

Predicting the readmission of diabetes patients

IN3062: Introduction to AI



Group 13

# Introduction

Hospital readmissions are a critical challenge in healthcare, particularly for diabetes management, a chronic disease affecting millions globally. Frequent readmissions, especially within 30 days of discharge, signal gaps in care, increase patient morbidity, and strain healthcare resources. Predicting these readmissions is essential for improving outcomes, reducing costs, and designing targeted interventions.

This study analyses a dataset of over 100,000 hospital stays (1999–2008) from 130 US hospitals, containing patient demographics, clinical records, laboratory results, medications, and outcomes. Using this data, the study aims to identify factors influencing 30-day readmissions and develop predictive models to enhance diabetes care. The dataset contains only discrete values, as even values such as glucose levels are give as over and under certain amounts.

Diabetes is a chronic condition characterized by high blood glucose levels due to the body’s inability to produce or effectively use insulin. It is categorized into two main types: Type 1 and Type 2. Type 1 diabetes, often diagnosed in childhood or adolescence, is an autoimmune disorder where the immune system destroys insulin-producing beta cells in the pancreas, requiring lifelong insulin therapy. Type 2 diabetes, more common and typically associated with adults, results from insulin resistance and a gradual decline in insulin production. Contributing factors include obesity, sedentary lifestyles, and genetic predisposition. Both types increase the risk of complications such as cardiovascular disease, kidney failure, and nerve damage (World Health Organization, 2021; American Diabetes Association, 2023). Effective management includes lifestyle interventions, medication, and continuous monitoring of blood glucose levels.

## Problem Domain

Diabetes management is complex, involving lifestyle, medication adherence, and comorbid conditions. Despite advancements, many diabetic patients face frequent readmissions, leading to increased morbidity and healthcare costs. Predicting 30-day readmissions is crucial for identifying care gaps, implementing preventive measures, and reducing hospital penalties for high readmission rates. This study will use machine learning and statistical analysis to identify patterns and improve predictions, enhancing patient care and resource optimization.

## Objectives and Research Questions

This study addresses the following:

1. Key Factors Influencing Readmissions: What demographic, clinical, and treatment variables drive early readmissions in diabetic patients?
2. Predictive Accuracy: Can machine learning models classify patients as "readmitted" or "not readmitted" within 30 days?
3. High-Risk Subgroups: Which patient characteristics (e.g., age, treatment, comorbidities) are linked to higher readmission risks?
4. Unsupervised Patterns: How can clustering techniques uncover shared risk factors or care needs among similar patient groups

# Preprocessing

Our dataset, while thorough, still requires some data preprocessing to handle missing values, encode categorical features, and prepare the data for the machine learning models we will be using. It contains features related to patient demographics, administrative details, clinical indicators, and hospital metrics, with the goal of predicting readmission outcomes. The end goal is the prediction of 3 discrete outcomes, readmitted >30 days, readmitted <30 days and not readmitted at all meaning that it is a classification problem.

## Preprocessing Steps

Initially, the dataset contained a lot of missing or null values, as well as many features containing values that were unsuitable to feed a model e.g. the race column was of type string. The first step was to remove columns that were unnecessary: encounter\_id, patient\_nbr and payer\_code all are unique identifiers for administrative purposes and had no details that would affect a diagnosis. The first column that was processed was the diagnosis column, with all Vs being replaced with “10”. Since the diagnosis represents a unique code, if the same is applied to every value the values should retain their uniqueness.

In the data we also had to manage many missing values in weight and medical specialty. Since weight is one of the most key factors in patients with diabetes (American Diabetes Association, 2024) we decided that it was appropriate to keep the column despite all of the missing information. Medical speciality is linked to the type of doctor that diagnosed the patient. While this may play an important factor in the initial diagnosis, there is no evidence to suggest different doctors may perform any better or worse so dropping the column is appropriate. For features that had a few hundred to a couple of thousand null values, the rows were deleted from the dataset.

## Exploratory Analysis Summary

For reference, we used data visualisations of correlation matrices and bar charts to dictate which data columns were important based on their relationship to the target variable (‘readmitted’). Highly correlated variables with one another were flagged for a dimensionality reduction or exclusion. While strongly correlated features were prioritised for model development. The correlation matrix below shows whether variables tended to increase/ decrease with one another and by how much.

The bar charts focus on the relationship between the individual features and the target variable (‘readmitted’). These charts highlighted the variables with the strongest positive or negative correlations so that we knew which features to prioritise for further analysis when aiming to improve the strength of our model.

### Standardisation and Data Splitting

Features were standardised to ensure consistent scaling across models. The data set was split into training and testing sets at a ratio of 75/25 with cross validation applied to aid in the evaluation of the model.

# Report Methodology

## Hypothesis:

The AI model will be able to accurately predict whether a patient that has had diabetes will be either readmitted or not at all. Both the accuracy and precision score must be above at least 60% to ensure predictions made are valid and suitable for use within a medical environment.

## Training and evaluation methodology:

The models used in this report are; Perceptrons, Decision Trees, K-Random Forests and Neural Networks. We have chosen the latter 4 models as they were the highest ranking in accuracy, precision, recall and f1-score in our preliminary tests. The preliminary tests consisted of using the same pre-processed dataset for all the models, initialising all models with no parameters, scaling the dataset to models that would require it and allowing unlimited computation time for models. We did this to filter out models that are not great for our use case, so we can devote more time to tuning the models that were performing well initially. Perceptrons, although not the highest ranking, we included it to act as a baseline and also to get us familiar with the dataset and make adjustments quickly, as the model only took 0.3 seconds to execute.

When training and evaluating, we ensured that all models began using the same dataset configurations (i.e. dropped columns, filled with average or median values, one-hot encoded. The dataset is split into training and testing subsets, with 75% dedicated to training and 25% to testing, this was kept consistent between all models. We would then initialise a baseline result for each model and iteratively improve by changing hyper parameters and modifying the dataset values. To evaluate each model we would measure performance by calculating the accuracy, precision, recall and f1 score, as well as plotting a confusion matrix for a visual representation of the results.

We also took readings of computation time when evaluating our models as the end model may be used in a less powerful system (i.e. old machines in GPs / hospitals) and performance may be vital to ensure doctors/nurses can receive the patients readings fast, allowing them to see more patients from the time saved in computing.

### Perceptron

An advantage of this model was that it was quicker to run than the other models by 98% , executing at just 0.30 seconds, meaning we could update the hyper parameters and adjust the dataset iteratively without much time being lost from computation. This would also be useful in a hospital situation, where old hardware will still be able to run the model without difficulty.

A disadvantage that we found early on, was that it struggled to learn from the dataset and make accurate predictions. The dataset being derived from human attributes (weight, age, glucose levels etc.) caused a lot of noise from the randomness in nature, as well as containing over 100 '000 values, this simple model struggled to find a correlation and could not exceed a score of 0.53 in accuracy. Even after extensive parameter tuning and modifications of the dataset using standard scalers, the model failed to make accurate predictions and maxed out at 0.53.

### Random Forest:

An advantage of random forest was its ability to handle large datasets <Ref to back this point>, which is particularly useful in this case as the dataset contained over 100 '000 values. Another advantage we found was that it handled the complexity and randomness of the data well, <add data backed reasoning>., as the model works by splitting the data into random sets, then train a single tree per split set and then takes the average, this allows the trees to account for the complexity and randomness of the data much more easily compared to other models, which is reflected in its low computation time and high accuracy score <ref from testing>.

A disadvantage would be that it, although it reached the highest accuracy score in the lowest computation time, it was still quite computationally expensive to run, as it used lot of processing power and memory to handle the 100’000 rows of data. It would take around <20s? Ref> to execute which meant that the iterative process of improvement through tuning was slow. <Add timings and resource usage here>.

### Neural Networks

Neural networks excel at capturing complex, non-linear relationships in data, making them powerful for large datasets with intricate patterns. Their ability to generalize improves with larger datasets, as they learn more robust features. Additionally, hyperparameter tuning allows for adaptability to specific data characteristics. However, neural networks are computationally intensive, requiring significant processing power and time, particularly with large datasets. They are also prone to overfitting without proper regularization and can be challenging to interpret compared to simpler models.

For our large dataset, a neural network is suitable due to its capacity to handle high-dimensional data and learn complex relationships. The primary drawback is computational cost, but with appropriate tuning and resources, it is a viable and effective choice.

## Results:

### Perceptron

We chose a perceptron model first as it is a simple neural network, which will allow us to focus on formatting the data and shaping it into a useable form (i.e. one hot encoding, dropping columns, etc.) instead of debugging and modifying the model itself. It works by performing binary classification on a set of input features and outputs the data into two categories, which is suitable for our objective as we need to predict whether the patient will be readmitted or not.

With this model we were able to achieve an average score of 0.53 across all metrics (accuracy, precision, recall and f1), the average execution time was 0.30 seconds.

## Random Forest

First we achieved a base line accuracy score of 0.64 for the Random Forest model. We then wanted to explore the n\_estimators, criterion and max\_depth, as we believed these hyper parameters would have the most impact on the models performance, namely accuracy score and execution time.

We first selected n\_estimators as it controls the number of trees used by the model. The number of trees plays a pivotal role in accuracy score <ref>, because as the number of trees increases, the accuracy score also increases. We tested the n\_estimators at 3 values; 30, 80 and 130. We found that at 30 trees, the accuracy score dropped to 0.63 and that after 80 trees the model does not increase in performance from 0.64 accuracy and only increases computation time, so the best parameter for n\_estimators is 80.

Next, we focused on modifying the criterion hyper-parameter with values gini and entropy. On average it is found that gini computes in a quicker time by splitting into trees more frequently <ref> and that entropy is more selective when splitting into new trees, this allows entropy to make better decisions and improve on the accuracy score <ref>. However, we found that entropy made no difference in accuracy score and only increased computation time <Add timings>. Due to this we decided to keep the default value of gini.

We chose max\_depth as third most impactful parameter and to explore it next because trees which are too deep can cause overfitting <ref>, since the default value is set to None, it would be worth well exploring this parameter as even in the event of no gain in accuracy score, the models training speed could be improved as lower depth trees can be computed in less time <ref>. When testing with the values 10, 20 and 50 we found no gain in accuracy, but saw a decrease in training speed when using X value by Y percent, so X was chosen as the optimal value.

Brief explanation of other parameters chosen value:

The hyper parameter bootstrap was set to true, this creates randomness as each tree is trained on a different subset of the data, reducing overfitting. Ccp\_alpha was set to 0.0, to avoid pruning of trees, this allows the trees to grow to their full depth and find trends within the data. Class\_weight was set to balanced, as the data contain more patients that were not readmitted compared to admitted <add numbers>. Max\_features was set to sqrt, as this would lead to fewer features being used when finding a split, so the model avoids picking the most prevalent feature in each split and over fitting. Max\_leaf\_nodes was set to None, this allows the tree to have an unlimited number of nodes so that it is not constrained when growing and can improve model accuracy. Max\_samples was set to None, so that the model can utilise the entire data set when training and therefore reduce under fitting. Min\_impurity\_decrease was set to 0.0, this ensures that the impurity is not a factor and the overall model's performance is prioritised. Min\_samples\_leaf is set to 1 to allow the tree to grow as the data requires and not be limited by a high minimum leaf count. Monotonic\_cst was set to false as we did not require constraints on the input features and instead wanted the model to find complex patterns in the data. N\_jobs was set to None, so that only one CPU core was used, this was done to replicate the lower end machines that hospitals might use. Oob\_Score was set to False as cross validation was already being performed. Random\_state was set to 42 to ensure all models use the same state and the scores were reproducible.Warm\_start was set to False, to stop the model from reusing previously fitted trees, reducing the chances of over-fitting. Verbose was set to 0, as we did not require output logs as the model was training.

<Discuss score here>

### Neural Network

The initial baseline accuracy of the Neural Network (NN) model was 0.53, which served as a reference point for further optimization. Neural networks are particularly effective for identifying non-linear relationships in data, making them suitable for our dataset's complexity. The most impactful features included key parameters related to patient admissions, demographic details, and medical history. These features were processed using one-hot encoding to ensure compatibility with the NN model.

To improve performance, we focused on optimizing three key hyperparameters: the number of hidden layers, the number of iterations, and the regularization parameter (alpha). These parameters were chosen due to their significant influence on both the model's accuracy and its computational efficiency.

The number of hidden layers was identified as a critical factor in model performance. Hidden layers allow the network to capture deeper patterns in the data. We tested configurations with 50, 100, and 200 nodes in the first layer, paired with 1 node in the second layer.

With 50 nodes, the model achieved the highest accuracy score of 0.61, compared to 0.53 for configurations with 100 or 200 nodes. The additional nodes did not improve accuracy but increased computational time, with execution times rising from 6.17 seconds (50 nodes) to 7.96 seconds (200 nodes). Based on these results, we fixed the hidden layer configuration at (100, 1), balancing accuracy with training efficiency.

The max\_iter parameter was tested at values of 10, 100, and 1000 to determine its effect on model convergence and performance. At 10 iterations, the model achieved an accuracy of 0.48 but produced convergence warnings, indicating insufficient iterations for optimization. Increasing the iterations to 100 improved accuracy to 0.57, with balanced precision and recall scores. Extending iterations to 1000 resulted in negligible performance gains, with an accuracy of 0.56, but significantly increased computation time to 23.14 seconds. Therefore, the optimal value for max\_iter was determined to be 100.

The alpha parameter controls regularization strength to prevent overfitting. We tested values of 0.0001, 0.001, 0.01, and 0.1. At 0.0001, the model achieved an accuracy of 0.54, while higher values improved accuracy. An alpha of 0.1 yielded the best performance, with an accuracy of 0.60 and an F1 score of 0.58. Beyond this value, performance plateaued, and computation times increased. Thus, an alpha value of 0.1 was selected.

Evaluation:

Conclusion:

References

- American Diabetes Association (2023) \*Standards of medical care in diabetes—2023.\* Diabetes Care. Available at: https://diabetesjournals.org

- World Health Organization (2021) \*Diabetes.\* Available at: https://www.who.int/news-room/fact-sheets/detail/diabetes