Predicting readmission of Diabetes patients within a year of admission using patient encounter information

MA429 Summative Project – Group 5

Candidate Examination IDs: 21730, 29510, 31594

Abstract

We use a dataset of Diabetes patients hospital records to predict likelihood of readmission within one year of current admission. This is a novel task on a dataset previously used only for predicting early (<30 days) readmission. We test models using pre-processing taken from literature and then using our own novel pre-processing, we produce a dataset of 25 features which we use to train six different models: a decision tree, a random forest, logistic regression, lasso regression, neural network, and XGBoost models.

Although our models don't perform extremely well (accuracy around 60% for all), we saw improvement to the literature models performance on our task. We also achieve a relatively high sensitivity (TPR) which is necessary given that our task concerns patient's medical care.

We offer advice to health care providers based on our findings. We suggest that older patients with frequent hospital admissions are more likely to be readmitted within a year of current admission and therefore extra care should be provided to prevent this.

Introduction

Diabetes impacts 11.6% of the US population, in 2021/2022 it was the eighth leading cause of death and had an estimated total direct and indirect cost of $413 billion [*CDC, Nov 2023*]. For these reasons, analysis of patient records from diabetes admissions to hospital is used to manage care and provide targeted intervention to those at risk of further complications, therefore enabling a reduction in readmissions and reduced strain on the health services. Using a dataset compiled between 1999-2008 of hospital records of diabetes patients, we will predict whether a patient would be readmitted within 1 year of their initial admission. This prediction task differs to most literature on this dataset, it is most common for researchers to predict early readmissions, which occur within 30 days of initial admission. The reasoning behind instead predicting 1-year readmission is that factors which relate to readmission within 30 days are often serious health problems that have not been resolved, whereas readmission within 1-year might suggest mismanagement of a patient's diabetes. For this reason, we are interested in exploring whether a relationship can be found between a patient's treatment and readmission within 1 year.

Literature

The dataset was initially compiled in 2014 for a research article written by John N. Clore and collaborators [*Clore et al. 2014*]. They were investigating the impact of HbA1c measurements (a reflection of blood glucose concentration) on early readmissions to hospital. They gathered data from a database containing over 74 million admissions of 17.8 million unique patients and extracted visits that satisfied a set of criteria identifying hospital admissions due to “diabetic encounters” in which the length of stay was between 1-14 days, and that laboratory tests and medications were administered. This dataset contains over 100,000 patient encounters. 55 features were extracted as having been identified by a clinical expert to potentially be associated with diabetic conditions and management. These included 23 medications and information on diagnosis, admission type and patient characteristics (weight, age, race).

In the original literature they were specifically exploring how one feature impacted readmission (HbA1c measurement). They found that this measurement was taken infrequently, having been performed in only 18.4% of cases, in the cases HbA1c was recorded it takes values <7, <8, and normal. They performed statistical analysis of the relationship between early readmission and HbA1c measurement by producing logistic models for 1. *All variables except HbA1c* and 2. *All variables including HbA1c* and 3. *All variables except HbA1c but including pairwise interactions* and finally 4. *All variables including HbA1c and pairwise interaction.* From these models they tested the significance of variables, analysed the deviance tables, and performed sensitivity analysis by removing one variable at a time and observing changes in beta coefficients. They discover that the relationship between readmission and HbA1c measurement depends quite significantly on the primary diagnosis, that with a primary diagnosis of diabetes mellitus and any measurement of HbA1c (as in the test was performed, not a requirement for a specific level), there resulted in a lower rate of early readmission than when the primary diagnosis is a circulatory or respiratory disease. They argue that when the test was ordered there is statistically significant change in medication in patients which may explain the relationships between HbA1c test performed, primary diagnosis and early readmission, especially given some diagnoses are more treatable than others.

In a paper published in 2023 by Dhiraj Kumar Sah Kanu and Madan Khanal [*Kanu and Khanal, 2023*], there is analysis of this dataset with the intention of predicting early readmission based on all features. Early readmission again means readmission within <30 days. They pre-process the data to reduce the number of features from 50 to 23 and explore relationships between these remaining features and readmission rates through data visualisation. They train three separate models: a logistic regression model, a decision tree, and a random forest model. Their logistic regression model had an 88.43% accuracy and a recall of 82.6%. Their decision tree had a test accuracy of 100%, to ensure the model was not overfitted they performed cross-validation resulting in an average score of 78.7%. Finally with the random forest model they got 100% accuracy on their testing set and 89.6% on cross-validation. They make suggestions for improving early readmission rates and discuss the fall backs of their paper. It should be noted that although the pre-processing and modelling procedure is explained in reasonable detail, the results obtained in this paper were not reproducible and they are by far the most optimistic in terms of success of the literature available. There was also no visualisation of explainable methods like the decision tree. This calls into question the validity of the models and the legitimacy of the work.

In another paper published by Clodagh Reid [*Reid, 2019*], the researcher again attempts to use machine learning techniques to predict early (<30 days) readmission. They perform extensive background research into work done on diabetes prediction and perform similar pre-processing to previous work. An addition to previous work attempted in this study is introducing a second target variable "Diabetes," the task which aims to predict whether a patient is given (in diagnosis columns 1,2,3) a diabetes diagnosis. The original task is performed with success, they achieve 86% accuracy and 87% recall using a boosted decision tree and 76% accuracy and 72% recall in logistic regression. In this paper there is also an attempt to build a neural network which achieves 76% accuracy and 78% recall. Their "Diabetes" diagnosis prediction is less successful with all models averaging around 65% accuracy and 62% recall. We suggest that this is because the existence of "Diabetes" as diagnosis 1,2 or 3 in this dataset would be solely dependent on the number of diagnoses that a patient is given and whether a doctor sees it fit to assign "Diabetes" as a primary diagnosis. Given that this dataset was specifically extracted to contain only incidences of "diabetic encounters" [*Clore et al. 2014*] and instances were extracted under very specific conditions, we believe that this task is difficult, which might suggest why the prediction models do not perform very well.

In reading and evaluating previous approaches to this readmission problem, we trained models for our novel classification task (readmission within 1 year, instead of 30 days) using the pre-processing as described in the mentioned papers. Doing this we achieved the following results:

Model	Accuracy	Sensitivity	Recall
Logistic Regression	57%	59%	57%
Decision Tree	55%	64%	48%
Random Forest	61.8%	81%	34%

Table [1] - Table showing the accuracy, precision and recall achieved on our novel prediction task, using pre-processing taken from literature.

Our aim for this task is to meet and exceed the results above using our own novel pre-processing and modelling.

Data and pre-processing

The dataset as published by Clore et al. contains 50 variables including the target "readmitted" and 101766 patient encounters. The variables include patient information, admission information, diagnoses, care plan records, test results and information on specific medication. They are a mix of categorical and numerical variables. We perform significant pre-processing to produce a dataset which will be used in our models. An overview of our pre-processing is given in Figure [1] and described in detail in the following subsections.

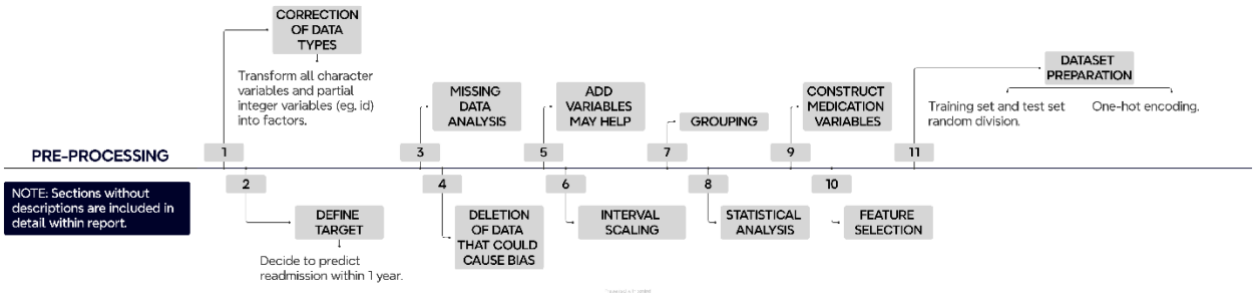

Figure [1] - Flow chart for data preprocessing.

Missing data analysis

Seven variables contain missing values, which are denoted by "?" in this dataset. Weight, payer code and medical specialty have a sizeable proportion of missing values (96.9%, 39.6% and 49.1% missing respectively). We believe it is justified to remove these variables entirely without impacting the quality of our models.

Weight must be removed because it contains too few entries to be useful in the model.

Payer code is related to the payment or insurance plan a patient is on, although this could provide some insight into the socio-economic background of a patient; which often has an impact on medical care; the proportion of missing values makes its inclusion impractical, especially given many available values are classified as "Unknown".

Medical specialty provides useful information about the severity and type of problem a patient is admitted with, for example a patient admitted to Cardiology or Emergency has different implications than those admitted to Dermatology or Orthopaedics. However, this information can similarly be found in the diagnoses a patient receives. Therefore, it is reasonable to remove without losing information.

The remaining "?" values are found in the "Race" and "Diag_X" variables. Approximately 2.7% contain missing values. We remove these encounters because it is not possible to predict race or diagnosis based on other data and it would be irresponsible to impute with common values which could negatively impact our models. In doing this we remove approximately 2100 patient encounters.

Discharge conditions

Discharge ID tells us the reason a patient is discharged. In most cases a patient is identified as being discharged to home or transferred to another facility. However, in some cases a patient is discharged because they have "Expired" or they are discharged to hospice, where it is assumed, they will pass away. In these cases, we can guarantee that the patient will not be readmitted to hospital. The observations with a patient discharge ID in [11, 13, 14, 19, 20, 21] are removed. Although these observations could contribute to the model's ability to predict readmission (or non-readmission), it does not provide useful insight to medical facilities when making decisions about care, therefore it is not helpful for our task. 2423 encounters are removed this way.

Identifying unique patient encounters

We want all observations to be independent of each other for our models. This is not the case for repeat patient encounters, so we identify unique encounters from the patient ID information. There are 69987 unique patient IDs and 99340 unique encounter IDs. We keep only the first encounter for each patient ID and remove the rest. The dataset is reduced to 69987 observations. We also remove the patient ID and encounter ID variables to completely de-identify the data.

Medication variables

The dataset contains 23 features which are diabetic medications, they take values: "No" meaning the medication is not used; "Steady" meaning the patient was already on this medication and remained on it; "Up" meaning the dosage of this medication was increased (from previous dosage or patient started taking medication); and "Down" meaning the dosage of this medication was decreased.

At this point three medications (citoglipton, glimepiride pioglitazone and examide) contain only one level ("No") meaning there is no variability. These variables have no predictive power and will add nothing to the models, so we remove them.

There are a few bits of information we want to extract from the remaining medication features. We begin by introducing a new variable which counts the number of medications that were changed during a patients stay. Intuitively, more medication changes suggest more intervention and so we believe this could be useful for our model. The distribution of this new variable is shown in Figure [2].

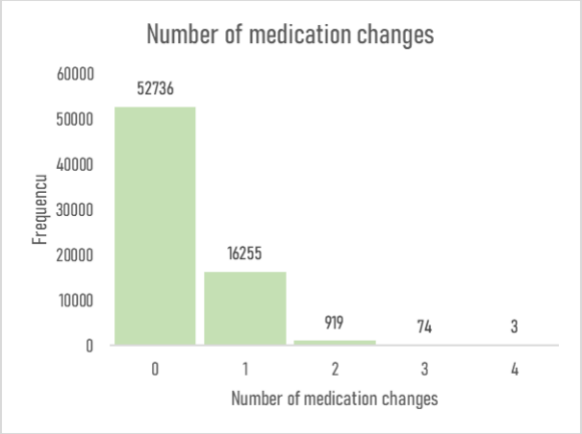

Figure [2] - Graph showing the distribution of variable "num_medication_change"

Next, we want to extract 2 important pieces of information. We want to know whether a specific medication is used and whether that medication dosage changed (for each medication). To do this, we create new variables for each medication called "[medication].used" and assign it value 0 for "No" and 1 for "Steady", "Up", or "Down". We then change the value of the original "[medication]" variables to either "Up", "Down" or "No Change". In doing this we have introduced 20 new "used" variables to the dataset.

Given that we have extracted the information about whether each medication is used, we now remove the following variables: acetohexamide, tolbutamide, acarbose, troglitazone, tolazamide, glipizide.metformin, metformin.rosiglitazone, metformin.pioglitazone because they all contain only one level in the change variable ("No Change").

Age

The age variable is given as an age range with values given as [0-10), [10,20), [20-30) etc. We extract the bounds and set age as the midpoint. This is important because readmission rates change significantly with age, making it a useful predictor. We want to ensure that age is treated as an ordinal variable, as the likelihood of readmission increases with age. To do this we make "Age" an ordered factor with 10 levels.

Age	Number of observations	% readmission
0-10	153	0.1764706
10-20	534	0.3146067
20-30	1121	0.3211418
30-40	2692	0.3257801
40-50	6828	0.3586702
50-60	12351	0.373573
60-70	15688	0.4055966
70-80	17749	0.4476309
80-90	11110	0.4555356
90-100	1761	0.3674049

Table [2] - Showing the distribution of age and the likelihood of readmission.

When encoded the age variable takes values for "age.L", "age.Q", "age.C" representing age to the powers of 1, 2 and 3 respectively and "age.4", "age.9" representing age to higher order powers.

Diagnoses categorisation

Initially the columns containing information about patients' core diagnoses (diag_1, diag_2, and diag_3 variables) are coded using ICD9 medical codes. ICD9 are alphanumeric codes which are used to classify diseases, injuries, symptoms, and medical procedures. For example, the code 250.00 represents “Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled”. To break the code down: 250 represents the category “Diabetes mellitus” and .00 represents the specific subcategory “without mention of complication, type II or unspecified type, not stated as uncontrolled”.

Initially there are 717 unique values in the diag_1 column, 749 in diag_2, and 790 in diag_3. We will group these into the following categories: [Circulatory, Respiratory, Digestive, Diabetes, Injury, Musculoskeletal, Genitourinary, Neoplasms, and Other]. These categories are chosen to represent the most commonly occurring groups in the dataset. The table of classification codes can be found in Appendix 1. These categorisations were taken from [*Clore et al. 2014*] and cross-referenced with [*ICD-9-CM].*

Classifying test results

The dataset contains two sets of test results. We want to know whether a test result was normal or abnormal. We also identify if the test was not performed, this gives an indicator of how thorough a patient's care plan is.

First, “A1Cresult” is the results of the HbA1C test; which was the variable of interest in [*Clore et al. 2014*]. It takes values “None”, “Normal”, “>7”, and “>8”. We reclassify these results as “None”, “Normal”, and “Abnormal”.

Next, “max_glu_serum” is the glucose serum level which measures of amount of glucose in the blood. It takes values “None”, “Normal”, “>200”, and “>300”. Again, we reclassify these results as “None”, “Normal”, and “Abnormal”.

The justification for recategorization is that most of the patients didn’t have the test run so the result is classed as “None”. We hope that by combining the two “abnormal” test results, any pattern is easier for the model to find.

>200	>300	None	Norm
936	712	66638	1701

>7	>8	None	Normal
2866	6239	57141	3741

Table [3] - Table showing the distribution of test results.

Categorising other variables

We reclassify four sets of variables: “Admission type”, “Admission source”, “Time in hospital” and “Discharge disposition”.

Admission type is grouped into: [“Not available”, “Emergency”, “Elective”, “Newborn”]. It is found that the nine encounters labelled “Newborn” have incorrectly labelled ages (infeasible for newborns), therefore we remove these encounters.

Admission source is grouped into: [“Emergency room”, “Referral”, “Not available”, “Transfer”, “Other”].

Time in hospital should be treated as an ordinal variable, there is a positive correlation between time spent in hospital for the current admission and likelihood of being readmitted. This is shown in Table [6]. Like age, time in hospital is treated as a polynomial regression model during encoding leading to the 14 encoded variables being time to the power of 1-14.

Discharge disposition is grouped into: [“Discharged to home”, “Transferred”, “Remain patient”, “Not available”].

The codes assigned to each category are listed in Appendix 2.

Anomaly detection

Considering our dataset contains numerical features, we felt compelled to check for statistical anomalies which could bias our modelling outcomes. With the most recent pre-processing we consider the overview of the following numerical features and their summary statistics:

Predictor	Skewness	Mean	Min	1Q	Median	3Q	Max
Age	-0.6331	65.44	5	55	65	75	95
Time in hospital	1.179	4.273	1	2	3	6	14
Number of lab procedures	-0.2195	42.88	1	31	44	57	132
Number of procedures	1.2276	1.425	0	0	1.425	2	6
Number of medications	1.4345	15.67	1	10	14	20	81
Number of medication changes	1.6036	0.2619	0	0	0	3	4
Number of visits	6.6930	0.5598	0	0	0	1	49
Number of diagnoses	-0.7250	7.224	1	6	8	9	16

Table [4] – Table showing the summary statistics of numerical features

We first checked for outliers using boxplots. We observe how two features have a low number of outliers compared to the sample size. Number of lab procedures has 98 outliers (0.14%) and Number of diagnoses has 235 (0.33%). We eliminate the respective instances considering the low proportions and after verifying that their distribution with respect to the target variable wasn't heavily skewed toward one value. This brings down the number of instances to 69645.

We also tested whether transforming the variables with high skewness impacted the model. This work is detailed in Appendix 4. There was not a significant improvement therefore this was not included in the final pre-processing, to improve the interpretability of the results.

Feature selection

We perform a statistical analysis by calculating the entropy of each variable. This gives us information about the distribution of variables which could help us assess how useful a variable will be as a predictor. The entropy ranges from 0.0003-6.1995.

First, we investigate variables with very low entropy values (<0.1). Features with low entropy have high predictability because they don't contain much variability. For example, "repaglinide" has an entropy of 0.0169. The distribution is as follows:

	Up	Down	No change
# encounters	68 (0.1%)	28 (0.042%)	66619 (99.85%)
% readmission	0.4128	0.4559	0.4286

Table [5] - Showing the distribution of repaglinide changes and likelihood of readmission.

We perform a likelihood ratio test to decide whether a variable with low entropy would be a good predictor for readmission. Comparing a logistic regression model of "readmission ~ repaglinide" with "readmission ~ 1" we achieve a p-value of 0.763. We have insufficient evidence to suggest that this variable will significantly improve our model.

We perform similar analysis on the remaining "low entropy" features and achieve similar conclusions. For this reason, we remove the following variables: ["repaglinide", "nateglinide", "chlorpropamide", "glimepiride", "pioglitazone", "rosiglitazone",

"acarbose", "miglitol", "glyburide.metformin", "nateglinide.used", "chlorpropamide.used", "acarbose.used", "miglitol.used", "glyburide.metformin.used"] which all have an entropy of <0.1.

Next, we explore the features with high entropy (>2). We extract the following features: ["age", "time_in_hospital", "num_lab_procedures", "num_procedures", "num_medications", "diag_1", "number_diagnoses"]. The distribution of one of these is as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
# encounters	9846	11499	11881	9014	6459	4851	3764	2738	1837	1456	1144	883	728	615
% readmission	0.356	0.39	0.401	0.424	0.429	0.443	0.453	0.466	0.469	0.467	0.445	0.468	0.441	0.478

Table [6] - Showing the distribution of time in hospital and readmission rate.

Again, we perform a likelihood ratio test and discover that variables with high entropy have very low p-values, meaning that they are statistically significant and will improve our models. We consider these variables important to our models.

Performing likelihood ratio test for feature selection

We perform likelihood ratio tests on the remaining features to remove insignificant ones. We make 2 logistic regression models: “readmission ~ all except one variable” and “readmission ~ all” and use a likelihood ratio test to determine whether there is sufficient evidence to suggest a variable improves the model. The p-values are shown in Appendix 5. As a result of this analysis, we remove the following variables: ["metformin", "glipizide", "glyburide", "diabetesMed_changed", "num_medication_change", "glimepiride.used", "glyburide.used", "pioglitazone.used"].

Dataset after feature selection

We are left with the following 23 features.

	Variable	Type	Levels
1	race	Factor	African American, Asian, Caucasian, Hispanic, Other
2	gender	Factor	Male, Female
3	age	Ordered factor	5 < 15 < 25 < 35 < 45 < 55 < 65 < 75 < 85 < 95
4	admission_type_id	Factor	Elective, Emergency, Not_available
5	admission_source_id	Factor	Emergency_room, Not_available, Other, Referral, Transfer
6	time_in_hospital	Ordered factor	1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < 11 < 12 < 13 < 14
7	num_lab_procedures	Integer	
8	num_procedures	Integer	
9	num_medications	Integer	
10	number_visit	Numerical	
11	diag_1	Factor	Circulatory, Diabetes, Digestive, Genitourinary Musculoskeletal, Neoplasms, Other, Respiratory
12	diag_2	Factor	Circulatory, Diabetes, Digestive, Genitourinary Musculoskeletal, Neoplasms, Other, Respiratory
13	diag_3	Factor	Circulatory, Diabetes, Digestive, Genitourinary Musculoskeletal, Neoplasms, Other, Respiratory
14	number_diagnoses	Integer	
15	max_glu_serum	Factor	Normal, Abnormal, None
16	A1Cresult	Factor	Normal, Abnormal, None
17	insulin	Factor	Up, Down, No Change
18	diabetesMed	Factor	Yes, No
19	metformin.used	Factor	Yes, No
20	repaglinide.used	Factor	Yes, No
21	glipizide.used	Factor	Yes, No
22	rosiglitazone.used	Factor	Yes, No
23	insulin.used	Factor	Yes, No
24	readmitted	Factor	TRUE, FALSE

Table [7] - Final feature dataset to be used in modelling.

Methodologies

For our predictive task, we have selected a suite of well-known machine-learning algorithms tailored to address a classification problem on our pre-processed dataset, which contains 66,715 observations. To ensure robust performance across our dataset, we chose a diverse array of models, each suited to different aspects of the analysis.

We initiated our modelling with logistic regression, both with and without a Lasso regulariser. Following this, we implemented a simple neural network. We also employed a decision tree model for its interpretive ease. Lastly, we employed some ensemble methods, specifically Random Forest and XG-Boost, to further enhance the predictive power and stability of our models.

We implemented 5-fold cross-validation, which allowed us to tune the hyperparameters and produce the best-possible performing models. Although this approach increased our computational load, it provided analytical benefits, enabling us to identify the most effective models for our dataset.

Logistic regression and Lasso classification

Initially we build a logistic regression model with all features. To improve on this, we employ a Lasso classification model. This is a variation of Lasso regression tailored for classification tasks. It combines feature selection and regularization by minimizing the loss function; like in logistic regression; with an added L1-norm penalty and therefore reducing less important features' coefficients to zero.

Lasso was chosen instead of ridge regularisation because a reduction of features is desirable, and in lasso regularisation some of the unimportant features should be reduced directly to 0. This approach is particularly beneficial for high-dimensional data where feature selection is necessary and can improve interpretability of the models. Given that we are reducing the number of features, we could also find a reduction in training time.

We choose to use one-hot encoded data for the Lasso model, this is because it can make it easier for the logistic regression model to capture nonlinear relationships between the categorical variables and the target variable. Using one-hot encoded data in the original regression model is problematic because it increases the dimensions of the data significantly however using a regulariser, this is no longer an issue. It is found that the number of features decreases gradually as lambda (the regularisation parameter) increases. This is shown in Figure [3].

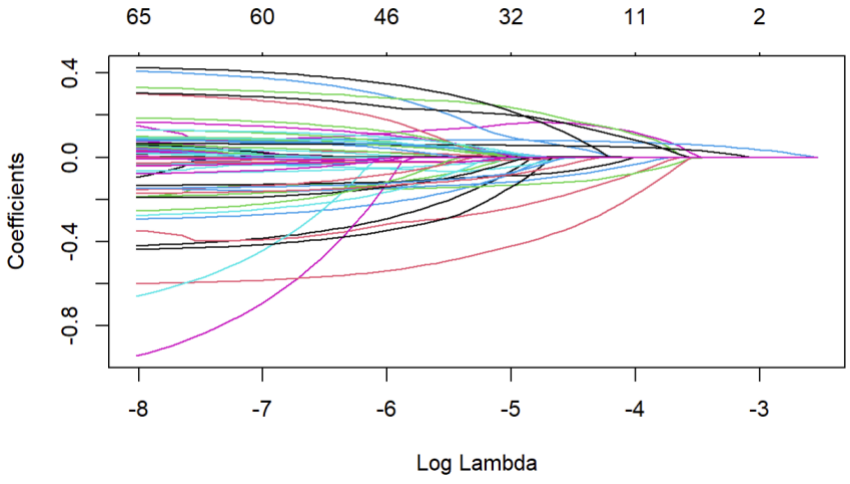

Figure [3] – Coefficients shrinkage in lasso model.

To choose the optimal lambda value, a 9:1 cross-validation method was applied, and the magnitude of lambda and misclassification error was plotted as follows.

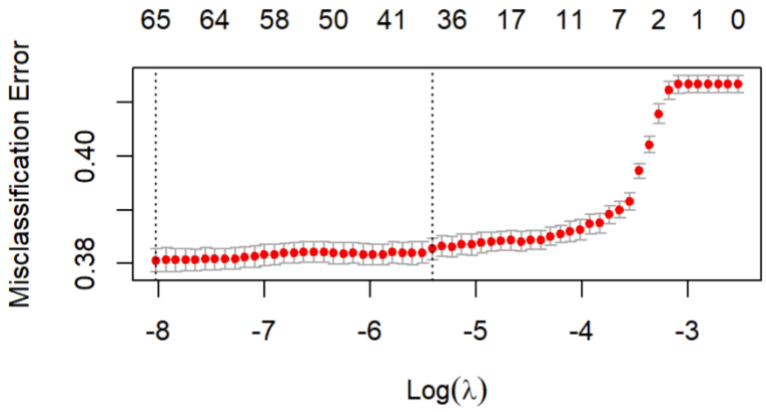

Figure [4] – Lasso model performance under different λ using cross-validation.

In lasso regression, as lambda increases, the bias of the model increases as the models become simpler. For our models four lambdas were selected and performances compared. The first chosen lambda is the lambda with the least error in cross-validation. It was so small that it reduced 72 features to 65 features. The second model uses the maximum lambda value that is within one standard error of the optimal performance in cross-validation. This lambda value allows for a significant reduction in the number of features to 39. Because 39 features are still too many to build an explainable model, we continue to increase the lambda to make more features shrink to 0. When setting lambda to 0.0183, the number of model features was significantly reduced to 11, much more suitable for an explainable model without much loss of accuracy. When lambda is increased further to 0.0302, we are left with only 3 features as non-zero, which are "age", "number_visit" and "number_diagnoses". This is the final model we build. All models and performances are explained in the Results section.

The initial logistic regression model took 17.35 seconds to train and achieved 61.97% accuracy. In comparison, as the number of features decreased the training time decreased significantly, with the final model taking under 1 second. The best performance of a lasso regression model was 61.92% and the model with just 3 features had 60.67% accuracy.

Neural Network

Neural networks (NN or ANN) are inspired by the complex architecture of the human brain. They are particularly effective for tasks requiring non-linear pattern recognition, such as natural language processing and dealing with large amounts of data.

A neural network is composed of interconnected units called neurons. Data in inputted through these neurons and is transformed by weighted sums at each node and activation functions, ending in a final prediction at the output layer. The model improves by adjusting the weights at the nodes during back-propagation, a process achieved by computing the gradient of the loss function relative to each weight throughout the training phase.

We selected this model for its ability to handle complex, non-linear relationships and interactions among the features in our dataset. Through hyperparameter tuning, we optimized the number of neurons in each hidden layer— determining that ten nodes were ideal— and the weight decay, set at 0.10 to prevent overfitting. Although the parameter optimisation is time and memory intensive, the final model training time was only 3 minutes 7 seconds, small in comparison and the prediction on the test set took less than 1 second.

The neural network achieved a 61.38% accuracy meaning it performed slightly worse than the logistic regression models. This is discussed later.

Decision Tree

The Decision Tree is a supervised, non-parametric method used for tackling both classification and regression challenges. It is straightforward and interpretable which makes it a useful model for our task, which should attempt to give advice to health care workers based on its findings. Decision trees partition the data into distinct regions, making predictions based on the

majority class within each region. One of the significant strengths of Decision Trees is their capability to process both numerical and categorical features, making them particularly suitable for datasets like ours. Additionally, they offer valuable insights into feature importance, highlighting which variables most significantly impact the target variable.

To optimize our model, we engaged in hyperparameter tuning, focusing specifically on the complexity parameter *(cp).* After careful adjustment, we determined the optimal setting for this parameter to be 0.01. The computational cost of fitting the model, measured in runtime, was 22.3 seconds and the test prediction time was 0.55 seconds. The decision tree gave a test accuracy of 61.36%.

Random Forest

Building on our experience with the standard decision tree model, we expanded our methodology to include ensemble techniques, starting with the Random Forest model. This approach uses bagging, where multiple decision trees are constructed during the training phase. Each tree is independently developed from a random subset of the training data, utilizing the bootstrapping method. During the construction of each tree, a randomly selected subset of features is used for making splits. This strategy reduces correlation among the trees, significantly enhancing the model's robustness and overall predictive power.

We opted for Random Forest as it serves as a logical extension of the decision tree model. Recognizing the computational intensity of Random Forest, we simplified our hyperparameter tuning process. We focused primarily on the 'mtry' parameter, which dictates the number of features considered at each split and is used for optimizing the model's accuracy. Despite the model's extensive computational demands, the tuning process was efficient, with the training time being 17.14 seconds and prediction time 1.63 seconds. The random forest method gave a 61.24% accuracy, a reduction in performance to the decision tree. This is unusual and will be discussed further later.

XGBoost

To further improve on the decision tree model, we used the XGBoost (eXtreme Gradient Boosting) algorithm. XGBoost is a refined version of gradient boosting, using decision trees as base learners. The trees are sequentially improved with regularization techniques, such as Ridge and Lasso, to enable model generalization. The model corrects errors from previous models through a process where each new tree is fitted against the negative gradient of the loss function.

We opted for XGBoost due to the lack of performance improvement provided by Random Forest model. In terms of hyperparameter tuning, we concentrated on the number of rounds (*nrounds)* to control the boosting iterations and maintained a low maximum depth of a tree (*max_depth)* to prevent overfitting, while other parameters were retained at their default settings. The fitting process was completed in 33 seconds and the prediction took 0.06 seconds. The XGBoost model gave a 62.30% accuracy, the highest achieved in our task.

Results

We wanted to better understand how the predictors impacted readmission within a year of a patient's initial admission to hospital. This was a new task performed on a dataset previously only used for predicting readmission within 30 days of initial admission. Table [8] show that most of our models have improved on the models built from literature pre-processing (found in Table [1]). Our novel pre-processing of the dataset lead to improved predictions with logistic regression and decision tree models but showed no improvement to the random forest. We consider the improvements a success and discuss the drawbacks which lead to no improvement for the random forest.

Model	Accuracy	Kappa	Sensitivity	Specificity
Logistic Regression	0.6197	-	0.830	0.313
Lasso	0.6192	-	0.834	0.309
Neural Network	0.6138	-	0.811	0.328
Decision Tree	0.6136	0.1284	0.861	0.255
Random Forest	0.6177	0.1464	0.810	0.339
XGBoost	0.6229	0.1664	0.816	0.344

Table [8] - Table comparing performance of all models

Logistic regression and lasso regression

The logistic regression model achieved an accuracy of 61.97%. We extended this with the lasso regression model which was used to reduce the number of features. As the penalty term lambda increases, we go from 72 to 3 features as shown in the table below. The performance of the model is not significantly impacted by such a huge reduction in features suggesting that most information the model is using is found in the "age", "number_visit" and "number_diagnoses" features which are the three remaining.

λ	3.28×10^{-4}	4.44×10^{-3}	1.83×10^{-2}	3.02×10^{-2}
Type	lambda with the least error in cross-validation	the maximum lambda value that is within one standard error of the optimal performance in cross-validation	customize lambda based on graph	customize lambda based on graph
Accuracy	61.92%	61.80%	61.74%	60.67%
Model length	65	39	11	3

Table [9] – Lasso models performance.

The simplest model gave the following equation:

$$-0.6877 + 0.0108 \times [\text{age}] + 0.1015 \times [\text{number of visits}] + 0.0377 \times [\text{number of diagnoses}]$$

Which can be treated as a simple way of establishing whether a patient is likely to be readmitted. Intuitively we see that as age, number of previous visits and number of diagnoses increases the patient has increased likelihood of being readmitted

and therefore steps should be taken to prevent this by spending extra time and resources on ensuring that patient is properly treated.

Neural Network

Our Neural Network 61.38% accuracy, which was not an improvement on the logistic regression model. From this we conclude that the model was unable to find any complex non-linear relationships in the training data. And therefore, the use of a complex neural network in this task is unnecessary, especially given that the lasso regression model with only 3 features performs just marginally worse.

We suggest that the reason for the neural network leading to no improvement is that our data is not detailed enough. Given many more features, the neural network may find non-linear relationships between variables such as medication dosage, health measurements (blood pressure, heart health etc.), diet, social class, income and a range of other bits of patient information but with a dataset of just 25 variables, there is not enough information to justify using a neural network as the primary model especially given that it is not explainable and therefore doesn't give use useful information to give to health care providers.

Decision tree and random forest

The decision tree model is shown in Figure [5] gives test predictions with an accuracy of 61.3%. The tree only has two splits using the number of visits and the number of lab procedures. The decision tree chooses not to add extra branches despite being below the maximum depth because there are no splits with high enough information gain to justify adding as a branch. This is because of the complexity parameter decides whether a split adds enough information, if not then the split isn't added to prevent overfitting the model.

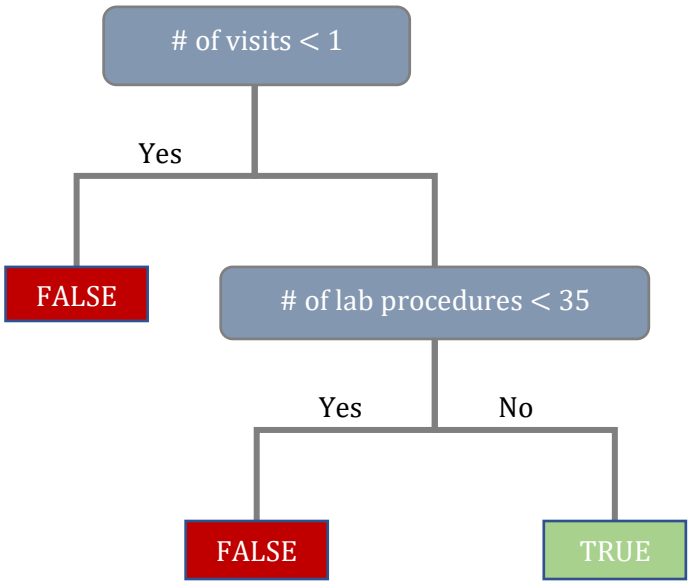

Figure [5] - Decision tree model which had 61.36% accuracy.

The decision tree again provides a simple framework for identifying patients whose care should be given extra attention. The results are again quite intuitive. A patient who has visited hospital within the last year and requires over 35 lab procedures likely has a serious health problem and their diabetes should be more carefully managed.

It is unusual for a Random Forest model to perform worse than the original decision tree. We suggest that our model's poor performance is due to the low Gini impurity decrease found in all variables. As shown in Figure [6], no variable has a mean Gini decrease of significant size. This means that each feature being considered for splitting is not very informative and doesn't have the ability to well-split the classification. This means that most of the trees made in the random forest perform the same or worse than the original meaning overall a worse classification accuracy. To test this theory, we build a decision tree without the features used in the original model. In doing this the "best" decision tree is a prediction of FALSE only (which

is the majority class). To further explore this issue, we trained an XGBoost model based on the misclassified classes in the original decision tree which did increase the accuracy.

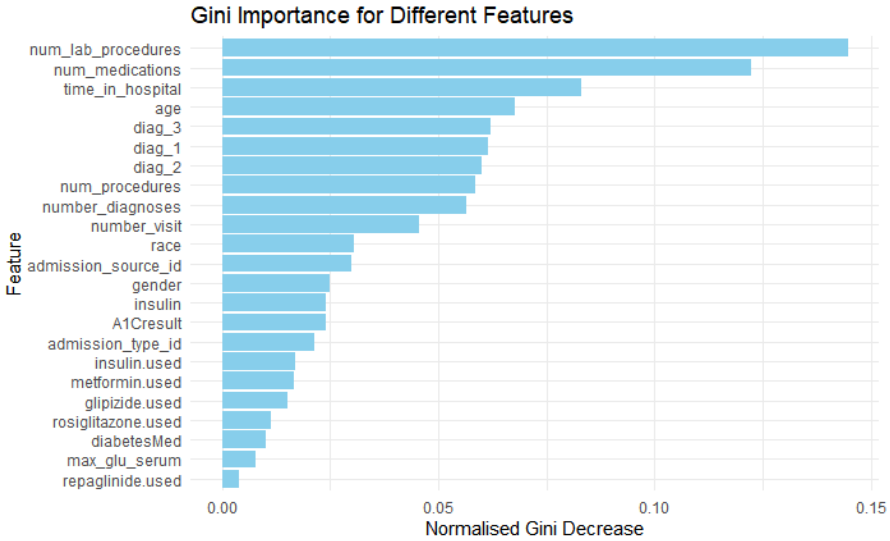

Figure [6] - Graph showing the normalised Gini decrease in information. All information gain from a split is small which is why the decision tree has only two splits and the random forest doesn't improve the model.

XGBoost

XGBoost had the highest accuracy out of the models with 62.3%. Through hyperparameter tuning the optimal number of trees to build was *nrounds = 500* and their maximum depth was *max_depth = 2.* The nrounds and depth was given as the lowest possible depth, suggesting that additional and longer trees were not increasing the model's performance. We believe the reason for this is that our data is not informative enough for extra, longer trees to add information without overfitting.

Figure [7] visualizes a single decision tree from the XGBoost ensemble method, which illustrates the decision-making process from the root node onwards. The root node split is number of visits, like in the original decision tree however the next splits differ, they use the age.C variable and the number of visits with a higher threshold. This is an example of one tree, the XGBoost model produces multiple trees in sequence, with each tree correcting the errors from the previous one. The final prediction is produced by aggregating all trees.

The metric *Cover* represents the number of instances seen by the split; the *Gain* is the information gain which provides insight into the importance of the node; the *Value* in the leaf nodes is the amount of value the leaf contributes to the final prediction.

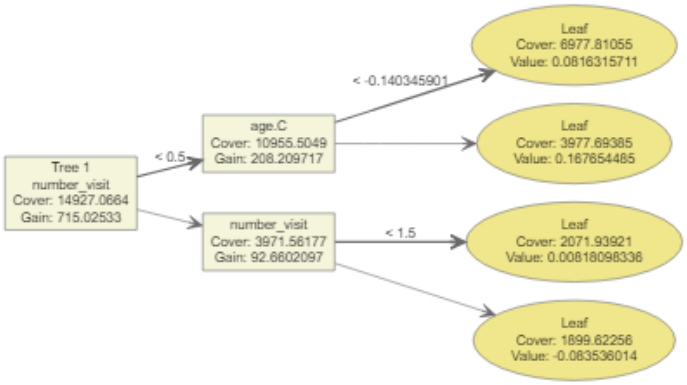

Figure [7] - Single decision tree from the XGBoost model

Sensitivity and specificity

Sensitivity is the proportion of positive encounters that are correctly predicted.

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity is the proportion of negative encounters that are correctly predicted

$$Specificity = \frac{TN}{TN + FP}$$

It is clear from Table [8] that the sensitivity of all models is high, and specificity is low.

	True Value	
	TRUE	FALSE
TRUE	3273	1887
FALSE	666	856

		True value	
Pred		TP	FP
		FN	TN

Sensitivity = 0.8309
Specificity = 0.3132

Figure [8] - Example of a confusion matrix and the sensitivity and specificity calculation.

The confusion matrices (all in Appendix 3) are all similar in that all models tend to over-predict TRUE, meaning overpredicting the number of patients who are readmitted. Given that these are models are being suggested to dictate the care of patients, it is ethically important that patients aren't put at risk and therefore over-prediction is preferred to under-prediction of readmittance.

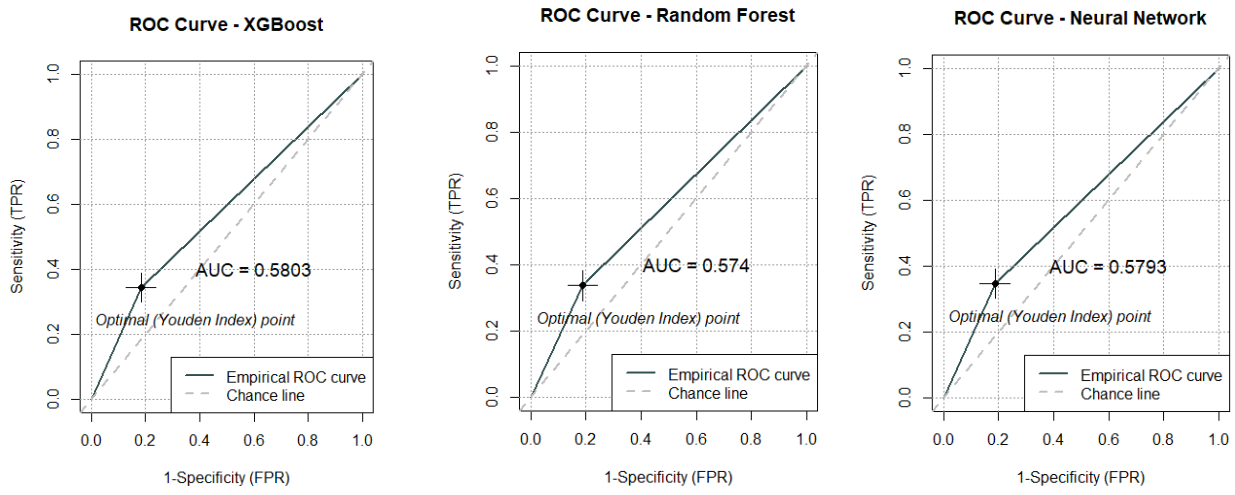

Figure [9] - ROC curves for XGBoost, Random forest and Neural network models

The ROC curves show the payoff between improving specificity and sensitivity. As expected, they are all very similar, given that the proportion of true and false positives are similar in all models. From the ROC curves we see that the cost of improving the false positive rate is a significant reduction in the true positive rate. The optimal point on the ROC curve is the point which maximises the area under curve which is an indicator of the model's ability to distinguish classes.

Drawing conclusions from models

For an insight into the factors influencing readmission, we analyse the important features from each model. Where possible the 2/3 most important features of our models are shown as below. Importance is judged based on appearance in the model and relative size of coefficients.

Model	Accuracy	Three most important features
Logistic Regression	0.6197	admission_source_idOther, diag_1diabetes, admission_source_idTransfer
Lasso	0.6192	number_visit, number_diagnoses, age
Neural Network	0.6138	-
Decision Tree	0.6136	number_visit, num_lab_procedures
Random Forest	0.6177	num_lab_procedures, num_medications, time_in_hospital
XGBoost	0.6229	-

Table [10] - Table comparing three most important features

The number of visits feature is the first ranked in both lasso and decision tree models. We performed a data presentation of the distribution of readmitted over different number of visits. We removed the observations with less than 20 instances to improve interpretability, given that there are a limited number of patients with visits between 20-49 it skewed the distribution.

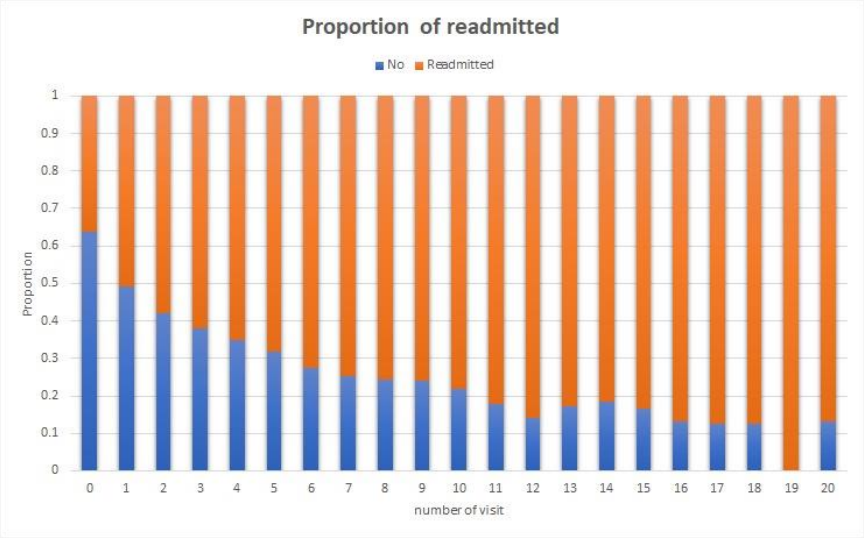

Figure [10] - Readmitted distribution on different number of visits.

As the number of visits increases, there is an obvious trend that the proportion of readmitted increases. This trend is reasonable since when the number of visits rises, it indicates that the patients are more likely to be in poor health so that they have a higher likelihood of being readmitted.

Age is also an important factor that we have previously analysed the distribution of in Table [2], there is a tendency for the proportion of patients readmitted to hospital to increase with age omitting age above 90 due to small amount of data. This is also intuitive that health problems increase with age, especially beyond 60.

The 'num_lab_procedures' ranks the first in the random forest importance gain and second in the decision tree. After visualising the data, no obvious quantitative relationship was found between it and readmission rates. From this we conclude that it is a good splitting feature alongside other splits.

Conclusion

We conclude that although our models do not perform as well as we would have liked, the novel pre-processing that we used lead to an improvement on the best-results using Literature pre-processing, which is a success for our models. We also saw improvements through hyper-parameter tuning and using more complex models (like going from a decision tree to XGBoost). We discuss the downfalls of models such as neural networks and random forests, which did not improve on the simpler logistic regression and decision tree models. We also suggest that our models having a high sensitivity (TPR) is a positive given the context of our problem in healthcare.

The advice that we would offer health care providers based on our findings is to pay careful attention to patients with a history of hospital admissions, especially patients over 60 years old. Frequent admissions to hospital could suggest that a patient's diabetes is poorly managed. It is suggested that extra tests and procedures are given to these patients to reduce their likelihood of readmission within 1 year. Although these results are intuitive, it is useful to have evidence of these relationships in justifying increased spending on certain patients.

References

[CDC, Nov 2023] Centre for Disease Control and Prevention (CDC). National Diabetes Statistics Report. Nov 2023. https://www.cdc.gov/diabetes/data/statistics-report/index.html. Accessed 09/04/24

[Clore et al. 2014] John N. Clore, Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios. Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records. 2014. DOI: 10.1155/2014/781670. https://doi.org/10.1155/2014/781670

[ICD-9-CM] ICD.Codes. ICD-9-CM Chapters. https://icd.codes/icd9cm. Accessed 24/03/24

[Kanu and Khanal, 2023] Dhiraj Kumar Sah Kanu, Madan Khanal. Implementation of Big Data Analytics on Diabetes 130-US Hospitals for year 1999-2008 for predicting patient readmission. 2023. DOI: 10.13140/RG.2.2.18564.30081. https://doi.org/10.13140/RG.2.2.18564.30081

[Reid, 2019] Clodagh Reid. Diabetes Diagnosis and Readmission Risks Predictive Modelling: USA. 2019. National College of Ireland. https://norma.ncirl.ie/4106/1/clodaghreid.pdf

Appendix 1

These ICD9 codes are used to group diagnoses in diag_1, diag_2 and diag_3. They were chosen to cover the most frequently found codes in the dataset.

Group Name	ICD9 Code	Specific information
Circulatory	390-459, 785	Diseases of the circulatory system
Neoplasms	140-239 780, 781, 784, 790-799 240-249, 251-279 680-709, 782 001-319	Neoplasms Other symptoms, and ill-defined conditions Endocrine, nutritional, and metabolic diseases Diseases of the skin and subcutaneous tissue Infectious and parasitic diseases
Respiratory	460-519, 786	Diseases of the respiratory system
Digestive	520-579, 787	Diseases of the digestive system
Diabetes	250.xx	Diabetes mellitus
Injury	800-999	Injury and poisoning
Other	290-319 E-V (Start with letters) 280-289 320-359 630-679 360-389 740-759	Mental disorders External causes of injury Diseases of the blood Diseases of the nervous system Complications of pregnancy or childbirth Diseases of the sense organs Congenital anomalies
Musculoskeletal	710-739	Diseases of the musculoskeletal system
Genitourinary	580-629, 788	Diseases of the genitourinary system

Appendix 2

The following codes are based on the ID-mappings provided by [*Clore et al. 2014*]

Admission type

Admission type	ID	# encounters	Notes
Emergency	1, 2, 7	48299	
Newborn	4	9	9 encounters - removed
Elective	3	13786	
Not available	5, 6, 8	7893	

Admission source

Admission type	ID	# encounters
Emergency room	7	37269
Referral	1, 2, 3	22790
Transfer	4, 5, 6, 10, 18, 19, 22, 25, 26	4836
Other	8, 11, 12, 13, 14, 23, 24	14
Not available	9, 15, 17, 20, 21	5069

Discharge disposition

Discharge disposition	ID	# encounters
Discharged to home	1, 6, 8	52675
Transferred	2, 3, 4, 5, 7, 10, 15, 16, 17, 22, 23, 24, 30, 27, 28, 29	14040
Remain patient	9, 12	11
Not available	18, 25, 26	3252

Appendix 3

Layout of confusion matrix:

		True value	
Pred		TP	FP
		FN	TN

Model	Confusion matrix	Accuracy	Sensitivity	Specificity	Training time*	Prediction time*
Logistic regression	TRUE FALSE TRUE 3273 1887 FALSE 666 856	0.6197	0.8309	0.3132	17.35 secs	0.55 secs
Lasso	TRUE FALSE TRUE 3285 1887 FALSE 654 846	0.6192	0.835	0.309	0.37 secs	0.013 secs
Neural Network	TRUE FALSE TRUE 3247 1850 FALSE 692 883	0.6138	0.8116	0.3286	4.01 min	0.087 secs
Decision Tree	TRUE FALSE TRUE 3395 2034 FALSE 544 699	0.6136	0.8619	0.2558	22.3 secs	0.55 secs
Random Forest	TRUE FALSE TRUE 3192 1839 FALSE 747 894	0.6177	0.8106	0.3396	21.76 secs	0.66 secs
XGBoost	TRUE FALSE TRUE 3201 1793 FALSE 738 940	0.6229	0.8162	0.3443	59.77 secs	0.075 secs

* Training/prediction time varies between runs. Time given as indication of efficiency.

Appendix 4

Table [4], we observe how the feature recording total number of visits (merged from outpatient, emergency and inpatient visits feature) is extremely positively skewed with a value of 6.69, which can be visualised on its histogram on the right of the figure below, the histogram for "Number of lab procedures" demonstrates a distribution with less skew. The large skew in visits is due to most patients not being admitted for any hospital visits within a year of their current admission.

Such a highly skewed feature might influence our modelling negatively. It could lead to an increased bias in the prediction, especially for regression problems. Moreover, for gradient-based methods – which we will be using later - skewed features might cause optimization issues, as the gradient might tend towards the more frequent values.

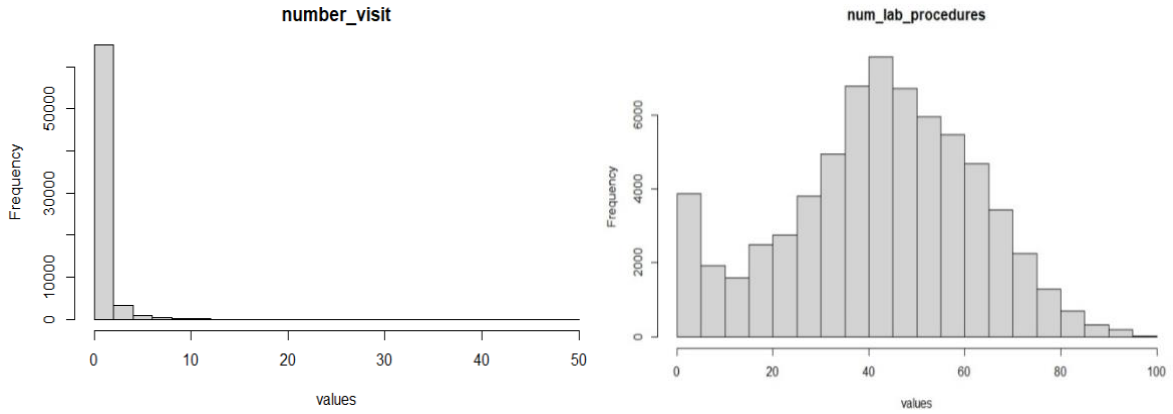

Figure - Graphs showing how distribution of data relates to skewness. Number of lab procedures has a skewness of –0.2195 (close to zero) compared to number of visits with skewness 6.69.

Therefore, we decided to rescale the feature using a simple logarithmic transformation. It is paramount to do such rescaling after the Training-Test set split as to avoid data leakage and biasing the predictions of the models. We first logarithmically rescaled the feature for the Training set which brought down the skewness from 6.83 to a reasonable value of 1.10 and then similarly for the Test set.

Appendix 5

Variable	p-value
race	2.35E-11
gender	5.26E-05
age	1.01E-38
admission_type_id	1.78E-28
discharge_disposition_id	0.002548517
admission_source_id	1.16E-62
time_in_hospital	8.97E-15
num_lab_procedures	3.85E-05
num_procedures	1.82E-08
num_medications	0.044534027
number_visit	7.54E-198
diag_1	1.91E-17
diag_2	3.51E-06
diag_3	5.87E-13
number_diagnoses	3.40E-46
max_glu_serum	0.040616752
A1Cresult	1.04E-13
metformin	0.939406833
glipizide	0.141597383
glyburide	0.454838289
insulin	0.004071102
diabetesMed_changed	0.313039452
diabetesMed	4.78E-28
readmitted	NA
num_medication_change	0.352752157
metformin.used	2.07E-10
repaglinide.used	0.002335352
glimepiride.used	0.634881546
glipizide.used	0.027731341
glyburide.used	0.676814621
pioglitazone.used	0.066907814
rosiglitazone.used	4.35E-05