*Lambton College at Queen’s College of Business, Technology and Public Safety*

**Program**: Big Data Analytics (DSMM)

**Project Name:**

Post-Discharge Readmission Prediction for Diabetic Patients

A Machine Learning Multiclass Perspective

**Course**: Big Data Framework (BDM 3603)

**Professor:** Ishant Gupta

**Presented By :**

Jennylynne Dominguez (C0929140)

Chaw Su Su Thin (C0916347)

Hazel Portia Elaine Santos (C0915982)

Joyce Ann Murillo (C0927648)

## Introduction:

Readmissions to hospitals are a significant source of overall medical costs and a new measure of service quality. Like other chronic illnesses, diabetes is linked to a higher risk of readmission to the hospital. Previous hospitalization, age extremes, and socioeconomic barriers are risk factors [(1)](#_svg6j8cbw36g).

In 2021, diabetes was listed as the eighth leading cause of death in the United States, appearing as the underlying cause in over 103,000 death certificates, and mentioned in nearly 400,000 total deaths. These numbers highlight the systemic burden of diabetes on both individual health and healthcare infrastructure [(2)](#_svg6j8cbw36g).

Readmission rates further underscore this challenge. Studies from 2017 report that 14% to 24.5% of diabetic patients are readmitted within 30 days of discharge — substantially higher than non-diabetic populations. For example, one U.S. hospital system reported a 24.5% 30-day readmission rate among diabetic patients, compared to 17.7% among those without diabetes. A multi-state Medicaid study found that 22.8% of diabetic hospitalizations resulted in readmission within 30 days[(3)](#_svg6j8cbw36g).

These statistics make clear that early readmission among diabetic patients is a clinically relevant and economically costly problem. Improving the ability to identify high-risk patients at discharge could allow healthcare providers to deploy proactive, targeted interventions — such as follow-up appointments, medication reconciliation, or home-based care — to reduce preventable readmissions.

This project addresses this challenge by developing a machine learning-based prediction model to assess the likelihood of a diabetic patient being readmitted within 30 days of discharge. Using a decade of real-world hospitalization data from 130 U.S. hospitals (1999–2008), we frame the prediction task as a multiclass classification problem: (1) no readmission, (2) readmission after 30 days, and (3) readmission within 30 days — with a focus on the last category as the most clinically actionable.

## Problem Statement

This project focuses on post-discharge readmission prediction for diabetic patients. Specifically, we aim to build a machine learning model that predicts whether a patient will be readmitted within 30 days of hospital discharge, using clinical and demographic data available at the time of discharge.

The prediction task is framed as a multiclass classification problem with three categories:

* No readmission
* Readmission after 30 days
* Readmission within 30 days (primary focus)

By identifying high-risk patients at discharge, the model can support healthcare providers in deploying targeted follow-up care, reducing avoidable readmissions, and improving chronic disease management for individuals living with diabetes.

## Objective

The goal of this project is to predict whether a diabetic patient will be readmitted within 30 days after discharge, using information collected throughout their hospital stay. This post-discharge prediction can help hospitals retrospectively identify factors that contribute to readmission risk, and optimize inpatient diabetes management strategies.

**Predict Readmission Timing:** Build a multiclass model to predict if a diabetic patient will be readmitted within 30 days, after 30 days, or not at all, using discharge-time data.

**Demonstrate Practical Usability:** Build an interactive UI that allows clinicians or discharge planners to input patient information available at the end of hospitalization—such as lab results, diagnoses, and medication changes—and receive *real-time predictions of 30-day readmission risk*, illustrating how machine learning can assist in discharge planning and post-care resource allocation.

**Support Clinical Decision-Making**: Support hospital decision-making processes by providing explainable, actionable risk scores that clinicians can use to prioritize patient follow-up care and reduce costly readmissions.

## Project Impact and Value In Healthcare

**Treatment Effectiveness Analysis**

By analyzing how different medications, dosage adjustments, or procedures correlate with readmission outcomes, predictive models can uncover patterns in treatment effectiveness. This allows researchers and clinicians to identify which care pathways are associated with reduced readmission risk.

* Valuable for clinical researchers seeking to understand the real-world impact of therapeutic decisions.
* Can inform updates to clinical protocols and evidence-based guidelines for inpatient diabetes management.
* Helps evaluate consistency across hospitals, revealing whether variation in care contributes to avoidable readmissions.

Such insights not only support retrospective analysis but also help shape future standards of care aimed at improving long-term patient outcomes.

**Care Pathway Optimization**

Predictive models can identify patterns in patient journeys that are associated with higher readmission risk — even when care followed standard clinical protocols. This enables healthcare teams to go beyond checklist compliance and examine whether certain discharge strategies are truly effective.

* Flags discharge plans that may appear appropriate but lead to frequent readmissions.
* Informs timing and structure of follow-up care, such as earlier post-discharge check-ins for high-risk patients.
* Reveals gaps in patient education, self-care understanding, or support systems that standard protocols may overlook.

These insights support continuous improvement of care pathways to reduce preventable readmissions and enhance overall care quality.

## Dataset Overview

The dataset used in this project consists of 101,766 hospital encounters of patients diagnosed with Type 1 or Type 2 diabetes, collected over a 10-year period (1999–2008) from 130 U.S. hospitals and integrated delivery networks. It includes 50 variables covering patient demographics, admission details, prior hospital utilization, lab results, diagnoses, medications, and treatment-related information. Each row represents a single hospital stay, making the dataset multivariate and encounter-level in nature. This rich clinical data enables the development of predictive models to assess 30-day readmission risk, providing insights into patient outcomes and opportunities for improving post-discharge care planning [(4).](#_svg6j8cbw36g)

### Why We Chose This Dataset

**Real-World Clinical Relevance**

This dataset is derived from actual hospital records across 130 U.S. hospitals, making it highly relevant for real-world healthcare analytics. As students, working with authentic medical data allows us to explore how machine learning can impact patient outcomes, particularly in chronic disease management like diabetes.

**Rich Feature Set for Predictive Modeling**

The dataset contains over 50 features, including demographics, lab results, diagnoses, and treatment details. This provides an excellent opportunity to practice preprocessing, feature selection, and model training on a complex, multidimensional dataset.

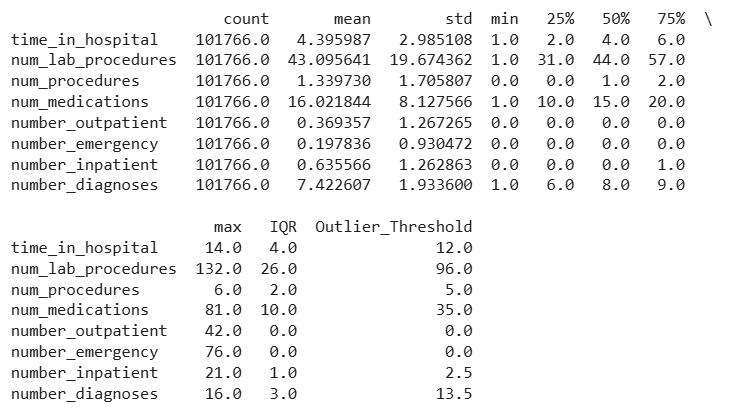
**Imperfect Dataset**

The dataset is imperfect, which reflects real-world healthcare data. It contains missing values, imbalanced classes, and ambiguous categories — all of which present realistic challenges. This allows us to apply what we've learned in data cleaning, preprocessing, and modeling in a more meaningful and practical way, just like in real-world projects.

## Preprocessing Steps

As part of the preprocessing phase, we performed descriptive analysis to understand the structure and distribution of both numerical and categorical variables in the dataset.

1. **Numerical Variables :** We used *.describe().T* to generate summary statistics such as mean, standard deviation, min/max, and quartiles for all numerical features. This helped us:

* Identify potential outliers
* Detect skewed distributions
* Understand value ranges across key clinical indicators (e.g., lab procedures, number of inpatient visits)

**Observation**:

Variables like number\_outpatient, number\_emergency, and number\_inpatient are heavily right-skewed, with most values at 0 but rare high values.

These high values can indicate chronically ill or complex patients and may warrant closer analysis or feature transformation (e.g., binning, log-scaling).

Outliers are present in most fields — especially in num\_lab\_procedures, num\_medications, and number\_emergency.

1. **Categorical Variables :** For categorical columns, we *applied .describe(include='object').T* to view:

* The number of unique values
* The most frequent category (top)
* The frequency of that top category

This helped flag variables with:

* Low cardinality, which may be suitable for one-hot encoding
* Unbalanced distributions (e.g., features where one category dominates)
* Potential data quality issues, such as inconsistent or missing values represented by "?", "NULL", or "Not Available”

This descriptive step was critical in guiding our decisions for:

* Imputation
* Encoding
* Feature selection and transformation

1. **Removed Non-Eligible Discharges**

We excluded records associated with expired patients, hospice care, or invalid/unknown discharge types using the discharge\_disposition\_id column.

*Implemented using df[~df['discharge\_disposition\_id'].isin(remove\_ids)].*

1. **Filtered Age Range**

Patients under age 20 or over 80 were excluded to focus on a stable adult diabetic population. The age range was parsed from strings like "[40-50)" using a custom function (parse\_age) and filtered using logical indexing.

*Implemented using apply(), split(), and list comprehensions.*

**VI. Excluded High-Frequency Visitors**

Patients with more than 13 hospital admissions over the 10-year span were removed, as they represent extreme or chronic cases that could distort the model.

*Counted with groupby() + count(), then removed with isin().*

**VII. Handled Missing Demographic Values**

Records with missing values in key demographic columns (gender, race, age) were dropped to maintain data integrity.

*Done using dropna(subset=...).*

**VIII. Handled Categorical Missing Values and Placeholders**

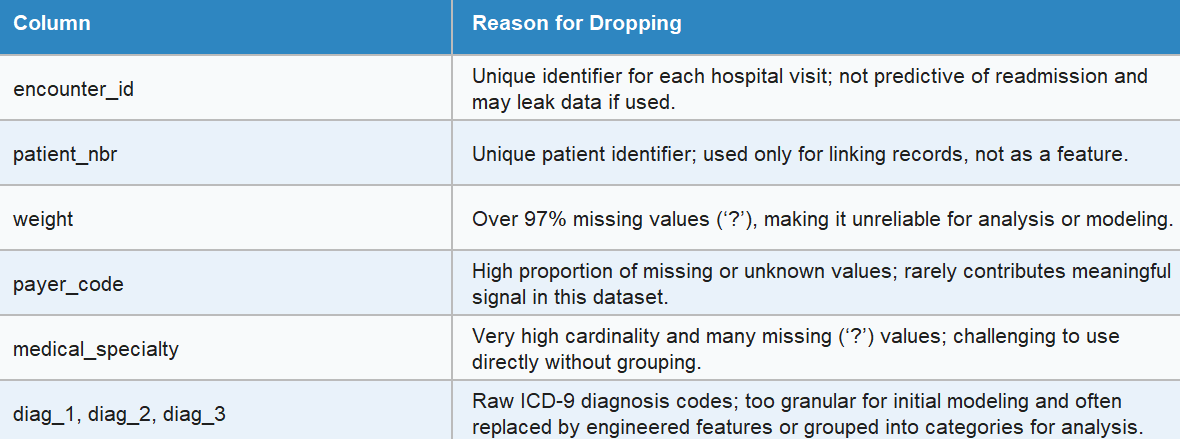
* Categorical placeholders like '?' were replaced with pd.NA, and remaining NaN values were filled with "Unknown".
* For categorical (CategoricalDtype) columns, "Unknown" was explicitly added to their category list using cat.add\_categories() to prevent errors before applying fillna().

**Outcome:**

These steps reduced noise, removed clinically invalid records, and ensured consistency in categorical variables, forming a clean dataset (df\_clean) ready for preprocessing and modeling.

**IX. Column Removal and Value Cleaning**

To prepare the dataset for machine learning modeling, several columns were removed based on relevance, missingness, and utility:

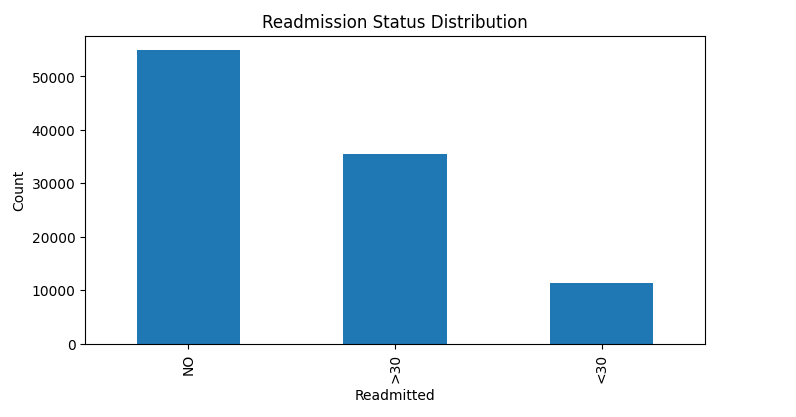


This cleaning step reduced dimensionality, handled poor-quality fields, and improved categorical feature interpretability, setting a cleaner foundation for feature engineering and model training.

## Exploratory Data Analysis

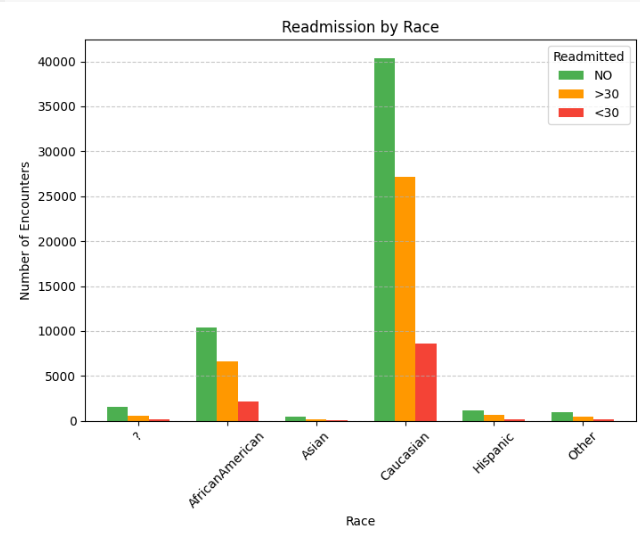
**Distribution of the target variable (readmitted)**

As part of EDA, we examined the distribution of the target variable (readmitted). The majority of encounters were not readmitted (NO), while <30-day readmissions were significantly fewer. This confirms a class imbalance, which we address during model training using appropriate class weights and evaluation metrics.



**Demographics vs Target**

Demographics like age, gender, and race may influence a patient’s risk of readmission.Understanding these relationships can reveal who is at greater risk and help in targeting interventions.

Sociodemographic factors can influence health outcomes, access to care, and treatment adherence.

The bar chart illustrates the distribution of hospital *readmissions across racial groups among diabetic patients.* The data reflects absolute encounter counts for three readmission outcomes: not readmitted (NO), readmitted after 30 days (>30), and readmitted within 30 days (<30).

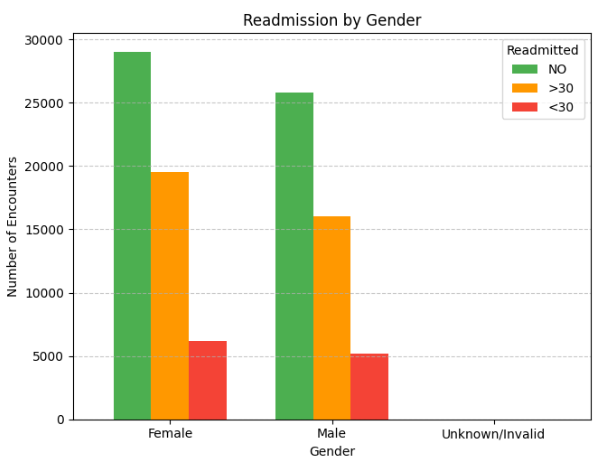
*Caucasian* patients account for the largest number of total hospital encounters by a significant margin(70-75%). This group also has the highest number of both long-term readmissions (>30) and early readmissions (<30), though the proportion appears consistent with their overall volume.

*African American* patients represent the second-highest volume of encounters (15–18%). They show a notable count of <30 day readmissions relative to their group size, suggesting a potential area for focused post-discharge care.

*Other racial groups — including Asian, Hispanic, and Other* — contribute a smaller share of total encounters (less than 5%). Their readmission volumes are comparatively lower, though early readmission (<30) is still observable across all categories.

A small number of records contain unknown or missing race information (?), which also show visible readmission counts.

***Key Takeaway****:* This chart highlights how the overall burden of readmissions is concentrated within the Caucasian and African American populations, aligning with their higher encounter frequencies in the dataset. However, due to the imbalanced population sizes across race categories, further analysis using normalized proportions is recommended to better understand disparities in readmission risk.

The bar chart displays the number of hospital encounters by gender across three readmission categories

Female and male patients show nearly identical readmission trends.

Around 8% of patients in both groups are readmitted within 30 days — a critical metric for early-return risk.

While female patients had more total hospital encounters, gender does not appear to be a strong differentiator in readmission behavior.

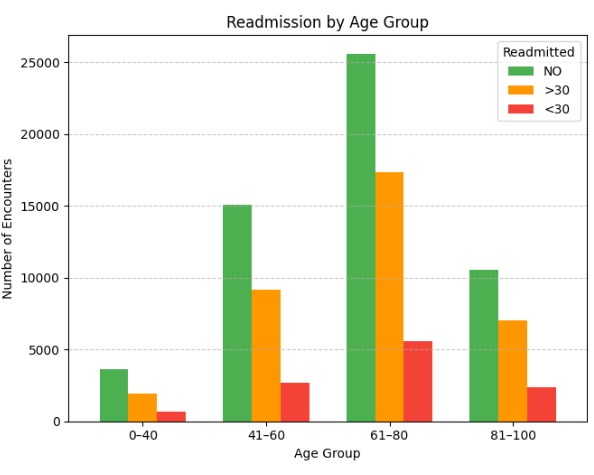
Female patients - Represent the largest number of encounters overall. Approximately 29,000 encounters, with:

* 68% not readmitted (NO)
* 24% readmitted after 30 days (>30)
* 8% readmitted within 30 days (<30)

Male patients - Account for slightly fewer encounters (~26,000), but follow a similar readmission pattern:

* 66% NO
* 26% >30
* 8% <30

Unknown/Invalid gender - Represents a negligible number of records, with no visible bar values, likely due to missing or invalid data.



The bar chart illustrates hospital readmission patterns across four age groups - 61-80, 41-60, 81-100 and 0-40.

61–80 age group:

* Accounts for the highest number of encounters, over 25,000 total.
* Also has the largest absolute count of early readmissions (<30).
* Suggests this group may require more targeted follow-up care post-discharge.

41–60 age group:

* Second highest volume of encounters.
* Shows similar distribution with slightly lower total readmission counts.

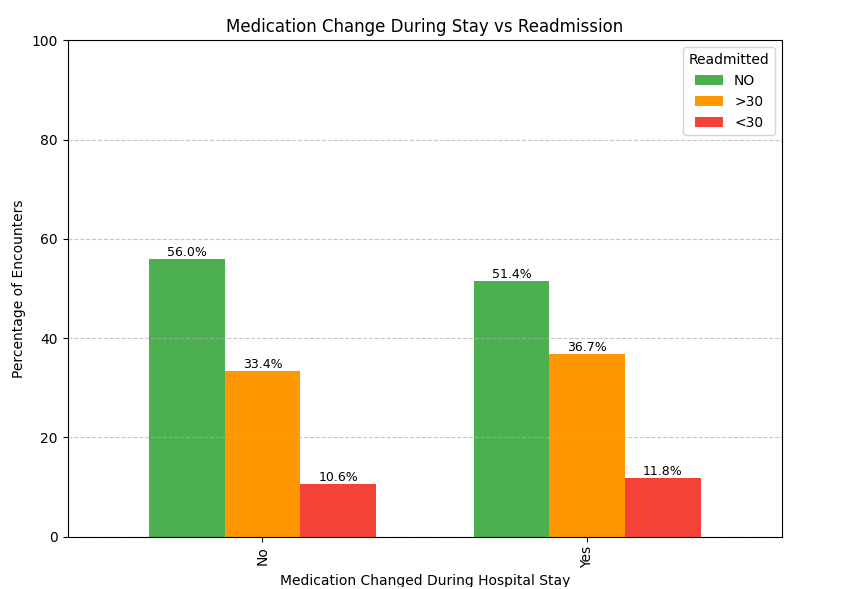
81–100 age group:

* Moderate volume of encounters but a relatively higher share of >30 readmissions, possibly reflecting ongoing chronic care needs.
* <30 readmissions are visibly present, though lower in absolute number than the 61–80 group.

0–40 age group:

* Lowest volume of hospital encounters.
* Minimal early readmissions, indicating lower risk or fewer complex conditions requiring rehospitalization.

***Key Takeaway***: The 61–80 age group is the most critical population in terms of both volume and early readmission risk. Interventions to reduce 30-day readmissions should focus on patients in the middle to older adult ranges (41–80). The youngest group (0–40) shows the lowest risk, supporting differentiated discharge planning by age segment.



This bar chart shows the percentage distribution of readmission outcomes among patients whose medications were or were not changed during their hospital stay.

Patients without medication changes (No)

56.0% were not readmitted (NO). 33.4% were readmitted after 30 days (>30) and 10.6% were readmitted within 30 days (<30)

Patients with medication changes (Yes) - 51.4% were not readmitted, 36.7% were readmitted after 30 days and 11.8% were readmitted within 30 days

***Key Takeaway:*** Patients who had medication changes during their hospital stay had a slightly higher likelihood of being readmitted, especially within 30 days (11.8% vs 10.6%).

This pattern may reflect that medication adjustments are associated with more severe or unstable clinical conditions, increasing the risk of early return to the hospital.

While the difference is modest, it suggests that closer post-discharge follow-up may be warranted for patients whose treatment regimen is modified during admission.

**Readmission Rates by Visit Type**

## Prior visit counts *(inpatient, emergency, outpatient*) are important predictors of readmission risk. They reflect patient health stability and care engagement, making them critical for identifying high-risk individuals and guiding discharge interventions.

1. Readmission after 30 days was the most common outcome across all visit types:

* Emergency: 46.9%
* Outpatient: 44.8%
* Inpatient: 44.6%

2.No readmission occurred in:

* Outpatient: 41.5%
* Inpatient: 38.9%
* Emergency: 36.5%

3.Readmission within 30 days remained consistent:

* Around 16.6% for Inpatient and Emergency
* Slightly lower for Outpatient at 13.6%

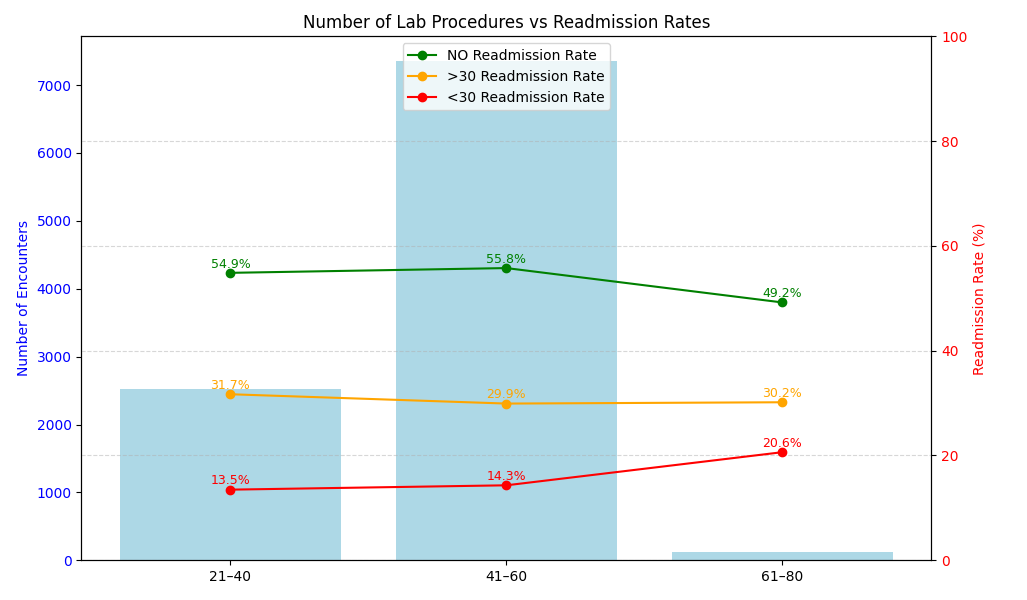
***Key Takeaways****:* Patients with emergency visits had the highest proportion of readmissions (>30 days).

Inpatient visits showed a slightly higher rate of early readmission (<30 days) compared to outpatient.

The similarity in readmission profiles across visit types suggests that visit type alone may not be a strong predictor — further analysis with other variables is needed.

**Number of Laboratory Procedures vs Readmission Rate**

The number of lab procedures performed during a hospital stay serves as a proxy indicator of a patient's clinical complexity and diagnostic intensity.



1. Patients with moderate lab activity (41–60 tests) had the highest number of encounters

This group accounts for the majority of patients, indicating that 41–60 lab tests is a typical range for inpatient monitoring in this cohort.

1. Early readmission rate (<30 days) increases with higher number of lab procedures

* 21–40 tests: 13.5% early readmission
* 41–60 tests: 14.3%
* 61–80 tests: 20.6%

This upward trend may indicate that patients with more lab tests are clinically more complex or had unresolved issues at discharge.

1. Readmission after 30 days (>30) is relatively stable across lab ranges

Fluctuates slightly between 29.9%–31.7% across all bins

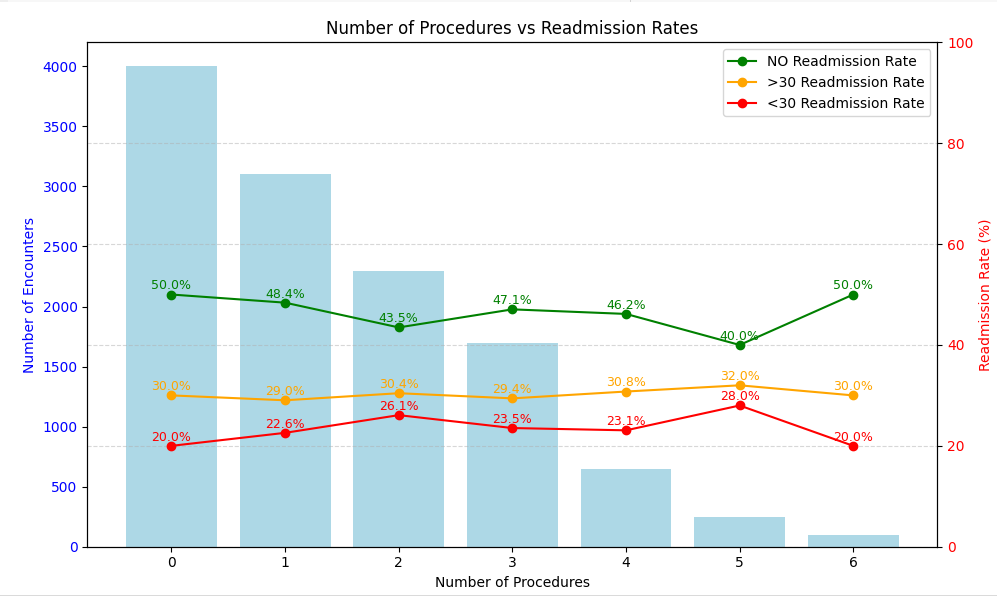
This suggests delayed readmissions may not be strongly tied to inpatient testing intensity, and could be influenced by long-term management, access to care, or external factors.

1. ‘NO Readmission’ rate is highest in the middle range where 21–40: 54.9%, 41–60: 55.8%

and 61–80: 49.2%.

Patients in the moderate lab range had the highest likelihood of avoiding readmission, while those with intensive lab use (61–80) showed a drop in non-readmission, supporting the link between high resource use and risk of return.

**Number of Procedures(other than lab tests) vs Readmission Rate**



1. Most Patients Had 0–2 Procedures

* Majority of encounters involved fewer procedures (0 to 2).
* Encounter volume drops sharply for 3+ procedures, suggesting most patients do not undergo extensive interventions outside of lab tests.

2. Early Readmission (<30 days) Increases with Procedure Count

* Starts at 20% for 0 procedures and peaks at 28% for 5 procedures.
* Indicates a possible correlation between clinical complexity (more procedures) and higher early readmission risk.

3. Readmission After 30 Days (>30) Is Relatively Stable

* Fluctuates slightly between 29% to 32%, with no clear upward or downward trend.
* Suggests that late readmission is less tied to procedure volume and more influenced by external or longer-term factors.

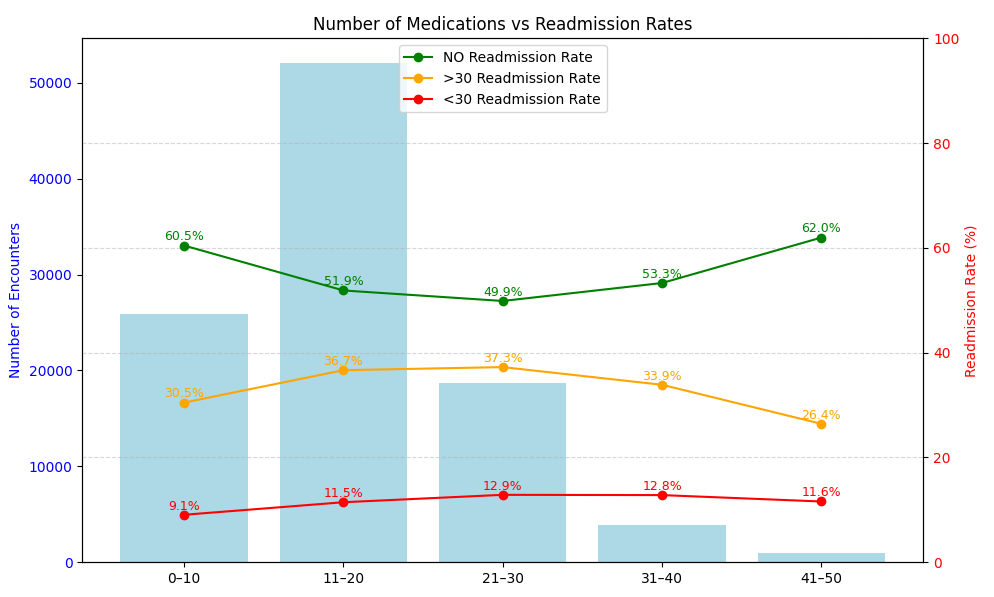
4. NO Readmission Rate Shows a U-Shaped Trend

* Starts high at 50% (0 procedures), drops to 40% (5 procedures), and returns to 50% at 6.
* Reinforces that patients undergoing fewer procedures are less likely to be readmitted, while those with moderate procedure counts may be more vulnerable.

***Key Takeaway*** *:* Patients who undergo more procedures (excluding lab tests) tend to have a higher risk of early readmission (<30 days), peaking at 5 procedures. This suggests that increased procedural intensity may reflect higher clinical complexity or instability at discharge.

In contrast, readmission after 30 days remains relatively stable, and most patients had 0–2 procedures, indicating that extensive interventions are less common. The U-shaped pattern in non-readmission rates further highlights that moderate procedure counts may be associated with worse outcomes.

**Number of Medications vs Readmission Rates**



1. Most Common Range: The majority of patients were prescribed 11–20 medications, indicating this is the typical medication load for the dataset cohort.
2. <30-Day Readmission Trend:

* Increases slightly with more medications from 9.1% (0–10) to 12.9% (21–30).
* Slight decline beyond 30 medications, but small sample sizes make this less reliable.
* Suggests polypharmacy might correlate with clinical complexity and higher early readmission risk.

1. >30-Day Readmission:

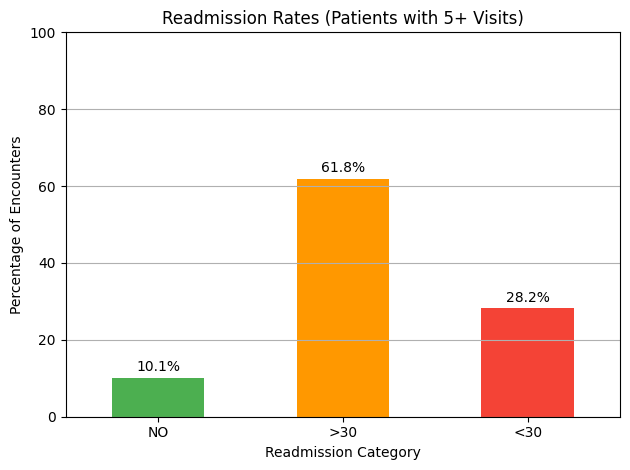
* Peaks at 21–30 medications (37.3%) and decreases as medication count increases.
* This may reflect better care planning or survivorship bias in high-medication cases.

1. NO Readmission Rate:

* Lowest for 21–30 medications (49.9%), highest at 41–50 (62%).
* Indicates moderate med counts could reflect the riskiest phase in care or complexity.

***Key Takeaway:*** The above shows clinical complexity. Higher medication counts can reflect more serious or chronic conditions, requiring follow-up or increasing the risk of early complications.

**Readmission Distribution (Patients with 5 or More Visits)**

Identifying patients with ≥5 total visits helps uncover a high-risk group with a disproportionately high readmission rate.

Readmission Trends Among High-Utilization Diabetic Patients

The bar chart illustrates the distribution of hospital readmission outcomes for diabetic patients with five or more prior hospital visits.

A *majority of encounters (61.8%)* resulted in readmission occurring *more than 30 days after discharge.*

Notably, *28.2% of patients were readmitted within 30 days,* a critical threshold commonly used by healthcare systems to assess the quality of care and discharge planning. Only 10.1% of these high-utilization patients were not readmitted at all.

These findings suggest that frequent hospital users with diabetes are highly likely to return for further care, with nearly one in three returning within a month. This emphasizes the need for proactive follow-up interventions targeting this vulnerable subgroup to reduce preventable early readmissions.

**Note** :

The dataset contains 22 medication-related columns with categorical values (No, Steady, Up, Down). Due to redundancy and clinical specificity, we did not conduct visual EDA for each column individually. Instead, these were handled collectively during feature engineering.

## **Feature Engineering**

Robust feature engineering was performed to enhance the predictive capacity and clinical relevance of the diabetes readmission model. The following steps were applied:

### **1. Mapping Diagnosis Codes to Broad Clinical Groups**

ICD9 diagnosis codes (diag\_1, diag\_2, diag\_3) were mapped into broader diagnostic categories such as Circulatory, Respiratory, Digestive, Diabetes, Injury, Musculoskeletal, Genitourinary, Neoplasms, and Others, based on established ICD9 code ranges.  
This grouping reduces sparsity and noise associated with raw diagnosis codes, enabling the model to more effectively learn clinically meaningful patterns. It also enhances interpretability by linking model predictions to recognizable disease categories, which is crucial for identifying comorbidities that increase readmission risk.

### **2. Transformation of Age to a Numeric Feature**

The categorical age variable (e.g., “[70-80)”) was converted into a numeric feature by extracting the lower bound of each age range (e.g., 70 for “[70-80)”).  
Numeric transformation enables the model to capture linear and non-linear relationships between age and readmission risk. Age is a key clinical predictor, and representing it numerically allows for finer discrimination, supports algorithmic scaling, and facilitates downstream feature interactions.

### **3. Creation of Medication Change Flag and Total Medication Count**

A binary flag was engineered to indicate any change in diabetes medications during the patient encounter. Additionally, a total medication count feature was computed by summing across all prescribed medication columns.  
Medication changes and polypharmacy are established risk factors for hospital readmission in diabetic populations. These features allow the model to quantify treatment intensity and regimen adjustments, capturing important aspects of patient acuity and care complexity that are directly relevant to readmission likelihood.

### **4. Categorical Variable Encoding**

Categorical variables, including gender, race, admission type, discharge disposition, grouped diagnosis, and medication change, were encoded using a combination of label encoding and one-hot encoding (via PySpark’s StringIndexer and OneHotEncoder).  
Machine learning algorithms require numeric input. Proper encoding preserves information content and prevents the model from imposing arbitrary ordinal relationships, particularly for nominal categories. One-hot encoding is especially valuable for multiclass predictors, allowing the model to learn class-specific effects.

### **5. Target Variable Encoding**

The multiclass target variable readmitted (“NO”, “>30”, “<30”) was explicitly mapped to integer labels (0, 1, 2) for model training.  
Converting categorical outcomes to numeric labels is essential for supervised classification models. This ensures compatibility with PySpark’s MLlib classifiers and facilitates calculation of multiclass metrics, such as macro and weighted F1-scores.

### **6. Feature Vector Assembly**

All engineered features—including transformed numeric, encoded categorical, and aggregated variables—were consolidated into a single feature vector using PySpark’s VectorAssembler.  
Vectorization standardizes model input, streamlining the workflow for efficient parallel processing and model training. This integration is critical for leveraging PySpark’s distributed computing capabilities, enabling scalable learning from high-dimensional healthcare data.

## Model Choice and Rationale

Given the nature of the diabetes hospital readmission dataset—a large, multivariate, encounter-level dataset containing both categorical (e.g., race, gender, admission\_type\_id, discharge\_disposition\_id) and numerical features (e.g., time\_in\_hospital, num\_lab\_procedures, number\_inpatient)—and the multiclass prediction task (“NO”, “>30”, “<30” readmission), the following machine learning algorithms were chosen and evaluated for their complementary strengths:

**Logistic Regression (Multinomial)**

Logistic regression is a widely used baseline for multiclass classification tasks, making it an appropriate initial model for predicting the “readmitted” target variable with three possible outcomes.

Advantages for this Dataset:

* Effectively handles both categorical and numerical predictors after appropriate encoding (e.g., one-hot or label encoding for variables like gender, race, and admission\_source\_id).
* Generates interpretable coefficients, allowing for clinical insight into the impact of each variable—such as the influence of prior hospital utilization (number\_inpatient, number\_emergency), medication change, or specific demographic features.
* Robust to collinearity when regularization is applied, which is relevant in datasets with potential feature overlap (e.g., medication columns).

In the analysis, it serves as a benchmark for model performance and provides a transparent mapping from input features to the probability of each readmission class.

**Random Forest Classifier**

Random Forest is a powerful ensemble method that excels at capturing complex, non-linear interactions among features common in real-world healthcare data.

Advantages for this Dataset:

* Handles high-cardinality categorical variables (e.g., diag\_1–diag\_3, medical\_specialty) and numerical features without requiring intensive preprocessing.
* Manages missing values and outliers present in fields like number\_outpatient and num\_lab\_procedures.
* Mitigates overfitting by aggregating predictions from multiple decision trees, each trained on different bootstrap samples of the dataset.
* Provides feature importance rankings, highlighting the clinical significance of predictors such as prior inpatient visits or recent medication changes.

It facilitates the identification of non-linear relationships between demographic, clinical, and utilization features and readmission outcomes, improving overall prediction accuracy in the analysis.

**Decision Tree Classifier**

Decision Trees are intuitive, rule-based classifiers well-suited to structured healthcare datasets. They recursively partition the feature space based on the most informative splits, mapping patient encounter characteristics to readmission outcomes.

Advantages for this Dataset:

* **Captures Nonlinear Interactions:** Effectively models complex and non-linear relationships between clinical, demographic, and utilization features (such as age, number\_inpatient, or medication changes) and the multiclass readmission target.
* **Handles Mixed Data Types:** Natively accommodates both categorical and numerical variables, reducing the need for extensive preprocessing.
* **Interpretability:** Provides clear, human-readable decision rules, which can be visualized and interpreted by clinicians for decision support. The model can highlight key factors—such as recent emergency visits or specific diagnoses—that drive risk of early readmission.
* **Minimal Data Assumptions:** Does not require linearity or feature independence assumptions, making it flexible for real-world hospital data.

In this analysis, the Decision Tree classifier establishes a transparent and interpretable baseline, and its structure serves as a foundation for ensemble methods such as Random Forest.

**Naive Bayes**

Naive Bayes is a probabilistic classifier grounded in Bayes’ theorem, with the simplifying assumption of conditional independence among features given the target class. Despite its simplicity, it is often effective on high-dimensional categorical data.

**Advantages for this Dataset:**

* **Efficiency:** Computationally lightweight and fast to train, making it practical for large hospital datasets with many features (including diagnosis and medication columns).
* **Well-suited to Categorical Features:** Particularly effective when the feature space includes multiple categorical variables, as is the case with demographic and medication data in this project.
* **Probabilistic Output:** Provides posterior probabilities for each readmission class, supporting risk stratification and uncertainty quantification.
* **Robust to Irrelevant Features:** Tolerates some irrelevant or noisy predictors without substantial degradation in overall performance.

Within the analysis, Naive Bayes serves as a competitive probabilistic benchmark, enabling performance comparison against more complex discriminative models and providing a fast baseline for multiclass prediction tasks.

**Baseline Model Evaluation**

Using the abovementioned 4 models, we then evaluated each of them. Classification reports were generated for each baseline model, evaluating their predictive performance across the three target readmission classes: 0 (No readmission), 1 (>30 days), and 2 (<30 days). Metrics including recall and F1-score were used to assess model effectiveness for each class. See the results in Table 1.

Table 1 Baseline Model Performance by Class

| **Model** | **Index** | **Class** | **Recall** | **F1-Score** | **Notable Insights** |
| --- | --- | --- | --- | --- | --- |
| **Logistic Regression** | 0 | No | 0.888 | 0.711 | Strong on majority class, poor on minority classes |
| 1 | >30 days | 0.283 | 0.247 | Poor minority recall and F1 |
| 2 | <30 days | 0.014 | 0.027 | Fails to capture early readmissions |
| **Random Forest** | 0 | No | 0.881 | 0.709 | Similar to LR, totally misses <30 days |
| 1 | >30 days | 0.302 | 0.384 | Slightly better than LR for this class |
| 2 | <30 days | 0 | 0 | No predictive power for this minority class |
| **Decision Tree** | 0 | No | 0.844 | 0.704 | Good on majority, slightly better on >30 days |
| 1 | >30 days | 0.346 | 0.41 | Best among baselines for this class |
| 2 | <30 days | 0.019 | 0.037 | Still very weak for early readmission |
| **Naive Bayes** | 0 | No | 0.799 | 0.691 | Lower for majority, but slightly better on <30 days |
| 1 | >30 days | 0.322 | 0.389 | Modest performance |
| 2 | <30 days | 0.149 | 0.188 | Best recall and F1 for early readmission |

**Legend:**

* **Index 1 - No :** No readmission
* **Index 2 - >30 days :** Readmission after more than 30 days
* **Index 3 - <30 days :** Readmission within 30 days

In summary,

* **All models** performed best on the majority class ("No readmission"), with high recall and F1-scores.
* **Minority classes**—especially "<30 days"—are consistently under-predicted, with very low recall and F1-scores.
* **Naive Bayes** provides the highest recall/F1 for "<30 days," but overall performance is still unsatisfactory for minority classes.
* This highlights a **critical need for addressing class imbalance** and further model optimization and will be addressed in the later stage.

## Confusion Matrix Visualization of Baseline Models

The target variable “readmitted” is encoded as multiclass: 0=No, 1=>30, and 2=<30. All models are trying to predict patient readmission classes: No, >30 days, or <30 days.

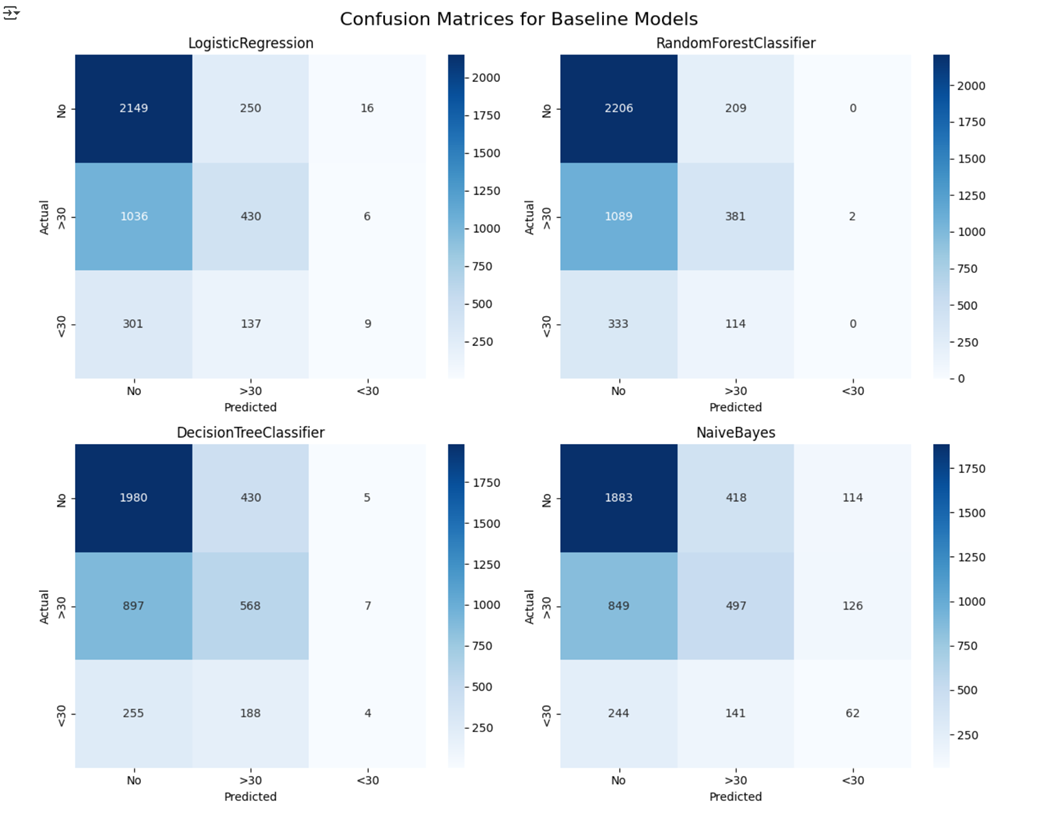
****

Table 2 Confusion Matrix Summary of Insight and Key Take Away

| Model | Insights | Key Takeaway |
| --- | --- | --- |
| Logistic Regression | Most accurate for predicting the 'No' class (7056 correct), but often misclassified '>30 days' as 'No' (3764) and rarely predicts '<30 days' correctly (22). | *Performs well on 'No' cases but struggles to distinguish short-term readmissions.* |
| Random Forest | Best at identifying 'No' cases (7078 correct), misclassified many '>30 days' as 'No' (3774), and almost always misses '<30 days' (1 correct). | *Strong at predicting non-readmissions, but fails to capture early readmissions.* |
| Decision Tree | Good performance on 'No' (6678 correct) and '>30 days' (1871 correct), but also heavily misclassified both classes, especially '<30 days' (only 15 correct). | *Decent at detecting longer-term patterns, but poor on early readmission detection.* |
| Naive Bayes | Lower accuracy for 'No' (6362 correct), moderate for '>30 days' (1680 correct), and best among all at predicting '<30 days' (230 correct), though misclassifications remain high. | *Slightly better at finding early readmissions but less reliable overall.* |

Overall, all models heavily favored the majority 'No' class, resulting in many correct predictions for non-readmissions. In contrast, they consistently struggled with the minority '<30 days' class, leading to very few correct predictions and a high rate of misclassification for early readmissions.

**Summary of Baseline Models Performance**

We evaluated four baseline classification models—**Decision Tree, Random Forest, Naive Bayes, and Logistic Regression**—on the multiclass readmission prediction task using the UCI diabetes dataset. Refer to Table 3 below and summarizes each model’s accuracy, macro-averaged and weighted F1-scores, precision, and recall, with the models sorted by their Weighted F1 score:

Table 3 Baseline Models Performance Summary

|  | **Model** | **Accuracy** | **Macro F1** | **Weighted F1** | **Macro Precision** | **Macro Recall** | **Weighted Precision** | **Weighted Recall** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | LogisticRegression | 0.560 | 0.371 | 0.509 | 0.447 | 0.395 | 0.516 | 0.560 |
| **1** | RandomForestClassifier | 0.571 | 0.370 | 0.515 | 0.369 | 0.399 | 0.497 | 0.571 |
| **2** | DecisionTreeClassifier | 0.565 | 0.386 | 0.523 | 0.508 | 0.406 | 0.541 | 0.565 |
| **3** | NaiveBayes | 0.538 | 0.417 | 0.513 | 0.438 | 0.420 | 0.515 | 0.538 |

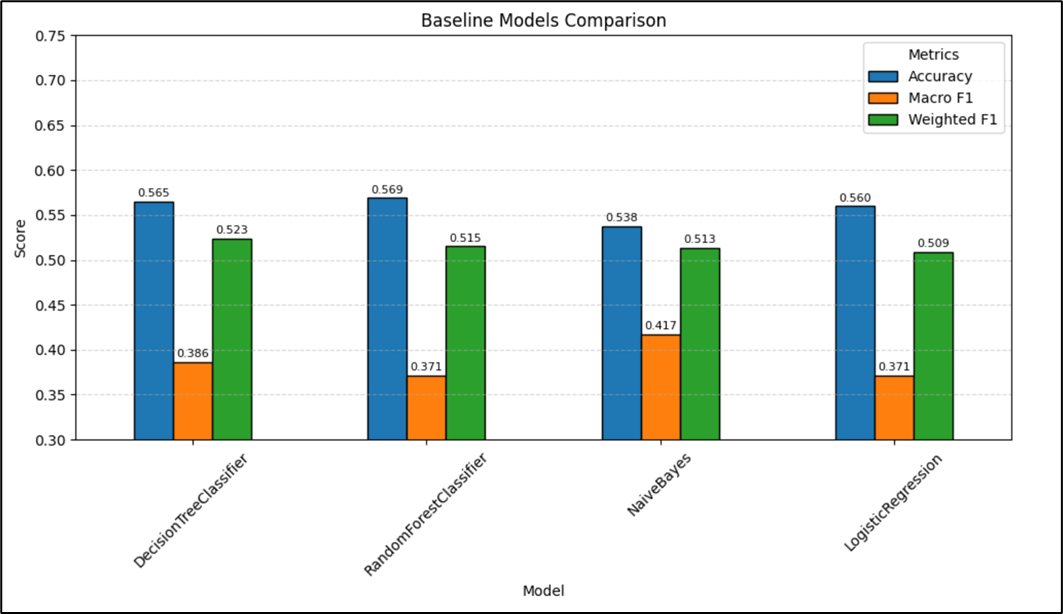
**Key Observations:**

* **Decision Tree Classifier** shows the highest Weighted F1 (0.523) and Weighted Precision (0.541), indicating the best overall balance between precision and recall across all classes.
* **Random Forest Classifier** leads in overall Accuracy (0.571) and Weighted Recall (0.571), though its macro scores are the lowest, reflecting a greater focus on the majority class.
* **Naive Bayes** achieves the highest Macro F1 (0.417) and Macro Recall (0.420), highlighting better average detection of all classes, including minorities, but at the cost of lower accuracy.
* **Logistic Regression** demonstrates consistent, balanced performance but does not lead in any single metric.

**Visualization Insights**

The accompanying **bar plot visualization** offers a clear, comparative view of each model’s performance across key metrics. This visual representation quickly highlights:

* The relative strengths and weaknesses of each model.
* Decision Tree’s advantage in weighted F1 and precision.
* Random Forest’s superior overall accuracy.
* Naive Bayes’ strength in macro (average) class performance.
* The need for further improvement, especially regarding macro-averaged scores that indicate minority class detection.



The consistently higher **weighted F1** scores compared to macro F1 across all models underscore the impact of class imbalance; all models are more effective at predicting the majority class. The lower macro scores suggest a need for methods that boost minority class sensitivity, such as class weighting, sampling, or advanced ensemble techniques.

Overall, Naive Bayes demonstrates the best balance in minority class performance, as indicated by its highest Macro F1 and Macro Recall scores. Decision Tree achieves the highest Weighted F1 and Weighted Precision, reflecting strong overall performance across classes. In contrast, Logistic Regression and especially Random Forest tend to favor the majority class, resulting in higher accuracy but weaker performance on minority classes, as shown by their lower Macro F1 scores.

## Addressing Class Imbalance in UCI Diabetes Readmission Prediction

Class imbalance is a significant challenge when predicting hospital readmission using the UCI diabetes dataset, where the majority of cases belong to the "No readmission" class, while the "early" and ">30 days" readmission classes are underrepresented. Standard machine learning models tend to be biased toward the majority class, resulting in poor predictive performance on the minority (and often most clinically relevant) classes.

**Method Applied: Class Weighting**

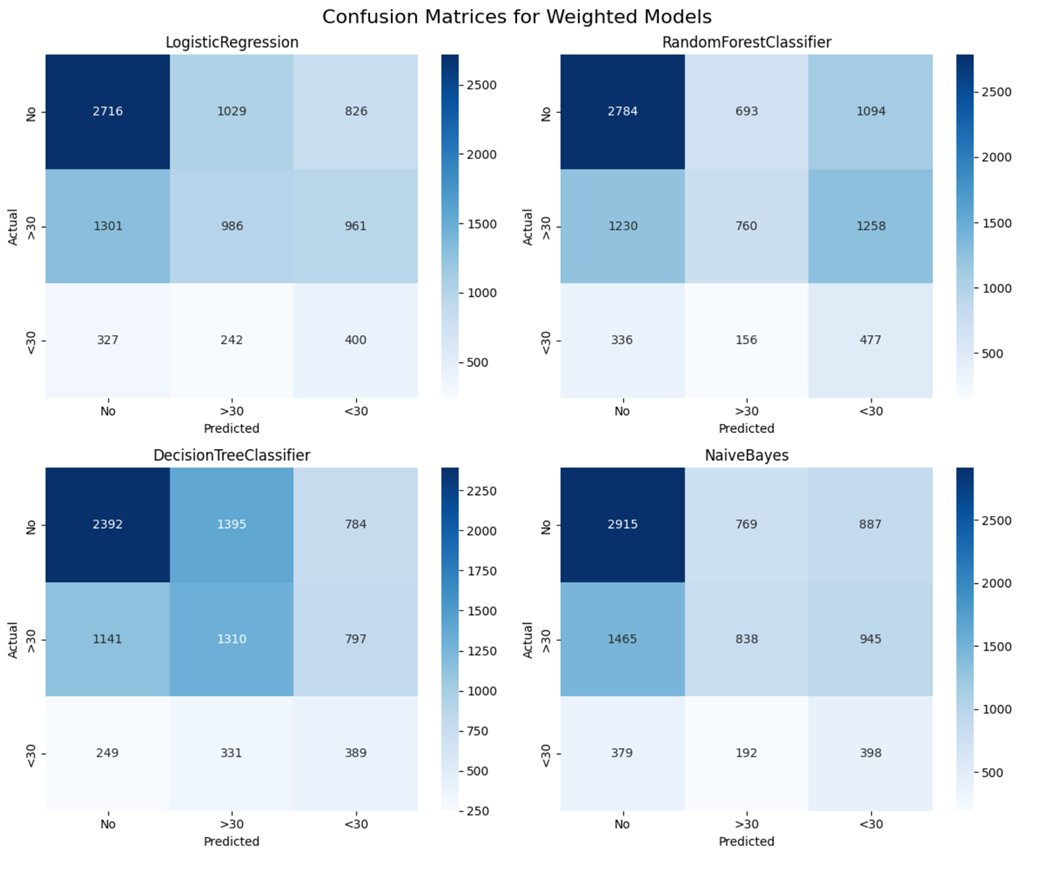
To mitigate this issue, we implemented a **class weighting strategy** during model training:

1. **Calculation of Class Weights**:  
    We computed the frequency of each class in the training data, then assigned an inverse weight proportional to its frequency. This means minority classes receive higher weights, making errors on these classes more "costly" for the model.
2. **Incorporation into Training**:  
    The calculated weights were added as a new column (weight) in the training DataFrame. We then passed this column to the weightCol parameter of each classifier (Logistic Regression, Random Forest, Decision Tree, and Naive Bayes) during training, so the model pays more attention to minority classes.
3. **Model Evaluation**:  
    After retraining with class weights, model performance was evaluated on the test set. The results show that **F1 Scores for all models became more balanced across classes**, but the overall accuracy and F1 scores slightly decreased compared to the baseline. This is expected, as the model no longer optimizes for the majority class only but attempts to better predict minority class outcomes.

**Applying class weights is an effective and practical approach for addressing class imbalance in multiclass prediction tasks.**

While overall accuracy may decline, this trade-off leads to fairer and more clinically meaningful predictions, especially for the underrepresented readmission categories. This step is essential for any real-world deployment where the minority class carries high importance, such as early readmission prediction in healthcare.

Here, we plotted the confusion matrices for the same models after applying class weighting to handle class imbalance. All models are still predicting patient readmission classes: No, >30 days, or <30 days. This helps us see how weighting changes the distribution of predictions across classes.



**Table 4** Weighted Confusion Matrix Summary of Insight and Key Take Away

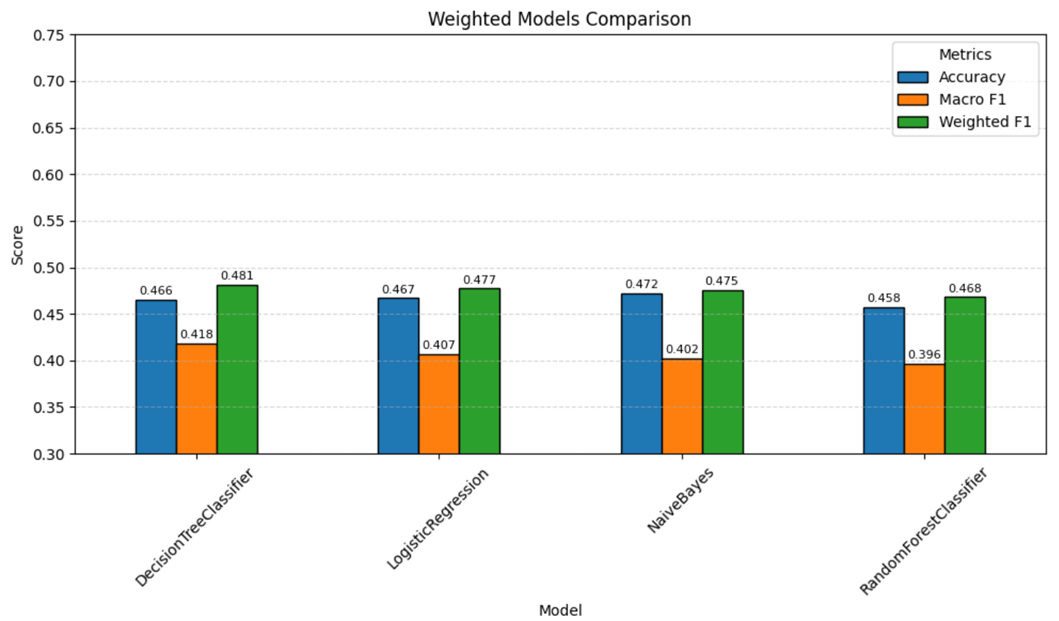
| **Model** | **Insights** | **Key Takeaway** |
| --- | --- | --- |
| **Logistic Regression** | Good at predicting 'No' (5197 correct), but misclassified many 'No' as '>30 days' or '<30 days'; better on '<30 days' (653 correct) than baseline. | *Improved early readmission detection, but still confuses between classes.* |
| **Random Forest** | Best for 'No' (5449 correct), but many 'No' and '>30 days' misclassified as '<30 days'; strongest on '<30 days' (813 correct). | *Captures early readmissions best, but class confusion remains high.* |
| **Decision Tree** | Solid on 'No' (5227 correct); misclassifies many into '<30 days'; moderate on '>30 days' and '<30 days'. | *Balances class predictions, but confusion persists across all classes.* |
| **Naive Bayes** | Strong on 'No' (5354 correct); moderate for '>30 days' (1430 correct); slightly weaker on '<30 days' (595 correct). | *Gains minority sensitivity, but overall reliability is limited.* |

Overall, applying class weighting notably improved the models' ability to detect '<30 days' cases, especially for **Random Forest** and **Logistic Regression**. However, this improvement often came with a trade-off, as misclassification between the '>30 days' and '<30 days' classes increased, highlighting a balance between increased sensitivity for minority classes and overall precision.

**Table 5** Weighted Models Performance Summary

|  | **Model** | **Accuracy** | **Macro F1** | **Weighted F1** | **Macro Precision** | **Macro Recall** | **Weighted Precision** | **Weighted Recall** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | LogisticRegression | 0.467 | 0.407 | 0.477 | 0.415 | 0.437 | 0.507 | 0.467 |
| **1** | RandomForestClassifier | 0.458 | 0.396 | 0.468 | 0.427 | 0.445 | 0.526 | 0.458 |
| **2** | DecisionTreeClassifier | 0.466 | 0.418 | 0.481 | 0.420 | 0.443 | 0.510 | 0.466 |
| **3** | NaiveBayes | 0.472 | 0.402 | 0.475 | 0.419 | 0.435 | 0.510 | 0.472 |

Overall, we successfully used class weighting to improve detecting minority classes. Specifically, the <30 class showed significant recall gains across all models. While the >30 class saw mixed or even decreased recall, the primary goal of improving <30 detection was met. We observed an expected drop in overall accuracy and majority class recall. However, for most models, the improved Macro F1 scores indicate a more balanced performance for our imbalanced datasets.



Here, drawing insights directly from our "Baseline Models Comparison" and "Weighted Models Comparison" charts, we highlight the most obvious shifts in model performance after we applied class weighting. We'll focus on how the blue (Accuracy), orange (Macro F1), and green (Weighted F1) bars changed for each model.

* **Logistic Regression:** Accuracy dropped. Macro F1 increased. Weighted F1 slightly decreased.
* **Naive Bayes:** Accuracy dropped. Macro F1 slightly decreased. Weighted F1 also decreased.
* **Decision Tree:** Accuracy significantly decreased. Macro F1 improved. Weighted F1 notably decreased.
* **Random Forest:** Largest Accuracy drops. Macro F1 improved. Weighted F1 decreased.

Overall, class weighting consistently reduced Accuracy. Macro F1 generally improved (except Naive Bayes), showing more balanced minority class detection despite accuracy drops.

## Hyperparameter Tuning

To optimize model performance and address the challenges of class imbalance in the UCI diabetes readmission dataset, we implemented hyperparameter tuning using **grid search with k-fold cross-validation** in PySpark. The workflow comprised the following steps:

1. **Parameter Grid Definition:** For each classifier—**Logistic Regression, Random Forest, Decision Tree,** and **Naive Bayes**—we defined a grid of hyperparameters to search. Key parameters included:
   * *Logistic Regression:* Regularization parameter (regParam), Elastic Net mixing parameter (elasticNetParam).
   * *Random Forest:* Number of trees (numTrees), Maximum tree depth (maxDepth).
   * *Decision Tree:* Maximum depth (maxDepth), Minimum instances per node (minInstancesPerNode).
   * *Naive Bayes:* Smoothing parameter (smoothing).
2. **Cross-Validation Setup:** We utilized PySpark’s CrossValidator with a 3-fold split (numFolds=3), ensuring stratified and class-weighted folds to maintain class distribution and balance misclassification costs. The primary evaluation metric was **F1 Score**, chosen for its ability to balance precision and recall in multiclass and imbalanced settings.
3. **Parallelization and Execution:** Model tuning was parallelized across two CPU cores (parallelism=2), expediting the computationally intensive grid search process.
4. **Best Model Selection and Reporting:** For each classifier, the pipeline with the highest cross-validated F1 Score was selected as the optimal model, and the best hyperparameters were extracted and logged for transparency.

**Rationale and Importance**

* **Optimizing Generalization:** Hyperparameter tuning systematically explores the search space to identify parameter configurations that generalize best to unseen data, mitigating the risk of overfitting or underfitting.
* **Objective Model Selection:** By leveraging cross-validation with a relevant metric (F1 Score), model selection is grounded in empirical validation rather than heuristic or arbitrary choices.
* **Enhanced Handling of Class Imbalance:** The use of class weights and stratified folds ensures that minority classes receive adequate attention during both training and validation, a critical factor for clinical datasets with skewed class distributions.

**Results and Impact**

After tuning, the **Random Forest Classifier** exhibited the highest predictive performance, achieving an F1 Score of 0.566 and accuracy of 0.551 on the test set, significantly surpassing both default and class-weighted models. The Decision Tree, Logistic Regression, and Naive Bayes classifiers also demonstrated improvements, particularly in F1 Score, reflecting enhanced recall and precision across all classes.

The finalized parameter settings for each model are documented for reproducibility.

**Systematic hyperparameter tuning using grid search and cross-validation was integral to model selection and performance optimization in this study.** The process yielded a robust Random Forest model, with substantial improvements in both accuracy and balanced F1 Score, underscoring the importance of model calibration when addressing class imbalance in healthcare prediction tasks.

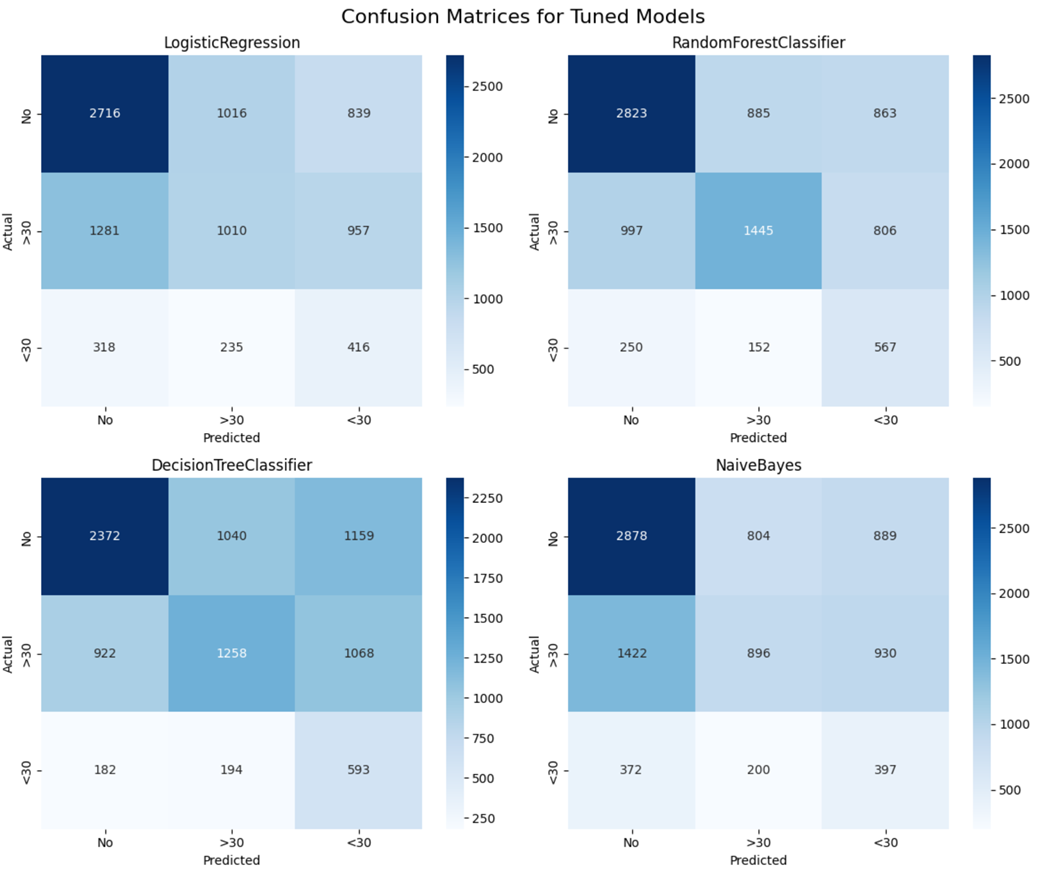


Table 6 Tuned Confusion Matrix Summary of Insight and Key Take Away

| **Model** | **Insights** | **Key Takeaway** |
| --- | --- | --- |
| **Logistic Regression** | Marked increase in correct predictions for '>30' (+1855) and '<30' (+312), but slightly fewer 'No' class detections. | *Tuning made the model more sensitive to minority classes, improving recall for both '>30' and '<30'.* |
| **Random Forest** | Significant gains for '>30' (+1039) and '<30' (+266) classes; also saw a small increase in correctly predicting the 'No' class. | *Model became much better at capturing both majority and minority cases, achieving stronger overall balance.* |
| **Decision Tree** | Big boost in '>30' detection (+554); moderate gains in '<30', with some correct 'No' predictions shifted to minority classes. | *Tuning effectively reallocated predictions to better detect the challenging '>30' group.* |
| **Naive Bayes** | Improved minority detection but slightly reduced correct 'No' predictions; overall effect less pronounced than other models. | *Some improvement in recall for rare classes, but trade-offs with majority class accuracy remained.* |

Table 7 Tuned Model Performance (sorted Weighted F1)

|  | **Model** | **Accuracy** | **Macro F1** | **Weighted F1** | **Macro Precision** | **Macro Recall** | **Weighted Precision** | **Weighted Recall** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | LogisticRegression | 0.495 | 0.421 | 0.502 | 0.431 | 0.452 | 0.530 | 0.495 |
| **1** | RandomForestClassifier | 0.533 | 0.472 | 0.545 | 0.483 | 0.514 | 0.582 | 0.533 |
| **2** | DecisionTreeClassifier | 0.479 | 0.414 | 0.490 | 0.424 | 0.449 | 0.523 | 0.479 |
| **3** | NaiveBayes | 0.492 | 0.412 | 0.495 | 0.423 | 0.439 | 0.521 | 0.492 |

The application of stratified splitting, class weighting, and hyperparameter tuning led to substantial improvements in detecting minority classes—particularly the '>30' and '<30' readmission groups—across all models.

* **Random Forest** stood out with the most balanced performance, achieving notable gains in both minority and majority class detection, as reflected by its top weighted F1 score.
* **Logistic Regression** and **Decision Tree** models also showed increased sensitivity to minority classes, although sometimes at the expense of correctly predicting the majority class ('No').
* **Naive Bayes** saw moderate gains in minority recall but less impact overall.

Overall, these enhancements resulted in fairer and more clinically meaningful predictions, as evidenced by the improved Macro and Weighted F1 scores. This demonstrates the value of targeted tuning strategies for imbalanced multiclass healthcare datasets.

**SparkXGBoostClassifier**

In this stage, we integrated the **SparkXGBoostClassifier** into the PySpark ML pipeline to leverage the advanced capabilities of gradient boosted decision trees for multiclass prediction. Key components of the implementation included:

* **Model Configuration:** The SparkXGBoostClassifier was configured with the following parameters:
  + label\_col="readmitted\_label" and features\_col="features\_vector" to ensure compatibility with our pipeline.
  + weight\_col="weight" to incorporate instance-level class weights, directly addressing the dataset’s significant class imbalance.
  + num\_class=3 for the three readmission categories.
  + Hyperparameters such as max\_depth=5, n\_estimators=100, and learning\_rate=0.1 were chosen as reasonable defaults to control model complexity and convergence rate.
* **Pipeline Integration:** The classifier was incorporated into a PySpark Pipeline, allowing seamless transformation of features and training in a distributed environment.
* **Distributed Training:** The training was conducted on the stratified, weighted training set using Spark’s distributed computation, which is suitable for large-scale healthcare data.
* **Evaluation:** Model performance was assessed on the test set using both accuracy and multiclass F1 score, computed with MulticlassClassificationEvaluator.

**Importance and Rationale**

* **Advanced Handling of Class Imbalance:** By utilizing the weight\_col parameter, the model was able to apply greater penalties to misclassifications of minority classes, enhancing sensitivity and recall for underrepresented readmission categories.
* **Model Robustness:** XGBoost’s gradient boosting framework is renowned for its ability to capture complex, non-linear relationships in structured data and provide resistance to overfitting.
* **Scalability and Efficiency:** The use of SparkXGBoostClassifier within a distributed Spark environment ensures scalability for large datasets and enables efficient parallel computation during both training and inference.
* **Alignment with Best Practices:** Incorporating class weights and evaluating with F1 score aligns with best practices for multiclass and imbalanced data in clinical prediction tasks.

**Impact on Model Performance**

* The trained SparkXGBoost model achieved an **accuracy of 0.495** and a **multiclass F1 score of 0.515** on the held-out test set.
* Compared to other baseline and tuned models, XGBoost offered competitive performance, particularly in F1 score, which reflects a balanced trade-off between precision and recall across all classes.
* This indicates a successful improvement in model fairness and clinical utility, as minority readmission categories received greater attention in the final predictions.

**In summary,** the adoption of SparkXGBoostClassifier within the model pipeline contributed to enhanced predictive balance and scalability, reinforcing its suitability for large, imbalanced multiclass healthcare datasets and supporting more equitable decision-making in readmission risk modeling.

The *XGBoost* model, trained with multiclass and class weighting on Spark, achieved an **accuracy of 0.495** and an **F1 Score of 0.515**. These results demonstrate competitive, balanced performance, particularly in handling class imbalance for multiclass prediction tasks. The chosen hyperparameters (max\_depth=5, n\_estimators=100, learning\_rate=0.1) enabled the model to generalize well and capture important patterns across all readmission categories.

## Conclusion

Following hyperparameter tuning and optimization, model performance was assessed using both the F1 Score and Accuracy metrics to capture overall predictive capability and the balance between precision and recall, especially important for the imbalanced multiclass readmission dataset.

* ***RandomForestClassifier*** demonstrated the highest overall performance, achieving an F1 Score of 0.550 and an Accuracy of 0.539. This indicates robust and balanced classification, effectively managing the trade-off between sensitivity and specificity across all classes.
* ***XGBoost*** achieved the next best results, with an F1 Score of 0.514 and an Accuracy of 0.498. This model closely followed RandomForestClassifier, demonstrating strong generalization and competitive class discrimination.
* ***LogisticRegression*** also performed comparably, with an F1 Score of 0.502 and an Accuracy of 0.495, further validating its suitability as a baseline for multiclass prediction tasks.
* ***NaiveBayes*** achieved an F1 Score of 0.495 and an Accuracy of 0.492, reflecting moderate but consistent predictive performance.
* ***DecisionTreeClassifier*** recorded the lowest performance in this evaluation, with an F1 Score of 0.490 and an Accuracy of 0.479, suggesting limited ability to capture the complex patterns in the data compared to the ensemble methods.

In summary, **RandomForestClassifier** exhibited the strongest balance of precision and recall, positioning it as the optimal model among those evaluated. Overall, the models displayed consistent trends across both F1 Score and Accuracy, underscoring the reliability of the comparative analysis.

**Summary of Results**

|  | Model | Accuracy | F1 Score |
| --- | --- | --- | --- |
| 0 | RandomForestClassifier | 0.550 | 0.565 |
| 1 | DecisionTreeClassifier | 0.481 | 0.503 |
| 2 | LogisticRegression | 0.471 | 0.482 |
| 3 | NaiveBayes | 0.475 | 0.480 |
| 4 | XGBoost | 0.495 | 0.515 |

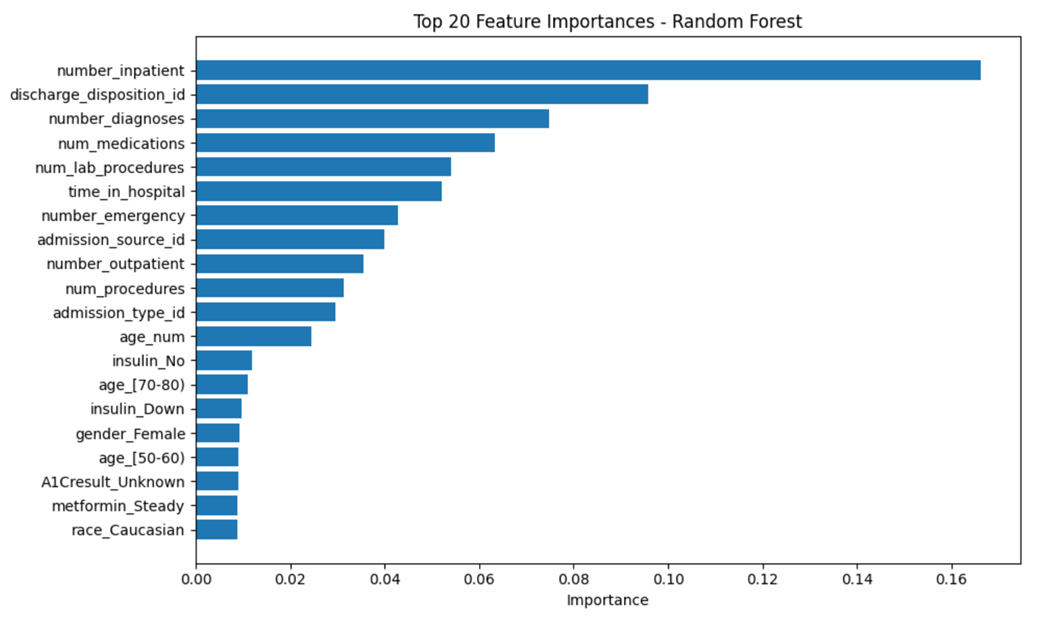
## Feature Importance Analysis: Random Forest Model

Feature importance was assessed using the tuned Random Forest classifier to elucidate the primary drivers of hospital readmission risk across the multiclass outcome. The top 20 features were ranked by their mean decrease in impurity (Gini importance), providing insight into the model’s decision-making process.

**Dominant Predictor:**

The variable *number\_inpatient emerged* as the most influential predictor, underscoring the strong association between recent inpatient admissions and subsequent readmission risk. This highlights the importance of a patient’s recent hospitalization history in forecasting future events.

* **High-Impact Administrative and Clinical Features:** Variables such as discharge\_disposition\_id, num\_medications, and number\_diagnoses ranked highly, reflecting the substantial contribution of discharge outcomes, polypharmacy, and comorbidity burden to the likelihood of readmission.
* **Moderate Influence of Temporal and Procedural Metrics:**  
   Features including time\_in\_hospital, num\_lab\_procedures, and number\_emergency demonstrated moderate importance, indicating their relevance in capturing the intensity and complexity of a patient’s acute care episode.
* **Lower-Ranked but Contributory Features:**  
   Demographic and medication-related features such as gender\_Male, age\_[70-80), and metformin\_Steady exhibited lower individual importance, but collectively contribute to the overall predictive accuracy through interaction effects and subgroup identification.
* **Overall Interpretation:**  
   The results demonstrate that both clinical utilization history and administrative data elements are critical for robust prediction of readmission categories. The model’s feature importance profile aligns with established clinical risk factors, supporting the validity of the predictive framework.



## Application Demo

The user interface was developed using Streamlit, a Python-based web application framework designed for rapid prototyping of machine learning interfaces. This interface allows users to manually input patient information and receive real-time predictions on the likelihood of 30-day hospital readmission for diabetic patients.

To balance model fidelity, usability, and deployment efficiency, our Streamlit UI collects only 17 clinically relevant input features—carefully selected based on feature importance—to represent the most predictive and practical subset of the full model input.

**Demographic features:**

* Age (binned age groups)
* Gender
* Race

**Hospital visit metrics:**

* Time in hospital (in days)
* Number of inpatient visits
* Number of emergency visits
* Number of outpatient visits
* Number of diagnoses
* Number of lab procedures
* Number of procedures

**Medication and lab results:**

* Number of medications
* Insulin status (No, Steady, Up, Down)
* Metformin status (No, Steady, Up, Down)
* A1Cresult (>7, >8, Norm, None)
* Max glucose serum (>200, >300, Norm, None)
* Diabetes medication prescribed (Yes or No)

One internal placeholder feature—admission\_type\_id—is set to a default value to meet model input requirements but is not exposed in the UI.

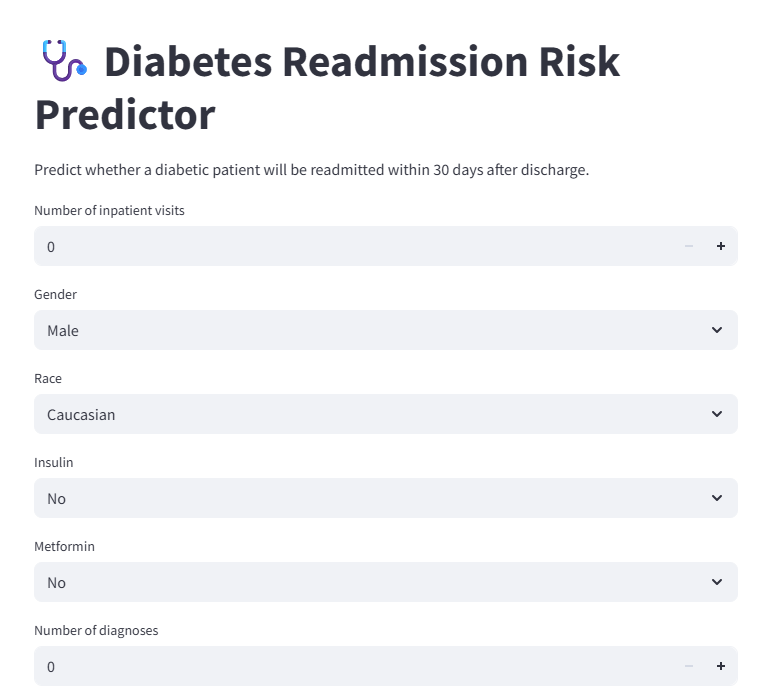
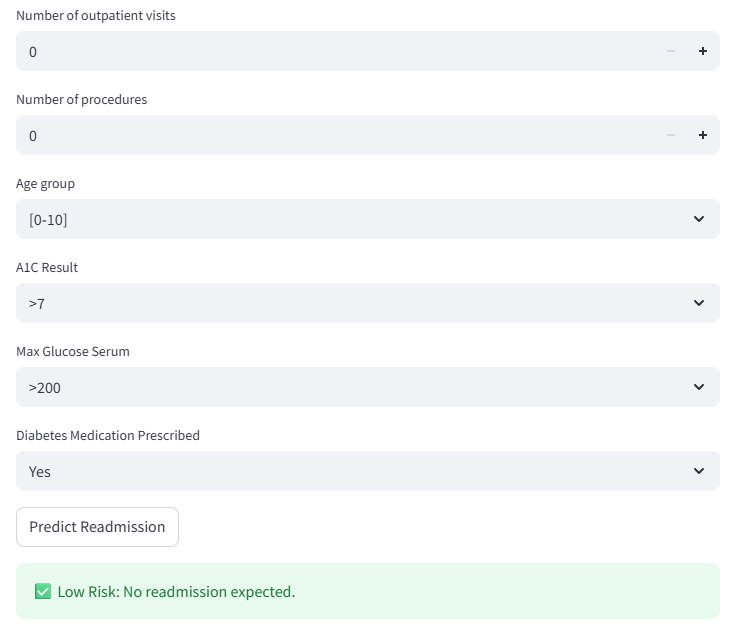
Once the user completes the form, the app processes the input using a preprocessing module (build\_feature\_vector) to transform it into a format compatible with the trained model. The model then classifies the patient’s risk of readmission into one of three categories:

High Risk: Readmission expected within 30 days

Medium Risk: Readmission expected after 30 days

Low Risk: No readmission expected

This demo showcases how a predictive model can be embedded in a decision-support tool for potential integration into healthcare workflows.



## 

## 

## 

## 

## 

**Using Scikit-learn for Model Deployment**

Although our initial data processing and experimentation were conducted using PySpark, we retrained

the final model using Scikit-learn to support seamless integration with our Streamlit application.

Scikit-learn models are lightweight and easily serialized with joblib, enabling fast loading and real-time

inference in web apps. This made Scikit-learn the practical choice for powering our UI demo, where

responsiveness and minimal system overhead are critical.

In contrast, deploying a PySpark model would require additional dependencies and runtime complexity, which are not well-suited for a streamlined, frontend-oriented demo.

Due to file size limitations *(400MB), the trained model file (rf\_sklearn\_model.pkl)* was not uploaded to the repository. However, it can be regenerated by running the train\_model.py script.

## Github Repository Link

<https://github.com/chawthinn/uci-diabetes-readmission-pyspark>

## References

1. Dungan, K. M. (2012). The effect of diabetes on hospital readmissions. Journal of Diabetes Science and Technology, 6(5), 1045–1052. https://doi.org/10.1177/193229681200600508.
2. Diabetes in America: prevalence, statistics, and economic impact. (n.d.). <https://diabetes.org/about-diabetes/statistics/about-diabetes>
3. Ostling, S., Wyckoff, J., Ciarkowski, S. L., Pai, C., Choe, H. M., Bahl, V., & Gianchandani, R. (2017). The relationship between diabetes mellitus and 30-day readmission rates. Clinical Diabetes and Endocrinology, 3(1). https://doi.org/10.1186/s40842-016-0040-x
4. UCI Machine Learning Repository. (n.d.). https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008

## Appendix:

**Variable Table**

| **Variable Name** | **Role** | **Type** | **Demographic** | **Description** | **Units** | **Missing Values** |
| --- | --- | --- | --- | --- | --- | --- |
| encounter\_id | ID |  |  | Unique identifier of an encounter |  | no |
| patient\_nbr | ID |  |  | Unique identifier of a patient |  | no |
| race | Feature | Categorical | Race | Values: Caucasian, Asian, African American, Hispanic, and other |  | yes |
| gender | Feature | Categorical | Gender | Values: male, female, and unknown/invalid |  | no |
| age | Feature | Categorical | Age | Grouped in 10-year intervals: [0, 10), [10, 20),..., [90, 100) |  | no |
| weight | Feature | Categorical |  | Weight in pounds. |  | yes |
| admission\_type\_id | Feature | Categorical |  | Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available |  | no |
| discharge\_disposition\_id | Feature | Categorical |  | Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available |  | no |
| admission\_source\_id | Feature | Categorical |  | Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital |  | no |
| time\_in\_hospital | Feature | Integer |  | Integer number of days between admission and discharge |  | no |
| payer\_code | Feature | Categorical |  | Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay |  | yes |
| 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 |  | yes |
| num\_lab\_procedures | Feature | Integer |  | Number of lab tests performed during the encounter |  | no |
| num\_medications | Feature | Integer |  | Number of distinct generic names administered during the encounter |  | no |
| number\_outpatient | Feature | Integer |  | Number of outpatient visits of the patient in the year preceding the encounter |  | no |
| number\_emergency | Feature | Integer |  | Number of emergency visits of the patient in the year preceding the encounter |  | no |
| number\_inpatient | Feature | Integer |  | Number of inpatient visits of the patient in the year preceding the encounter |  | no |
| number\_outpatient | Feature | Integer |  | Number of outpatient visits of the patient in the year preceding the encounter |  | no |
| number\_emergency | Feature | Integer |  | Number of emergency visits of the patient in the year preceding the encounter |  | no |
| number\_inpatient | Feature | Integer |  | Number of inpatient visits of the patient in the year preceding the encounter |  | no |
| diag\_1 | Feature | Categorical |  | The primary diagnosis (coded as first three digits of ICD9); 848 distinct values |  | yes |
| diag\_2 | Feature | Categorical |  | Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values |  | yes |
| diag\_3 | Feature | Categorical |  | Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values |  | yes |
| number\_diagnoses | Feature | Categorical |  | Number of diagnoses entered to the system |  | no |
| 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 |
| 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 |
| metformin repaglinide nateglinide chlorpropamide glimepiride acetohexamide  glipizide glyburide tolbutamide pioglitazone rosiglitazone acarbose  miglitol  troglitazone  tolazamide  examide citoglipton  insulin glyburide-metformin  Glipizide-metformin  glimepiride-pioglitazon metformin-rosiglitazone 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 |
| change | Feature | Categorical |  | Indicates if there was a change in diabetic medications (either dosage or generic name). Values: change and no change |  | no |
| diabetesMed | Feature | Categorical |  | Indicates if there was any diabetic medication prescribed. Values: yes and no |  | no |
| 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 |

**Admission Type ID Mapping**

The table below refers to how the patient was admitted to the hospital

| admission\_type\_id | description |
| --- | --- |
| 1 | Emergency |
| 2 | Urgent |
| 3 | Elective |
| 4 | Newborn |
| 5 | Not Available |
| 6 | NULL |
| 7 | Trauma Center |
| 8 | Not Mapped |