A Case Study of U.S. Hospitals

Hospital Readmission Prediction for Diabetes Patients

Report

Group 2

ROSSILLON Timothée, FAOUZI Walid, SHAN Jiameng,

NDAW Abdoulaye, THEODORE Mary Sheeba

1. DATASET OVERVIEW

The dataset includes information from 1999 to 2008 about patient care in 130 hospitals across the U.S. Each row represents the hospital records of a diabetes patient who received lab tests, medications, and stayed in the hospital for up to 14 days. **The goal is to predict whether a patient will be readmitted to the hospital within 30 days after being discharged.** This problem matters because, even though there is strong evidence that proper care improves outcomes for diabetes patients, many still don't receive adequate treatment in hospitals. This is often due to inconsistent diabetes management, particularly in controlling blood sugar levels. Poor care not only increases hospital costs (due to readmissions) but also puts patients at higher risk of complications, which can harm their health and even be life-threatening.

2. CONTEXT OF THE PROBLEM

The dataset includes features that provide both demographic and biographical details about the patients, as well as information related to their diagnoses, treatments, and medications. Our primary objective is to develop a model that predicts whether a patient will be readmitted within 30 days of discharge.

Qualitatively, certain factors are likely to influence the probability of readmission. For instance, elderly patients may have a higher likelihood of returning, as might patients with conditions requiring ongoing care. Additionally, individual differences in how patients manage their health—such as varying levels of attentiveness to medical needs—could play a role. Each patient’s unique physical condition and health status also add layers of complexity.

Before delving into data exploration, it is crucial to consider these contextual factors and reflect on questions that could impact our modeling approach. For example, issues like the statistical independence of data points or feature collinearity are likely to arise. Taking the time to assess the context qualitatively provides valuable guidance for these decisions later in the process

3. CONCEPT OF SOLUTION

Having grasped the context of the dataset and the problem, we recognized that data cleaning would be a critical step in our solution. Key decisions had to be made regarding the encoding of features to ensure that the raw data was synthesized in a concise and efficient manner. This was essential to minimize redundancies and ensure our features were robust enough for modeling.

Given that the dataset consisted primarily of medical data with categorical features, we made several assumptions about which models would be most suitable. Based on our research, we understood models like Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, Gaussian NB, XGBoost and Logistic Regression are frequently applied in such cases. Medical data often incorporates prior knowledge such as a patient’s health or medical history which can be a powerful predictor. For example, elderly patients with severe diagnoses

are likely to have a higher probability of readmission compared to younger patients with less severe issues. In the medical field, Naive Bayes is particularly relevant for addressing such conditional probabilities, where the goal is not just to predict disease Y but to assess the likelihood of disease Y given specific attributes like age, medical history, or family history.

From the outset, we ruled out Linear Regression due to the binary nature of our target variable, which Linear Regression is not well-suited for. Logistic Regression, however, is specifically designed for binary classification problems, making it a natural choice.

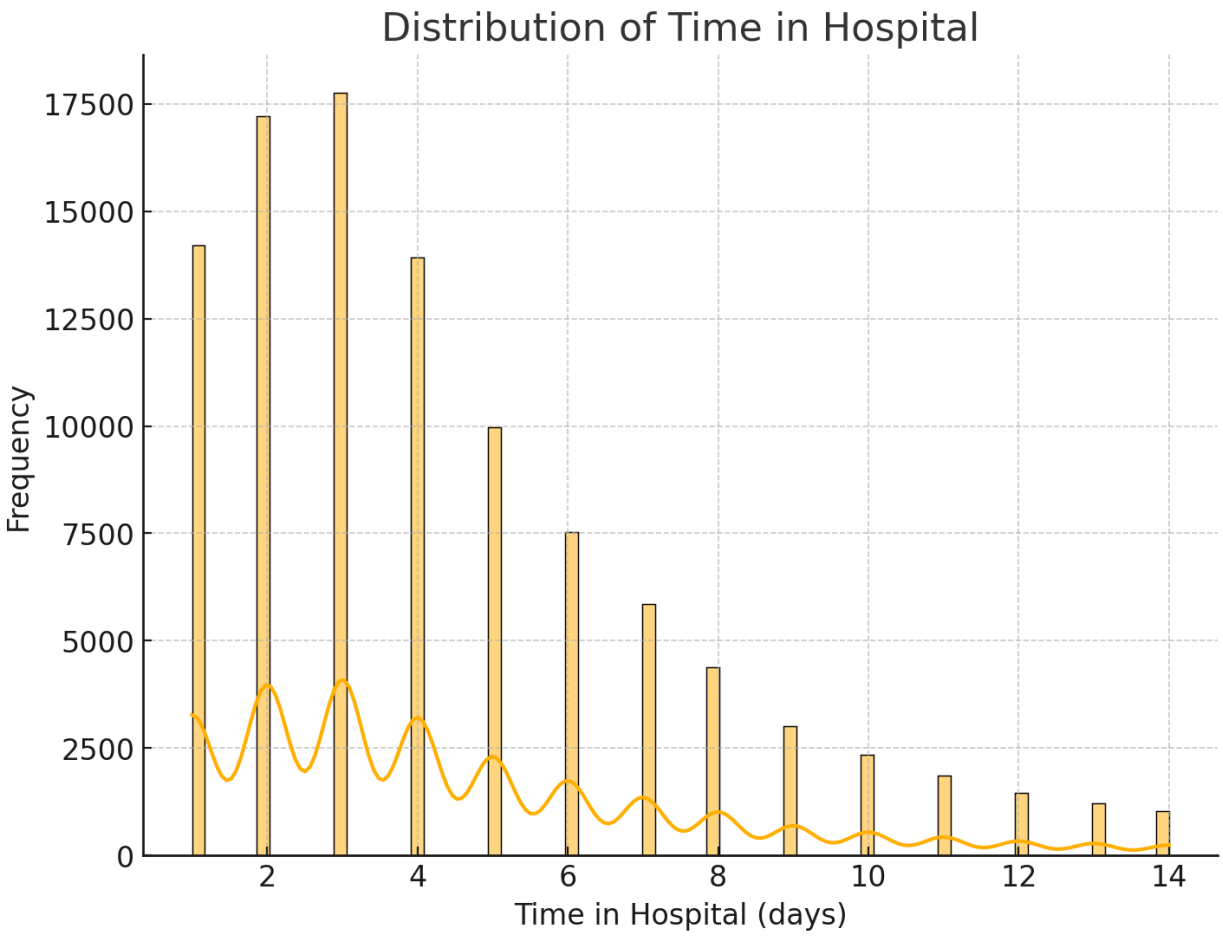
4. EXPLORATORY DATA ANALYSIS

The analysis conducted through exploratory data analysis (EDA) provided meaningful insights into the dataset's structure and its implications for machine learning performance and interpretability. Here's a detailed interpretation:

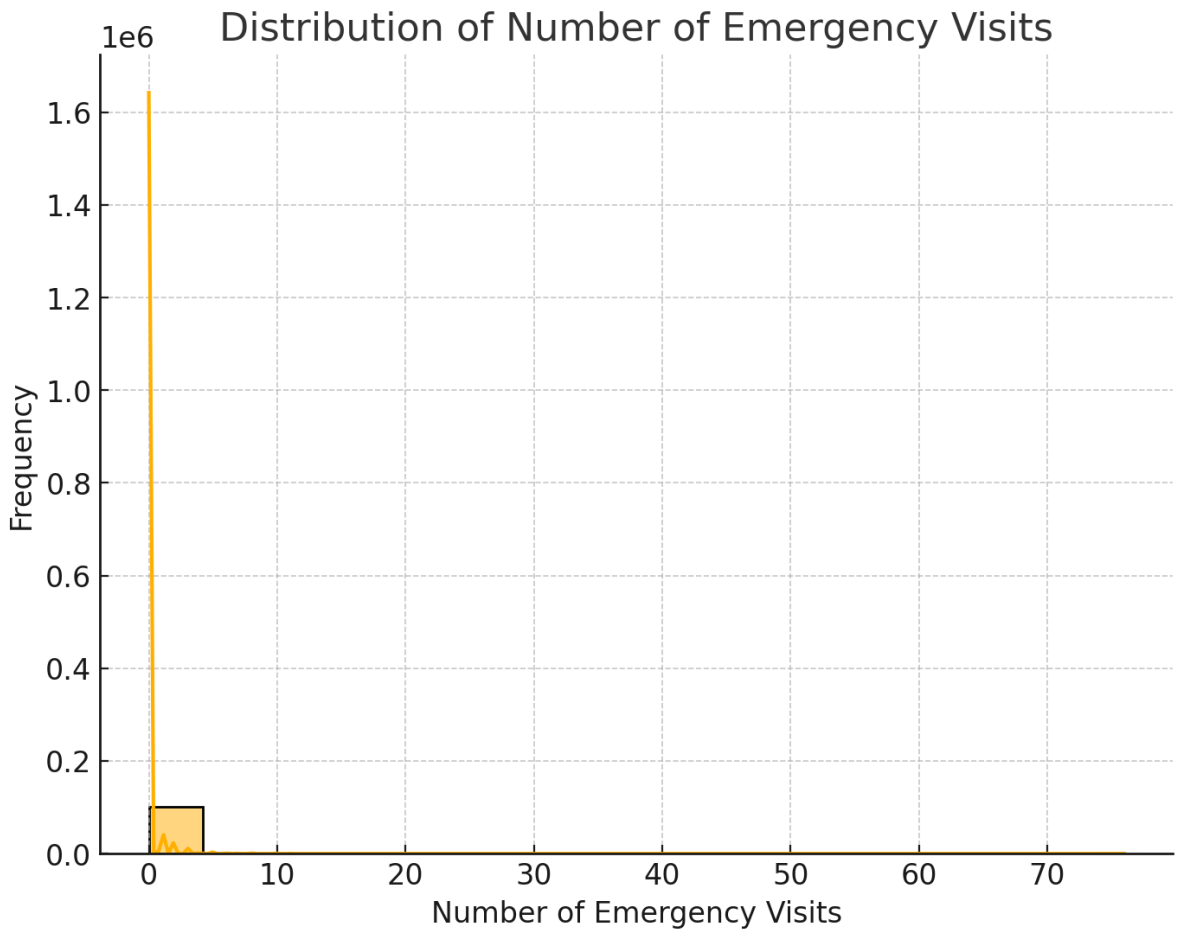
4. 1 Numerical Features: Analysis and Implications

Key numerical variables such as time_in_hospital, num_lab_procedures, num_medications, and number_diagnoses demonstrate distinct central tendencies and variability, indicating their roles in describing patient profiles and resource utilization. Some features, like number_emergency and number_inpatient, exhibit extreme outliers, which can significantly impact machine learning models sensitive to scaling, such as Logistic Regression or k-Nearest Neighbors.

From a performance perspective, high variability among these features points to the need for normalization or scaling to ensure algorithmic stability. Outlier management, such as capping or robust transformations, may further enhance the model's robustness. Interpretability-wise, attributes like time_in_hospital and num_medications are intuitively tied to patient care metrics, offering insights that resonate with stakeholders.

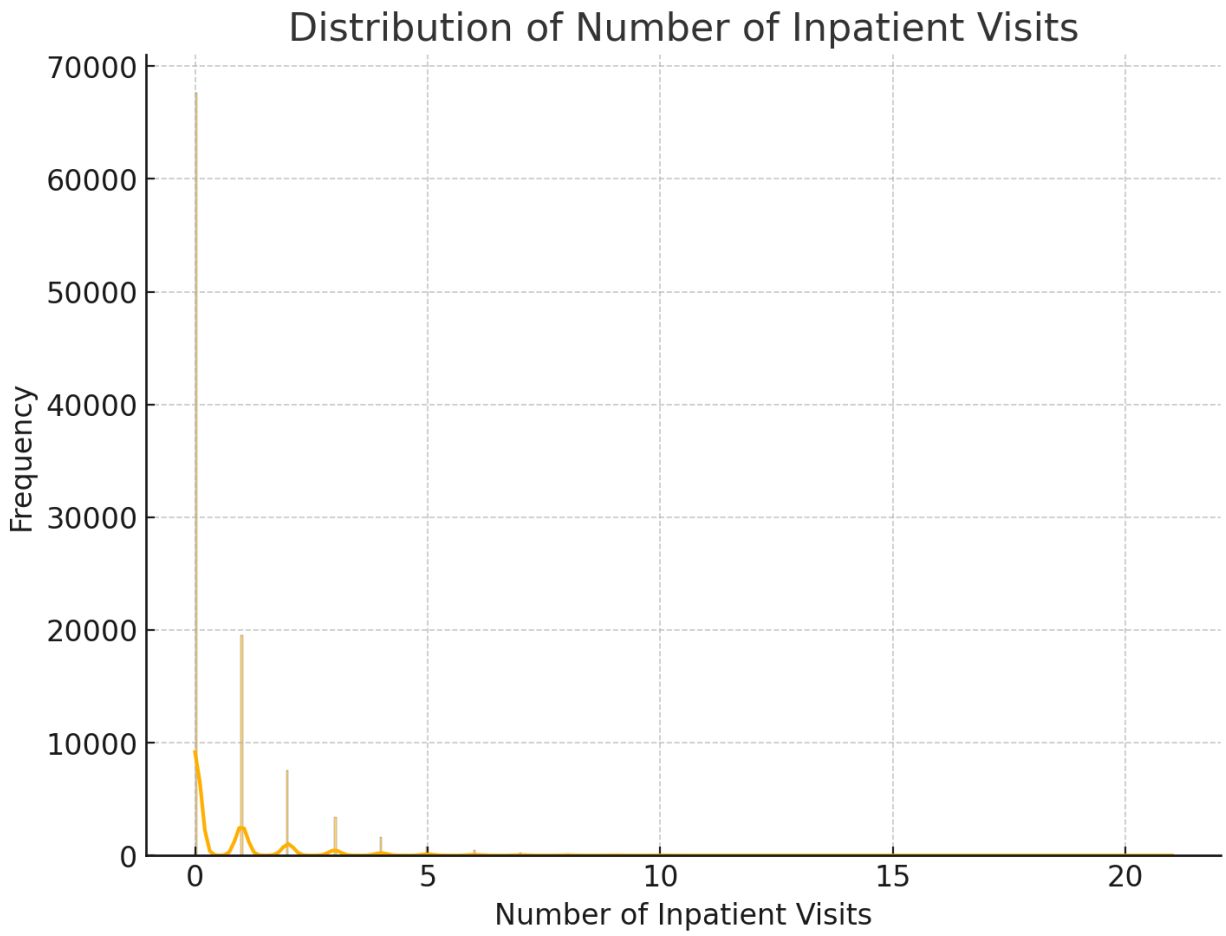


- **Observation**: Often approximates a normal distribution, showing most patients stay a moderate duration, with fewer very short or very long stays.
- **Insight**: Aligns well with machine learning models, as its natural distribution requires minimal preprocessing. Can serve as a direct predictor of resource usage.



- **number_emergency**:

- **Observation**: Often highly skewed, where most patients have zero or very few emergency visits. A long tail indicates a small subset of patients with frequent emergency visits.
- **Insight**: Highlights population-specific healthcare needs. These outliers might require special attention (e.g., transformations) to prevent bias in regression-based models.



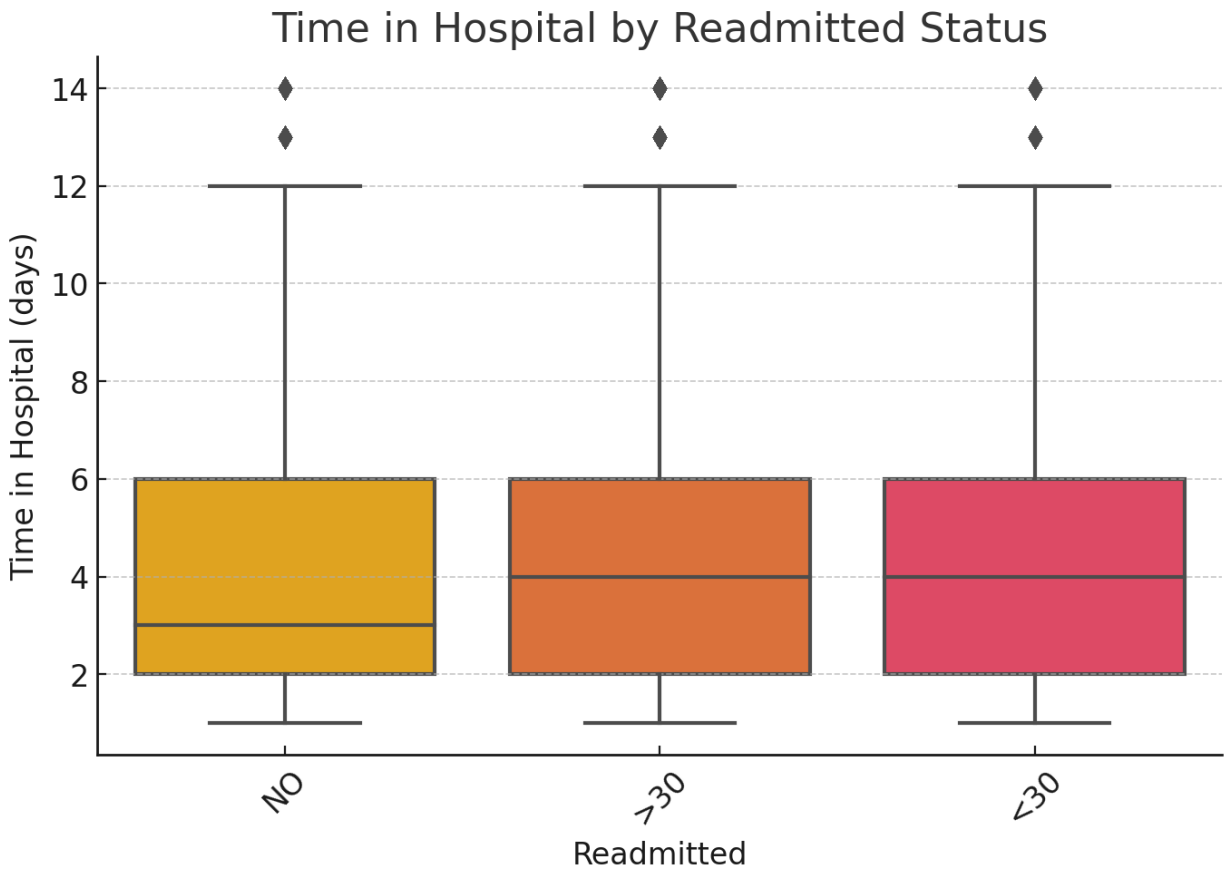
- **number_inpatient**:

- **Observation**: Likely exhibits a similar skewed pattern as number_emergency, with most patients having few inpatient admissions and a few outliers with many.
- **Insight**: Reflects variability in patient profiles and care intensity. Could significantly impact models due to its range and sparsity.

4.2 Categorical Features: Challenges and Relevance

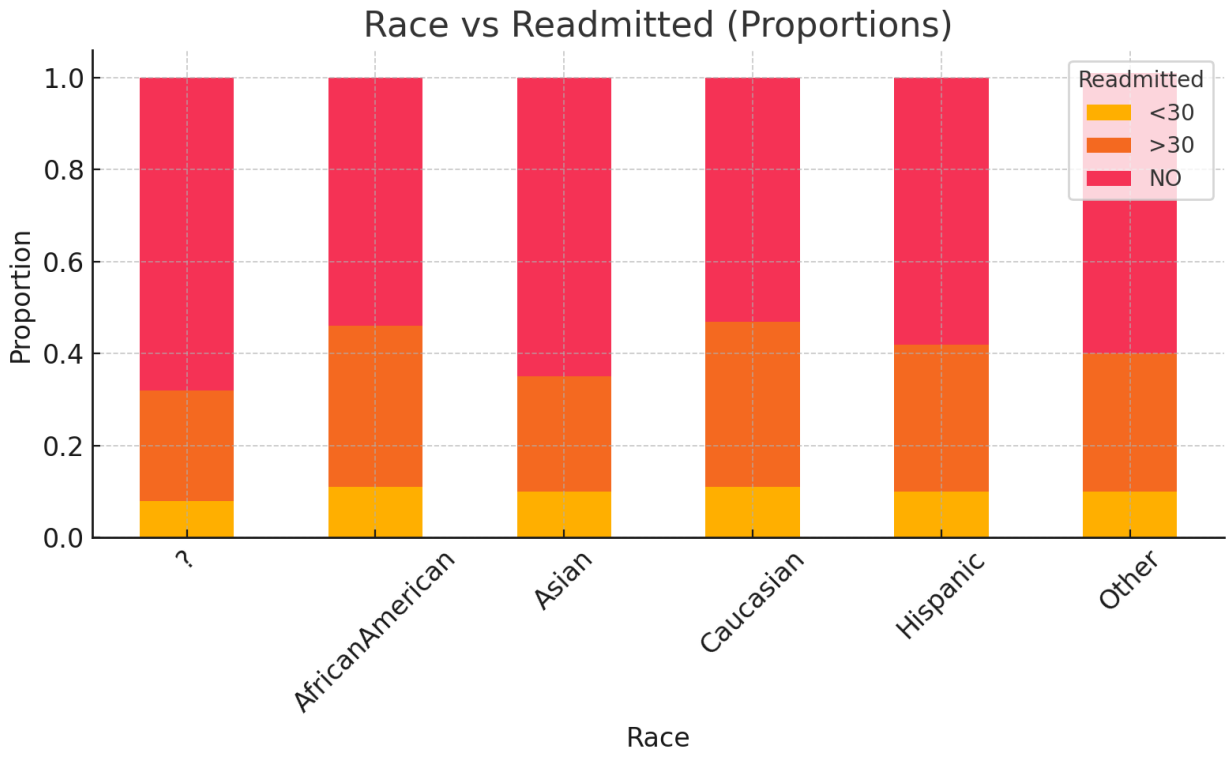
The dataset includes several categorical variables with substantial missing data, such as weight (96.86%) and payer_code (39.56%). These gaps could result from systematic exclusions or irrelevance in certain scenarios. Additionally, features like diag_1, diag_2, and diag_3 exhibit high cardinality, complicating encoding strategies and increasing the risk of overfitting.

For performance optimization, high missingness in variables such as weight suggests they may be excluded unless reliable imputation strategies are available. High-cardinality features demand careful preprocessing, such as grouping or dimensionality reduction, to mitigate sparsity in one-hot or target encoding. From an interpretability standpoint, collapsing diagnoses into broader categories can simplify communication of results and align with actionable insights.



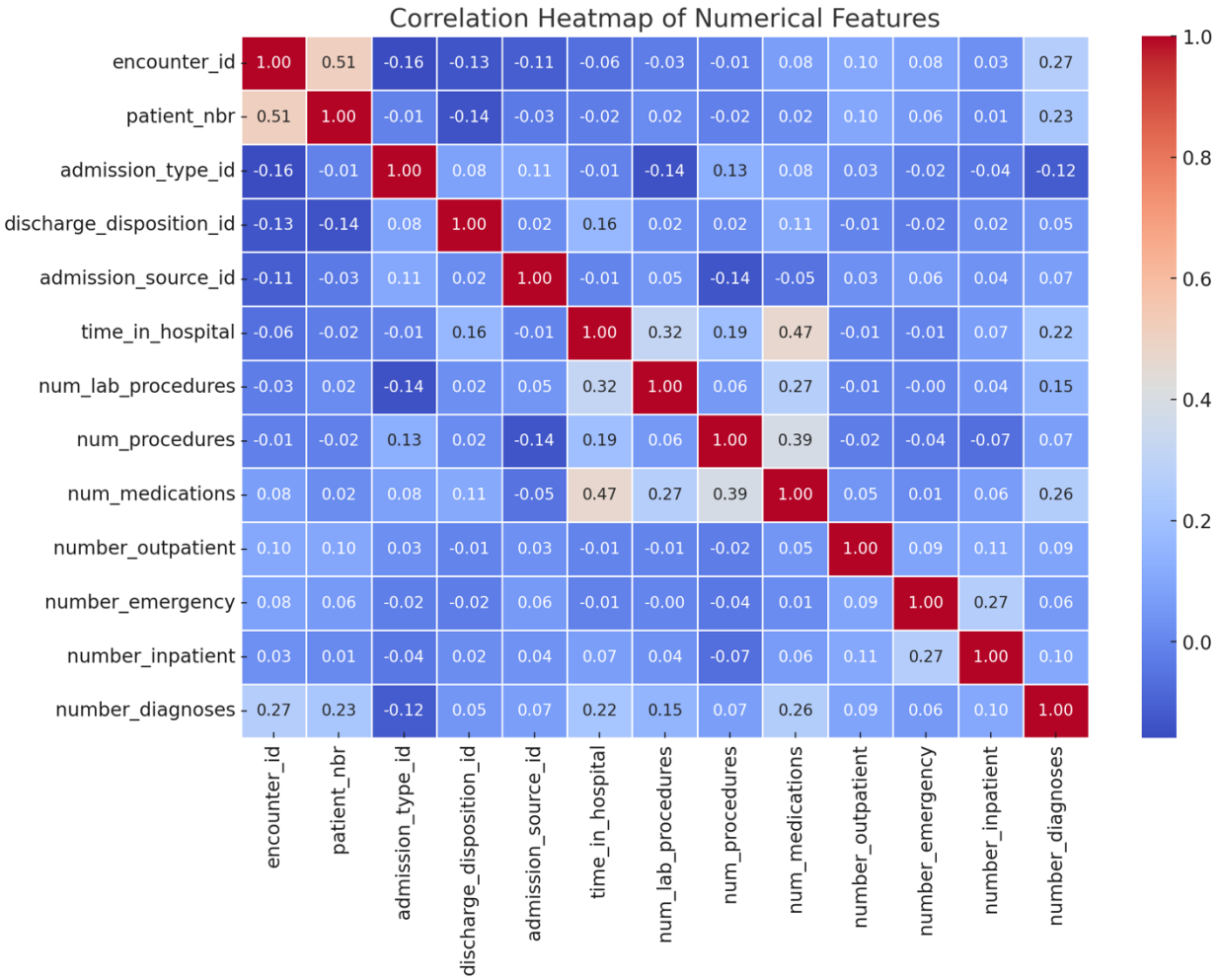
Box Plot: Time in Hospital by Readmitted Status:

- **Observation**: Patients readmitted within 30 days (<30) tend to have longer hospital stays compared to those not readmitted (NO).
- **Insight**: Time in hospital could be a strong predictor of readmission, reflecting higher care intensity or complexity of cases.



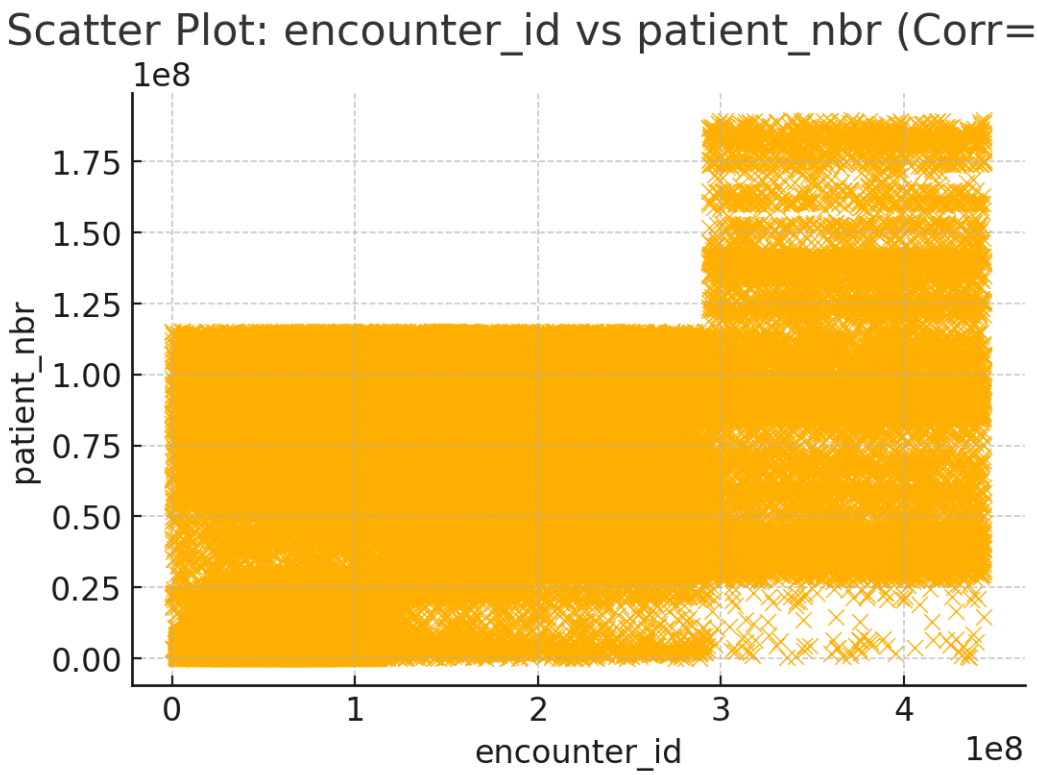
Stacked Bar Plot: Race vs Readmitted:

- **Observation**: Variations in readmission rates are evident across racial categories. For instance, some racial groups have a higher proportion of <30 readmissions.
- **Insight**: Demographic factors like race may interact with healthcare access or treatment outcomes, requiring careful consideration in modeling.



Correlation Heatmap:

- **Observation**: Strong correlations exist between certain numerical features, such as num_medications and num_lab_procedures.
- **Insight**: These correlations suggest potential redundancy or feature interaction opportunities, guiding feature selection and engineering.



Scatter Plot: Strongly Correlated Numerical Features:

- **Example**: A scatter plot between two strongly correlated features highlights their linear relationship.
- **Insight**: Correlated variables may convey overlapping information, which could impact model interpretability and performance.

4.3 Variable Distribution Patterns

The analysis revealed significant skewness in features like number_emergency and number_inpatient, highlighting population-specific trends where most patients have limited occurrences, with a few outliers. Conversely, features like time_in_hospital display near-Gaussian distributions, which align well with algorithms that assume normality in data.

Skewed distributions necessitate transformations, such as logarithmic or square root scaling, to stabilize model behavior, particularly in regression tasks or classifiers with

threshold-dependent sensitivity. Gaussian-like distributions, being naturally well-suited for statistical and machine learning models, require minimal preprocessing. These distributional insights underscore unique patient behaviors and care patterns.

4.4 Outliers and Their Impact

Outliers in variables such as number_emergency, number_inpatient, and num_lab_procedures extend beyond the interquartile range and could distort model predictions. Models prone to outlier influence, like linear regression, can benefit from strategies such as trimming or robust scaling.

Addressing outliers also aids interpretability by ensuring that predictions remain within feasible ranges, maintaining credibility in derived conclusions for stakeholders.

4. 5 Missing Data and Its Implications

The extensive missingness observed in variables such as weight and payer_code highlights structural or procedural gaps. Features with excessive missingness, particularly weight, may need exclusion to avoid introducing noise. Meanwhile, moderately missing data, as seen in payer_code, can be addressed through statistical imputation or machine learning-based methods.

Efforts to manage missing data contribute to both performance and interpretability, ensuring that the final dataset remains comprehensive and the insights trustworthy.

4. 6 Recommendations for Optimization

Feature engineering is critical to enhancing the dataset's utility. High-cardinality variables like diag_1, diag_2, and diag_3 should be grouped into broader diagnostic categories, reducing sparsity and improving interpretability. Missing data in features such as payer_code can be imputed using domain-relevant statistical methods or predictive algorithms.

The EDA revealed a moderate positive correlation between num_lab_procedures and num_medications. The race and gender features show some potential relationship with readmission rates. We also observed a significant class imbalance in the target variable, readmitted, which will require handling during preprocessing. Outliers were detected in time_in_hospital, which may need to be addressed. These insights will be used to guide the feature selection and modeling phases of the project

Preprocessing steps, including normalization and scaling for features like num_medications, are essential to manage variability. Transformations addressing skewed features such as number_emergency will stabilize inputs for sensitive algorithms.

In terms of model selection, tree-based algorithms such as Random Forest are recommended for handling categorical data with high cardinality effectively. Features like weight may be excluded if imputation fails to yield meaningful patterns.

To enhance interpretability, techniques such as SHAP values or permutation importance can be employed to validate the contributions of key variables like time_in_hospital and num_lab_procedures, ensuring the model's insights align with domain expectations.

This structured approach ensures that the dataset is prepared optimally for predictive modeling while maintaining transparency and trust in its outputs. Further exploration into visualization, multivariate relationships, or feature importance analysis can provide additional depth if needed.

5. DATA PRE-PROCESSING

- The dataset contains 48 features and 101766 observations.

- 11 numerical features and 37 categorical columns

Feature	Description	Data Type
race	Patient's race (e.g., Caucasian, African American, etc.)	Categorical
gender	Patient's gender (Male, Female, Unknown/Invalid)	Categorical
age	Age group in 10-year intervals (e.g., '[0-10)', '[10-20)', etc.)	Categorical
weight	Patient's weight in pounds (frequently missing)	Categorical
admission_type_id	Type of admission (e.g., emergency, elective)	Integer
discharge_disposition_id	Discharge status (e.g., to home, expired)	Integer
admission_source_id	Source of admission (e.g., physician referral, emergency room)	Integer
time_in_hospital	Length of hospital stay in days	Integer
payer_code	Payer type (e.g., Medicare, Medicaid)	Categorical
medical_specialty	Specialty of the admitting physician (e.g., cardiology, internal medicine)	Categorical
num_lab_procedures	Number of laboratory procedures during the encounter	Integer
num_procedures	Number of other procedures (excluding labs) during the encounter	Integer

num_medications	Number of unique medications administered	Integer
number_outpatient	Number of outpatient visits in the previous year	Integer
number_emergency	Number of emergency visits in the previous year	Integer
number_inpatient	Number of inpatient visits in the previous year	Integer
diag_1	Primary diagnosis code (ICD-9)	Categorical
diag_2	Secondary diagnosis code (ICD-9)	Categorical
diag_3	Additional diagnosis code (ICD-9)	Categorical
number_diagnoses	Total number of diagnoses	Integer
max_glu_serum	Maximum glucose serum test result ('None', 'Norm', '>200', '>300')	Categorical
A1Cresult	Hemoglobin A1c result ('None', 'Norm', '>7', '>8')	Categorical
diabetesMed	Whether diabetes medications were prescribed ('Yes', 'No')	Categorical
change	Whether there was a change in medication ('Ch', 'No')	Categorical
insulin	Insulin prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
metformin	Metformin prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
repaglinide	Repaglinide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
nateglinide	Nateglinide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
chlorpropamide	Chlorpropamide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
glimepiride	Glimepiride prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
acetohexamide	Acetohexamide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
glipizide	Glipizide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
glyburide	Glyburide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
tolbutamide	Tolbutamide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
pioglitazone	Pioglitazone prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
rosiglitazone	Rosiglitazone prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
acarbose	Acarbose prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
miglitol	Miglitol prescription status ('Up', 'Down', 'Steady', 'No')	Categorical

troglitazone	Troglitazone prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
tolazamide	Tolazamide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
examide	Examide prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
citoglipton	Citoglipton prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
glyburide-metformin	Glyburide-metformin combination prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
glipizide-metformin	Glipizide-metformin combination prescription status ('Up', 'Down', 'Steady', 'No')	Categorical
readmitted	Whether the patient was readmitted within 30 days	Binary

5.1 Dropped Features

features_drop_list = ['weight', 'payer_code', 'medical_specialty', 'repaglinide', 'nateglinide', 'chlorpropamide', 'acarbose', 'miglitol', 'troglitazone', 'tolazamide', 'examide', 'citoglipton', 'glyburide-metformin', 'glipizide-metformin', 'glimepiride-pioglitazone', 'metformin-rosiglitazone','metformin-pioglitazone', 'acetohexamide', 'tolbutamide']

Features	Reasons
weight	97% of null value
Payer_code	Around 40% of missing values, and the payer code has no significant abbreviation, also not relevant to the outcome
medical_specialty	Around 50% of missing values
Repaglinide, nateglinide, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, glyburide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-pioglitazone, acetohexamide, tolbutamide	These features are dominated by the "No" value, indicating that these medications were either not prescribed or not changed during most encounters.

We retained 30 features after the drop

Race:

- Grouped Asian, Other and Hipsanic to one category called ‘Others’ as they were very less in proportion, when compared to Caucasian and AfricanAmerican.

Gender

- 3 rows with value ‘Unknown/Invalid’ was dropped as they do not have any significance

Age

- In the dataset, age was not a continuous variable. It was defined as range eg, [0-10] upto [70-80]. We decided to compute the mid points(5, 10, 15 etc.) and keep them as values for age, as it would be an important factor for our prediction

5.2 Recoded and Grouped Features

Feature	Values and Recoding
Gender	Male: 1, Female: 0
max_glu_serum	Labels: Null: 0, Norm: 1, >200:2, >300: 3
A1Cresult	Labels: Null:0, Norm:1, >7: 2, >8: 3
Change	No: 0, Yes: 1
DiabetesMeb	Yes: 1, No: 0
Readmitted	<30: 1, >30: 0, No: 0 (Target variable for binary classification)
Race	Labels: Caucasian: 1, AfricanAmerican: 2, Others: 3
Dosage Changes	For each drug: metformin, glimepiride, glipizide, glyburide, pioglitazone, rosiglitazone, insulin: - No: 0, Steady: 2, Down: 1, Up:3

admission_type_id column has been grouped and remapped using the admission_type_mapping dictionary to simplify and consolidate similar admission categories into broader groups. Here's the rationale behind the grouping:

Emergency (1), Urgent (2), and Trauma Center (7): These represent high-priority admissions requiring immediate or urgent care. They have been grouped together under the same category (1) to reflect their similarity in urgency.

Elective (3): This category is distinct because it represents planned admissions that are non-urgent, so it retains its unique category (3).

Newborn (4): Admissions for newborns are fundamentally different from other types and are retained as their unique category (4).

Not Available (5), NULL (6), and Not Mapped (8): These categories represent missing, undefined, or unclassified admission types. They are grouped under a single category (5).

discharge_disposition_id column has been remapped using the discharge_mapping dictionary to group similar discharge dispositions into broader, more interpretable categories. Here's the rationale behind the grouping:

Home and Related Categories (1, 6, 8, 9, 13):

a. These represent scenarios where patients are discharged to their home or a similar setting (e.g., under home health service or home IV provider care).
b. Grouped under category 1 to indicate a return to home or equivalent.

Transfer to Another Facility (2, 3, 4, 5, 14, 22, 23, 24, 27, 28, 29, 30):

c. Includes discharges to other hospitals, long-term care facilities, rehabilitation centers, nursing homes, psychiatric hospitals, and other inpatient or federal institutions.
d. Grouped under category 2 to consolidate all types of institutional transfers.

Neonate and Outpatient Follow-Up (10, 12, 15, 16, 17):

e. Represents cases such as neonatal aftercare or discharges where patients are expected to return for outpatient services.
f. Grouped under category 10 to indicate specialized or follow-up care.

Expired (11, 19, 20, 21):

g. Includes all cases where the patient expired, whether at home, in a medical facility, or an unspecified location.
h. Grouped under category 11 for all mortality-related outcomes.

Left Against Medical Advice (7):

i. Indicates patients who left the hospital against medical advice.
j. Retains its own category (7) due to its distinct nature.

Missing, Not Mapped, or Invalid Data (18, 25, 26):

k. Represents missing, invalid, or unclassified discharge outcomes.
l. Grouped under category 18 to handle undefined or unknown values consistently.

admission_source_id column has been remapped using the discharge_mapping dictionary to group similar discharge dispositions into broader, more interpretable categories. Here's the rationale behind the grouping

Referral Sources (1, 2, 3):

a. **Description:** Includes patients referred by a physician, clinic, or HMO.
b. **Mapped to Category 1:** This grouping consolidates all types of referral sources to represent admissions initiated by medical professionals or organizations.

Transfers (4, 5, 6, 10, 22, 25):

c. **Description:** Includes transfers from another hospital, skilled nursing facility (SNF), critical access hospital, ambulatory surgery center, or other healthcare facilities.
d. **Mapped to Category 2:** This grouping consolidates all transfer-related admissions to represent patients arriving from another healthcare institution

Emergency (7, 8):

e. **Description:** Includes admissions from the emergency room or court/law enforcement.
f. **Mapped to Category 3:** This grouping highlights emergency or involuntary admissions, which are distinct due to their urgency or external legal factors

Births (11, 13, 14):

g. **Description:** Includes normal deliveries, sick baby admissions, and extramural births.
h. **Mapped to Category 4:** This grouping represents admissions related to childbirth and neonatal care.

diag –1, diag-2, diag-3 are basically codes of International Statistical Classification of Diseases and Related Health Problems, They data type is changed to float, missing values are replaced with –1 and Strings starting with E and V are replaced with 0

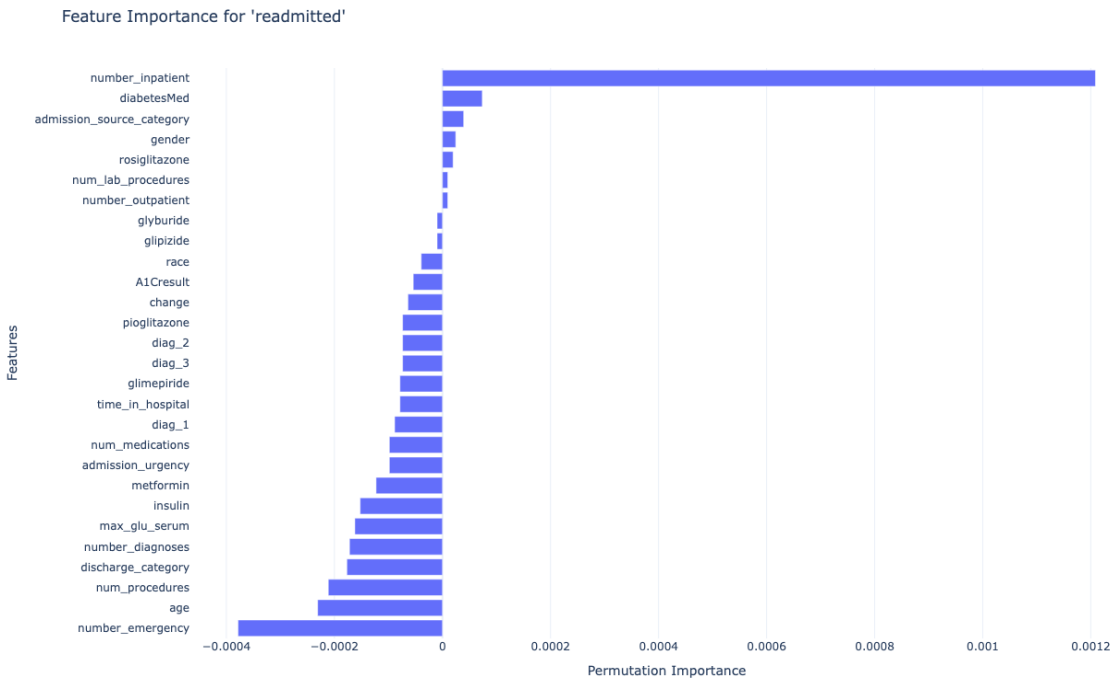
- ICD-9 codes related to diabetes are typically found within the range **240-279**, specifically under **Endocrine, nutritional, and metabolic diseases**. These codes cover different types of diabetes, including Type 1, Type 2, and gestational diabetes

ICD-9 Range	Label	Description
1-139	1	Infectious and parasitic diseases
140-239	2	Neoplasms
240-279	3	Endocrine, nutritional, and metabolic diseases; immunity disorders
280-289	4	Diseases of the blood and blood-forming organs
290-319	5	Mental disorders

320-389	6	Diseases of the nervous system and sense organs
390-459	7	Diseases of the circulatory system
460-519	8	Diseases of the respiratory system
520-579	9	Diseases of the digestive system
580-629	10	Diseases of the genitourinary system
630-679	11	Complications of pregnancy, childbirth, and the puerperium
680-709	12	Diseases of the skin and subcutaneous tissue
710-739	13	Diseases of the musculoskeletal system and connective tissue
740-759	14	Congenital anomalies
760-779	15	Certain conditions originating in the perinatal period
780-799	16	Symptoms, signs, and ill-defined conditions
800-999	17	Injury and poisoning
E	0	External causes of injury
V	0	Supplemental classification

5.3. PERMUTATION IMPORTANCE with Logistic Regression Model

Permutation importance is a model-agnostic method to assess feature importance by measuring the change in model performance when the values of a feature are randomly shuffled.



Rank	Feature	Permutation Importance Value	Interpretation
1	number_inpatient	0.02462	Most important feature; frequent inpatient visits strongly predict readmission risk.
2	diabetesMed	0.00477	Presence or absence of diabetes medication significantly affects readmission.
3	insulin	0.00278	Insulin prescription impacts readmission likelihood significantly.
4	number_emergency	0.00245	Emergency visits contribute to the likelihood of readmission.
5	num_procedures	0.00220	More procedures performed correlate with higher readmission likelihood.
6	glyburide	0.00206	Glyburide prescription is associated with readmission outcomes.
7	metformin	0.00193	Metformin usage has a moderate influence on predicting readmissions.
8	rosiglitazone	0.00058	Prescription of rosiglitazone moderately influences readmission.
9	pioglitazone	0.00044	Pioglitazone usage has minor predictive power for readmission.
10	glipizide	0.00025	Prescription of glipizide provides a small predictive value.
11	glimepiride	0.00020	Glimepiride usage slightly influences readmission prediction.

Key Observations:

1) **Top Predictors**:
    a. number_inpatient is the most significant feature, confirming its importance in predicting readmissions.
    b. diabetesMed and insulin follow, highlighting the importance of medication-related features.
2) **Low or Negative Impact Features**:
    a. Features with **negative permutation importance** suggest they introduce noise, are redundant, or overlap with more predictive features. These include:
        i. **Demographic Features**: age, race, and gender showed little to no predictive power for readmission.
        ii. **Clinical Measures**: num_lab_procedures and A1Cresult were minimally impactful, indicating they may not provide strong signals for readmissions.
        iii. **Tertiary Diagnoses**: Features like diag_3, diag_1, and diag_2 had low or negative contributions, suggesting weak correlation with the target variable.
3) **Features with the Most Negative Impact**:

a. **time_in_hospital (-0.00962)** and **number_diagnoses (-0.00917)**: These features had the largest negative importance values, suggesting they likely overlap with or add redundancy to other features like number_inpatient.
b. **num_medications (-0.00727)**: Too many medications might introduce noise rather than predictive power.

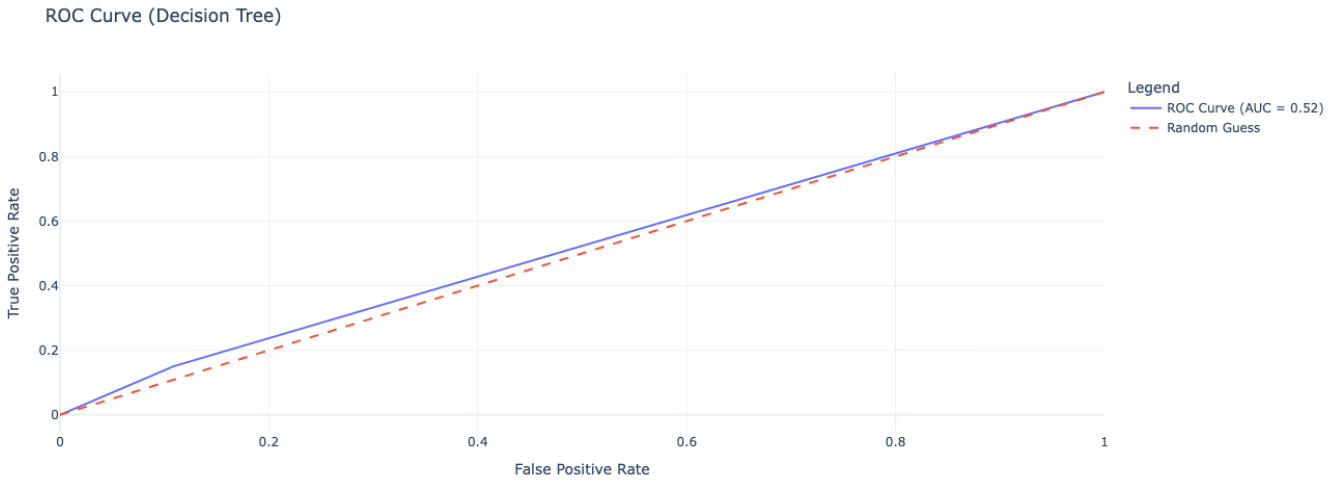
4) **Discharge and Admission Factors**:
   a. **discharge_category (-0.00412)** and **admission_source_category (-0.00019)** had little to no contribution, possibly due to a lack of granularity or relevance in the context of readmission.

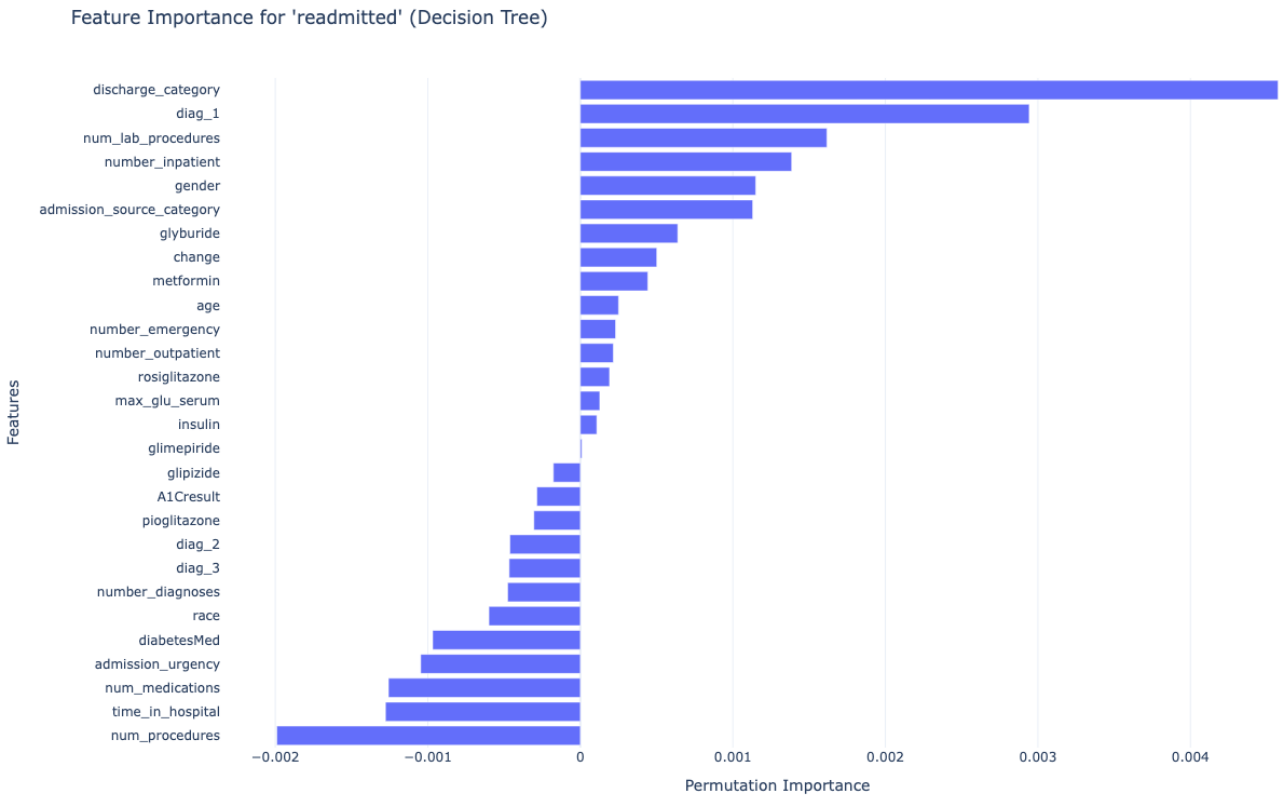
5.4. PERMUTATION IMPORTANCE with DecisionTree Classifier in comparison to Logistic regression Model.

Feature	Importance (Logistic Regression)	Importance (Decision Tree)	Commonality
number_inpatient	**0.02468**	**0.00139**	**High in both**
discharge_category	-0.00416	0.00457	**Moderately important in both**
diag_1	-0.00077	0.00294	**Present in both**
num_lab_procedures	-0.00210	0.00162	**Moderately important in both**
insulin	0.00283	0.00011	**Present in both**
diabetesMed	**0.00493**	-0.00097	Unique to Logistic Regression
number_emergency	0.00244	0.00023	Unique to Logistic Regression
metformin	0.00194	0.00044	Unique to Logistic Regression
admission_source_category	-0.00030	0.00113	Unique to Decision Tree
diag_3	-0.00273	-0.00047	Present in both but less important
time_in_hospital	-0.00965	-0.00128	Present in both but less important
num_medications	-0.00719	-0.00126	Present in both but less important
age	-0.00808	0.00025	Present in both but low impact

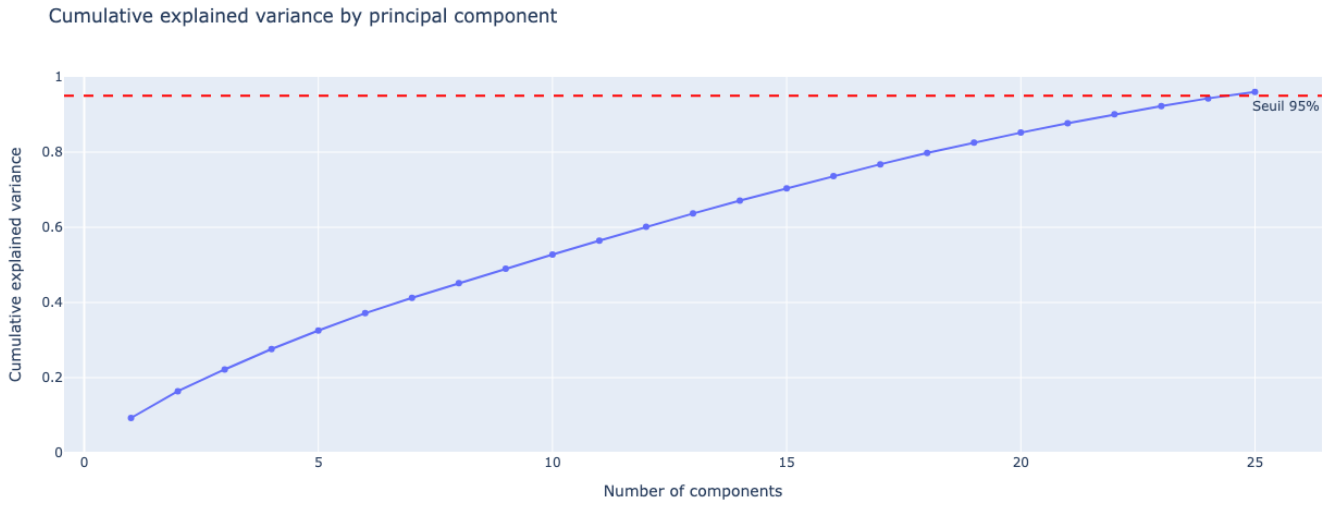
Key Observations from the Table

1. **Highly Important Features in Both Models**:
    a. number_inpatient is the most important feature in Logistic Regression and still carries weight in Decision Tree.
    b. discharge_category and num_lab_procedures are moderately important in both models.
2. **Features Unique to Logistic Regression**:
    a. diabetesMed has the highest importance specific to Logistic Regression.
3. **Features Unique to Decision Tree**:
    a. admission_source_category and diag_1 are highlighted by the Decision Tree.
4. **Low Impact Features in Both**:
    a. Features like time_in_hospital and num_medications have low importance in both models.





5.5 Principal Component Analysis (PCA)



1. **Cumulative Explained Variance**:

a. The graph indicates how many components are needed to explain increasing amounts of variance.
b. At **25 components**, 95% of the total variance is captured.

2. **Dimensionality Reduction**:
   a. From the graph, using fewer components (e.g., **12 from the Kaiser criterion**) simplifies the data while still capturing significant variance.

5.5.1 Explained Variance by Retained Components (Kaiser Criterion)

Component	Key Contributing Features	Interpretation
PC1	time_in_hospital, num_lab_procedures, num_medications, change, diabetesMed	Reflects medical intensity (hospital stay, labs, and medication management).
PC2	age, discharge_category, num_procedures, admission_urgency	Patient demographics (age) and procedural factors dominate.
PC3	admission_source_category, diag_3, A1Cresult, time_in_hospital	Admission sources and secondary diagnoses influence this component.
PC4	time_in_hospital, num_procedures, insulin, change	Hospital duration and insulin treatments are significant.
PC5	age, discharge_category, num_lab_procedures, rosiglitazone	Age and lab procedures drive this component, along with specific medications.
PC6	admission_source_category, num_lab_procedures, glyburide, metformin	Focus on admission sources and diabetes medications.
PC7	time_in_hospital, glyburide, diabetesMed, glipizide, change	Diabetes-specific treatments and hospital stays dominate.
PC8	glimepiride, metformin, rosiglitazone, num_medications	Highlights specific diabetes medications and their use patterns.
PC9	admission_source_category, rosiglitazone, glimepiride, diag_3	Influenced by secondary diagnoses and specific medications.
PC10	rosiglitazone, glipizide, num_lab_procedures, metformin	Diabetes treatments and lab procedures dominate this component.
PC11	glimepiride, discharge_category, num_procedures, age	Focus on discharge processes, patient age, and specific medications.
PC12	diag_1, time_in_hospital, num_diagnoses, insulin	Driven by primary diagnoses, insulin use, and the number of diagnoses per patient.

- **Medical Intensity**: time_in_hospital, num_lab_procedures, num_medications, and change consistently appear in the top components.
- **Diabetes Management**: Features like insulin, diabetesMed, metformin, and rosiglitazone highlight the impact of diabetes treatment
- Each component captures a unique aspect of the data, with PC1 capturing the most variance (9.28%) and PC12 the least (3.66%).

PC	Summary
PC13	Driven by **discharge processes**, **age**, and minor interactions with medications.
PC14	Reflects patient demographics (gender and race) and minor effects of specific medications.
PC15	A mix of **admission urgency** and minor effects of medications.
PC16	Balances **discharge categories** and **medication-specific effects** (e.g., metformin, insulin).
PC17	Dominated by race, age, and **interaction effects across diagnoses**.
PC18	Driven by a combination of **diagnoses and diabetes treatments** (e.g., rosiglitazone).
PC19	Focuses on the interplay between **race**, **discharge category**, and **medications**.
PC20	Highlights **admission urgency**, **minor medication effects**, and noise.
PC21	Captures more complex patterns of **age** and **demographics** interactions with medications.
PC22	Reflects **minor effects of medications**, particularly glipizide and glyburide.
PC23	Focuses on secondary diagnoses (diag_2, diag_3) and **medication contributions**.
PC24	Represents complex **interactions across diagnoses, medications, and demographic features**.
PC25	Primarily driven by noise and redundant effects.

5.5.2 Summary of Most Important Features Across the 12 PCs

Feature	Relevance Across PCs
time_in_hospital	Appears in **PC1, PC2, PC4, PC12**; strongly linked to medical intensity.
num_medications	Highly important in **PC1**; indicates overall treatment complexity.
change	Significant in **PC1, PC4**; captures changes in medication during hospital stay.
age	Strongly represented in **PC2, PC5**; highlights patient demographics.
num_lab_procedures	Important in **PC1, PC6**; associated with diagnostic efforts.
admission_source_category	Significant in **PC3, PC6**; related to how patients access care.
insulin	Relevant in **PC4, PC12**; captures diabetes-specific management.
rosiglitazone	Strongly featured in **PC5, PC9, PC10**; relates to specific diabetes treatments.
glyburide	Dominant in **PC6, PC7**; indicates treatment choices for diabetes.
glimepiride	Appears prominently in **PC8, PC11**; linked to specific diabetes medications.

The dataset was organized and partitioned into training and testing subsets, with 80% allocated for training and 20% for testing. To predict the readmission rates among patients with diabetes, we developed models utilizing six widely used binary and multiclass analysis, as follows: Logistic regression, Decision tree, Random Forest tree, Gradient boosting, GaussianNB and XGBoost.

Table 1 Model description

Method	Description	Implementation Details
Decision Tree Classifier	A tree structure where each node represents a feature condition. Data is split recursively until leaf nodes represent final classifications.	Constructs a decision tree that maximizes information gain, supports hyperparameters, makes predictions by traversing the tree based on feature values.
Random Forest Classifier	An ensemble method using multiple decision tree. Combines the outputs of individual trees (via majority voting) to enhance predictive performance.	Combines multiple decision trees trained on bootstrapped samples of the dataset and random subsets of features, aggregates predictions via majority voting for classification tasks.
Gradient Boosting Classifier	Sequentially builds a series of weak learners (usually shallow trees), each correcting errors of the previous model by optimizing a loss function.	Sequentially trains weak learners and uses a differentiable loss function to optimize predictions
GaussianNB	A probabilistic classification technique that assumes each feature independently predicts the output variable. The final classification is determined by selecting the class with the highest calculated probability	Implemented using the GaussianNB class from scikit-learn. Assumes that features follow a Gaussian distribution; calculates mean and variance for each class; predicts based on the highest calculated probability.
XGBoost	An advanced form of gradient boosting that improves speed and performance by leveraging system optimizations and additional regularization techniques.	Implemented using the XGBoost from xgboost library, supports efficient memory usage and parallelized tree construction.

Performance Evaluation

To comprehensively access the prediction effect on readmission rates, this project employed the following five distinct evaluation metrics: the area under the receiver operating characteristic (AUROC) curve; precision; recall; F1 score; and accuracy.

$$precision = PPV = \frac{TP}{TP + FP}$$

$$Recall = TPR = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{(2 \cdot precision \cdot Recall)}{precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Four common metrics in binary classification tasks include true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Precision measures the proportion of correct predictions among positive samples, indicating the ratio of true positive samples to all samples classified as positive. Recall, or sensitivity, reflects the proportion of actual positive samples that are correctly identified, representing the ratio of true positive samples to all actual positive samples. These metrics are integral to identification and prediction algorithms. The F1 score is often employed to evaluate the precision of classification algorithms. Another critical performance metric is accuracy, which quantifies the proportion of correct predictions across all samples, calculated as the ratio of correctly classified samples to the total number of samples in the test dataset.

Prediction of the Readmission

Table 1. The model performance metrics (original data)

	Accuracy	Precision	Recall	F1-score	ROC RUC score	Average precision
Decision Tree	0.79	0.14	0.18	0.17	0.53	0.12
Random Forest	0.89	0.52	0.01	0.01	0.64	0.19
Gradient Boosting	0.89	0.5	0.01	0.01	0.68	0.22
Gaussian Naive Bayes	0.86	0.25	0.13	0.17	0.62	0.18
XGBoost	0.89	0.40	0.02	0.03	0.66	0.20
Logistic Regression	0.89	0.47	0.01	0.02	0.64	0.20

Table 2. The model performance metrics (PCA, 95% variance explained)

	Accuracy	Precision	Recall	F1-score	ROC RUC score	Average precision
Decision Tree	0.79	0.14	0.17	0.15	0.52	0.12
Random Forest	0.89	0.52	0.01	0.02	0.60	0.17
Gradient Boosting	0.89	0.53	0.01	0.01	0.64	0.20
Gaussian Naive Bayes	0.87	0.30	0.09	0.14	0.60	0.17
XGBoost	0.89	0.41	0.02	0.04	0.61	0.17
Logistic Regression	0.89	0.48	0.01	0.03	0.64	0.20

Table 3. The model performance metrics (PCA, Kaiser criterion)

	Accuracy	Precision	Recall	F1-score	ROC RUC score	Average precision

Decision Tree	0.80	0.13	0.15	0.14	0.51	0.11
Random Forest	0.89	0.56	0.01	0.02	0.60	0.17
Gradient Boosting	0.89	0.77	0.00	0.01	0.63	0.19
Gaussian Naive Bayes	0.89	0.32	0.05	0.09	0.60	0.17
XGBoost	0.89	0.33	0.01	0.02	0.61	0.17
Logistic Regression	0.89	0.41	0.01	0.01	0.63	0.19

The reduction from 28 columns to 25 columns to retain 95% of the variance is not considered a significant reduction in dimensionality. In this case, the application of PCA does not bring sufficient benefits for the following reasons:

- **Small reduction**: the reduction of only 3 columns does not significantly simplify the dataset
- **Added complexity**: The application of PCA adds a pre-processing step that can complicate the interpretation of results.
- **Loss of interpretability**: Principal components are linear combinations of the original variables, which may make it more difficult to interpret the results.

6. MODELING

6.1 Resampled the target value

Using SMOTE (Synthetic Minority Over-sampling Technique) to resample the target value is crucial in addressing the significant class imbalance observed in your dataset, with a distribution of 72,324 for the majority class and only 9,086 for the minority class. This imbalance leads to poor model performance, particularly in terms of recall for the minority class, as evidenced by low scores across various models. By generating synthetic examples for the minority class, SMOTE enhances its representation in the training set, allowing models to learn more effectively from these instances. Consequently, this approach not only improves the overall predictive performance but also reduces bias towards the majority class, resulting in better balance between precision and recall metrics. The desired objective is to enable the models to better detect the positive class.

Table 1. Shape and Distribution before and after using SMOTE

Description	Data shape	Class Distribution (0 / 1)
Before SMOTE	(81410, 28)	72324 / 9086
After SMOTE	(144628, 28)	72324 / 72324

Table 2. Results after using SMOTE

	Accuracy	Precision	Recall	F1 Score	ROC AUC	Average Precision
Decision Tree	0.198	0.115	0.925	0.205	0.516	0.115
Random Forest	0.232	0.117	0.897	0.207	0.541	0.124
Gradient Boosting	0.119	0.112	0.996	0.201	0.560	0.128
Gaussian Naive Bayes	0.697	0.158	0.397	0.226	0.599	0.179
XGBoost	0.127	0.113	0.993	0.202	0.503	0.111
Logistic Regression	0.640	0.163	0.537	0.250	0.636	0.195

Interpretation:

Decision Tree: The decision tree model shows low accuracy (19.8%) and precision (11.5%), indicating that it struggles to correctly classify instances of both classes, despite a high recall (92.5%) for the minority class (class "1"). This suggests that while it identifies many true positives, it also misclassifies a significant number of true negatives, leading to poor overall performance.

Random Forest: The random forest model performs slightly better with an accuracy of approximately 23%. It maintains a high recall (89.7%) for the minority class but has low precision (11.7%), indicating that many predicted positives are false positives.

Gradient Boosting: This model exhibits very low accuracy (11.9%) and precision (11.2%), while achieving near-perfect recall (99.6%) for the minority class, which indicates it is almost always predicting the minority class correctly but fails to generalize well, resulting in poor performance on the majority class.

Gaussian Naive Bayes: This model stands out with the highest accuracy (69.7%) among the tested models and shows a more balanced performance with moderate precision (15.8%) and recall (39.7%). It suggests that this model can better distinguish between classes compared to others.

XGBoost: Similar to gradient boosting, XGBoost has low accuracy (12.7%) and precision (11.3%), but it achieves high recall (99.3%). This again indicates a tendency to predict the minority class excessively, leading to poor classification of the majority class.

Logistic Regression: Logistic regression shows reasonable performance with an accuracy of about 64% and better precision (16.3%) and recall (53.7%) compared to other models, suggesting it strikes a better balance between correctly identifying both classes.

Overall, while SMOTE helps in increasing the recall for minority classes across all models, it often comes at the cost of precision and overall accuracy, highlighting challenges in effectively

balancing class representation without introducing noise or overfitting in predictions for the majority class and the results still not sufficient.

6.2 Models Selection and tuning

We decided to tune models such as XGBoost, Gradient Boosting, Random Forest, and Logistic Regression for improving their predictive performance, because of their high upgrade potential especially in the context of imbalanced datasets. XGBoost and Gradient Boosting are powerful ensemble methods that can significantly benefit from hyperparameter tuning. By adjusting parameters like learning rate, maximum depth of trees, and the number of estimators, we can optimize their ability to learn complex patterns in the data while preventing overfitting. This is particularly important given that these models can easily become too complex if not properly tuned. Random Forest is another ensemble method that combines multiple decision trees to improve accuracy and control overfitting. Tuning parameters such as the number of trees, maximum features, and minimum samples required to split an internal node can enhance its performance on both majority and minority classes, making it more robust in handling class imbalance. Logistic Regression, while simpler than the aforementioned models, still requires careful parameter tuning to maximize its effectiveness. Adjusting parameters like regularization strength can help improve the model's ability to generalize from training data to unseen data, especially in scenarios with imbalanced classes. In summary, tuning these models is crucial not only for optimizing their predictive capabilities but also for ensuring they effectively handle class imbalances and generalize well to new data. This process ultimately leads to better decision-making based on the model outputs.

We decided to tune our models using **RandomizedSearchCV** followed by **GridSearchCV** to optimize their performance effectively. Initially, RandomizedSearchCV allows us to explore a wide range of hyperparameter values without exhaustively searching through all possible combinations. This method is particularly useful when dealing with a large parameter space, as it can identify promising regions quickly while requiring significantly less computational time compared to a full grid search. For instance, in the case of the XGBoost model, we defined a diverse parameter distribution for hyperparameters like ‘n_estimators’, ‘max_depth’, and ‘learning_rate’, enabling us to capture various configurations that could enhance model performance. Once we identify the best parameters from the RandomizedSearchCV, we narrow down the search space and utilize GridSearchCV for fine-tuning. This approach allows us to systematically evaluate specific values around the best parameters found earlier, ensuring that we achieve optimal performance. For example, by adjusting ‘n_estimators’ and ‘max_depth’ around the previously identified best values, we can refine our model's capabilities further. This two-step tuning process not only improves efficiency but also enhances the likelihood of finding the most effective hyperparameter settings for our models, ultimately leading to better predictive accuracy and robustness.

The results for the optimized models show slightly better performance, but not much different from previous models. This shows that tuning was not a great added value in this case, and that the use of SMOTE can also be questioned.

6.3 Resampled without fake data (class_weight)

We therefore observe that the results of the different models are not conclusive. Furthermore, in the context of medical data analysis, the use of SMOTE can be problematic due to the introduction of synthetic data that may not accurately reflect real-world scenarios. This can lead to misleading conclusions and potentially damaging decisions in clinical settings. Consequently, it is more prudent to abandon the SMOTE technique and use the class_weight parameter in models such as Random Forest, XGBoost and Logistic Regression.

By applying 'class_weight', we can assign different weights to classes during model training, effectively penalizing the model more for misclassifying instances of the minority class. This adjustment encourages the model to focus on correctly identifying minority class instances without artificially inflating the dataset with synthetic examples. As a result, this method maintains the integrity of the original data while addressing class imbalance issues.

The objective of doing this is for better handling of imbalanced datasets, particularly in medical applications where accurate predictions for minority classes are crucial. It can lead to improvements in recall and F1 scores for these classes without compromising the overall model performance. This strategy not only enhances model reliability but also aligns with ethical considerations in healthcare analytics by ensuring that predictions are based on authentic data rather than artificially generated instances. This approach could potentially lead to improved model performance and better decision-making results in clinical practice.

In this way, for the Random Forest we use class_weight='balanced_subsample'. For the xgboost we compute the scale_pos_weight by dividing the length of class 0 by the length of class 1. For the Logistic Regression we use the parameter class_weight='balanced'.

Table 1: Results for models with class weight applied

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1 Score (Class 1)	ROC AUC	Average Precision
Random Forest	0.8886	0.6154	0.0035	0.0070	0.6547	0.2000
XGBoost	0.6951	0.1803	0.4883	0.2633	0.6474	0.1980
Logistic Regression	0.6663	0.1705	0.5152	0.2562	0.6433	0.1977

The results for the models utilizing class weights reveal varying levels of performance, particularly in their ability to identify the minority class. For Random Forest, the model achieved an accuracy of approximately 88.86%, demonstrating strong classification of the majority class (0) with a precision of 61.54% for the minority class (1). However, it exhibited a critically low recall of 0.35%, indicating that it identified very few true positives. This resulted in an F1 score of 0.0070, reflecting a significant imbalance in performance between classes. The ROC AUC score of 0.6547 suggests moderate discriminatory ability, but the model still struggles with minority class identification. XGBoost showed a lower accuracy of about 69.51%, with a precision of 18.03% for the minority class and a recall of 48.83%. This indicates better identification of minority cases compared to Random Forest, but still highlights considerable room for improvement, as evidenced by an F1 score of 0.2633 and a ROC AUC of 0.6474. In contrast, Logistic Regression achieved an accuracy of approximately 66.63%, with a precision of 17.05% for the minority class and a recall of 51.52%. The F1 score stands at 0.2562, indicating similar challenges in balancing precision and recall as seen in XGBoost and Random Forest, while its ROC AUC score of 0.6433 suggests limited effectiveness in distinguishing between classes. We can see that the results obtained in this way are better than when using SMOTE, but the results are still not good enough.

Given these results, while Random Forest has the highest accuracy, its inability to effectively identify minority cases makes it less suitable for applications where detecting positive instances is critical. XGBoost offers a more balanced approach with better recall and a reasonable F1 score, making it potentially more useful in scenarios where identifying minority class instances is essential. So, while Random Forest shows strong overall accuracy, XGBoost emerges as the better model in terms of identifying minority class instances effectively (even more in the case of imbalanced dataset), making it more suitable for applications where such detection is crucial.

Table 2: Results for XGBoost after optimization

Metric	**Value for XGBoost**
Accuracy	0.6575
Precision (Class 1)	0.1843
Recall (Class 1)	0.6041
F1 Score (Class 1)	0.2824
ROC AUC Score	0.6780
Average Precision	0.2178
Confusion Matrix	[[12010, 6072], [899, 1372]]

The results for the optimized XGBoost model show a notable change in performance compared to the previous iteration. The optimized model achieved an accuracy of 65.75%, which is a decrease from the earlier accuracy of 69.51%. This indicates that while the model still performs reasonably well overall, it has lost some predictive power in classifying instances correctly. In terms of precision for the minority class (Class 1), the optimized model shows a slight

improvement with a precision of 18.43%, compared to 18.03% previously. However, this still reflects a low ability to accurately predict positive cases when they are identified. The recall for Class 1 has improved significantly to 60.41%, up from 48.83% in the earlier results, indicating that the model is now better at identifying actual positive instances. The F1 score for Class 1 stands at 0.2824, which is an increase from the previous score of 0.2633, suggesting a better balance between precision and recall in the optimized model. The ROC AUC score has also improved to 0.6780, compared to 0.6474 previously, indicating enhanced capability in distinguishing between classes.

In conclusion, while the optimized XGBoost model shows improvements in recall and F1 score, it experiences a decrease in overall accuracy compared to its earlier performance. The enhancements in recall indicate that the model is more effective at identifying positive cases, which is crucial in applications where detecting minority instances is essential. However, the low precision still suggests that there is significant room for improvement in accurately predicting these cases. Overall, these results highlight the importance of continued tuning and exploration of strategies to enhance performance, particularly for minority class detection in critical applications such as medical diagnostics.

Cross-validation:

The implementation of cross-validation in our analysis served to rigorously evaluate the performance of the optimized XGBoost model by ensuring a robust assessment across multiple subsets of the training data. By utilizing StratifiedKFold, which maintains the proportion of classes in each fold, the approach aimed to provide reliable estimates of key performance metrics, such as precision, recall, and F1 scores.

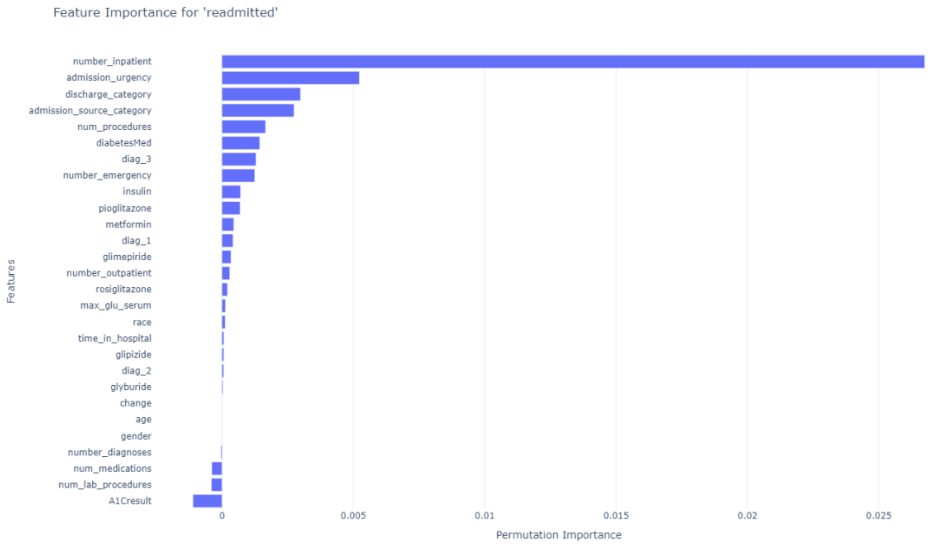
The results from cross-validation revealed better metrics: the precision scores averaged around 84%, indicating a strong ability of the model to accurately identify positive instances across different folds. The recall scores were approximately 65%, suggesting that the model effectively detected a substantial portion of actual positive cases. The F1 scores ranged from 0.7141 to 0.7169, reflecting a solid balance between precision and recall. In comparison, the previously reported results for the optimized XGBoost model indicated an accuracy of 65.75%, with a precision for Class 1 at 18.43%, recall at 60.41%, and an F1 score of 0.2824. These earlier metrics highlighted significant challenges in accurately predicting minority class instances.

Overall, the cross-validation results demonstrate that the optimized XGBoost model exhibits significantly improved performance metrics compared to its previous evaluation, particularly in terms of precision and F1 scores. This suggests that the model is now better equipped for applications requiring accurate detection of rare events, making it a more viable option for medical diagnostics. Further tuning and validation may continue to enhance its effectiveness in real-world applications.

Permutation Importance:

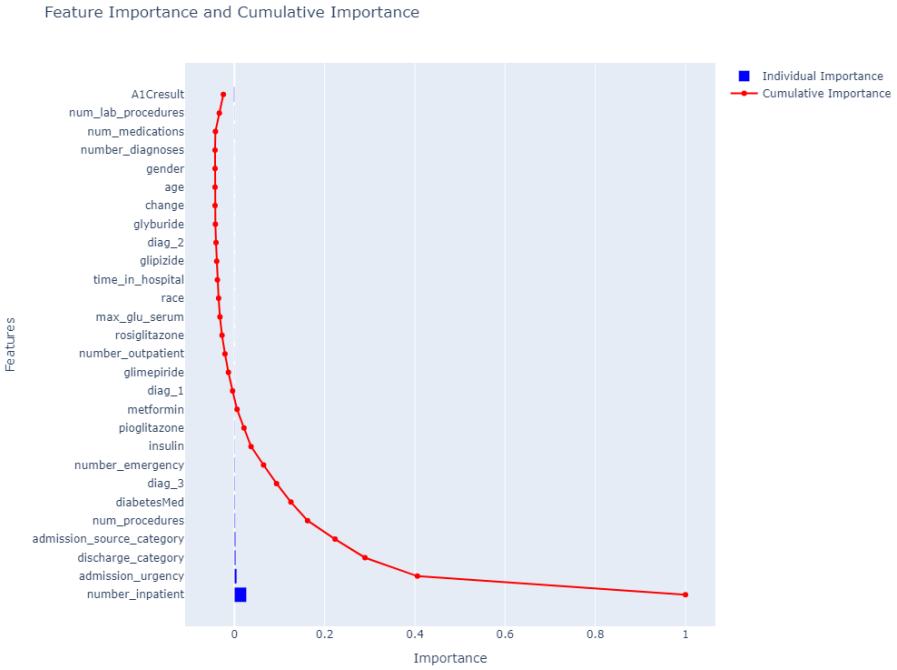
Then we compute permutation importance for the XGBoost model to see how the model work.

Fig1. Feature Importance for the target value



Then we compute the cumulative importance for the XGBoost model.

Fig2. Cumulative Importance and Feature Importance



Focusing on a subset of the top ten features can significantly enhance the interpretability and explainability of a model. By narrowing down the number of features, we simplify the model, making it easier to understand how each feature contributes to the predictions. This reduction can help in identifying which factors are most influential in driving outcomes, thereby providing clearer insights for stakeholders and facilitating better decision-making.

The results obtained with only the top ten features indicate an accuracy of 65.75%, with a precision for Class 1 at 18.43%, and a recall of 60.41%. The F1 score for Class 1 stands at 0.2824, suggesting that while the model can identify a reasonable proportion of positive cases, its precision remains low, leading to a high rate of false positives. The ROC AUC score is 0.6780, indicating moderate ability to distinguish between classes. In comparison, using the full dataset features after cleaning leads to similar results.

6.4 Multi-class classification

We set the target variable into three modalities to apply the multi-class classification, transitioning from binary classification to multimodal analysis (not readmitted, readmitted within 30 days, readmitted after 30 days) allows for distinguishing characteristics and risk factors for patients across different timeframes. Compared to the binary model, multiple classification could capture more detailed information and enhance predictive power and can better showcase the distribution of data and differences in influencing factors, training a multi-class model (e.g., multinomial logistic regression, decision trees) for the three modes improves predictive accuracy and practical applicability.

Table1 The model performance metrics (multi-class)

	Accuracy	Precision	Recall	F1 Score	ROC AUC
Decision Tree	0.464	0.472	0.464	0.468	0.542
Random Forest	0.578	0.553	0.578	0.530	0.660
Gradient Boosting	0.583	0.555	0.582	0.530	0.674
Gaussian Naive Bayes	0.553	0.508	0.553	0.486	0.624
XGBoost	0.581	0.551	0.581	0.538	0.678
Logistic Regression	0.568	0.529	0.568	0.496	0.637

Interpretation:

Decision Tree: The Decision Tree model shows the weakest performance across all metrics compared to other models, with the lowest Accuracy (0.464), Precision (0.472), Recall (0.464), F1 Score (0.468), and ROC AUC (0.542), which indicate that this model performs well for the majority class (class 0) and its performance for minority classes (1 and especially 2) is poor, which showcases a strong class imbalance. It may suffer from overfitting and lack generalization ability for multi-class classification tasks. Thus, decision tree is not applicable to multi-class models to predict whether a patient will be readmitted within 30 days.

Random Forest: Random: Random Forest performs significantly better than Decision Tree in accuracy, F1 score and ROC AUC. However, its Precision (0.553) and Recall (0.578) are moderate, it is an ensemble method, which reduces overfitting and increases stability. However, it still shows the class imbalance challenges, performs well on the majority class (class 0) but struggles with the minority classes (1 and especially 2).

Gradient Boosting: Gradient Boosting achieves slightly better results than Random Forest, with a slightly higher score in Accuracy (0.583), Precision (0.555), Recall (0.582), and ROC AUC (0.674). However, the F1 Score (0.530) is identical to Random Forest. The high recall for class 0 and poor recall for classes 1 and 2 points to the model's tendency to focus more on the majority class while neglecting the minority classes. Overall, Gradient Boosting is particularly effective in handling class imbalances and capturing non-linear patterns, making it a strong contender.

Gaussian Naive Bayes: Gaussian Naive Bayes has moderate Accuracy (0.553), but its Precision (0.508) and F1 Score (0.486) are relatively low compared to other models. Its ROC AUC (0.624) is higher than Decision Tree but falls short of the ensemble methods. Gaussian Naive Bayes assumes feature independence, which may oversimplify the problem and result in subpar performance for this multi-class task.

XGBoost: XGBoost delivers the best overall performance, with high Accuracy (0.581), Precision (0.551), Recall (0.581), and F1 Score (0.538). Its ROC AUC (0.678) is the highest among all models. In terms of class imbalance, this imbalance skews the performance. Though this model is impacted by class imbalance, its imbalance problem is relatively improved compared to other models. All in all, its robustness, regularization techniques, and ability to handle imbalanced data make it the top-performing model for this task.

Logistic Regression: Logistic Regression performs moderately well, with Accuracy (0.568), Precision (0.529), Recall (0.568), and F1 Score (0.496). Its ROC AUC (0.637) is better than Gaussian Naive Bayes but lower than ensemble methods. It provides a baseline for comparison but lacks the ability to capture complex non-linear patterns, which limits its performance.

Hyperparameter Tuning of XGBoost for Multi-Class Classification:

To improve the performance of the XGBoost model, we conducted an extensive hyperparameter tuning process. The tuning consisted of two main steps: a randomized search to explore a broad range of parameter combinations, followed by a grid search to refine and optimize the selected parameters. This approach ensures that the model is robust, captures complex patterns in the data, and mitigates the impact of class imbalance.

Initially, we performed a randomized search to efficiently explore the parameter space. We defined a wide range of distributions for key hyperparameters such as the number of estimators (n_estimators), maximum tree depth (max_depth), learning rate (learning_rate), subsampling rate (subsample), column sampling rate (colsample_bytree), regularization term (gamma), and minimum child weight (min_child_weight). The randomized search was conducted using 10

iterations with 3-fold cross-validation to balance exploration and computational efficiency. The scoring metric was set to the weighted F1 score (f1_weighted) to emphasize performance across all classes, especially the minority classes.

The best parameters identified from the randomized search were:

colsample_bytree: 0.91, gamma: 0.61, learning_rate: 0.18, max_depth: 7, min_child_weight: 1, n_estimators: 861 & subsample: 0.90. These results yielded a weighted F1 score of 0.544, showcasing the model's ability to balance precision and recall across all classes while mitigating overfitting through regularization and subsampling techniques.

Building on the results of the randomized search, we refined the hyperparameters using grid search. A narrower range of values centered around the best parameters was defined for n_estimators, max_depth, learning_rate, and subsample. Grid search allowed us to fine-tune the model's performance by systematically evaluating combinations of these parameters with 3-fold cross-validation, ensuring a comprehensive yet focused optimization process.

7. CONCLUSION

To ensure consistency within the dataset, manual encoding of the categorical variables was used. These steps may be a point of improvement. Also, key features for diabetics' people like 'weight' are missing. This kind of important informations in diabetic diagnoses could have a significant impact and help models differentiate classes.

Class imbalance management could be improved. The models systematically show low recall and precision for the minority class (class 1). The implementation of advanced techniques to manage class imbalance, or more strategic adjustment of class weights, could help improve detection of minority instances. Nevertheless, in a medical context, it is important to pay attention to the addition of synthetic data and to the relevance of certain methods.

We could optimize more the hyperparameters: Although some adjustments have been made, further optimization of hyperparameters could result in better-performing models.

For the model selection, considering alternative algorithms that are known to perform well with imbalanced datasets, such as LightGBM or CatBoost could be an improvement.

The integration of predictive models in healthcare, particularly for the prediction of readmissions, can transform the management of patient care. The ability to anticipate readmissions within 30 days of discharge enables hospitals to classify patients according to their risk (high, medium, low) and implement targeted interventions, such as frequent follow-ups and personalized education. This improves follow-up and reduces readmission rates.

Economically, this approach optimizes resource management by proactively identifying high-risk patients, enabling early intervention programs and reducing the costs associated with prolonged care. Ensuring adequate follow-up also improves patient satisfaction.

In conclusion, the use of predictive models not only strengthens clinical decision-making but also fosters innovation in medical organizations. This leads to more efficient resource allocation and better care coordination, which is essential in a sector were improving care and reducing costs are crucial.