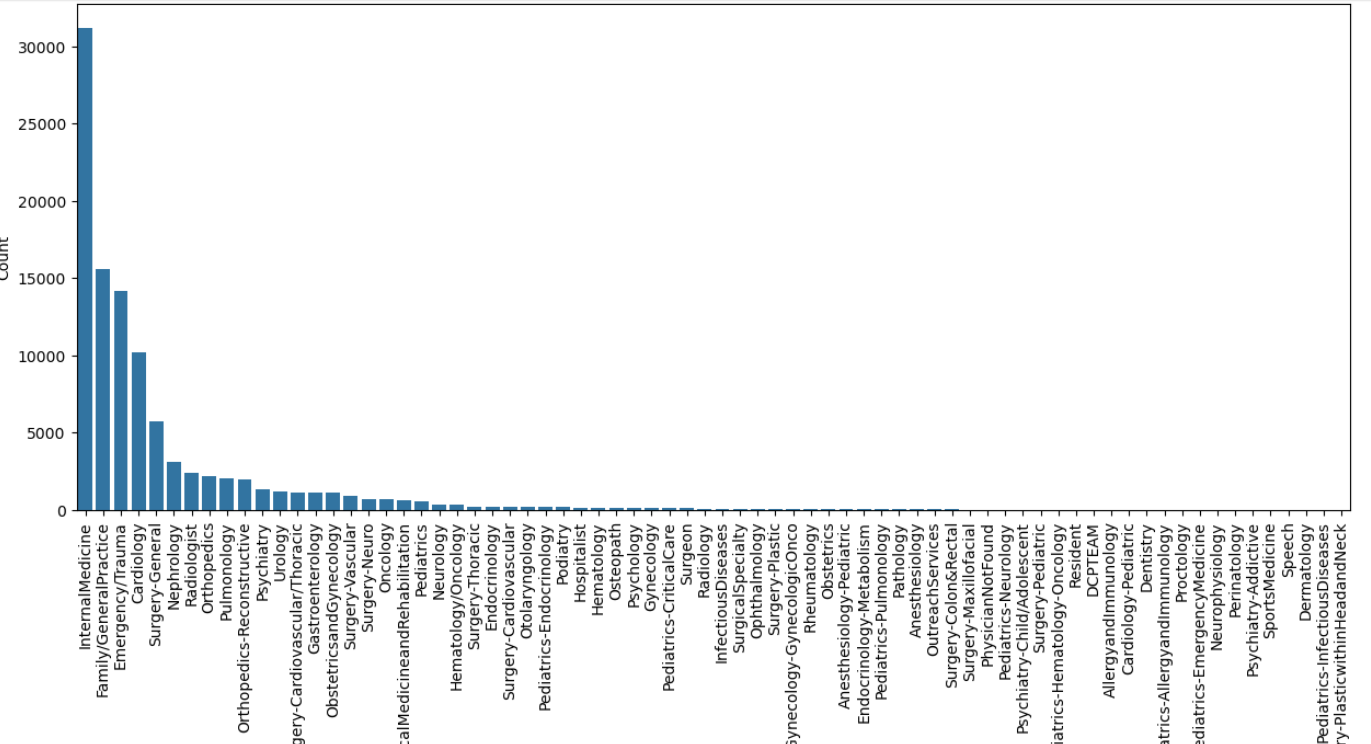
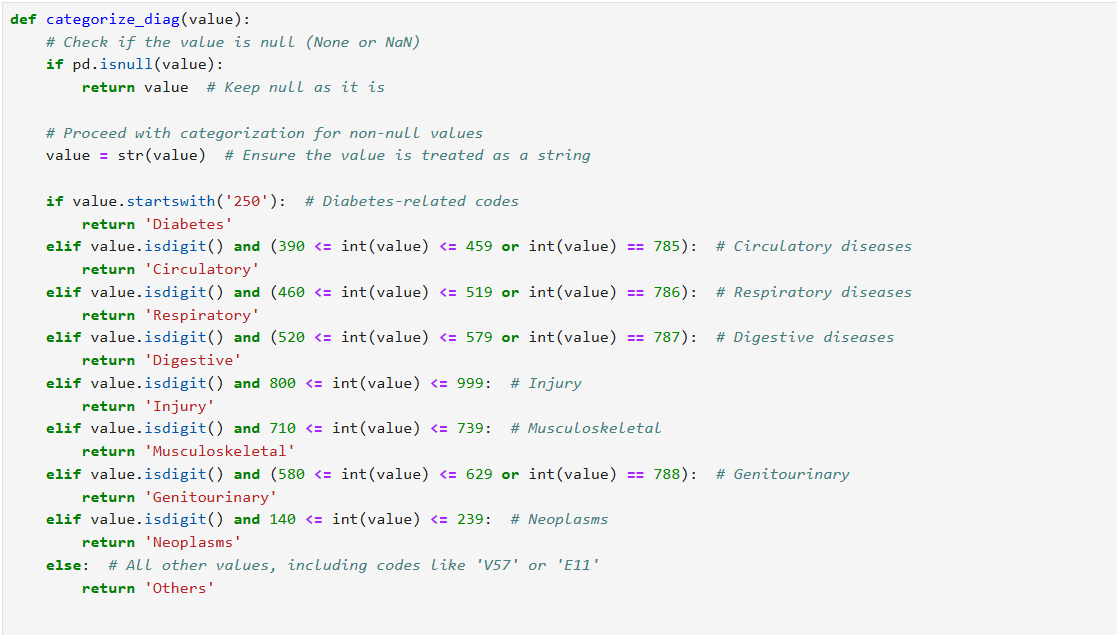
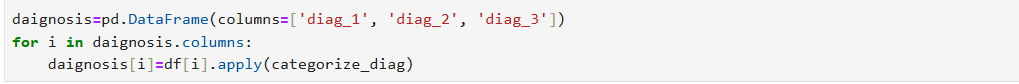
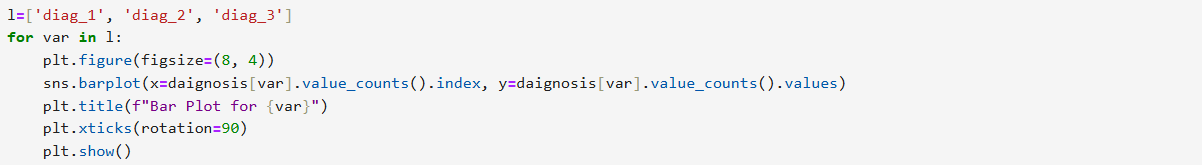
****

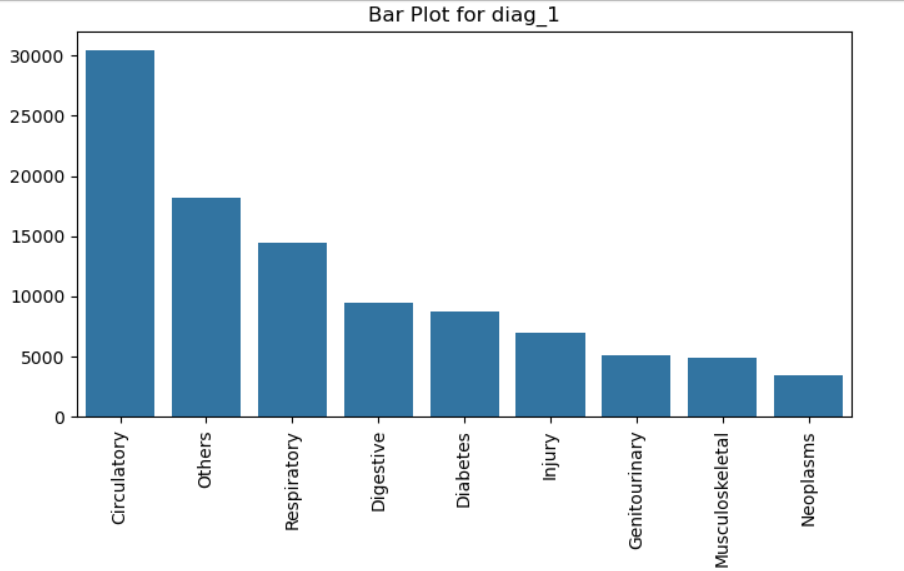
Fig 3.3.4: Distribution of Medical speciality

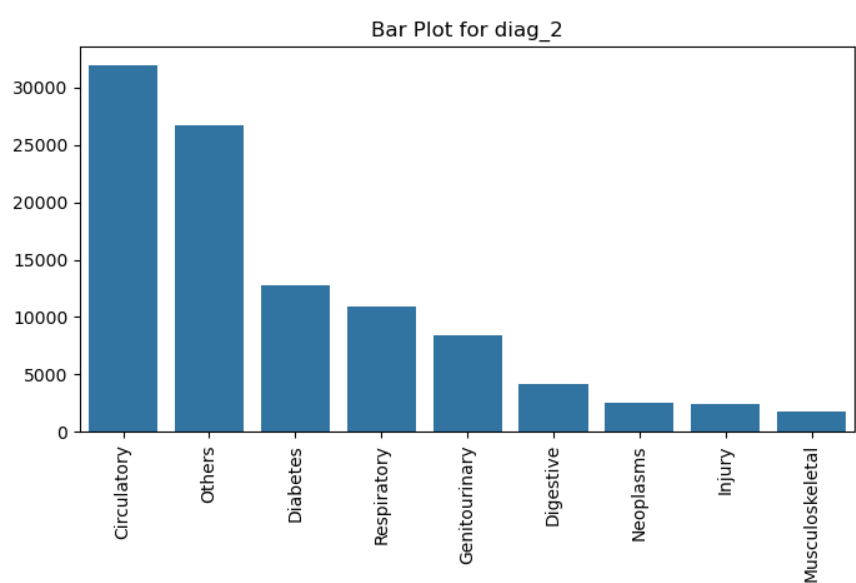
* **Distribution of diag\_1,diag\_2,diag\_3**

****

****

****

****

****

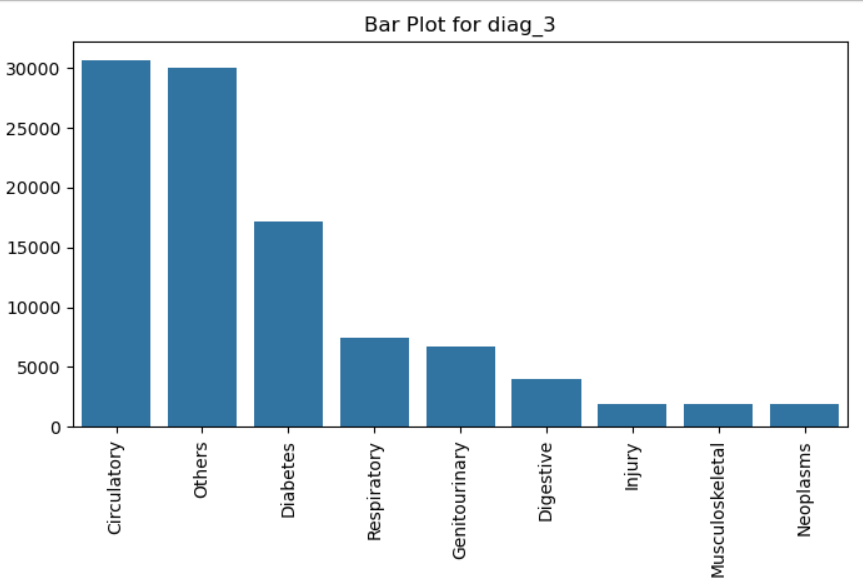
****

Fig3.3.5: Distribution of diag\_1, diag\_2, diag\_3

* 1. **Checking for 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.**

* 1. **Checking for statistical significance of variables**

#### 3.5.1 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. **discharge\_disposition\_id**:
   * A statistically significant difference was observed between the mean values of the two groups.
   * This suggests that the discharge disposition plays a role in predicting readmission.
3. **admission\_source\_id**:
   * A statistically significant difference exists between the two groups.
   * The source of admission has an influence on patient readmission.
4. **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.
5. **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.
6. **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.
7. **num\_medications**:
   * A statistically significant difference was observed.
   * The number of medications prescribed significantly affects the likelihood of readmission.
8. **number\_outpatient**:
   * A statistically significant difference exists between the two groups.
   * The number of outpatient visits is related to readmission rates.
9. **number\_emergency**:
   * A statistically significant difference was found.
   * The frequency of emergency visits is strongly associated with patient readmission.
10. **number\_inpatient**:
    * A statistically significant difference was observed.
    * The number of inpatient visits significantly influences the likelihood of readmission.
11. **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.

#### 3.5.2 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**: 0.1488 (greater than 0.05)
   * **Conclusion**: race and readmitted are independent.
   * **Expected Counts**: No cells with counts < 5.
2. **Variable: gender**
   * **p-value**: 0.3605 (greater than 0.05)
   * **Conclusion**: gender and readmitted are independent.
   * **Expected Counts**: No cells with counts < 5.
3. **Variable: medical\_specialty**
   * **p-value**: 8.14×10−328.14 \times 10^{-32}8.14×10−32 (very small, less than 0.05)
   * **Conclusion**: medical\_specialty and readmitted are dependent.
   * **Expected Counts**: 28.47% of cells with counts < 5 (violates assumption).
4. **Variable: diag\_1**
   * **p-value**: 4.88×10−924.88 \times 10^{-92}4.88×10−92 (very small, less than 0.05)
   * **Conclusion**: diag\_1 and readmitted are dependent.
   * **Expected Counts**: 51.26% of cells with counts < 5 (violates assumption).
5. **Variable: diag\_2**
   * **p-value**: 2.80×10−392.80 \times 10^{-39}2.80×10−39 (very small, less than 0.05)
   * **Conclusion**: diag\_2 and readmitted are dependent.
   * **Expected Counts**: 55.95% of cells with counts < 5 (violates assumption).
6. **Variable: diag\_3**
   * **p-value**: 1.01×10−521.01 \times 10^{-52}1.01×10−52 (very small, less than 0.05)
   * **Conclusion**: diag\_3 and readmitted are dependent.
   * **Expected Counts**: 55.58% of cells with counts < 5 (violates assumption).
7. **Variable: acetohexamide**
   * **p-value**: 1.0 (greater than 0.05)
   * **Conclusion**: acetohexamide and readmitted are independent.
   * **Expected Counts**: 50% of cells with counts < 5 (violates assumption).
8. **Variable: tolbutamide**
   * **p-value**: 0.4799 (greater than 0.05)
   * **Conclusion**: tolbutamide and readmitted are independent.
   * **Expected Counts**: 25% of cells with counts < 5 (violates assumption).
9. **Variable: troglitazone**
   * Results cut off in the output. Based on previous results, it's likely similar issues with expected counts will be present.

**Key Points:**

1. **Assumptions**:
   * The Chi-Square test requires that no more than 20% of cells have expected counts < 5. This is violated for many variables like medical\_specialty, diag\_1, diag\_2, diag\_3, etc.
   * Results for these variables should be treated with caution or supplemented with Fisher’s Exact Test if possible.
2. **Significance**:
   * Variables with a p-value less than 0.05 are dependent on readmitted, such as medical\_specialty, diag\_1, diag\_2, diag\_3.
3. **Next Steps**:
   * Address assumption violations by collapsing categories or using alternative methods.
   * Focus on significant variables while ensuring assumptions are met.
   1. **Checking for class imbalance and its treatment**



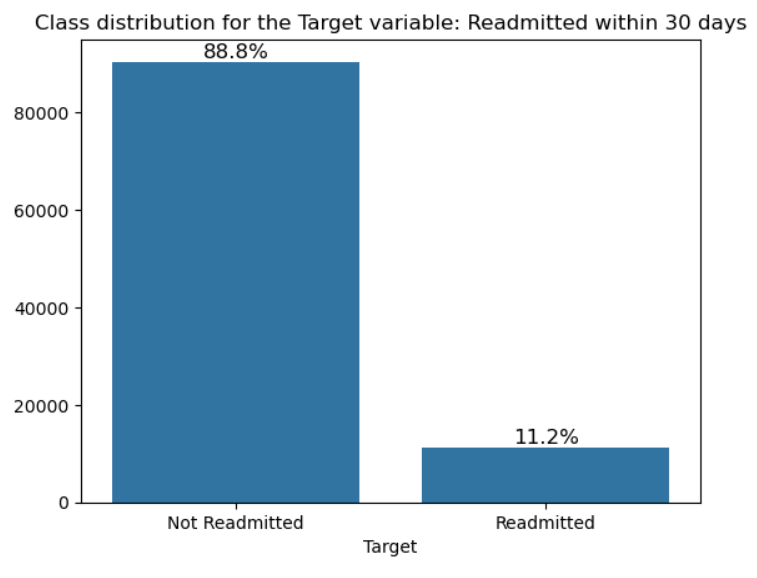


Fig 3.6.1: Barplot readmission class imbalance

* **Observations**

As visible, our data is highly imbalanced. Imbalanced datasets can lead to a bias towards the majority class constituting 88.8% 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.

We prefer Over-sampling by SMOTE method.



# Feature Engineering

**4.1. Transformations required**

* **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’

**4.2. 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.

**4.3. Feature selection**

* **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.

* **Steps to Use RFE:**

1.Choose an Estimator:

RFE requires an estimator (like DecisionTreeClassifier, LogisticRegression, etc.) that can calculate feature importance.

2.Specify Number of Features:

Define the number of features you want to keep using the n\_features\_to\_select parameter.

3.Fit the Model:

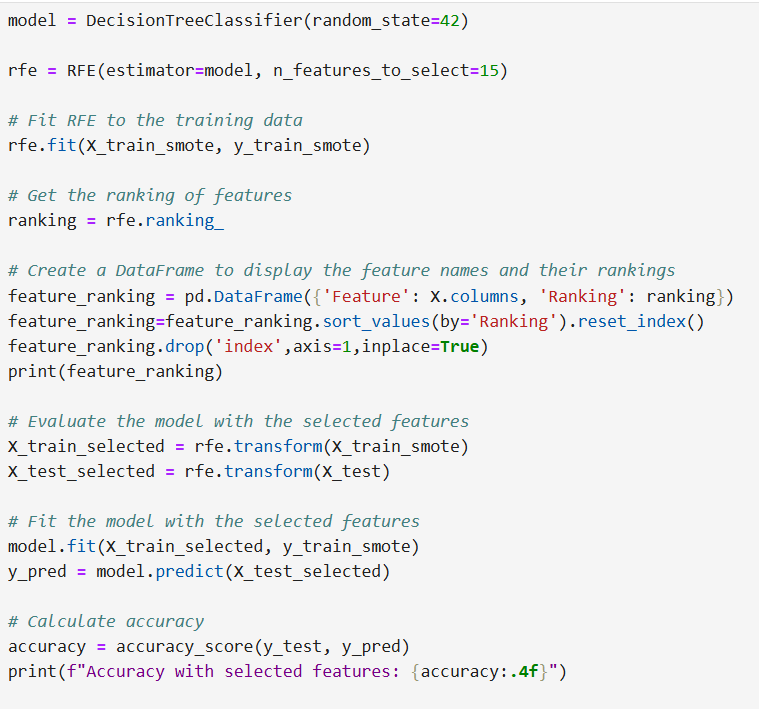
Fit the RFE model to the training data using fit(). It evaluates the features and eliminates the least important ones.

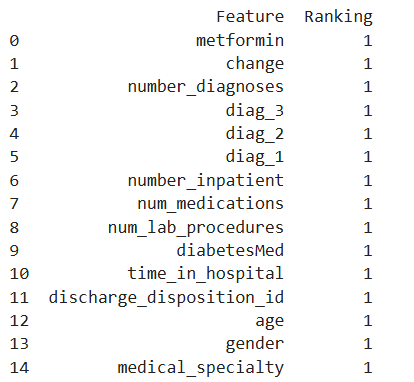
4.Evaluate Feature Selection:

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).

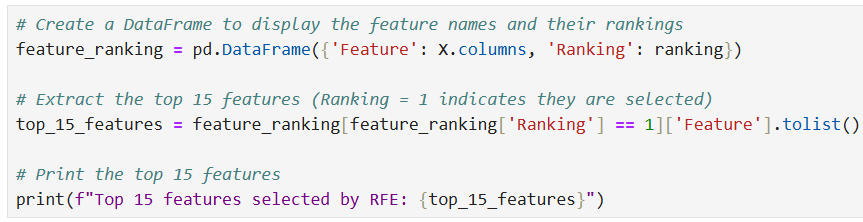
5.Transform the Data:

Use transform() to apply the feature selection to the dataset, keeping only the most important features.





Tab 4.3: Top 15 Important Features



**Top 15 features selected by RFE**

'gender', 'age', 'discharge\_disposition\_id', 'time\_in\_hospital', 'medical\_specialty', 'num\_lab\_procedures', 'num\_medications', 'number\_inpatient', 'diag\_1', 'diag\_2', 'diag\_3', 'number\_diagnoses', 'metformin', 'change', 'diabetesMed'

**Basic Patient Information**

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

**Hospital Stay Details**

* discharge\_disposition\_id: 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.
* 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.

**4.4. 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.

# Assumptions

**Assumptions for Machine Learning Supervised Classification Models**

Machine learning models rely on specific assumptions that can significantly impact their performance. Understanding these assumptions is critical for selecting the right model and preparing the data accordingly. Just as with most things in life, assumptions can directly lead to success or failure. Similarly in machine learning, appreciating the assumed logic behind machine learning techniques will guide you toward applying the best tool for the data.

Ref: <https://www.kdnuggets.com/2021/02/machine-learning-assumptions.html>

Below are the key assumptions for Classification models:

**Logistic Regression**

**Assumption 1** - Binary logistic regression requires the target / dependent variable to be binary.

For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome (such as Success etc..).

We have two classes in the target variable, 0 representing No readmission within 30 days and 1 representing readmission within 30 days.

**Assumption 2** - Only the meaningful variables should be included.

We have ensured that there are no unwanted variables selected for model building.

**Assumption 3** -The predictor variables should not be correlated to each other meaning the model should have little or no multicollinearity.



|  |  |
| --- | --- |
|  | **Inference:**  The following 20 variables are highly collinear having VIF value > 5:  Miglitol  Chlorpropamide  Acarbose  glyburide-meformin  Nateglinide  Repaglinide  max\_glu\_serum  Glimepiride  number\_diagnoses  Rosiglitazone  Age  Pioglitazone  A1Cresult  Glyburide  Glipizide  Metformin  num\_medications  num\_lab\_procedures  diabetesMed  diag\_1 |
|  |  |

Tab 5.1: VIF for Features

|  |  |
| --- | --- |
|  | **Observations:**  We find all the 25 independent variables are non-collinear and thus satisfying the condition of absence of multi-collinearity. |

Tab 5.2: VIF for Features that are non-collinear

**Assumption 4** - The independent variables are linearly related to the log odds.

We need to check the assumption of independent variables are linearly related to the log odds.

One way to checking this is to plot the independent variables in question and look for an S-shaped curve.

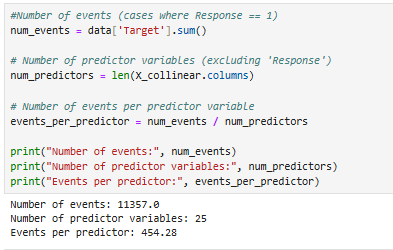
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Fig 5.1: Log odds linear plot for numerical variables

**Observations:**

Log odd linear plot of some of the variables resemble S-shaped curve.

**Assumption 5** - Logistic regression requires quite a large number of observations.



**Observations:**

* We calculate the number of events by summing the 'Target' column, which represents the cases where the outcome of interest occurs.
* We calculate the number of predictor variables by counting the number of columns in the DataFrame and excluding the outcome variable.
* We divide the number of events by the number of predictor variables to get the events per predictor.
* We can then compare the calculated events per predictor with the recommended guideline of 10-20. If the ratio is below this guideline, it may indicate a potential violation of the assumption of a sufficiently large sample size.
* With 11357 events and 25 predictor variables, the calculated number of events per predictor is approximately 454.28. This exceeds the commonly recommended guideline of having at least 10-20 events per predictor variable.

**Inference:** The dataset appears to meet the assumption of having a sufficiently large sample size for logistic regression. Having a high number of events per predictor variable suggests that there should be adequate statistical power and precision in estimating the model parameters, enhancing the reliability of the logistic regression analysis. Therefore, the dataset likely provides a robust basis for fitting a logistic regression model and conducting statistical inference.

**Assumptions for Decision Tree:**

* No Assumptions on Data Distribution: Decision trees can handle both linear and non-linear relationships.
* Handling of Multicollinearity: While less sensitive to collinearity, redundant features may impact performance.

**Random Forest:**

* No Assumption of Data Distribution: As a non-parametric model, it handles skewed or multi-modal data.
* Multicollinearity: Robust to collinearity, but excessive multicollinearity may reduce performance.

**Support Vector Machine (SVM):**

* Linear Separability: The data should ideally be linearly separable (relaxed with kernel functions).
* Feature Scaling: SVM is sensitive to feature scales, requiring normalization or standardization.
* No Multicollinearity: High collinearity can affect margin placement.
* Noise Sensitivity: Outliers can heavily influence SVM's decision boundary.

**Bagging & Boosting (e.g., Random Forest, XGBoost):**

* No Assumption of Data Distribution: These models are flexible, handling diverse data distributions.
* Outlier Sensitivity: Bagging (e.g., Random Forest) is more robust to outliers than boosting methods.
* Multicollinearity: While bagging models like Random Forest handle multicollinearity well, boosting models can struggle with it.

**Conclusion**

Understanding model assumptions helps ensure better performance and more reliable results. For regression models, assumptions like linearity, independence, and homoscedasticity are crucial. Classification models like Decision Trees and Random Forests are more flexible but still benefit from managing multicollinearity and scaling. SVM requires feature scaling and can be sensitive to noise and outliers.

For more details on these assumptions and their implications, [KDnuggets' article](https://www.kdnuggets.com/2021/02/machine-learning-assumptions.html).

# Building and evaluation of Base model

This Python function compare\_models is designed to evaluate and compare the performance of multiple classification models using a specific evaluation metric (in this case, recall) via cross-validation. It leverages several common machine learning algorithms and visualizes their performance using a box plot. The goal of the function is to help assess how different models perform on a given dataset (X and Y) with the specified number of cross-validation splits (n\_splits) and random seed (random\_state).



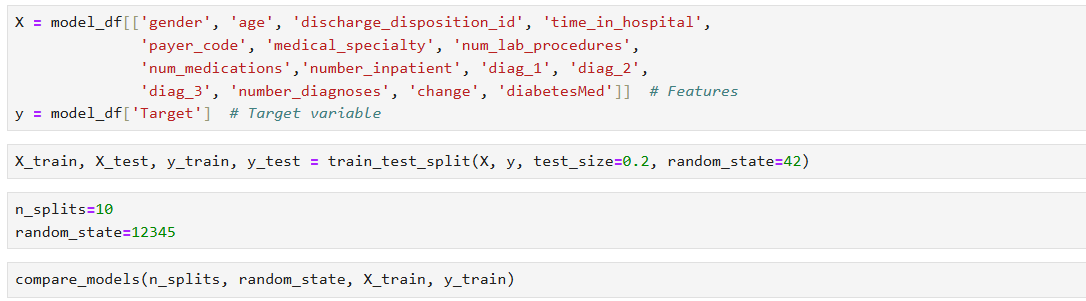
* **Steps the Code is Performing**

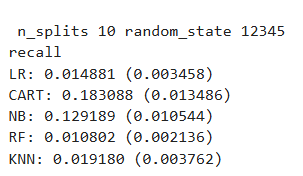
1. **Importing Required Libraries**:
   * The function begins by importing necessary libraries:
     + matplotlib for plotting the results.
     + StratifiedKFold from sklearn.model\_selection to perform stratified cross-validation.
     + cross\_val\_score from sklearn.model\_selection to evaluate models using cross-validation.
     + A selection of classification models including Logistic Regression, Decision Tree, Naive Bayes, Random Forest, K-Nearest Neighbors, and XGBoost.
2. **Preparing the List of Models**:
   * Several machine learning models are instantiated and stored in a list:
     + **Logistic Regression (LR)** with max\_iter set to 20000 to ensure convergence.
     + **Decision Tree Classifier (CART)**.
     + **Gaussian Naive Bayes (NB)**.
     + **Random Forest Classifier (RF)**.
     + **K-Nearest Neighbors (KNN)**.
   * These models are defined with short names (e.g., 'LR', 'CART') for easy identification in the comparison.
3. **Selecting the Evaluation Metric**:
   * The code defines a list of evaluation metrics (scores\_req), and in this case, **recall** is selected to evaluate model performance.
   * **Recall** is a suitable metric for imbalanced datasets, focusing on the ability of a model to identify all relevant instances of a class.
4. **Cross-validation Setup**:
   * The function uses Stratified K-Fold cross-validation (StratifiedKFold), which ensures that each fold has a proportional distribution of class labels. This is useful for classification tasks where class distribution may be imbalanced.
   * The number of splits for cross-validation is fixed at 10 (n\_splits=10), and shuffling is enabled to randomize the data split.
5. **Evaluating Models**:
   * For each model in the list, the function performs cross-validation:
     + It uses cross\_val\_score to compute the recall score for each model over 10 folds.
     + The mean and standard deviation of the recall scores for each model are computed and printed for comparison.
6. **Visualizing Results**:
   * After collecting the results for each model, the function generates a box plot using matplotlib to visually compare the recall performance of the models.
   * The box plot provides an overview of the variability and distribution of the recall scores for each model.
   * The plot title is dynamically generated based on the evaluation metric (e.g., "Algorithm Comparison using recall").
7. **Output**:
   * For each model, the mean and standard deviation of recall scores are printed in the format:  
     Model Name: Mean Recall Score (Standard Deviation)
   * A box plot is displayed showing the recall performance of each model, allowing a visual comparison.

* **Model Comparison for Baseline Classification (Without treating data imbalance)**

The **compare\_models** function is used to evaluate baseline classification models on a dataset related to diabetic patient care. The target variable predicts a specific outcome, and the features include demographic, clinical, and treatment-related variables. The dataset is split into training and testing subsets, with the models evaluated using stratified 10-fold cross-validation.

The focus of the evaluation is on the **recall** metric, which is particularly important in scenarios where minimizing false negatives is critical, such as in healthcare applications. For this baseline analysis, no oversampling techniques like SMOTE were applied, allowing us to assess the models' raw performance on the imbalanced dataset.





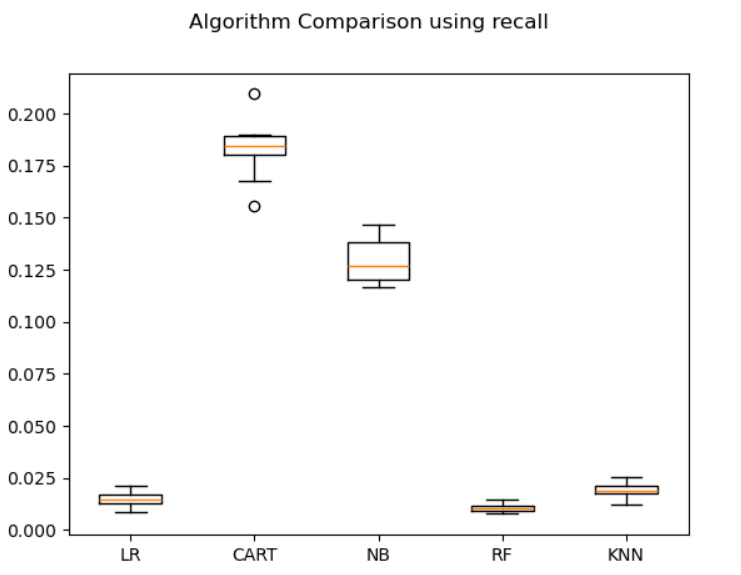


Fig 6.1 : Comparison of Model Performance without SMOTE using Recall

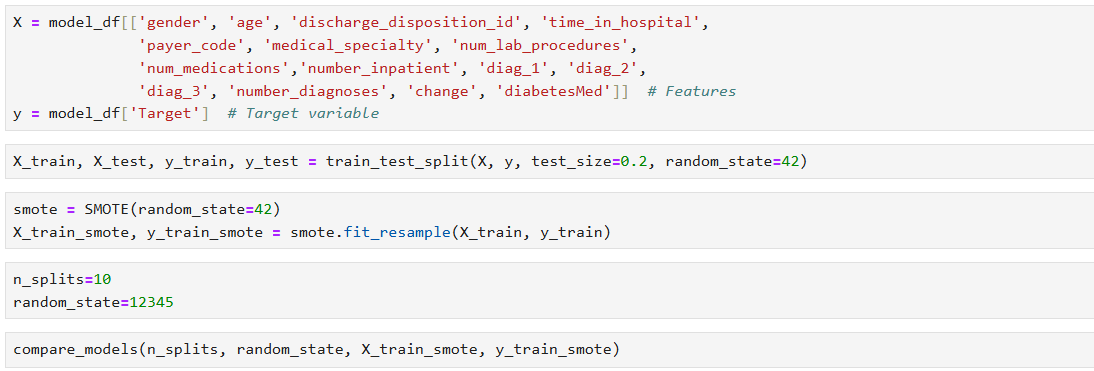
**Inference**

Key insights from the recall scores are as follows:

* **Decision Tree Classifier (CART)** achieved the highest recall (0.183), indicating it captures the most positive cases among the models.
* **Naive Bayes (NB)** performed moderately well, with a recall of 0.129.
* **Logistic Regression (LR)** and **Random Forest (RF)** had low recall values of 0.014 and 0.010, respectively, indicating they struggled with the imbalanced nature of the dataset.
* **K-Nearest Neighbors (KNN)** had a slightly better recall than LR and RF but still performed poorly overall, with a recall of 0.019.

The results obtained so far suggest that **CART** is the most suitable baseline model in terms of recall. However, all models exhibit limitations in handling class imbalance effectively. This indicates a need for advanced techniques, such as oversampling with SMOTE, to enhance model performance further.

* **Model Comparison After Applying SMOTE**



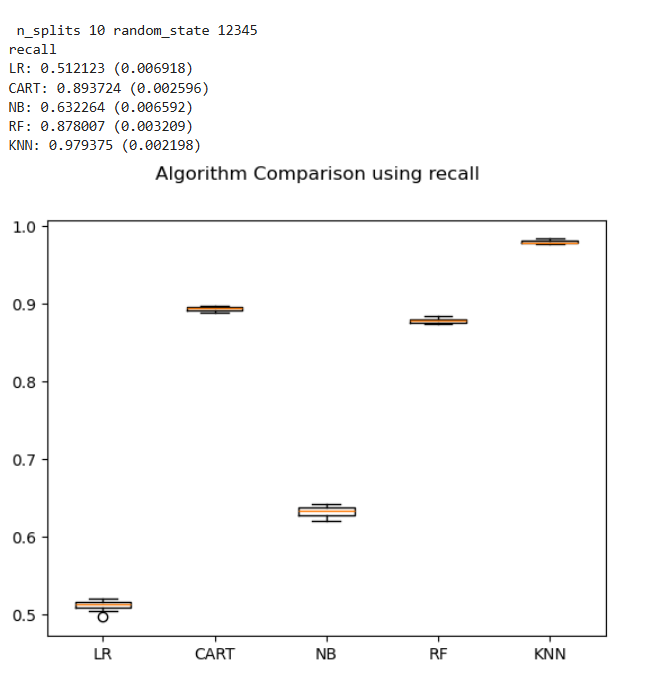


Fig 6.2: Comparison of Model Performance with SMOTE using Recall

**Inference**

After applying SMOTE to address class imbalance, the recall scores of all models showed significant improvement, demonstrating the effectiveness of resampling in boosting model sensitivity to the minority class. Below is the updated analysis and a comparison with the results from the baseline models (without SMOTE):

1. **Performance Analysis (With SMOTE):**
   * **Logistic Regression (LR):** Recall improved significantly to **0.512** from the previous 0.014, showing better sensitivity to positive cases after SMOTE.
   * **Decision Tree Classifier (CART):** Recall surged to **0.893**, retaining its position as one of the top-performing models and demonstrating high effectiveness.
   * **Naive Bayes (NB):** Recall improved to **0.633**, a substantial increase from the earlier 0.129, indicating its ability to leverage resampled data effectively.
   * **Random Forest (RF):** Recall increased to **0.877**, a significant jump from 0.010, making it a strong competitor post-resampling.
   * **K-Nearest Neighbors (KNN):** Recall reached the highest value of **0.978**, showcasing the model's strong performance after addressing class imbalance.
2. **Comparison With Baseline (Without SMOTE):**
   * **General Improvement:** All models demonstrated dramatic improvements in recall, underscoring the importance of addressing class imbalance.
   * **KNN's Dramatic Improvement:** KNN's recall soared from 0.019 to **0.978**, showing the most notable gain after SMOTE.
   * **CART and RF:** Both CART (0.183 → 0.893) and RF (0.010 → 0.877) showed excellent adaptability to resampled data, making them reliable choices.
   * **Naive Bayes:** Recall improved moderately but still lags behind CART, RF, and KNN, highlighting its relatively weaker performance despite resampling.
   * **Logistic Regression:** While its recall increased significantly, it still lags behind other models, suggesting further optimization is needed.
3. **Insights:**
   * SMOTE was highly effective in improving recall for all models, with **KNN** emerging as the top performer.
   * Decision Tree (CART) and Random Forest also delivered exceptional recall scores, making them competitive options.
   * Logistic Regression, despite improvement, requires additional tuning or feature engineering to compete with more complex models.

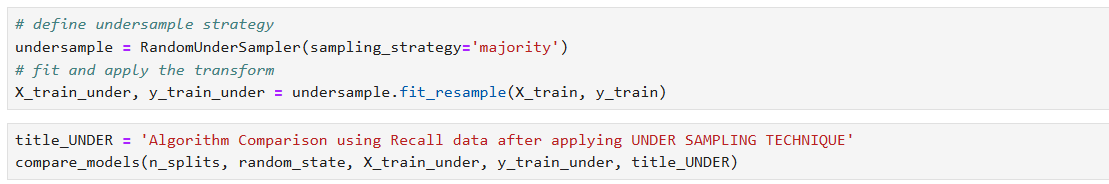
* **Model Comparison After Applying Random Undersampling**

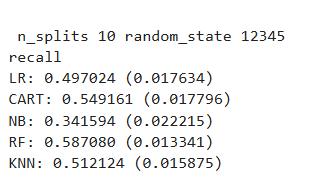
**Introduction**

Imbalanced datasets often lead to biased models favoring the majority class, making it challenging to achieve reliable predictions. Random under sampling offers a solution by reducing the size of the majority class through random removal of instances, creating a balanced dataset for training. This process can be repeated until the desired class distribution is achieved, such as an equal representation of all classes.

While effective in addressing class imbalance, this approach has limitations. Randomly removing instances may lead to the loss of critical information, potentially weakening the model's ability to form a robust decision boundary. It is particularly suitable for datasets where the minority class has sufficient examples to train a meaningful model despite these constraints.

For more details on random under sampling and other techniques for imbalanced classification, refer to the article: [Random Oversampling and Undersampling for Imbalanced Classification](https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/).

****

****

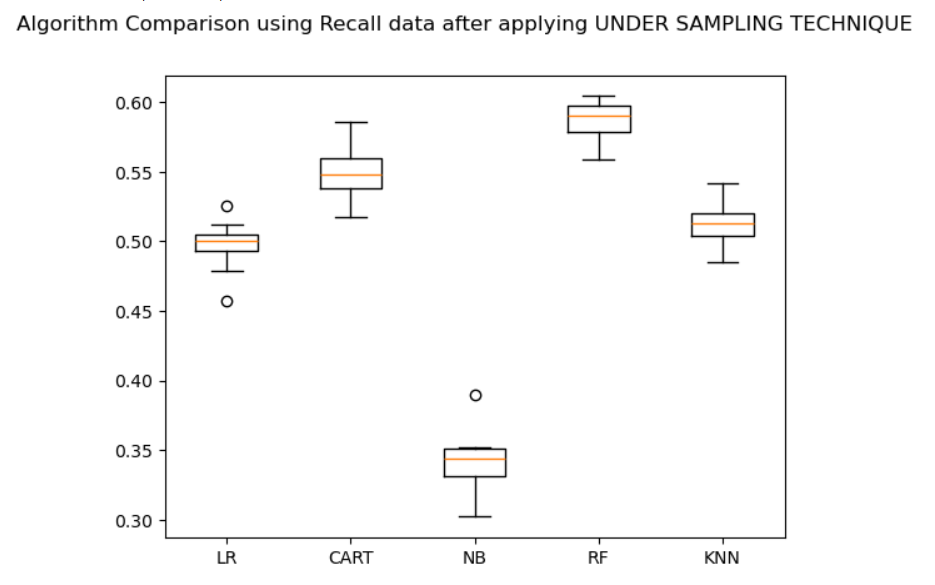
****

Fig 6.3: Comparison of Model Performance with Random Undersampling

**Inference:**

After applying random under sampling and evaluating models using 10-fold cross-validation with random\_state=12345, the recall scores indicate the following insights:

1. **Random Forest (RF)** achieved the highest recall score of **0.5871**, demonstrating its superior ability to correctly identify positive instances despite the under sampled dataset.
2. **CART (Classification and Regression Tree)** followed with a recall score of **0.5492**, showing a competitive performance.
3. **K-Nearest Neighbors (KNN)** achieved a recall score of **0.5121**, indicating moderate performance.
4. **Logistic Regression (LR)** scored **0.4970**, showing slightly lower performance compared to KNN.
5. **Naive Bayes (NB)** had the lowest recall score of **0.3416**, suggesting challenges in handling the under sampled data effectively.

Overall, Random Forest and CART stand out as the most effective models for maximizing recall in this under sampled dataset, making them ideal candidates for scenarios where correctly identifying the minority class is critical.

**Conclusion**

**Recall has increased to 89% when we used SMOTE technique on the data; 55% when we used random under-sampling from 18% without treating the imbalance in the data.**

SMOTE dramatically enhanced the recall of all models, with KNN achieving the highest recall (0.978). CART and RF also performed exceptionally well. These results highlight the importance of resampling techniques in addressing class imbalance.

We observe that Recall is 89% in ten-fold cross-validation and it is good. Moreover, decision trees remain a popular choice in many applications due to their simplicity, interpretability, and versatility.

**We choose Decision Tree (CART) model as our base model.**

Future steps could include trying more applicable models, hyperparameter tuning or ensemble methods to further refine model performance.

# References and Bibliography

For your project on predicting early readmissions of diabetic patients, here are some valuable references and resources:

* 1. **References**

1. **Primary Dataset**

UCI Machine Learning Repository’s Diabetes 130-US Hospitals for years 1999–2008 dataset. This dataset includes over 50 features related to patient demographics, medical history, and hospital outcomes, making it ideal for analyzing readmission patterns.

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

1. **Research Articles**

“Predicting diabetic patient readmission using machine learning” provides insights into various algorithms for modeling early readmissions. Published by IEEE.

URL: <https://ieeexplore.ieee.org/document/8781796>

URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3570838/>

1. **Implementation Resources**

GitHub repository showcasing a project on predicting diabetic patient readmissions using Python and machine learning techniques. It provides code and step-by-step guides for feature engineering and model building.

URL: GitHub Repository: <https://github.com/pmacinec/diabetes-patients-readmissions-prediction>

* 1. **Bibliography**
  2. UCI Machine Learning Repository. Diabetes 130-US Hospitals for years 1999–2008 Data Set. Available at: <https://archive.ics.uci.edu/ml/> .
  3. IEEE Xplore. Prediction of diabetic patient readmission using machine learning. Available at: <https://ieeexplore.ieee.org> .
  4. Arine, Bruno. Predicting diabetic patient readmission with machine learning. Available at: <https://brunoarine.com> .
  5. GitHub. Diabetes Patients Early Readmissions Prediction. Available at: <https://github.com/pmacinec> .

# Appendix

8. 1. **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 | DiabetesMed | 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 8.1: Data dictionary