Predicting Diabetes Readmission

Aysha Allahverdiyeva

Brown University, Data Science Institute (DSI)

https://github.com/aysha-alv/Predicting-Diabetes-Readmission/tree/main

---

Introduction:

This project aims to tackle issues surrounding ambiguity of readmission for diabetes mellitus. Currently, diabetes mellitus (DM) readmission rates are around 26%, 30 day rehospitalization rates are between 14.4-22.7%. These kinds of readmissions are attributed to one or more of the following:

- Hyperglycemia or hypoglycemia
- Onset/worsening of diabetes-related symptoms
- Inability to adhere to care recommendations (often due to factors outside of the patient's control)
- Lack of understanding of treatment
- Inaccurate/incomplete care instructions

(Ostling et al., 2017). With this said, detangling the ambiguity surrounding readmission is a significant step in lowering readmission rates. To accomplish this, it is important to understand demographic and medical history patterns to predict who is at risk for readmission so that specialized treatment can be delivered to them preemptively. This kind of exploration could mitigate issues contributing to readmission and could reduce the number of rehospitalizations.

The data used is from a research paper entitled "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records" by Strack et al. (Strack et al., 2014). Researchers used the Health Facts Database, containing compilations of clinical records across the US, and preprocessed the raw data.

Given the dataset has been preprocessed by the authors, it is tailored to heavily rely on medication use patterns influencing blood sugar. I believe this means a classification model built off this data would be similarly limited. Factors such as lifestyle choices and body composition are not included in the feature space, which likely undermines the models performance. Strack el al. claimed that there is a

relationship between probabilities of readmission and the HbA1c measurement depending on the primary diagnosis. In contrast, this exploration aims to solely focus on the readmission variable to attempt to correctly classify new observations into one of three classes: Not readmitted (No), readmitted for less than 30 days (<30), and readmitted for over 30 days (>30). In implementing Machine Learning (ML) techniques similar to those described in this report, data practitioners should expect above-baseline classification scores, keeping in mind limitations put forth by the available data.

Exploratory Data Analysis:

When conducting exploratory data analysis, I investigated the demographic and medical history features in relation to the target variable, encoded as "readmitted" with the 3 aforementioned classes. When conducting general exploratory analysis, I found relationships between the target variable and age, medical history, and time spent in hospital.

When investigating a breakdown of age groups by readmission status (figure 1), I created a stacked bar plot with proportions, rather than counts, of instances in each age group. Surprisingly, there were more >30 day readmissions across the board, implying more gravity to the DM condition than I initially anticipated. Otherwise, the data behaved as expected with lower readmission rates for younger ages and a normal-seeming distribution otherwise - mimicking the general distribution of the age data in the dataset.

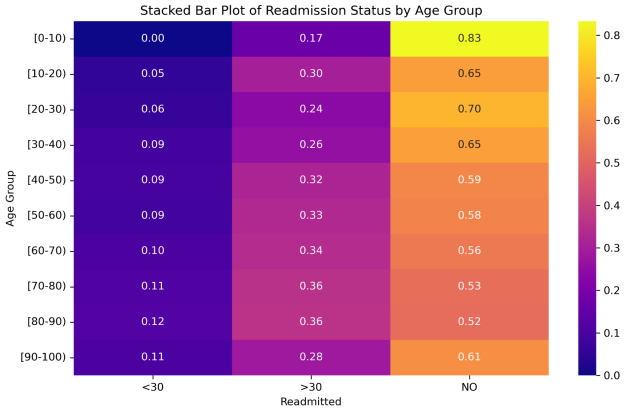


Figure 1

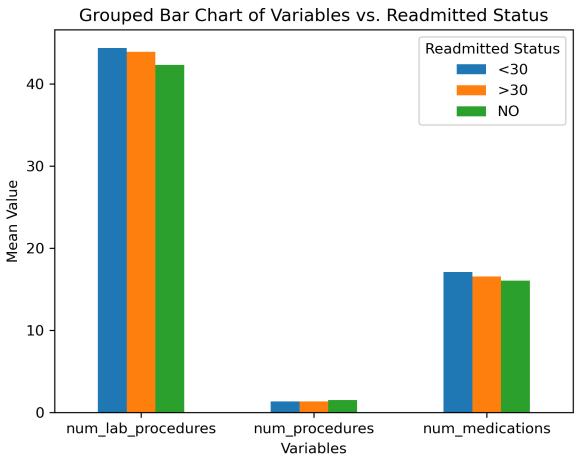


Figure 2

When exploring medical history variables like number of lab procedures, general procedures, and medications (figure 2) the data revealed that the numbers of lab procedures were generally high - which makes sense since these are all diabetes patients and must have had some testing done to determine the preliminary DM diagnosis. The number of medications was relatively high for similar reasons. The number of procedures was very low for all classes of readmission which I found to be surprising. Those with <30 day readmissions had the highest numbers for all procedures,

which surprised me since I assumed longer readmissions require more hands-on medical treatment.

When exploring the number of days previously spent in the hospital, the box-and-whisker distribution (figure 3) indicated similar maximum and minimum values across readmission statuses - however, the median number of days was higher for readmitted patients. This exploratory data analysis provided insight into the integrity of the data, affirming relationships that I assumed had existed before beginning the preprocessing and algorithm selection.

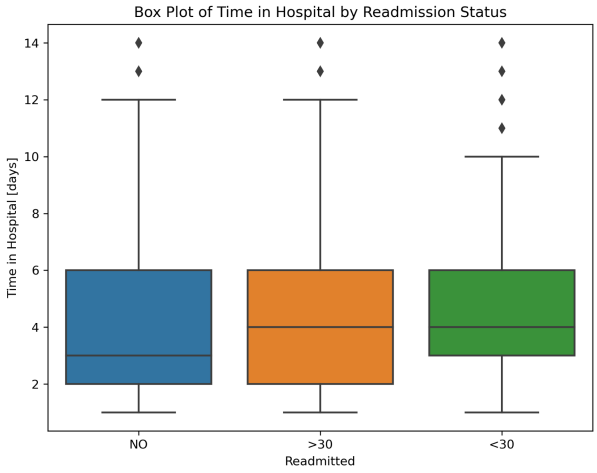


Figure 3

Methods:

I split the data using a stratified basic split (train_test_split). This strategy was most appropriate since the target variable was unbalanced (figure 4), so stratifying on that variable seemed appropriate to distribute the classes equally among training, testing, and validation data. In preprocessing the data, I used a StandardScaler for any numeric features such as time spent in hospital, a OneHot encoder for categorical features - which made up most of the feature space - and the OrdinalEncoder for ordered categorical features such as age and weight ranges. Missing features were transformed into an "Other" category or removed entirely from the dataset, depending on the percentage of missing values per feature. Diag_1, 2 and 3 features were difficult to preprocess with >2500 unique values, so these were collapsed into the 20 most popular values, and the rest grouped into an "Other" category.

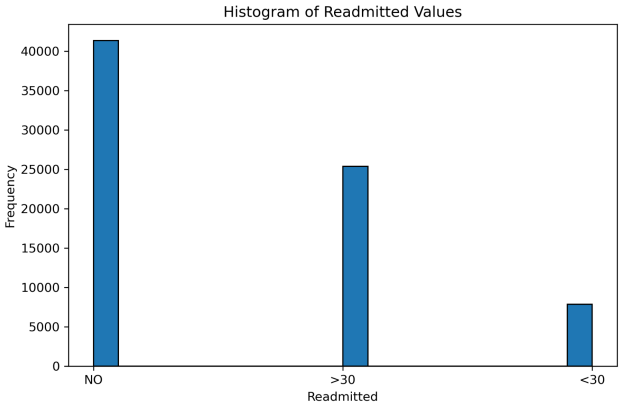


Figure 4

In defining the ML pipeline, I implemented a function using GridSearchCV through which I could input different algorithms (both linear and nonlinear) and parameter grids before selecting the most effective one based on mean performance across random states. This function also extracted the best metric and parameters for every model's run. I used a weighted F1 score as the evaluation metric for all my models as weighted averaging would take into account the frequencies of every class letting

each one contribute equally to the overall score, which is crucial for imbalanced data. Problems in medical diagnosis also yield very different consequences, with false positives costing patients thousands in treatment and false negatives meaning that a diagnosis is missed and patients don't get the care they need. With that said, the weighted F score metric balances recall and precision, capturing the nuances of a model's performance with tasks as sensitive as this one. Expanding on the models I used, the table below summarizes the parameters that were fed into the function, and ones outputted as the optimal ones.

ML Algorithm	**Param Grid**	**Best Params**
LogisticRegression	logreg_C: np.logspace(-3, 3, 7) Logreg_penalty: l2	logreg__C: 1.0, logreg__penalty: 'l2'
RandomForest	class_weight: [None, 'balanced'] n_estimators': [100, 300, 500], max_depth': [None, 30, 50], min_samples_split': [2, 10, 20], min_samples_leaf': [1, 4, 6], max_features': ['sqrt', 'log2', None]	class_weight: 'balanced', max_depth: None, max_features: 'sqrt', min_samples_leaf: 1, min_samples_split: 20, n_estimators: 500
KNearestNeighbors	n_neighbors: [3, 5, 7, 10], weights: ['uniform', 'distance'], metric: ['euclidean', 'manhattan']	metric: 'euclidean', n_neighbors: 10, weights: 'uniform'
GradientBoost	n_estimators: [100, 200] learning_rate: [0.01, 0.1] max_depth: [3, 5] min_samples_split: [2, 4] min_samples_leaf: [1, 2]	learning_rate: 0.1, max_depth: 5, min_samples_leaf: 2, min_samples_split: 2, n_estimators: 200
SVC	C: [1, 10], kernel: ['rbf'], gamma: ['scale']	C: 10, gamma: 'scale', kernel: 'rbf'

In testing different parameter grids to optimize performance, I used a subset of my data and pivoted the parameter ranges based on existing discourse, but found that the results were not as promising as when I expanded the range of the parameter values without compromising computational efficiency too much. Capturing a larger range allowed for better performance overall. Trying a large range of regularization values settled a balanced class weight - speaking to the imbalance of data, and the other tree-oriented parameters allow for maximum exploration without overbearing computational power.

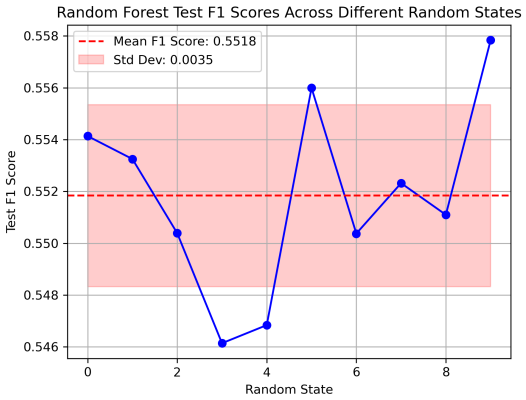


Figure 5

This includes a large number of trees, and high depth. I was conscious of the computational power that my chosen parameter grids would require so I had to balance efficiency and accuracy. In looking at the uncertainty of the evaluation metric in relation to my model performance I plotted the mean and standard deviation of the scores (figure 5). This showed some variance but not to an alarming degree.

Results:

The baseline weighted F score was 0.3955. The table below summarizes each model's performance compared to this baseline:

ML Algorithm	Best F Score	Difference (vs baseline)	Standard deviations away from baseline mean
LogisticRegression	0.5223	0.1268	2.44
RandomForest	0.5537	0.1582	3.01
KNearestNeighbors	0.5159	0.1204	2.29
GradientBoost	0.5345	0.1390	2.73
SVC	0.5243	0.1288	2.45

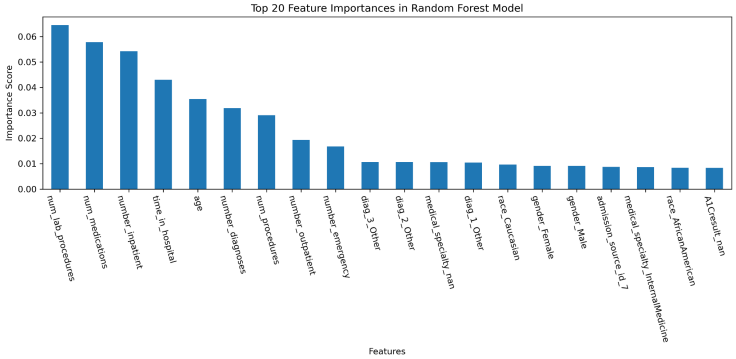


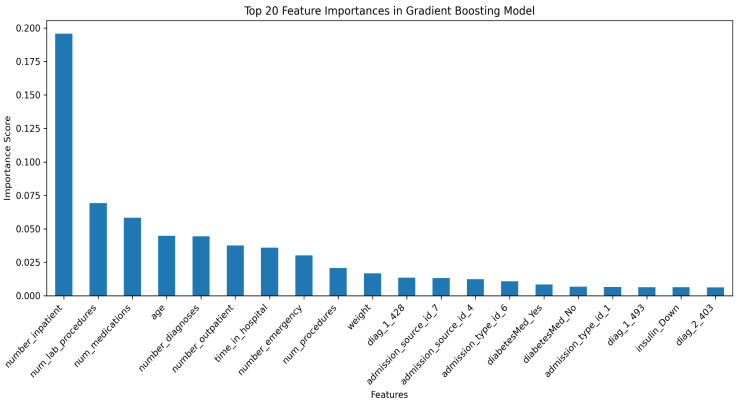
Figure 6

The RandomForest model was the most predictive and performed 3 standard deviations above the baseline mean.

I explored global feature importance in the RandomForest, GradientBoosting, and LogisticRegression models. For the random

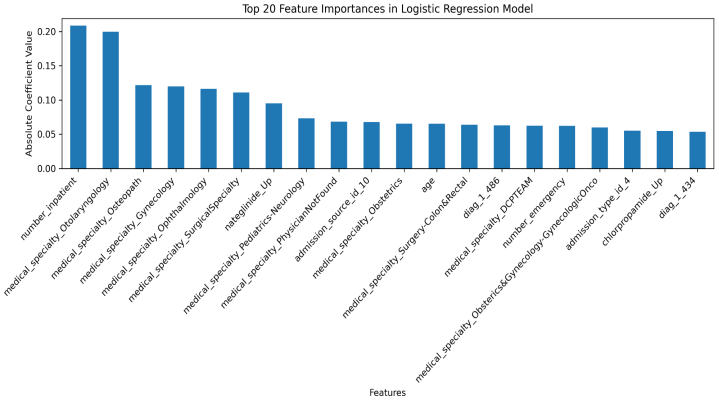
forest (figure 6), the medical history variables were most significant in aiding predictive power. This is in line with some of the exploratory data analysis, wherein the box and whiskers plots as well as the days in hospital graphs demonstrated some relationship with the target variable. There were also >50 variables corresponding to medication adherence - all of which were not deemed important globally.

Figure 7



In the logistic regression (figure 7), the medical speciality of the physicians diagnosing the patients carried much more weight in adding to the predictive power of the model. This was surprising as these variables did not stand out in any of the exploratory analyses.

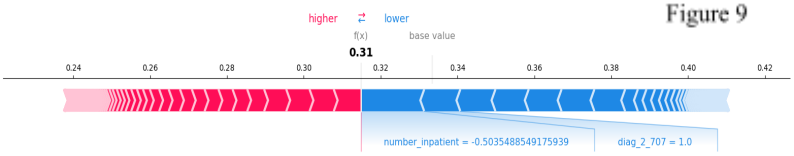
Figure 8



For the gradient boosting algorithm (figure 8), the medical history variables were important, similar to the random forest.

Local feature importance (figure 9) in the random forest using SHAP and a single prediction point showed that number of inpatient procedures pushes the prediction towards a lower probability of readmission, as does diag_2_707, or the presence of skin ulcers (decoding the “707” ICD disease code).

Figure 9



In interpreting the models, I created confusion matrices to investigate where the models underperformed (figure 10). It became evident that the models had difficulty distinguishing between Class 1 and 2 (readmission for <30, >30 days). They also had difficulty classifying Class 0 which was the No readmission class. This all made sense as the training data was imbalanced with the least number of Class 0 instances. Class 1 and 2 both correspond to some readmission value which explains why they are easily confused.

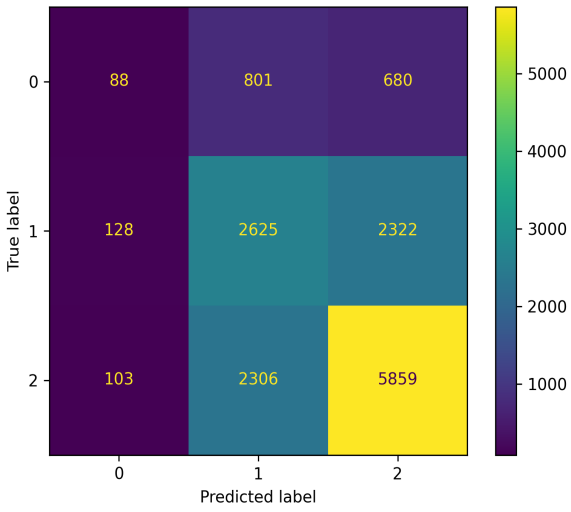


Figure 10

In terms of feature importances in the context of model interpretation, it speaks volumes about patient data that features such as medical history were most valuable in predicting readmission. There also appears to be a surprising gap in useful biometric data that could supplement the models performance. This kind of analysis would benefit from inclusion of more biometric and demographic patient data, as I believe resorting to physician specializations for feature selection speaks to an absence of useful patient information that could be used to predict diabetes readmission status.

Outlook

To improve model performance, reclassifying the <30 and >30 variables into a single "Yes" variable to create a bivariate classification model could resolve the ambiguity causing poor performance in classifying Class 1 and 2. The diag features could be expanded completely without collapsing non-top-20 values. In improving performance and interpretability, I could perform feature selection after understanding the impotence of every feature to the analysis. This would reduce noise in the model's function, and would make the reasoning behind every prediction slightly more interpretable. I am also curious about the application of deep learning methodologies like neural networks in solving medical problems like this one - from my reading, I believe this would be useful and would provide a new angle on predictive modeling for multivariate classification. If I had more time, I would definitely try this approach and would expand my parameter grids further to see if increased computation time serves to improve the model performance significantly.

Speaking to my remarks at the end of the last section, I argue that if the data was slightly more informative, features informing blood sugar levels, muscle mass, or BMI would be ranked highly in terms of contribution to the model. This, in my opinion, is the biggest flaw in the problem I am attempting to solve. Without balanced and detailed biometric data, very little can be said about disease-related readmission. With illnesses like DM, lifestyle factors are most important in determining how someone manages their condition. Having such insight would likely improve model performance.

Works Cited:

Ostling, S., Wyckoff, J., Ciarkowski, S.L. et al. The relationship between diabetes mellitus and 30-day readmission rates. Clin Diabetes Endocrinol 3, 3 (2017). https://doi.org/10.1186/s40842-016-0040-x

Strack, B., et al. (2014). "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records." BioMed Research International 2014: 781670.