Hospital Patients Readmission Prediction using Logistic Regression ML Model

Authored by Yuvaraj C U – yuvarajcu@gmail.com

Introduction

Hospital readmissions are a significant challenge for healthcare systems, often associated with high costs. The task focuses on predicting whether a patient will be readmitted within 30 days based on demographic and clinical data. The dataset includes features like age, medical specialty, number of procedures, diagnosis codes, and medication details. **Logistic Regression** was selected as the primary model due to its simplicity and strong performance in binary classification tasks, also as part of the assignment.

Data:

The dataset, sourced from kaggle, contains no missing values, and the target variable (readmission status) is well-balanced. Further details and analysis are available in the accompanying notebook.

Exploratory Data Analysis:

Though optional, EDA was conducted to better understand the data. Graphs and correlations were explored, and insights are documented in the notebook.

Data Preprocessing:

The ***'age' variable***, stored as an object, was ordinally encoded, while binary encoding was applied to the ***'readmission', 'change', and 'diabetes_med'*** columns. Other categorical features were label-encoded to reduce feature complexity. The target variable (readmission) was assigned to ***'y'***, and relevant features to ***'X'***, excluding variables with low correlation or relevance.

Model Selection

As a part of the task, we initially start off with the logistic regression, which has better interpretability and efficiency in binary classification problems. The first variant of this model had just the max iterations set to standard value of 1000. We then added optimization techniques, one using standard scaling and other using GridSearchCV feature which produced - Best parameters: {'logreg__C': 0.01, 'logreg__solver': 'liblinear'} and Best cross-validation accuracy: 0.6076571428571429
Despite trying to utilize these tuning and optimization techniques the classification report for all three were similar to the following, which can also be referred to in the notebook.

	precision	recall	f1-score	support
0	0.60	0.79	0.68	4000
1	0.62	0.40	0.49	3500
accuracy			0.61	7500
macro avg	0.61	0.59	0.59	7500
weighted avg	0.61	0.61	0.59	7500

We then explored **Random Forest Regression**, which showed improvements in **recall**, the most critical metric in this healthcare scenario. Recall ensures that more true readmissions are identified, allowing preventive measures to be applied.

	precision	recall	f1-score	support
0	0.62	0.69	0.65	4000
1	0.59	0.52	0.55	3500
accuracy			0.61	7500
macro avg	0.60	0.60	0.60	7500
weighted avg	0.61	0.61	0.60	7500

Performance Metrics

Given the goal of reducing patient readmissions, **recall** was prioritized over precision. Recall captures the proportion of actual readmissions correctly predicted. While precision is secondary, the slight trade-off in Random Forest was acceptable given the improved recall, meaning it was able to correctly identify more patients who were likely to be readmitted. However, this came at the cost of a slight drop in precision, which is acceptable given our goal.

Theoretical Explanation of the Chosen Model

The chosen model, **Logistic Regression**, is based on the principle of **log-odds**. It assumes a linear relationship between the independent variables (features) and the log-odds of the dependent variable (readmission). The mathematical formulation is as follows:

$$y^{\{hat\}} = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n)}}$$

Where:

- y^ is the probability that a patient will be readmitted.
- b0 is the intercept (bias).
- b1, b2 ,...,bn are the coefficients learned during training that represent the contribution of each feature x1, x2 ,....,xn to the log-odds of readmission.

The output of this formula is a probability score, which is thresholded (typically at 0.5) to classify patients as readmitted or not. This formulation was implemented manually using Python's matrix operations and sigmoid function, allowing us to calculate the probabilities and make predictions.

Random Forest, in contrast and in short, uses a collection of decision trees to generate predictions. Each tree makes a decision based on a subset of features, and the final prediction is the average of all trees. The advantage of Random Forest is its ability to model non-linear interactions, but it does not provide a straightforward mathematical formula like Logistic Regression.