Hospital Readmission Prediction for Patients with Diabetes

Data 1030 Final Report – Aryaman Dutta

Brown University

https://github.com/Ary-d/DiabetesReadmission_Ary-d

Introduction:

This project attempts to create a machine learning classification tool that as accurately as possible classifies whether a person with diabetes will be readmitted within 30 days, in more than 30 days, or not readmitted at all based on certain features describing the patient and their current situation. Thus, the target variable for the classification is “readmitted”, which is a nominal variable that is “<30” if the patient was readmitted within 30 days, “>30” if the patient was readmitted in more than 30 days, and “NO” if the patient was not readmitted according to the records. The dataset has 101766 data points and 50 columns including the target variable.

This is a very important problem to solve as the cost of hospital readmissions is extremely high for hospitals [1] which reduces the amount of time and resources that can be allocated to other patients. Patients with diabetes in particular face a much higher chance of readmission. This can be seen in a statistical brief of the 2018 Nationwide Readmissions Database (NRD) [1], which shows that people with a principal diagnosis of “Diabetes mellitus with complication” are the group with the third highest number of hospital readmissions within 30 days. In fact, septicaemia, heart failure, diabetes, and COPD together accounted for one in five readmissions in 2018 [1]. Therefore, this is a very important classification problem to solve in healthcare.

This dataset was used in a study to fit the relationship between the measurement of HbA1c, a test that measures the amount of blood sugar attached to haemoglobin, and early readmission [2]. In order to do so the authors used multivariable logistic regression. The results are that the measurement of HbA1c was performed infrequently (18.4%) in the inpatient setting and that the decision to obtain this measurement is a useful predictor of readmission rates. The dataset was also used to create a gradient boosted tree (GBT)-based classification model that classified the three diabetes type diagnoses using multiple patient features [3]. This classification model showed a maximum average precision of 91.64%, a recall of 97.46%, an accuracy of 99.93%, an F-score of 94.19%, and a kappa of 96.61%.

Exploratory Data Analysis:

Below are several figures that stand out from the exploratory data analysis performed on the dataset.

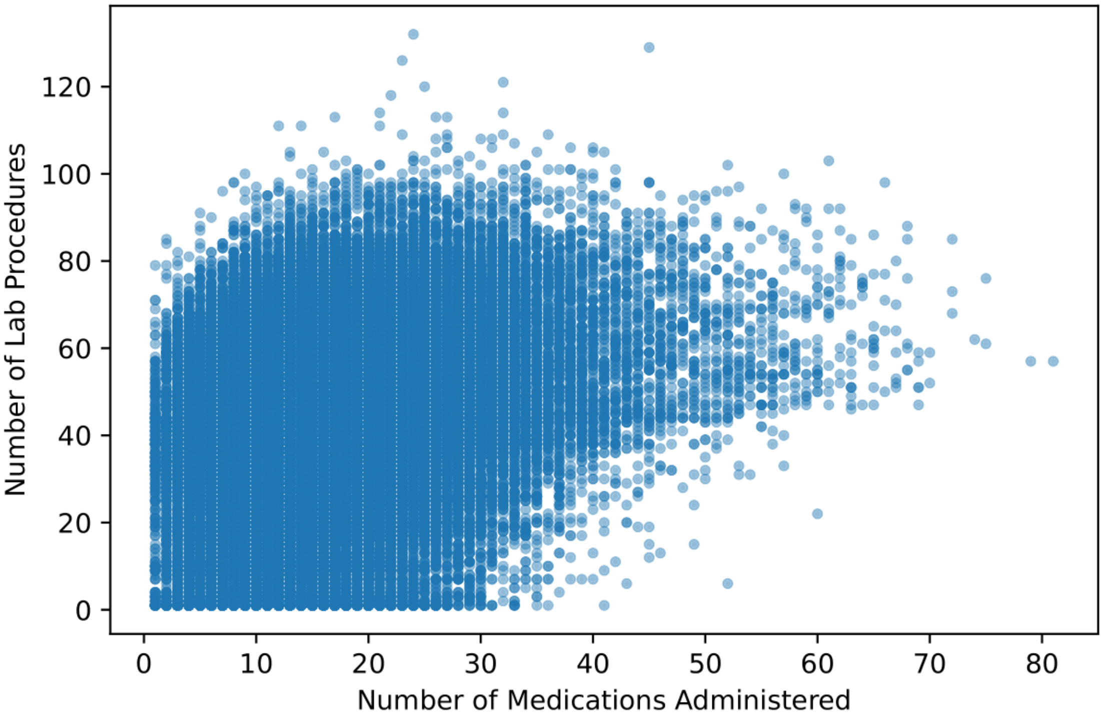

Figure 1 This figure displays a scatter plot of the number of lab tests performed during the encounter (num_lab_procedures) vs the number of medications administered during the encounter (num_medications). It appears there is a strong positive correlation between these two variables. In fact, from plotting a scatter matrix of all the numeric features, it appeared as though these were the two variables with the clearest positive correlation. This correlation could be possibly explained by the fact that it is likely the same patients who are more unwell than average who would get more lab tests taken and more medications administered. It is also the case that if you have more lab tests carried out on a patient you are likely to discover more about the patient’s affliction and then in turn administer more medications.

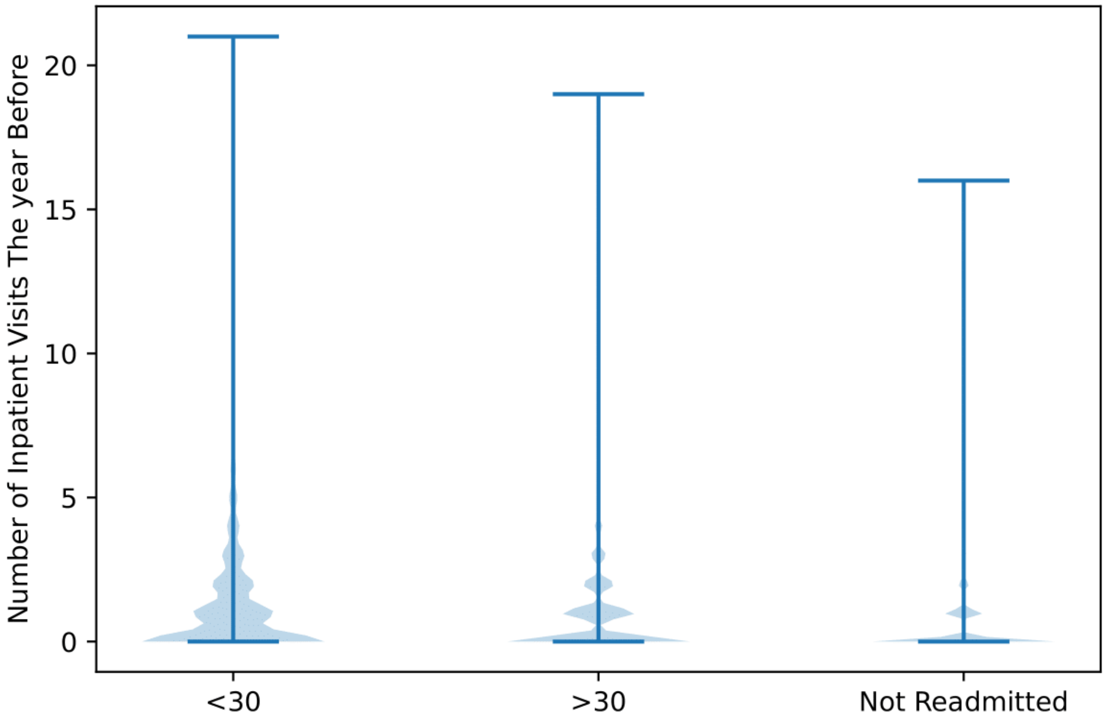

Figure 2 This figure displays a violin plot of the number of inpatient visits the year before this encounter (number_inpatient) by the readmission status of the patient (readmitted). From the plot it is evident that the more inpatient visits (hospitalisations) you have the previous year, the more likely it is that you will be readmitted within 30 days of being discharged. It is clearly the least likely outcome of the three that you will not be readmitted at all if you have had more than one inpatient visit the year before this encounter. This is significant as it implies that number_inpatient will be an important variable when predicting the readmission class of a patient.

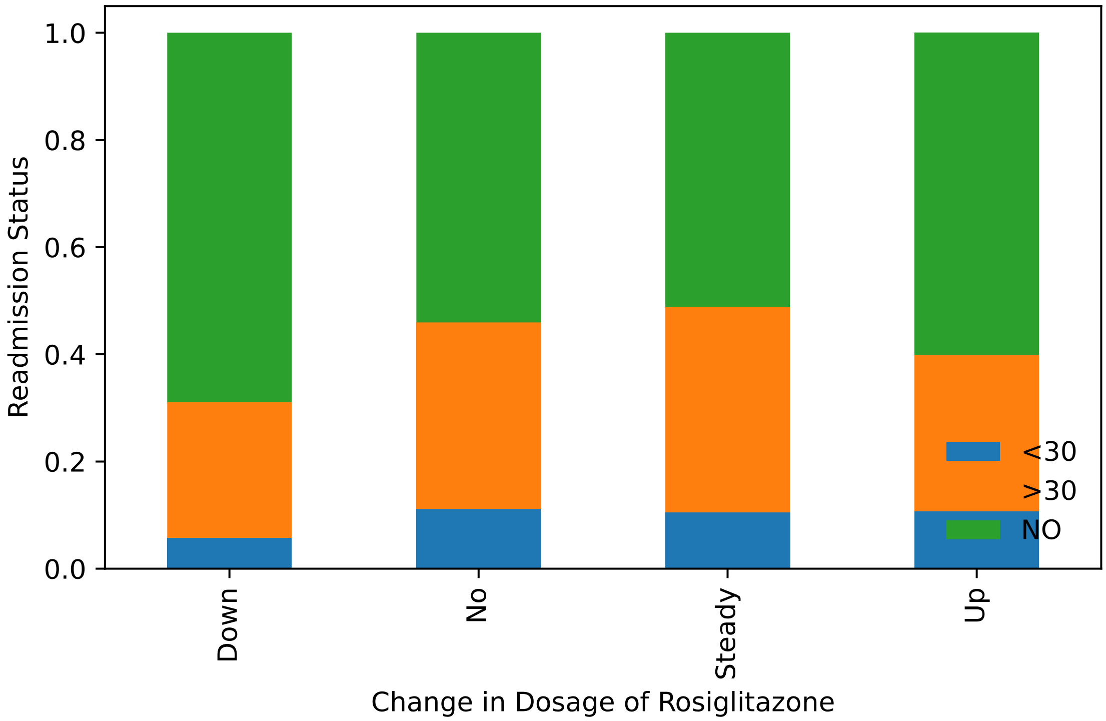

Figure 3 This figure displays a stacked bar plot with the change in the dosage of Rosiglitazone (rosiglitazone) in the x-axis and the readmission status (readmitted) in the y-axis. The x-axis labels mean the following- “Up” if the dosage of rosiglitazone was increased during the encounter, “Down” if the dosage was decreased, “Steady” if the dosage did not change, and “No” if rosiglitazone was not prescribed. Thus, from the plot it seems as though the probability of readmission falls if either the dosage is increased or the dosage is decreased during the encounter. This suggests that the change in dosage of rosiglitazone might be an important variable when predicting the readmission class of a patient and that rosiglitazone might be an important drug to look at when treating diabetic patients.

Methods:

The examide and citoglipton features have the same value for all datapoints and hence are dropped from the feature matrix. The encounter_id and patient_nbr columns are also dropped from the feature matrix as they are identifiers for encounters and patients respectively. Finally, the diag1, diag2 and diag3 columns are dropped because if these are included, we end up with 2297 features after preprocessing because these features are categorical features with a very high cardinality (diag1, diag2 and diag3 have 717, 749 and 790 unique entries respectively). Predicting in this very high dimensional space led to much worse results from all the models attempted, and hence these columns were omitted.

In this dataset the string “?” is used to represent certain missing values. These are replaced with NumPy nan values as this facilitates using the isnull() function to find out which columns have missing values, and if so how many missing values. The dataset initially has 8 numerical features and 37 categorical features. There are no missing numerical features, however there are 7 categorical features with missing values. These values are replaced with the string ‘Unknown’ which will be treated as a new categorical class.

A StandardScaler is used for the numerical continuous data as none are reasonably bounded enough to use a MinMaxEncoder. An ordinal encoder is used for only ‘age’, which is categorical not numerical in this dataset, as this is the only categorical feature which can be sensibly ordered (‘weight’ had too many missing values). Finally, a OneHotEncoder is used on the rest of the categorical data. There are 255 features in the preprocessed data.

The dataset is non-IID as there are 101766 encounters and 71518 patients, which means that there are multiple data points associated with the same patient. Thus, it has group structure and while splitting the data we must make sure that the validation score is based on patients not included in training, that the test score is based on patients not included in training and validation, and that points of one patient are not be distributed over multiple sets as this would lead to an erroneous generalization error. In order to do so, GroupShuffleSplit is used on the data to split it in the ratio 80:20 where 20% of the data goes to the test set and 80% goes to an ‘other’ set. This other set is then further split into train and validation sets using GroupKfold with four folds.

This second split is done within sklearn’s GridSearchCV method. GridSearchCV is applied to a range of models to find the optimal parameters for each model. This is done over five different random states for the splitting and the best model and corresponding test score for each random state are collected in lists. Then the mean and standard deviation of the list of test scores is calculated to quantify the uncertainty due to splitting. This is highlighted in ‘Table 1’ in the results section. ‘Table 1’ also contains the parameters experimented with for each model and what the optimal parameter values came out to be. Uncertainties due to non-deterministic machine learning methods used are also measured- the test scores for different random_state parameter values for the RandomForestClassifier model are computed. Then the mean and standard deviation of these test scores are calculated and included in ‘Table 2’. Changing the value of the random_state parameter for the SVC and logistic regression models did not change anything and hence these are omitted from ‘Table 2’. K-nearest neighbors does not have a random_state parameter and is a deterministic method, and hence it is also omitted from ‘Table 2’.

Accuracy is used as the evaluation metric as the goal of this project is to create a model that can accurately predict the class of as many datapoints as possible. The dataset is not imbalanced and hence there was no need to use sklearn’s balanced_accuracy_score and hence sklearn’s accuracy_score is used as the evaluation metric. The baseline accuracy score is thus just computed by predicting all points to belong to the most populous class.

Results:

The performance of all the algorithms tried are summarised in the tables below.

Model Name	Mean of Test Scores	Standard Deviation of Test Scores	Model Parameters Tuned	Best Model Parameters
Logistic Regression	0.5848802408249523	0.002938389399951688	NA	NA
Logistic Regression with l1 Penalty	0.5849992192031344	0.0030831131362543886	C: [0.001, 0.01, 0.1, 1, 10, 100]	C=0.1
Logistic Regression with l2 Penalty	0.58516780628014	0.003343065364278162	C: [0.001, 0.01, 0.1, 1, 10, 100]	C=1
Logistic Regression with elasticnet Penalty	0.5850980527641281	0.0031322561164506247	C: [0.01, 0.1, 1, 10, 10] l1_ratio: [2, 4, 5, 6, 8]	C=1 11_ratio=0.6
K-Nearest Neighbors	0.5774205454408003	0.0034581444436599345	n_neighbors: [10, 30, 100]	n_neighbors= 100

Random Forest Classifier	0.5889310356282	0.003716284423141351	max_depth: [10, 12, 15, 30, 100]
			max_features: [0.1, 0.175, 0.25, 0.5, 0.75, 1]
				max_depth=12
				max_features=0.175

SVC	0.593142509237481	0.0026521117330162245	C: [0.01, 0.1, 1, 10, 100]
			gamma: [0.01, 0.1, 1, 10]
				C=10
				gamma=0.01

Table 1 *Mean and standard deviation of test scores for multiple algorithms with over different random_state values for the splitting of the data.*

Model Name	Mean of Test Scores	Standard Deviation of Test Scores
Random Forest Classifier	0.5878224770188791	0.0007093357761395622

Table 2 *Mean and standard deviation of test scores for a random forest classifier with optimal parameters over different random_state parameter values.*

SVC is clearly the best model, as it has both the highest mean test score and the lowest standard deviation of test scores as we can see in Table 1. The baseline model has an accuracy score of 0.5391191557101586. The SVC model is thus around 20 standard deviations above the baseline mode, which is a respectable improvement in score.

It is however important for interpretability purposes, particularly for a prediction that strongly affects the lives of patients, that we can also get an idea of how our models came to their decisions, and for that reason three different global feature importances were computed. The permutation feature importance over 10 different shuffles for each feature was computed for logistic regression models and SVC model. According to this number_inpatient was very clearly the most influential feature in prediction, followed by discharge_disposition_id. Computing feature importance as the coefficient of scaled logistic regression model led to similar results, as did computing global feature importance using Python's SHAP package on my random forest model ('Figure 4'). In all methods number_inpatient is an extremely important feature while predicting, which was what was predicted during EDA ('Figure 2').

Local feature importances, again found using the SHAP package on the random forest model to calculate SHAP values, were also computed for a range of different datapoints (patients) to see how the classification worked on a patient-to-patient basis. These results also showed the features number_inpatient and different discharge_disposition_id's as most influencial for the classification of each patient.

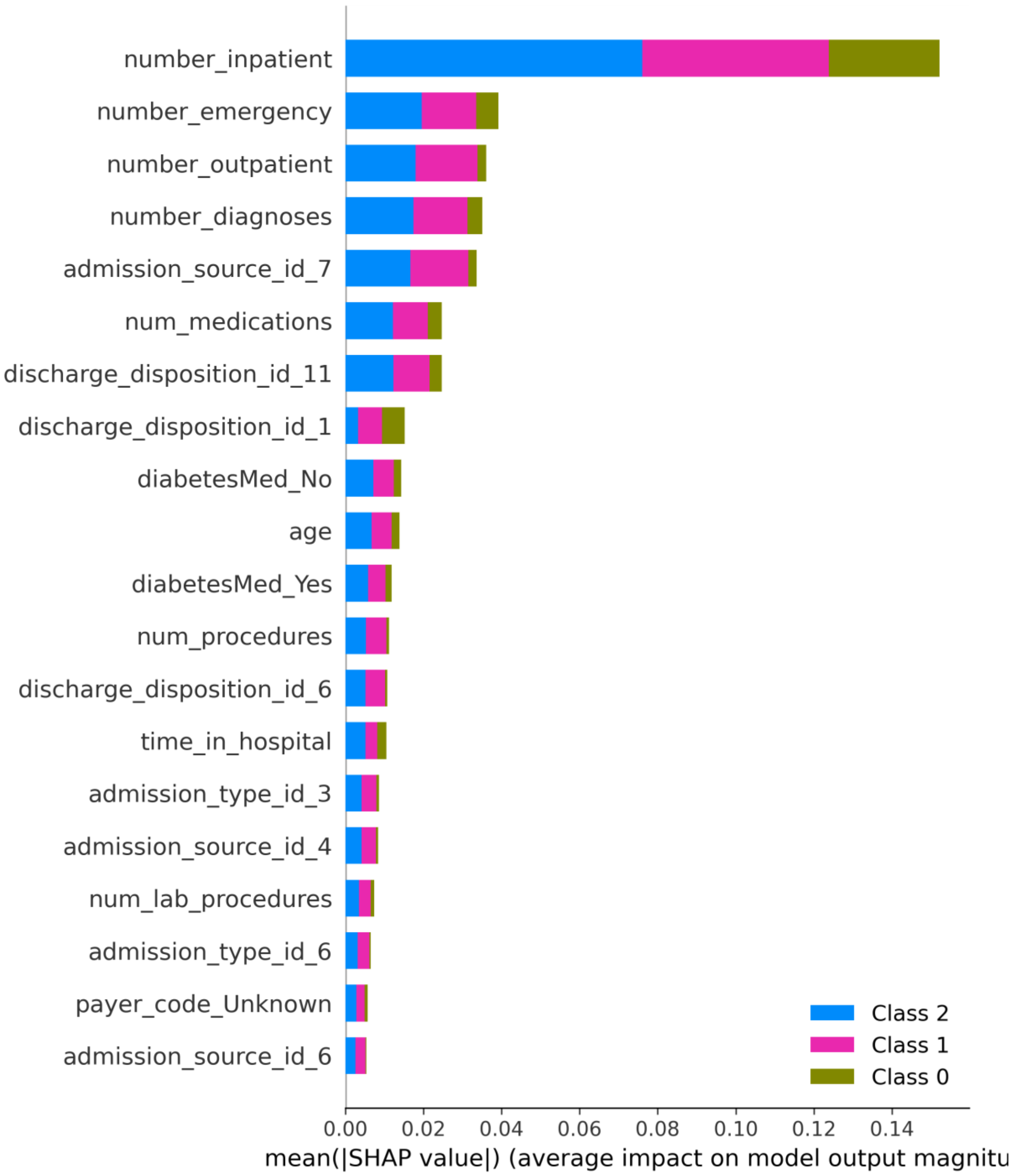

Figure 4 *Global feature importances using SHAP values and the RandomForestClassifer*

Outlook:

Possible improvements to the model include incorporating the diag1, diag2 and diag3 columns in some regard- for instance by combining them into a new feature that contains just a number between 0 and 3 to denote the number of diagnoses for each patient. Other models could also be tried out on the dataset to see whether this would improve performance. XGBoost in particular might lead to great results, as the RandomForestClassifier was the second-best performing model. While testing out why discharge_disposition_id values were consistently so high up in feature importance, a corresponding IDS_mapping CSV file was found, and it turns out some IDs correspond to patients who passed away or are on hospice, and hence clearly would not be able to be readmitted. Thus, excluding datapoints corresponding to these IDs (ID 11,13,14,19,20 and 21), would lead to a more robust model.

References:

[1] Weiss AJ (IBM Watson Health), Jiang HJ (AHRQ). Overview of Clinical Conditions With Frequent and Costly Hospital Readmissions by Payer, 2018. HCUP Statistical Brief #278. July 2021. Agency for Healthcare Research and Quality, Rockville, MD www.hcup-us.ahrq.gov/reports/statbriefs/sb278-Conditions-Frequent-Readmissions-By-Payer-2018.pdf.

[2] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records,” BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

[3] P N, P D, Mansour RF, Almazroa A. Artificial Flora Algorithm-Based Feature Selection with Gradient Boosted Tree Model for Diabetes Classification. Diabetes Metab Syndr Obes. 2021;14:2789-2806 https://doi.org/10.2147/DMSO.S312787