Predictive Modeling for Hospital Readmission

1 Introduction

1.1 Background

Diabetes requires careful monitoring during hospitalization to reduce readmission risks. This study predicts whether diabetic patients may return within 30 days post-discharge. The dataset covers 10 years (1999-2008) from 130 US hospitals, focusing on diabetic diagnoses. Accurate readmission predictions promise significant improvements for these patients.

1.2 Dataset Overview

The dataset used in this study covers ten years (1999-2008) and includes information from 130 hospitals and healthcare networks in the US. The dataset encompasses 50 features with 101,766 samples, consisting of both categorical and continuous variables. It includes patient identifiers, demographics, medical histories, medications, and diagnostic codes (ICD9). Categorical features entail patient characteristics like race, gender, and medical procedures, while continuous features cover numerical attributes like time spent in the hospital, lab procedures, and medications. Each entry in the dataset relates to hospital records of patients diagnosed with diabetes. The main aim is to figure out if these patients might need to go back to the hospital within 30 days after leaving.

Table 1. Small snippet of actual data

encounter_id	patient_nbr	race	gender	age	medical_specialty	diag_1	readmitted
2278392	8222157	Caucasian	Female	[0-10)	Pediatrics-Endocrinology	250.83	NO
149190	55629189	Caucasian	Female	[10-20)	?	276	>30

1.2.1 Readmission categorization

In the dataset, there are three distinct classes used for readmission categorization:

- '<30' indicates patients who were readmitted in less than 30 days.
- '>30' represents patients who were readmitted in more than 30 days.
- 'No' signifies no record of readmission.

2 Exploratory Data Analysis

Initial analysis identified categorical and continuous features, revealing a high incidence of missing 'weight' values. Further examination through violin plots illuminated distributions across significant continuous variables, uncovering insights into patient stay durations, lab procedures, and medication frequencies.

2.1 Medication Counts Across Age Groups by Readmission

Using a grouped bar chart, this analysis showcases medication distributions among different age groups, categorized by readmission status. The chart illustrates two bars per age group: one for readmitted patients and the other for non-readmitted. Across all ages, a consistent trend emerges—readmitted patients generally have a higher medication count compared to those not readmitted. This pattern suggests a possible link between increased medications and higher readmission likelihood, cutting across various age groups.

2.2 Imbalanced Classes and Data Insights:

Preliminary exploration involved assessing imbalanced classes. Moreover, violin plots elucidated distributions of major continuous variables like 'admission_type_id', 'time_in_hospital', 'num_lab_procedures', and others, revealing their respective distributions and potential outliers.

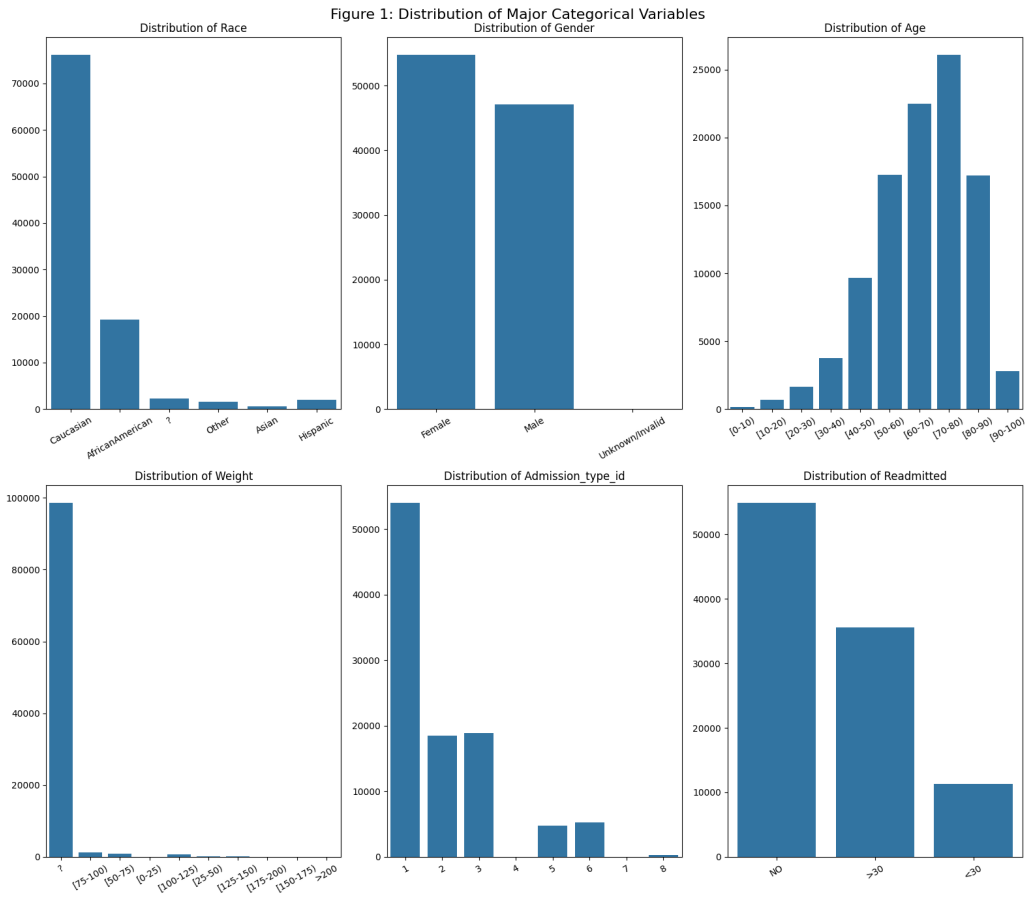

Figure 1. Distribution of Major Categorical Variables

3 Data Cleaning and Pre-processing

Data preparation involved dropping columns with substantial missing values `'weight'`, `'payer_code'`. Encoding categorical variables e.g. `'diag_1'`, `'diag_2'`, `'diag_3'` and scaling continuous variables were executed using pipelines and column transformers. We use StandardScaler to scale continuous variables which standardizes features by removing the mean and scaling to unit variance. This is commonly done to normalize features before applying machine learning algorithms. For categorical variables we handle missing values by imputing the most frequent value, then apply one-hot encoding to convert the categoricals into binary indicator columns. For some variables we use ordinal encoding which encodes the categories to integer values representing a meaningful order if one exists.

4 Machine Learning Model Evaluation and Hyperparameter Tuning

4.1 Model Selection and Performance Metric Justification

We use the F1 score, which specifically helps track the number of false negatives in our overall predictions by considering the number of true positives. The F1 score is a combined measure of Precision and Recall, both of which are metrics for monitoring the ratio of true positives concerning total positives in predictions and actual true positives, respectively.
We run our analysis comparing three models:

- K Nearest Neighbors Classifier
- Logistic Regression Classifier
- Random Forest Classifier

Hyperparameter tuning via GridSearchCV aimed to optimize model parameters for improved predictive performance. Cross-validation strategies (Stratified K-Fold) provided robust estimations across different k-fold values.

4.2 Model Performance Analysis

We summarize performance using two different cross-validation strategies for hyperparameter tuning and evaluation. The first table shows 3-fold cross-validated results. The random forest classifier achieved the best F1 score of 0.623, while logistic regression and KNN lag slightly behind at 0.603 and 0.588 F1. Increasing to 7-fold cross-validation, the second table shows logistic regression achieving a slightly higher F1 score of 0.615 compared to 0.607 for random forest and 0.575 for KNN. Precision, recall, and accuracy metrics follow similar patterns. Overall, random forest and logistic regression exhibit strong and comparable performance, with random forest doing better under 3-fold CV while logistic regression prevails under 7-fold. The KNN lags noticeably across metrics and CV strategies. These findings will help guide final model selection and evaluation on hold-out test data.

Table 2. Results for 3-fold CV

Fold	Model	precision	recall	f1	accuracy
3	RandomForestClassifier	0.637933	0.617065	0.622894	0.617065
3	LogisticRegression	0.610920	0.599443	0.602998	0.599443
3	KNeighborsClassifier	0.596609	0.584442	0.588279	0.584442

Table 3. Results for 7-fold CV

Fold	Model	precision	recall	f1	accuracy
7	LogisticRegression	0.628441	0.609499	0.614927	0.609499
7	RandomForestClassifier	0.614939	0.603472	0.606999	0.603472
7	KNeighborsClassifier	0.579792	0.572028	0.574666	0.572028

Project Link: https://github.com/14Richa/Patient-Readmission-Analysis