Predictive Modelling for Early Hospital Readmission Among Diabetes Patients

Abstract

Diabetes is a chronic disease that can make the individual more vulnerable to other infections and health conditions (*Diabetes and your immune system*. 2023). The prediction of readmission for diabetic patients will benefit the healthcare system as well as the patient themself. We aimed to investigate the efficacy of machine learning algorithms in predicting patient readmission rates within 30 days of discharge from the hospital. We used a comprehensive dataset containing patients' demographics, clinical variables, and prior health history to implement a two-tiered methodology. The dataset was first cleaned and pre-processed and then K-means clustering was applied to segment the data into distinct groups based on patient characteristics. Each cluster was then modelled with local random forest classifiers. The study highlighted the importance of cluster-based classification in enhancing predictive accuracy. A comparative analysis between a baseline imbalanced logistic regression model, a balanced logistic regression model, and a global random forest model demonstrated that the cluster-based model had significantly good performance achieving higher F1, precision, recall and ROC AUC scores. The model cross-validation scores ranged from 87-89% highlighting the generalisability of the model across new data and different clusters. Our findings support the idea of integrating machine learning algorithms in the healthcare system. The study offers insights for healthcare providers to implement targeted interventions for patients and effective resource allocation using cluster-based algorithms. Future research directions include further exploring clustering algorithms and integrating more detailed and specific patient data to further refine the predictive accuracy.

Introduction

Diabetes affects more than 500 million people worldwide, and in the United States, it accounts for a quarter of healthcare spending (Parker et al., 2024). The US has the world's highest estimated expenditure on diabetes, followed by China and Brazil. It is predicted that the global expenditure on diabetes will increase to 825 billion USD by 2030. A large part of this cost is due to the high readmission rate, which was found to be significantly higher in diabetic patients compared to those without diabetes (Ostling et al., 2017). Readmission can contribute to expenditure due to having to repeat the same tests and procedures on the same patient, as well as spending on beds, medication, and personnel. If readmission occurs between different healthcare facilities, then additional costs can also be incurred due to poor coordination between the 2 facilities. Diabetics have a higher readmission rate due to the chronic nature of the disease, meaning they require ongoing management of diet, lifestyle and medication. This coupled with their propensity to have coexisting medical conditions makes them more likely to be readmitted.

Predicting the readmission rate of a diabetic patient would be greatly beneficial to healthcare providers as it would allow them to make necessary arrangements to reduce further spending. Prediction models can be built using a range of input data about patients. However, to build an accurate model you need many predictors, as well as a large sample population. This is an ideal application for machine learning. This study aims to use machine learning to create a model that predicts early readmission (<30 days) of patients with diabetes. This will be done with a dataset containing information about clinical care at 130 US hospitals between 1999 and 2008. It contains data on more than 100,000 patients and has 47 predictors.

This classification model could prove useful for hospitals worldwide, as the same features used in the model could be applied to predict the readmission rate of diabetic patients in other hospitals. The model could also be further developed to predict more than just readmission. For instance, it could predict the treatments and medications required for readmitted patients. This would further reduce the burden on healthcare providers, as well as improve patient outcomes post admission.

Data Cleaning and Transformation

Method:

The original dataset was imported from a csv file into a pandas dataframe so that it could be cleaned before the model could be made. The shape of the dataframe was checked using pandas, and then the ‘encounter ID’ column was dropped, as it was just an identifier used by the hospitals and had no correlation with

readmission. The original dataframe characterised missing values with '?', which we replaced with NaN, since NaN is recognised by python libraries. We then made a summary table showing the number of missing values for each feature, before and after cleaning.

The next step was to turn readmission into a binary response variable. In the original dataset, this feature had 3 levels, <30, >30 and NO. We only needed 2 levels, Readmitted and not readmitted, so we converted <30 into 1, and >30 and NO into 0. Following this we checked the datatypes of each column. Then we calculated the percentage of missing values in each column and deleted columns with more than 90% missing values.
Since columns of low variance would also not be of much use to the model, we identified and deleted the near-zero variance columns. We also deleted all rows that had NaN values. After this we printed summary statistics of the remaining 13 features. The statistics included counts, mean, standard deviation, minimum value, Q1, Q2 and Q3, as well as maximum value. These were then used to pick relevant features. Following this, the outliers were identified in the remaining columns, with the boundaries being set using the interquartile method.
At this point it was noted that the different features had highly varied ranges. This can make it difficult to establish correlations between features. For example, The number of diagnoses ranged between 3 and 9, but the number of lab procedures ranged between 13 and 65. To resolve the differences between features, scalar normalisation was applied. This allowed the features to be more easily compared, as they now had similar ranges and means. Following this, the shape of the final cleaned data frame was checked, in order to compare it to the starting dataset.

Results:

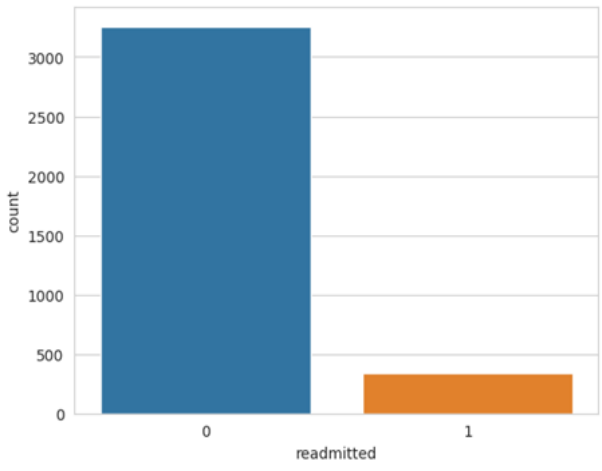
The initial data frame was shown to have a shape of 101766 rows and 50 columns.
Only 7 out of 42 features showed any difference in the number of missing values after cleaning. We deleted 19 columns which had zero or near zero variation. The biggest increase in missing values after cleaning was for weight, medical specialty, payer code, and race. At this stage, 13 of the features contained integer type data, and the rest were of datatype object, meaning they were categorical. When we calculated which columns had more than 90% missing values, we found max_glu serum to have 94% missing values, and weight to have 96% missing values, so these columns were removed from the dataset. 573 outliers were found and removed from the dataset as well. The 'num_lab_procedures' feature has the highest number of outliers and 'admission_source_id', 'admission_type_id', 'num_procedures', 'number_diagnoses' and 'time_in_hospital' have no outliers. The final cleaned data frame had 3593 rows and 30 columns.

Data Visualisation

Method:

Matplotlib (3.7.1) and Seaborn (0.13.1) python packages were implemented to visualise the relationships between the different features with the target variable in the cleaned dataframe. Using these packages we plotted the distribution of unique classes of the target variable (readmitted), a graph that shows the count of target variable against the number of medications, the scatter matrix plot, the correlation matrix and additional plots.

Results:



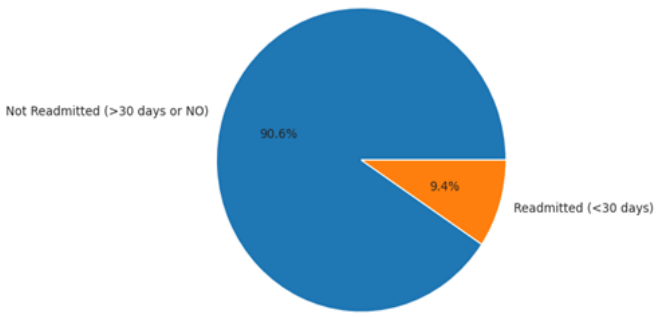


Figure 1. A comparative visualisation of the hospital readmission rates within 30 days. Left: histogram showing the number of patients that were not readmitted within 30 days or at all (0) compared to the number of patients that were readmitted to the hospital within 30 days (1). Right: pie chart showing the percentage of patients that were and were not readmitted within 30 days to the hospital, revealing a large proportion were not readmitted (90.6%) and a small proportion were (9.4%).

We observed that the data was imbalanced with 90.6% samples in the 'Not Readmitted (>30 days or NO)' and only 9.4% samples in the 'Readmitted (<30 days)' group (Figure 1).

Visualisation of Categorical Features:

To visualise the categorical features, count plots and pie charts were implemented. We observed that the readmission rates differed for different patient age groups (figure 2). The highest number of readmissions occurs in the 70-80 age bracket, both when looking at the total counts and when looking proportionally with the other age brackets. This suggests that age is a key factor for hospital readmission, with older patients being more likely to be readmitted within 30 days. The Pie chart further supported the findings of the countplot showing that the 70-80 age bracket accounts for 22.0% of patients that were readmitted. Moreover, the 60-70, 70-80 and 80-90 age brackets collectively accounted for over half (56%) of the readmitted patients, highlighting that age is a significant determinant for hospital readmission.

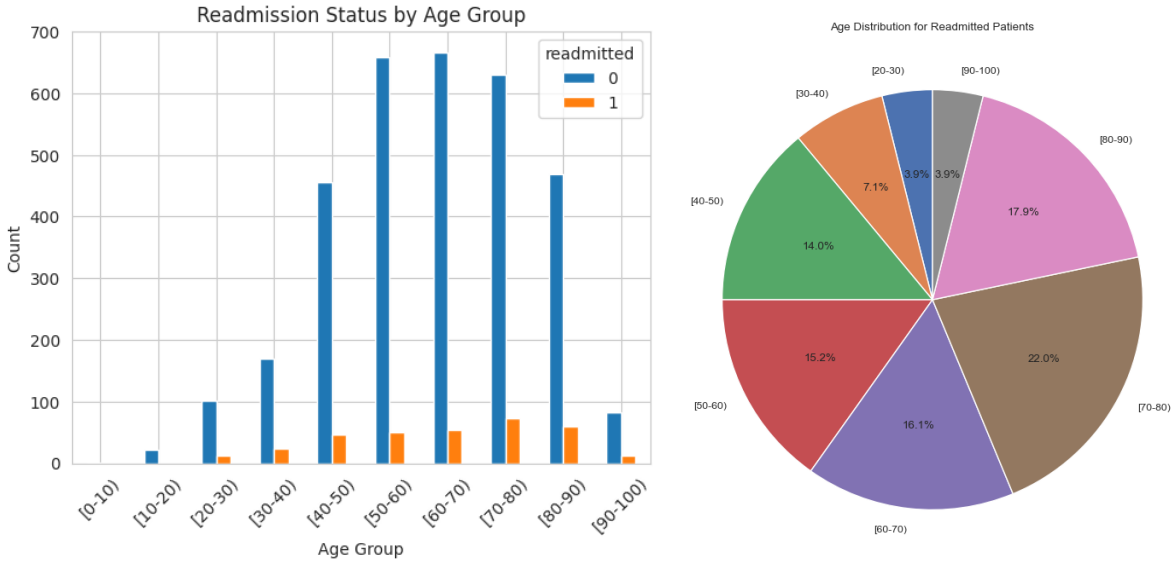


Figure 2. A comparative analysis of readmission rate across different patients' age groups. Left: Countplot showing patient readmission status categorised by age group. The blue bars represent patients that were not readmitted at all or not readmitted within 30 days (0). The orange bars represent patients that were readmitted within 30 days (1). Right: Pie chart showing the age group distribution of readmitted patients providing a percentage breakdown of each age group. The size of the slices in the pie chart are representative to the relative proportion of patients within each age group.

Figure 3 shows count plots for each categorical variable in the cleaned dataset. Through the analysis of these graphs we concluded that 'race', 'admission_type_id', 'discharge_deposition_id', 'admission_source_id', 'A1Cresult', 'metmorfin', 'glipizide', 'glyburide', 'pioglitazone', 'rosiglitazone', 'insulin', 'change' and 'diabetes_med' could be potential predictors for hospital readmission. These features showed varied counts across the different groups depending on readmission status, indicating a change in the feature increased the chances of readmission. For instance, 'race' shows that if the patient is Caucasian or African American they are likely to be readmitted and if the patient is Hispanic, Asian or Other they will not be readmitted. Likewise, having a change in your diabetes medication ('change') increases the chances of the patient being readmitted to the hospital within 30 days.

On the other hand, the feature 'payer_code' is unlikely to be a predictor for hospital readmission. This feature relates to the billing and insurance of the patient which may be less directly related to the patient's health outcome and therefore less likely to be a strong predictor for hospital readmission. While administrative factors can impact the patients care in the hospital, they might not directly cause the hospital readmission itself. This

gives this feature low predictive power for hospital readmission.

Visualisation of the Numerical Features:

Our examination of the dataset using violin plots revealed some trends in the numerical attributes concerning hospital readmission rates. Factors such as 'num_procedures', 'num_lab_procedures', 'num_medications', and 'num_diagnoses' emerged as indicators for readmission demonstrating notable variations in how these values are spread out their central tendencies and their quartile ranges between patients who were readmitted within 30 days and those who were not. In contrast, variables like 'time_in_hospital' 'num_emergency' 'num_outpatient' and 'num_inpatient' displayed less predictive power, with only slight variations in their distributions observed between the two groups.

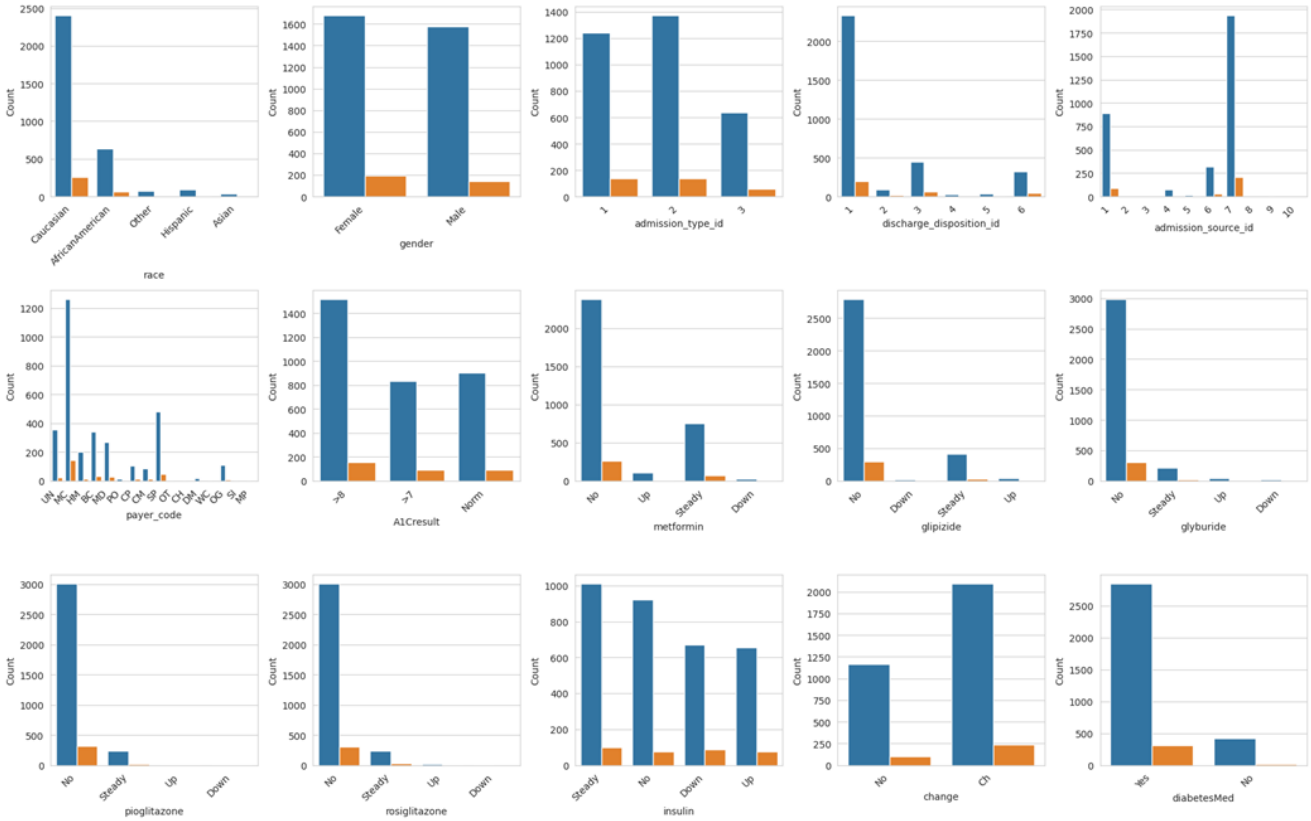


Figure 3. Count plots of various categorical features against readmitted status, readmitted within 30 days (orange) or not readmitted within 30 days/not readmitted at all (blue). For admission_type_id, 1: Emergency, 2: Urgent, 3: Elective. For discharge_deposition_id, 1: discharged to home, 2: discharged/transferred to another short-term hospital, 3: discharged/transferred to SNF, 4: Discharged/transferred to ICF, 5: discharged/transferred to another type of inpatient care institution, 6: discharged/transferred to home with home health service, 7: left AMA, 8: discharged/transferred to home under care of Home IV provider. For admission_source_id, 1: physician referral, 2: clinic referral, 3: HMO referral, 4: transfer from a hospital, 5: transfer from a skilled nursing facility, 6: transfer from another health care facility, 7: emergency room, 8: court/law enforcement, 9: not available, 10: transfer from critical access hospital.

Another figure we made was the correlation matrix plot (figure 4) and the scatter matrix, based on these matrices during the previous data analysis and experiment, a large insight number can be taken and can be seen in relation to the variables in our data set. In the picture, the scatter plot matrix displays the spread and possible patterns between two variables. You can see that the data points are spread out and don't follow a clear pattern or trend. This shows that most of the variables have a weak linear relationship with each other. If the data points are not grouped together or aligned, it means that changes in one variable don't always lead to predictable changes in another variable.

On the other hand, there is a moderately positive correlation (r = 0.47) between the number of diagnoses (number_diagnoses) and the length of stay in the hospital (time_in_hospital). These results suggest that having more diagnoses is slightly linked to having to stay in the hospital longer. There is also a strong negative

correlation of (-0.8) between the admission source (admission_source_id) and the admission type (admission_type_id). This means that certain types of admission are less likely to come from certain sources of admission.

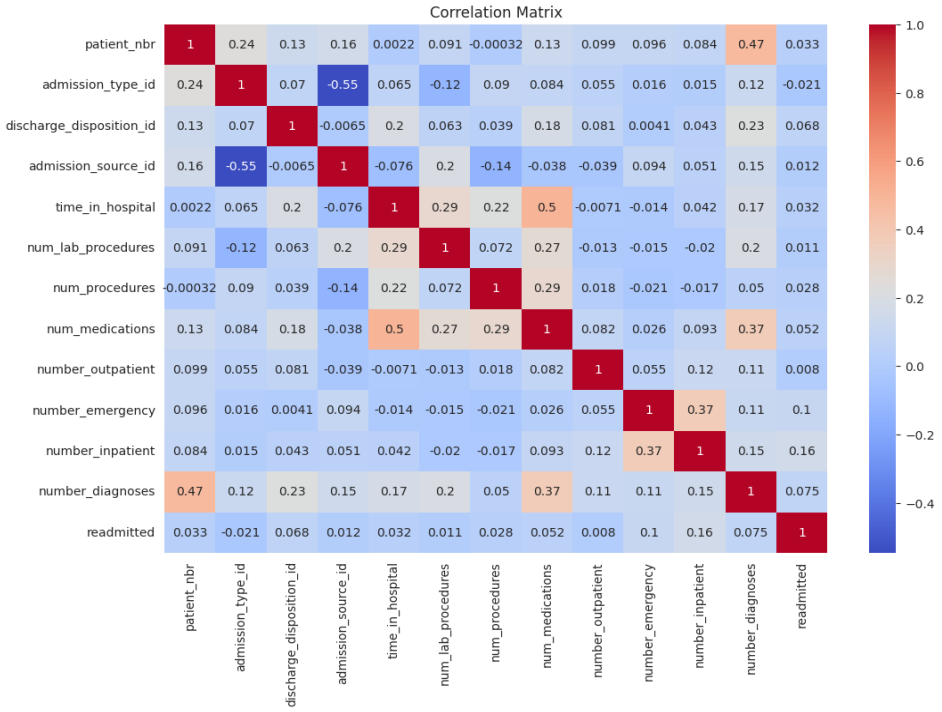


Figure 4. Correlation Matrix plot- This colourful heatmap shows how hospital admission factors (admission type, source, etc.) relate to discharge factors (length of stay, procedures, etc.). Deeper blue squares indicate stronger positive correlations (factors move together), while red squares suggest negative ones (opposite directions). Researchers use this to understand how admissions might influence discharges, improving resource allocation and patient care.

The correlation plot shows that most of the variables in the dataset are weakly connected in a straight line. A lot of data points have been separated away and the correlation coefficients are usually low, which means there isn't a strong linear relationship between them. But some pairs of variables, like number of diagnoses and length of stay, have weak negative correlations, while others, like admission type and admission source, have strong positive correlations.

Model Building

Method:

Our initial model was constructed using logistic regression. The first step was to select the appropriate predictors. This was done with the correlation matrix made during the data visualisation stage, using scores between each predictor and readmission. The predictors selected included time in hospital, number of diagnoses, age, admission type ID, discharge disposition ID, admission source ID, number of lab and non-lab procedures, and number of inpatients.

These predictors were then all stored in a single variable X, and the target variable (readmission) stored in a variable Y. Since several of the predictors were categorical, they were converted into numeric format, which is more accessible to machine learning algorithms. This was done with one-hot encoding, which is when each category in a feature is assigned a binary vector as a label. We did this with the .get_dummies() method from pandas. We then split the variables x and y into their own training, validation, and test sets, where 80% of the data is used for training and validation, and the remaining 20% for testing. We also set a random seed to ensure reproducibility when splitting the data into sets.

These sets, as well as the same random seed, were then used to train and test the logistic regression model. The sets were divided into 5 subsets, 4 of which would be used for training and the fifth used for testing. This was done 5 times, with each of the 5 sets used as a test set and the cross-validation score recorded and the mean taken. The accuracy of the training and test sets was also calculated.

The model would then be evaluated with different metrics such as y pred, accuracy, precision, accuracy, recall, f1 score and ROC scores. After this, the data was balanced and the model was retrained. Balancing was required due to the readmitted patients being in the minority in the dataset, resulting in a class imbalance for readmission. We rebalanced the dataset using synthetic minority over-sampling technique (SMOTE), a processing technique that generates synthetic samples for the minority class, in this case, readmitted patients. It does this using differences between readmitted samples and their nearest neighbours, multiplying the difference by a random number between 0 and 1, and adding that to the sample. This is repeated until enough readmitted samples are generated to have equal numbers of readmitted and non-readmitted patients in the dataset.

After this, a logistic regression model is fitted to the balanced X and y training sets, with the same random seed being used as before. This model is then used to make predictions about the test set. As before, the model was run 5 times, allowing for every combination of training and test sets. The 5 cross-validation scores and their mean were recorded. Following this, the performance metrics for the balanced, retrained model were calculated, to determine the improvement of the model.

Results

The cross-validation scores of the 5 sets for the original unbalanced model were 0.9065, 0.9, 0.9087, 0.9043 and 0.9063. This gave a mean score of 0.9052. After balancing, the model gave cross-validation scores of 0.6695,0.8029, 0.8089, 0.8195 and 0.8195, giving a mean of 0.7841. The performance metrics of the initial and retrained models are given in Table 1. The change in ROC curves can also be seen in figure 9. Balancing and training the model has resulted in a higher AUC score, which can be seen as a larger area under the graph. This suggests that the model has improved in its ability to distinguish between positive and negative results.

	Imbalanced Model			Balanced Model		
	Train set	Val set	Test set	Train set	Val set	Test set
Accuracy	0.9056	0.9043	0.9138	0.7855	0.8765	0.8873
Precision	0.6667	0.0000	0.3333	0.9316	0.1364	0.2059
Recall	0.0273	0.0000	0.0164	0.6162	0.0.545	0.1148
F1 Score	0.0524	0.0000	0.0312	0.7417	0.0779	0.1474
ROC AUC	0.5129	0.5000	0.5067	0.7855	0.5090	0.5369
Cross-val Score (mean)		0.9052			0.7840	

Table 1. Performance metrics of the initial model, before balancing and after balancing and retraining. Shown in the table are the different evaluation metrics (accuracy, precision, recall, F1, ROC AUC and cross-val (cross-validation) scores) used to evaluate the training, validation and test set to evaluate the models performance.

Improved Model Building

Method:

For the enhanced model we took two approaches, one model was a clustered-based Random Forest model and the second was a global Random Forest model.

Data Cleansing:

Before building the models, we reinserted the original dataframe to preprocess and cleanse the data. The ‘encounter_id’ and ‘patient_nbr’ were removed and the ‘?’ values were replaced with NaN values. Consequently, the percentage of missing values were calculated and features that contained more than 95% missing values were removed. Features that had an extremely near-zero variance of a threshold of 99% rounded

to two decimals removing 15 features. And "gender" entries with the value of "Unknown/Invalid" were removed. For feature engineering, we changed the 'age' feature values from the age ranges to integers (each age range was assigned to its midpoint value, i.e., [10-20] was assigned the integer 15) and the 'readmitted' feature was made binary (0 and 1 values).

We checked if the data was balanced or not and decided to use RandomUnderSampler to under sample the data so it balances the two 'readmitted' classes. Categorical features were then imputated using 'most frequent' imputation method and 'mean' imputation was used for numerical features. RobustScaler was used to normalise the data as this normalisation technique is not heavily influenced by the presence of outliers. Label encoding was used to encode the various categorical features in the data.

Clustering and Modelling:

To identify the features with strong predictive power for the target variable ('readmitted') we initiated a base RandForestClassifier to extract the feature importance values (these were converted to percentages to increase interpretability). We also used the recursive feature elimination with cross-validation (RFECV) from the scikit-learn package to identify the optimal number of features to include in our data to achieve optimal cross-validation results (the scoring parameter was specified to accuracy and the cross-validation parameters was 5). A graph was plotted showing the cross-validation scores against the number of features.
The first model we built was the global random forest model. For this model, we split the data into a training, testing and validation set (training set being 60% of the data and the testing and validation sets being 20% of the data). The random forest classifier was built with the following parameters: n_estimators = 300, max_depth = 10, max_features = 23, min_samples_leaf = 50. This model was built on the entire dataset acting as an unclustered model. The aim of this was to use it to compare it to the clustered-based model to see if clustering affects the model performance or not. The confusion matrix, accuracy, precision, recall, F1, and roc auc scores for the testing, validation and training set were printed for analysis.

For the cluster-based model, we implemented the elbow method to identify the value of k. Based on the plotted elbow analysis graph, the cluster distance graph and the pca graph we decided the value of k was 4. Having identified the k-value, we implemented k-means clustering (with the parameters as; 'init = k-means++, n_init = 10, max_iter = 300, random_state = 42) and added the cluster information back into the dataframe as a new column. A for loop was used to build local random forest classifiers for each cluster present in the data. The parameters for the random forest classifier were the same as those used in the global mode.

Evaluation of the Models:

The confusion matrix, accuracy, precision, recall, F1, and ROC AUC scores for the testing, validation and training set were printed for analysis for both the global and cluster-based random forest model. Printing the training set metrics allowed us to ensure overfitting or underfitting was not occurring as we could compare those metrics to the testing and validation set metrics. Learning curves were also plotted showing the increase or decrease of the model accuracy against the data frame size for the training set, validation set and the cross-validation results.

Results:

Data Cleansing:

The original shape of the data has 101,766 rows with 50 columns. After cleaning and processing this was reduced down to 101,763 rows with 32 columns. Initial analysis of the 'reamitted' classes showed a large imbalance with 90,406 not readmitted and 11,357 readmitted. After balancing using the RandomUnderSampler we had 9,088 not readmitted and 9,088 readmitted. A third insignificant class was detected for the 'gender' feature which only occurred in 3 instances. Consequently, this class; 'Unknown/Invalid' was removed from the data. Detection of the near-zero variance columns resulted in the removal of 15 features. 23 features were identified as being the optimal number of features for the model by the RFECV analysis. Coincidentally, these 23 features also showed the highest importance, the 23 features are:- race: 1.82%, gender: 1.68%, age: 4.57%, admission_type_id: 2.68%, discharge_disposition_id: 4.42%, admission_source_id: 2.39%, time_in_hospital: 5.32%, payer_code: 3.11%, medical_specialty: 3.64%, num_lab_procedures: 8.87%, num_procedures: 3.76%,

num_medications: 7.52%, number_outpatient: 1.93%, number_emergency: 1.51%, number_inpatient: 5.27%, diag_1: 8.60%, diag_2: 8.40%, diag_3: 8.25%, number_diagnoses: 3.63%, metformin: 1.38%, glipizide: 1.25%, insulin: 2.73%, and change: 1.22%.

Model Evaluation:

The global random forest model mean values were 0.9948 (validation set) and 0.8894 (test set). Table 2 shows the different evaluation metrics for the global random forest model. The clustered-based random forest model obtained mean cross-validation scores of 0.8853 (cluster 0, validation set), 0.8837 (cluster 0, test set), 0.8882 (cluster 1 validation set), 0.8874 (cluster 1, test set), 0.8786 (cluster 2, validation set), 0.8833 (cluster 2, test set), 0.8967 (cluster 3, validation set), and 0.8923 (cluster 3, test set). The evaluation metrics for the cluster-based random forest model are shown in figure 5.

Global Random Forest Model			
	Training Set	Validation Set	Test Set
Confusion Matrix	[[4,383, 2,380] [2,220, 4,543]]	[[10,944, 7,066] [879, 1,464]]	[[10,872, 7,230] [821, 1,430]]
Accuracy	0.6599	0.6096	0.6044
Precision	0.6600	05486	0.5474
Recall	0.6599	0.6163	0.6180
F1 Score	0.6599	0.5015	0.4960
ROC AUC	0.6599	0.6163	0.6180

Table 2. Performance metrics of the global random forest model. Shown in the table are the different evaluation metrics (accuracy, precision, recall, F1, ROC AUC and cross-val (cross-validation) scores) used to evaluate the training, validation and test set to evaluate the models' performance.

Figure 5. A grouped bar chart visualising the evaluation metrics for each cluster's test set from the cluster-based random forest model. The accuracy, precision, recall, F1, ROC AUC and cross-val score is analysed for each cluster's test set. The label in the bar explicitly

shows the value of the bar. Performance across each cluster was maintained well across each cluster (this can be seen as each metric has similar values for the same metric in a different cluster).

Discussion

Initial Model

The imbalanced logistic regression model showed a high accuracy score of 90.43% for the validation set and 91.38% for the test set. However, when we analysed the f1 score we noticed significantly low scores of 0.000 (validation set) and 0.000 (test set). The precision (0.000 for validation and 0.3333 for the test set) and recall scores (0.000 for the validation and 0.0164 for the test set) were also significantly low. This discrepancy is due to the model favouring the majority class ("Not readmitted") so the model fails to capture the complex relationships in the minority class ("readmitted"). This suggested to us that the model may be overfitting, so it fails to predict the readmitted patients in the unseen data. The poor ROC AUC scores (0.5090 for validation and 0.5369 for test set) further confirms the model's inability to distinguish between the classes effectively in new unseen data. We computed the confusion matrix which revealed the model predicted 656 true negatives, 2 false positives, 60 false negatives and 1 true positive. This provides evidence that the model is underperforming on the minority class compared to the majority class for the target variable. The learning plot shows that overfitting is occuring in the data. The training F1 score is consistently higher than the validation F1 score and the score fluctuates as the training data size increases. The F1 scores for the training and validation sets do not converge indicating overfitting is present.

The balanced logistic regression model shows a high accuracy, 87.65% for the validation set and 88.73% for the test set. However, just like the imbalanced model there is a significant drop in the other evaluation metrics; F1 scores, validation set was 0.0779 and test set was 0.1474, the presion was 0.1364 for the validation set and 0.2059 for the test set, the recall was 0.0545 for the validation and 0.1148 for the test set. Finally, the ROC AUC was 0.5090 for the validation set and 0.5369 for the test set. We computed the confusion matrix which revealed the model predicted 631 true negatives 27 false positives, 54 false negatives and 7 true positives. This provides evidence that the model is underperforming on the minority class compared to the majority class for the target variable. Despite applying the oversampling to balance the data the model still failed to predict the minority class accurately. The learning curve plotted suggests that there is a certain degree of overfitting taking place as the accuracy of the training set decreases significantly as training data size increases. However, towards the end of the plot the training and validation accuracy scores begin to converge which implies to us that overfitting is present but at a smaller extent compared to the imbalanced logistic regression model.

Enhanced Model

For the enhanced model we decided to keep the outlier in the data. This is because given the fact that the data is medical patient information what may classed as an outlier may just be a rare medical case that could help predict future cases. Due to the fact that the 'outliers' are present in the data and the initial model used oversampling and performed poorly we decided to use an undersampling method (RandomUnderSampler) and RobustScaler for normalisation as it is not that easily influenced by outliers (Nitin Vashisth, 2020). We also decided to change the threshold for near zero variance columns from 99% instead of the original 95% threshold as well as changing the threshold from 90% to 95% for dropping the columns with a high percentage of missing values. These threshold changes were made so we could get a larger set of data for the enhanced model compared to the initial model but is still cleaned. We analysed the 'patient_nbr' column and noticed that there were duplicate entries that had different 'encounter_id' values. This implied to us that the same patient was readmitted in the same hospital so we decided to keep these duplicates and just delete the 'patient_nbr' columns as they could provide significant relationships to the model. Also, to maximise the amount of data, appropriate imputation was carried out instead of deletion of rows. LabelEncoder was used to encode the categorical data into integers instead of OneHotEncoder because the use of OneHotEncoder increased the dimensions of the dataframe significantly which would make our model vulnerable to overfitting.

The evaluation metrics of the global random forest model indicates moderate performance on the validation and test set (60.96% accuracy for the validation and 60.44% accuracy for the test set). The confusion

matrix indicates that the model can effectively identify the different readmission statuses in unseen data. The ROC AUC is higher than 0.5 (0.6163 for the validation and 0.6179 for the test set) showing that the model's prediction is based on relationships it has learnt from the training set and not just guessing. The pression, recall and F1 are relatively high indicating the model can accurately detect true positive cases and the learning curve shows the model performs well with no underfitting or overfitting occurring.

The cluster-based local random forest classifier model shows commendable performance across all four clusters. The precision metrics and recall metrics reveals that the model can efficiently predict true positives from unseen data with a notable precision metric of up to 67.39% indicating a significant ability to minimise false positives. This is a significant capability in the medical field where overlooking a positive instance may have severe consequences. The F1 scores across all of the clusters are above 0.5 and have ROC AUC values of up to 0.6207 indicating the model can effectively distinguish between the positive and negative instances for the target variable 'readmitted'. The cross-validation scores 0.88 to 0.89 between the four clusters. This shows the model can effectively generalise and apply the predictions to unseen data. The learning curve plot and consistency in the metrics from the training sets to the validation set and testing set indicates the model performs well with no overfitting or underfitting occurring.

Global Model versus Cluster-based Model

Transitioning from the global modal to the cluster-based model with local classifiers shows an enhancement in model performance for the cluster-based model. By incorporating the K-Means clustering into the preprocessed dataset we can group the data into different distinct groups based on the patterns and similarities found amongst the patients. Then the implementation of a local random forest classifier for every individual cluster within the data allows us to tailor that classifier for that specific cluster. This allows the model to learn the complex relationships for the cluster more efficiently. Also, medical data, such as the data used to build our model, is known for being diverse and complex so shifting from a global model to a tailored specific clustered-based model addresses the complex and diverse nature of medical data.

The clusters-based model offers a more nuanced understanding of patients' readmission by leveraging the different characteristics of each cluster present in the data to predict outcomes compared to the more generalised global model that uses random forest on the data as a whole entity. While the global model showed improvements in performance (compared to the initial logistic regression models) through careful predictor selection and data cleaning, it was still limited in the information it could learn about the cluster-specific relationships.

The cluster-based models show a moderately higher performance than the global model performance through several evaluation metrics. Tailoring the classifiers to specific clusters enabled the models to achieve higher precision and recall scores indicating a more effective prediction of true positives and better handling of the minority class. The cross validation scores are slightly higher in the cluster-based model indicating these models are better generalised to unseen data as they can maintain good performance without overfitting and underfitting occurring. Finally, the ROC AUC evaluation metric is higher in the cluster-based approach indicating that the cluster-based model has a stronger ability to distinguish between the two classes which is a critical advantage when it comes to predicting patient readmission, where the cost of false negatives can be high.

In essence, the clusters-based model represents a significant improvement compared to the initial logistic regression and global random forest model showing marked improvements in performance metrics and the ability to generate actionable insights.

References

Diabetes and your immune system. (2023) Available at: https://www.cdc.gov/diabetes/library/features/diabetes_immune_system.html (Accessed: 27.03.2024).

Nitin Vashisth. (2020) 'Dealing with imbalanced dataset (UnderSampling)', *medium.com*, May 26,. Available at: https://medium.com/analytics-vidhya/dealing-with-imbalanced-dataset-undersampling-4e9b488a97c6 (Accessed: 27.04.2024).

Ostling, S., Wyckoff, J., Ciarkowski, S.L., Pai, C., Choe, H.M., Bahl, V. & Gianchandani, R. 2017, "The relationship between diabetes mellitus and 30-day readmission rates", *Clinical diabetes and endocrinology*, vol. 3, no. 1, pp. 3.

Parker, E.D., Lin, J., Mahoney, T., Ume, N., Yang, G., Gabbay, R.A., ElSayed, N.A. & Bannuru, R.R. 2024, "Economic Costs of Diabetes in the U.S. in 2022", *Diabetes care*, vol. 47, no. 1, pp. 26-43.