Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 2, 1803-1829
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i2.4929

Comparison of machine learning algorithms in predicting hospital readmissions for diabetic patients

Shefqet Meda[1,2*], Lily Cuku[1], Zhilbert Tafa[2]
[1]Canadian Institute of Technology, Tirana, Albania; shefqet.meda@cit.edu.al (S.M.) lily.cuku@cit.edu.al (L.C.)
[2]International Balkan University, Skopje, North Macedonia; tafaul@t-com.me (Z.T.)

Abstract: Hospital readmission is a substantial burden on healthcare systems, increasing both costs and patient strain. Current traditional methods for identifying patients at high risk for readmission are often based on clinicians' judgment. This issue is especially concerning for patients with chronic conditions like diabetes, where the pressure to reduce readmissions may lead to worse outcomes. Unlike traditional methods, machine learning algorithms can analyze complex datasets to identify patterns and risk factors, resulting in more precise predictions of readmission risk. They can also facilitate better resource allocation and personalized patient care. Previous studies have applied various algorithms to predict readmissions in healthcare institutions. In this study, we apply and compare the optimized versions of different machine learning (ML) models to predict 30-day readmissions and identify important predictors driving these outcomes. Based on the predefined metrics, the analyses identify the Stochastic Gradient Descent Classifier (SGDC) as the best-performing model for the available dataset and the applied ML parameter optimization. Although ML models demonstrate potential for predicting readmissions, they are not yet fully reliable.

Keywords: *Classification models, Diabetes readmission, Healthcare, Machine Learning, Predictive modelling, Readmission risk prediction.*

1. Introduction

1.1. Background

The increasing number of diabetes cases is alarming, as this condition is rapidly becoming a global epidemic and a public health issue. As reported by IDF Diabetes Atlas [1] in 2021, 537 million adults were diagnosed with diabetes and that number is expected to increase to over 783 million by 2045 [2]. The alarming number of people suffering from diabetes is likely to lead to an increase in the number of patients being readmitted to hospitals. Persistent readmissions are a significant concern due to their adverse effects on patient health outcomes and the additional expenses they incur. To address this issue and improve quality and outcomes in the health care system the Centres for Medicare & Medicaid Services (CMS) [3] established the Hospital Readmission Reduction Program (HRRP) [4].

In order to reduce hospital readmissions rates, the HRRP applied penalties to hospitals with excessive readmissions which in turn is believed to enhance care for patients. It aims to lower costs while better serving patients by encouraging hospitals to improve care coordination. While the HRRP does not specifically target diabetes, the diabetes is a condition that puts patients at a high risk of being readmitted to healthcare institutions. Readmission rates for diabetics are significantly higher ranging from 14.4% to 22.7%, compared to 8.5% to 13.5% in the general population [5]. Only in 2018, diabetes with complications accounted for 122,400 readmissions, with an average cost per readmission was approximately $15,200 [6].

History: Received: 18 December 2024; Revised: 27 January 2025; Accepted: 30 January 2025; Published: 20 February 2025
* Correspondence: meda.shefqet@ibu.edu.mk

The complications and costs of diabetes-related readmissions highlight the need for improved identification, planning, and healthcare management.

Currently, Diabetes Early Readmission Risk Index (DERRI) is a widely used method for readmission estimation. However, this methodology, based on the serious diabetic complications and patients' most recent HbA1c, lacks the reliability.

Predicting readmissions remains challenging and relies heavily on hospital resources and discharge planning efficiency [7]. The unintended results of the HRRP, on the other hand, make these challenges even harder. Although the program has effectively lowered readmittance rates, there are worries that hospitals may choose to not allow readmission for those patients whom they feel require additional attention so that they are not penalized.

This is a serious problem particularly in complex patients with conditions such as diabetes where incentivizing readmission reductions can lead to adverse events [3].

Under these limitations, ML [8] appears to be a remarkably unique resource for prediction and a motivating approach to explore. In contrast to conventional techniques, learning algorithms [9] can examine intricate datasets to uncover patterns and risk elements, leading to more accurate forecasts of readmission risk. Simultaneously, it can enhance resource distribution and individualized patient treatment.

1.2. Purpose

This study aims to evaluate the ML predictive models for hospital readmissions among diabetic patients by using historical healthcare data from the "Diabetes 130-US Hospitals for Years 1999-2008" dataset [10].

The main objectives of this study are:

- To find the most significant risk factors for readmissions in diabetic patients.
- To develop and refine various ML models (such as Logistic Regression, SGD Classifier, Decision Tree, Random Forest, Gradient Boosting Classifier, XGBoost, SVM, and RNN) to ascertain the likelihood of 30-day readmissions.

The evaluation is based on standard statistical, graphical, and computational methods, such as Accuracy, Recall, Precision, Specificity, AUC, F1 Score, Prevalence, Mean Squared Error, and R2 [11].

1.3. Scope of the Study

The study focuses on developing ML models for predicting 30-day hospital readmissions among diabetic patients, leveraging the "Diabetes 130-US Hospitals for Years 1999-2008" dataset. The dataset includes 50 features encompassing ten years of inpatient encounters from 130 U.S. hospitals. In preparing the data for modelling, the study addresses missing values and applies many pre-processing techniques.

However, we recognize certain research limitations, including data integrity issues (e.g., inconsistent categorizations), incomplete data, and computational resource constraints. Although the study gives valuable insights into predicting readmissions within 30 days, the current scope excludes the practical application of these ML models in clinical settings.

1.4. Contributions

We refine the process of data pre-processing and model building through careful feature engineering and handling of data challenges such as class imbalance, optimisation and comparisons. Additionally, the study provides broader insights into how ML can be applied across different patient populations. The methods used here, particularly in data preparation and model optimization, can be adapted to predict readmissions for patients with other conditions, such as heart or lung diseases.

In this study we have also advanced with some other findings referring to machine learning methods including SVM as more effective in high dimensional data and RNN (which is not a proper one for tabular datasets) and the parameters used to compare the performance of the models and the variance in the target variable which is readmission. So, in addition to AUC scores, we have also applied R2 and MSE measures in model evaluation for analyses and conclusions.

Scaling these techniques could enable healthcare providers to develop targeted interventions for various patient populations.

The findings provide healthcare practitioners with meaningful insights into improving discharge protocols and post-discharge care for diabetes patients at high risk of readmission.

The rest of this paper is organized as follows. The proceeding section is focused on reviewing the previous work in predicting hospital readmissions using ML. Section III presents the methodology, including preparation of datasets, preprocessing and processing, analysis, trainings, modelling and evaluation. The results are presented in section IV. Section V concludes the paper.

2. Literature Review

The application of ML methods on diabetes datasets has been mainly used to classify the patients as diabetic or non-diabetic, or in predicting the likelihood of diabetes occurrence. One such hybrid system, build on Naive Bayes Classifier and SVM is presented in a study for Implementation of Supervised Learning Algorithms in Disease Detection [12]. The challenges in diabetes management, the importance of diabetic patients' readmissions, and the ML-based possibility of readmission prediction have been extensively explored in literature.

2.1. Importance of Hospital Readmissions

Healthcare readmission rates are among the most important measures of quality within healthcare services, as they reflect the quality of care provided during inpatient stays and after discharge. Medicare and other healthcare providers perceive a high readmission rate as a lack of adequate care coordination and as an indicator of where improvement is needed so that hospitals do not operate ineffectively [13]. This indicator is especially significant where chronic diseases, such as diabetes, are concerned owing to their financial and clinical consequences.

As Association [14] notes, the cost of diagnosed diabetes in the United States in 2017 was estimated at $327 billion and 30% of that were spent on hospital inpatient care directly related to diabetes. In addition, the emphasis on cost containment still concentrates on the prevention of hospital readmissions because it is where improvement of healthcare quality is targeted. Data from 2016 to 2020 reveal that, although overall readmission rates have moderately declined, a significant proportion of readmissions are still preventable [15]. This shift emphasizes the need of focused strategies on readmissions which would increase patient benefit and minimize costs incurred.

2.2. Challenges in Diabetes Management

The effective management of diabetes during hospitalization is essential to reduce the likelihood of repeated hospitalizations; however, this is a challenging endeavour due incomplete glycaemic control, insufficient patient education, and the absence of standardized care protocols. Additional nurse challenges are compounded by the underuse of Certified Diabetes Educators (CDE) and the lack of a standard protocol between hospitals for blood glucose control, which increases the likelihood of readmissions, as noted in the recent American nurse article [16]. Similarly, the systematic review conducted in Soh, et al. [17] highlights the challenges associated with chronic diseases and diabetes which increase the rate of readmission along with socioeconomic challenges. Resolving these concerns requires multidisciplinary intervention involving proper education of the patients, effective discharge planning, and collaboration of multidisciplinary teams of healthcare personnel to improve patient outcomes and reduce hospital readmissions.

2.3. Machine Learning in Predicting Hospital Readmissions

With advancements in technology and computing, ML models have become a powerful quantitative approach for making predictions, offering significant benefits in pattern recognition and outcome prediction. These models have become popular in healthcare, specifically in predicting readmissions in hospitals for conditions such as diabetes. In this context, an attempt was made to study Logistic Regression (LR), Artificial Neural Network (ANN), and Easy Ensemble (EE) techniques for predicting 30-day readmissions for diabetes patients. The results of these studies revealed that while ANN can model non-linear patterns and relationships which are complex in the data, it did not consistently outperform LR and EE, particularly in imbalanced datasets.

The ANN model was more prone to overfitting, highlighting the need for careful model tuning and selection of appropriate features to ensure generalizability [18].

In another study, a few ML classifiers, such as Naive Bayes, Bayesian Networks, Random Forest, AdaBoost, and Neural Networks, were evaluated to predict high readmission risk in diabetic patients. Random Forest and AdaBoost models outperformed the others, highlighting the importance of using features like inpatient visits and discharge disposition. Despite these successes, there were issues with data imbalance, suggesting that improving feature selection and combining classifiers might enhance model performance [19].

Similarly, a study used Random Forest models to predict 30-day readmission rates in diabetic patients [20]. Random Forest outperformed the other models, including Naive Bayes and Tree Ensemble, having the highest AUC. According to the study, the key to Random Forest's effectiveness is its capacity to handle a large number of variables without overfitting. However, like with the previous study, it recognized limitations such as the need for more thorough clinical variables and the issue of data imbalance, which could bias predictions towards non-readmission.

3. Methodology

The methodology used in this paper includes data analysis and preprocessing, feature engineering, model development, model application, metrics, comparisons, and results analysis.

3.1. Data Pre-processing

This study utilized "Diabetes 130-US Hospitals for Years 1999-2008" dataset which is publicly acquired from the CI Machine Learning Repository. This dataset contains clinical records from 130 US hospitals over a span of ten years (1999-2008). The dataset contains 101,766 records with 50 features related to patient demographics, hospital details, and medical outcomes, such as admission details, lab results, and treatments received. Despite some missing values, it can be considered a reliable source for predictive modelling.

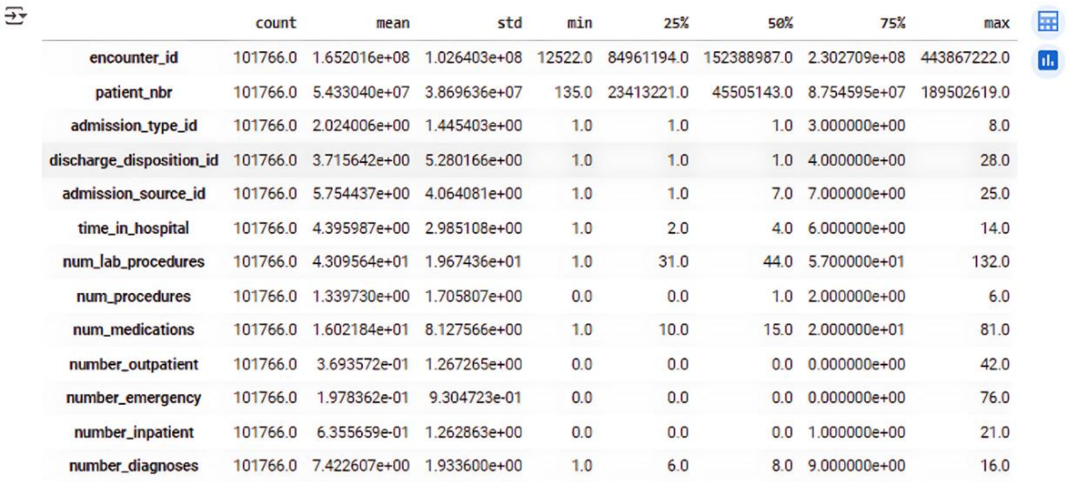
Key Features and ID Mappings are very important in structuring data sets, ensuring data consistency and efficient retrieval. Some key attributes include patient number, race, gender, age, admission type, time spent in the hospital, medical specialty of the admitting physician, number of lab tests performed, HbA1c test results, primary and secondary diagnoses, number of medications, use of diabetic medications, and the number of outpatient, inpatient, and emergency visits in the year preceding the hospitalization. Aside from the main dataset, there is a distinct file that includes ID mappings for categorical features, such as *admission_type_id*, *discharge_disposition_id*, and *admission_source_id*. These mappings are useful for interpreting the categorical data in the main dataset and defining specific types of admissions, discharge dispositions, and admission sources.

We applied *YData Profiling Report* for data profiling, describing types, distributions, missing values, and correlations. The overview part of the report shows basic information such as the number of variables, the number of observations, etc. The methodology also allows the identification the data areas that require focused attention for data cleaning purposes.

In terms of missing values, the dataset contains considerable gaps in several columns. For example, the weight column has 96.86% missing data, while *medical_specialty* has 49.08% and *payer_code* has 39.56%. These high percentages of missing data could have an impact on the results. Other columns like race (2.23%), diag_1 (0.02%), diag_2 (0.35%), and diag_3 (1.40%) have much lower missing percentages. Features such as *examide* and *citoglipton* have only one value in every row, suggesting they have no predictive potential.

When we analyzed Data Consistency and Anomalies, we found that the dataset has 71,518 unique *encounter_id* values, indicating that each hospital visit is unique. There were 30,248 instances where patients have multiple encounters, which is typical in healthcare. Importantly, no duplicates were found when considering *patient_nbr* and *encounter_id*, so each patient encounter is uniquely documented. No instances of negative values were found in the integer columns, which suggests that the numerical data is within expected ranges and does not contain obvious entry errors.

Following paragraph and data provides an overview of descriptive statistics including numerical features.



	count	mean	std	min	25%	50%	75%	max
encounter_id	101766.0	1.652016e+08	1.026403e+08	12522.0	84961194.0	152388987.0	2.302709e+08	443867222.0
patient_nbr	101766.0	5.433040e+07	3.869636e+07	135.0	23413221.0	45505143.0	8.754595e+07	189502619.0
admission_type_id	101766.0	2.024006e+00	1.445403e+00	1.0	1.0	1.0	3.000000e+00	8.0
discharge_disposition_id	101766.0	3.715642e+00	5.280166e+00	1.0	1.0	1.0	4.000000e+00	28.0
admission_source_id	101766.0	5.754437e+00	4.064081e+00	1.0	1.0	7.0	7.000000e+00	25.0
time_in_hospital	101766.0	4.395987e+00	2.985108e+00	1.0	2.0	4.0	6.000000e+00	14.0
num_lab_procedures	101766.0	4.309564e+01	1.967436e+01	1.0	31.0	44.0	5.700000e+01	132.0
num_procedures	101766.0	1.339730e+00	1.705807e+00	0.0	0.0	1.0	2.000000e+00	6.0
num_medications	101766.0	1.602184e+01	8.127566e+00	1.0	10.0	15.0	2.000000e+01	81.0
number_outpatient	101766.0	3.693572e-01	1.267265e+00	0.0	0.0	0.0	0.000000e+00	42.0
number_emergency	101766.0	1.978362e-01	9.304723e-01	0.0	0.0	0.0	0.000000e+00	76.0
number_inpatient	101766.0	6.355659e-01	1.262863e+00	0.0	0.0	0.0	1.000000e+00	21.0
number_diagnoses	101766.0	7.422607e+00	1.933600e+00	1.0	6.0	8.0	9.000000e+00	16.0

Figure 1.
Descriptive Statistics of numerical features

The figure 1 summarizes the central tendency, dispersion, and shape of the numerical features. Although *admission_type_id*, *admission_source_id*, and *discharge_disposition_id* appear as numerical variables, they should be treated as categorical variables because they represent ID mappings rather than continuous or ordinal data.

The analysis of variables helped us understand the intricacies, distribution, categorization, trends, and imbalances within the dataset. The readmitted variable indicates whether a patient was readmitted to the hospital after discharge and within what timeframe. It has three distinct categories:

NO: Indicates that the patient was not readmitted.

>30: Indicates that the patient was readmitted more than 30 days after discharge.

<30: Indicates that the patient was readmitted within 30 days after discharge

The distribution in the figure 2 indicates a substantial class imbalance, particularly concerning the <30 category, constituting only 11.2% of the data. This imbalance poses challenges for predictive modelling. To address this issue, strategies will be implemented during the data preprocessing phase. Since this study's goal is to predict less than 30-day readmissions, we will combine the NO and greater than 30 into a single category so that we have a binary classification problem, as seen in the figure 3. This will also help in getting a better picture of this study's problem in the bivariate analysis.

In the data pre-processing phase, identifying missing values is the first step, guaranteeing data integrity for analysis or model training. During the initial exploratory data analysis (EDA), missing values were identified in several features in the dataset. The missing values, represented by the character '?', were prevalent in features such as Payer Code, Medical Specialty, and Race.

3.1.1. Imputation Strategies

Missing values in these ICD9 codes (diag_1, diag_2, diag_3) were imputed using the mode. This decision was based on the lack of significant correlations between missing data in these columns and other features, suggesting that the missingness was random. Mode imputation ensures that the most frequent diagnosis code fills in the gaps, maintaining data consistency without introducing bias.

In Excluding rows or columns due to high missing values we have we have taken great care to note that if too many data are removed, especially key features, this can reduce the usefulness of the data.

The Payer Code, which indicates the specific health insurance type, was removed from the dataset because it had a significant number of missing values. Retaining this feature may provide unnecessary interference and potentially skew the model's results. Similarly, the 'medical specialty' feature was excluded because nearly 48.9% of its values were missing. The 'weight' feature was also omitted because 96.88% of its values were missing. Given the minimal data available, imputing or analysing this feature meaningfully was not feasible.

Considering Category Exclusion process, the 'Newborn' category within the *'admission_type_id'* feature was excluded due to the inconsistencies it introduced in the dataset. Specifically, the age values associated with entries categorized as 'Newborn' were found to be questionable and did not. align with patient age ranges. Likewise, the 'Delivery and Birth' category in *'admission_source_id'* was also excluded because it contradicted the values in column 'age'.

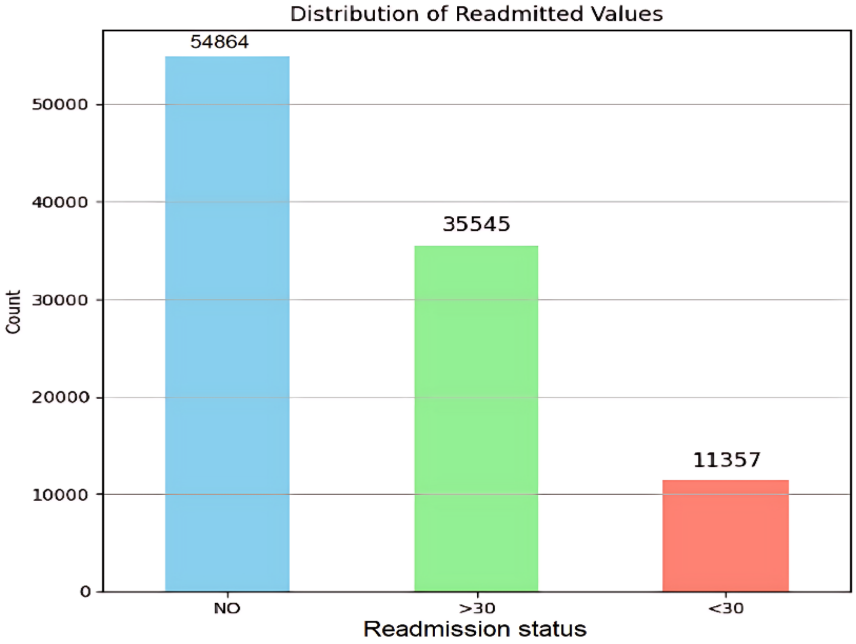


Figure 2.
Distribution of readmitted values.

After grouping categories, 'Neonate' category within the *'discharge_disposition_id'* feature was excluded. Since neonates who are discharged for specialized neonatal care have different follow-up and readmission patterns and there were a small number of cases (5), we dropped this category to maintain the model's focus on more relevant patient data. Specific categories like 'Expired' and 'Hospice' in '*discharge_disposition_id*' were excluded because they represent patients who are not expected to be readmitted.

The data had to be prepared for ML models and the others. The target variable 'readmitted' was binary encoded to distinguish between patients readmitted within 30 days (encoded as 1) and those not (encoded as 0). This method simplifies the model's task of predicting readmission.

One-hot encoding was applied to the 'race' variable, after adding a new category "UNK" (Unknown) introduced to handle missing values.

This ensures the model treats each race as a separate binary feature, avoiding unintended ordinal relationships.

The 'gender' variable, initially consisting of three categories (female, male, unknown/invalid), was simplified by dropping the 'unknown/invalid' category because it had just one value, making it insignificant. Binary encoding was applied for the remaining categories, where Female was encoded as 0 and 'Male' was encoded as 1.

The 'age' feature, originally categorical, was converted to numerical values representing the lower bounds of each age range. This preserves the ordinal nature of the data, making it suitable for modelling.

For the *'max_glu_serum'* feature, the categorical values were numerically encoded. 'None' was replaced with a placeholder value (-99) to signify no test was conducted. The values '>200' and '>300' were both encoded as 1 to represent high glucose levels. 'Norm' was encoded as 0 to indicate normal glucose levels. Additionally, an indicator feature named *'max_glu_serum_not_taken'* was created to flag records where no glucose serum test was conducted. This binary feature was set to 1 when the test was not taken and 0 otherwise, ensuring that the absence of data was captured in the model.

A1Cresult feature was similarly transformed to ensure it was compatible with the model. 'None' was replaced with a placeholder value of -99 to signify missing or exceptional cases. '>8' and '>7' were encoded as 1, representing elevated A1C levels indicative of diabetes. 'Norm' was encoded as 0 to represent normal A1C levels. Also, an indicator feature named A1Cresult_not_taken was created. It was set to 1 when the A1C test was not conducted and 0 otherwise.

The *'change'* and *'diabetesMed'* features were encoded to binary values for model compatibility. The change feature was encoded to 0 for 'No' and 1 for 'Ch,' representing whether a change in medication was made. Similarly, the *diabetesMed* feature was encoded to 1 for 'Yes' and 0 for 'No,' indicating whether the patient was on diabetes medication.

Grouping features was the next step following preprocessing.

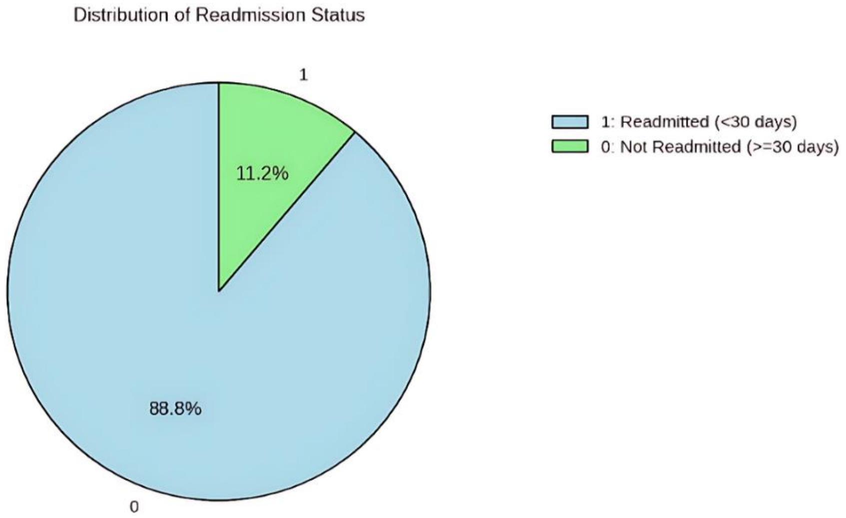


Figure 3.
Distribution of Readmission Status after encoding.

Features were grouped into categories based on their relevance and nature. An EDA report, including a distribution plot was created by the YData Profiling for each feature of these grouped categories:

- Identifiers for Patients ('encounter_id', 'patient_nbr').
- Visit Frequency *('number_outpatient', 'number_inpatient', 'number_emergency'*).
- Patient Admission Details *('medical_specialty','diag_1','diag_2','diag_3', 'time_in_hospital','number_diagnoses','num_lab_procedures','num_procedures','num_medications'*).

In the *Univariate Analysis* for Patient Demographics, we considered: race, gender, age, payer code, and weight. During this analysis we found as follow:

Race - The majority of the population in the dataset is Caucasian, followed by African American being the second largest racial group in this dataset. A category labeled *'?'* also indicates missing values, which will be addressed during the data cleaning phase.

Gender - The data is evenly distributed among male and female groups. In addition, "Unknown / Invalid entries" can be seen.

Age - The dataset consists primarily of older patients, with the largest age categories being 70-80 (25.6%) and 60-70 (22.1%). Younger age groups are under-represented, indicating a skew towards an older population.

Payer code - The majority of payer codes are missing or unknown, represented by the '?' category, which accounts for the highest count. Among the known payer codes, 'MC' (Medicare) is the most common, followed by 'HM' (Health Maintenance Organization) and 'SP' (Self-pay). Weight - 96.9% of rows in the dataset have a missing or unknown weight. This lack of recorded weight data will need to be addressed during data cleaning.

In the *Bivariate Analysis* for Patient Demographics, we have considered relationship and find out:

Race vs Readmitted - The Caucasian group has the highest number of readmissions, followed by African Americans.

Gender vs Readmitted - Most readmissions occur in females, with males also showing substantial numbers.

Age vs Readmitted - The age group 70-80 has the highest number of readmissions, followed closely by 60-70 and 80-90. Younger and older age groups show fewer readmissions.

Insulin vs Readmitted - Patients not on insulin (No) have the highest rates of readmission. The Steady category also has many readmissions.

For clinical results in the bivariate we grouped:

Glucose Serum Test vs Readmitted -The 'None' category dominates the readmissions, indicating most patients did not have a recorded Max Glu Serum value. The categories >300 and >200 have much lower readmission counts.

A1Cresult vs Readmitted - The body's A1C levels are a reliable predictor of glucose levels. The 'None' category represents the majority of readmissions in the data set. The lack of data on this potentially valuable feature might limit its predictive power.

DiabetesMed vs. Readmitted - Patients who received diabetes medication were more likely to not be readmitted (69,252) than those who were readmitted (9,111). These results suggest that diabetes medication might be associated with a lower readmission rate.

Now let's give in this bivariate analysis the findings for Admission and Discharge Details according to these relations:

Admission_source_id vs Readmitted - The number of patients being readmitted is highest for Admission Source ID of 7 (Emergency Room), followed by 1 (Physician referral).

Admission_type_id vs Readmitted - The number of patients being readmitted is highest for the Admission Type ID of 1 (Emergency), followed by 2 (Urgent) and 3 (Elective).

Discharge_disposition_id vs Readmitted - The chart shows that most readmissions occur in Discharge Disposition IDs 1 (Discharged to home), 2 (Discharged/transferred to another short-term hospital), 3(Discharged/transferred to SNF), with ID 1 having the highest number of total discharges but relatively few readmissions. The other IDs have fewer readmissions, indicating that readmission rates are generally low across all categories.

Grouping Categorical Features:

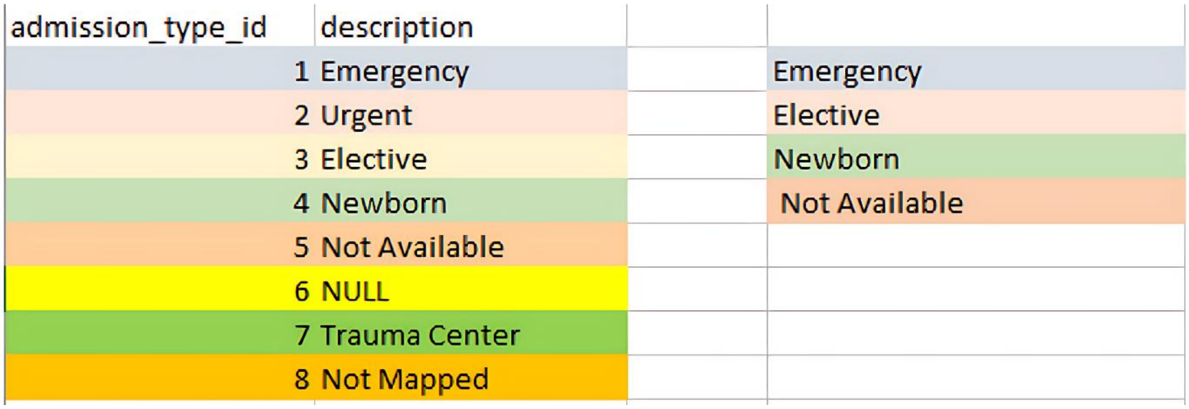
Grouping the *admission_type ID* into fewer categories simplifies the data and captures meaningful patterns without losing too much detail (see figure 4).

Emergency/Urgent/Trauma Center → Emergency These categories often involve urgent or critical situations.

Not Available/NULL/Not Mapped → Not Available These categories indicate missing or unspecified information.

Elective → Elective

Newborn → Newborn



admission_type_id	description		
1	Emergency		Emergency
2	Urgent		Elective
3	Elective		Newborn
4	Newborn		Not Available
5	Not Available		
6	NULL		
7	Trauma Center		
8	Not Mapped		

Figure 4.
Grouped categories of *'admission_type_id'*

For *disposition_id*, each id represented by the number was grouped into 8 categories such as: Discharged to home, Transferred to Another Medical Facility, Left AMA (Against Medical Advice, Still Patient/Referred, Expired, Hospice, Not Available, Neonate discharged.

Each Admission source id was grouped into these categories: Referral, Transfer, Emergency Room, /Law Enforcement, Not Available, Delivery and Birth, Readmission to Same Home Health Agency

As explained below, some features are dropped. This is done for increasing model performance, computational, and possibly overfitting. Features like 'encounter_id' and *'patient_nbr'* were dropped as they were simply identifiers without any predictive value.

After grouping the categories in both *'admission_type_id'* and *'admission_source _id'*, we decided to drop the *admission_type_id* feature because it shared a moderate correlation of 0.6234 with *admission_source_id*, meaning they overlap substantially in the information they provide. The categories in *admission_source_id* already cover what *admission_type_id* represents, making it unnecessary to keep both. Retaining both would introduce redundancy without adding significant value to the model.

The medications *'examide'* and *'citoglipton'* were dropped because they only contained the value *'No,'* indicating that they were not prescribed at all, making them non-informative for prediction purposes.

Medications that were prescribed to less than 1% of the population or had no significant impact on the outcomes were excluded. This decision was supported by earlier statistical tests (Chi-Square and ANOVA) that identified these features as insignificant. Removal of these features, helped the model focus on more relevant variables. The specific medications removed were: nateglinide, chlorpropamide, acetohexamide, tolbutamide, acarbose, miglitol, troglitazone, tolazamide, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone.

Since both '*number_changes*' and 'change' features were p-value invariant and equally informative, '*number_changes*' was retained instead of 'change' as it had a lower p-value which is preferred. Lastly, features such as '*number_outpatient*', '*number_emergency*', and '*number_inpatient*' were dropped after the new feature 'total_visits' was created.

3.2. Feature Engineering

Initially, categorical features in the dataset were under the object data type, while numerical features were represented as integers. By the end of the feature engineering process, all features were converted to the integer data type to ensure compatibility and consistency before proceeding to the modelling stage. Preparing the data in a uniform format allows ML algorithms to process the features without encountering type-related difficulties.

During the feature engineering phase, our goal was to create additional features to strengthen the predictive capability of the model. One of these features, 'total_visits,' was created by summing the outpatient, emergency, and inpatient visit. This metric encapsulated all visit types within a year and offered a broader perception of a patient's healthcare spending. Combining these visits into one feature simplified the data.

Additionally, *'num_changes'* was created to quantify the number of changes in a patient's medication regimen during their encounter. It combined the number of medications whose dosage was either increased or decreased, providing a single measure of medication management intensity. This feature captured the complexity of a patient's treatment plan.

Finally, the patients using insulin and the prescribed diabetes medication were classified in the '*insulin_treatment*' feature. This feature provides a range of treatment levels for patients: some can be insulin dependent only, others can be insulin supplemented with medication, and last but not the least, those who do not take any medications for diabetes. After this, all the categories were ascribed different numerical values so the model could separate various levels of treatment.

Feature Transformation and Handling Outliers is done in several steps. The first step in the feature transformation process was identifying and managing outliers to prevent them from negatively impacting model performance. Outliers were identified using Z-scores, with a threshold of 3, across the numerical features. The Interquartile Range (IQR) method was also used to validate these outliers further. Visualizations like histograms and boxplots were used for a visual valuation.

Initially, we applied the Yeo-Johnson transformation to address skewness and normality of the data distribution. While it was used to correct skewness, the transformation also played a role in bringing inaccessible outliers closer to the center of the distribution. After this transformation were completed, the results were reassessed and outliers were recalibrated like num_medications, number_diagnoses, and num_lab_procedures were handled by removing data points with Z-scores greater than 3.

Some models do not need much normalized data, while others, including logistic regression, perform better when they have features normally distributed. To ensure that the numerical features conformed to a normal distribution, we first assessed each feature's skewness and kurtosis. Variables such as *num_medications*, *time_in_hospital*, and *total_visits* exhibited noticeable skewness, with total_visits being particularly strongly skewed with a value of 5.682. The Yeo-Johnson transformation was utilized to correct the skewness, resulting in the normalization of these features.

Post-transformation results showed a reduction in skewness across all targeted features. For example, the skewness of total_visits decreased from 5.682 to 0.421, and num_medications shifted from 1.341 to 0.020. This transformation helped align the features closer to a normal distribution, which benefits many ML algorithms.

The Yeo-Johnson transformation, applied earlier to correct skewness, was also saved using Joblib to ensure consistency in its application to the validation and test datasets. By saving the transformer, the exact transformation applied during training was replicated during validation and testing.

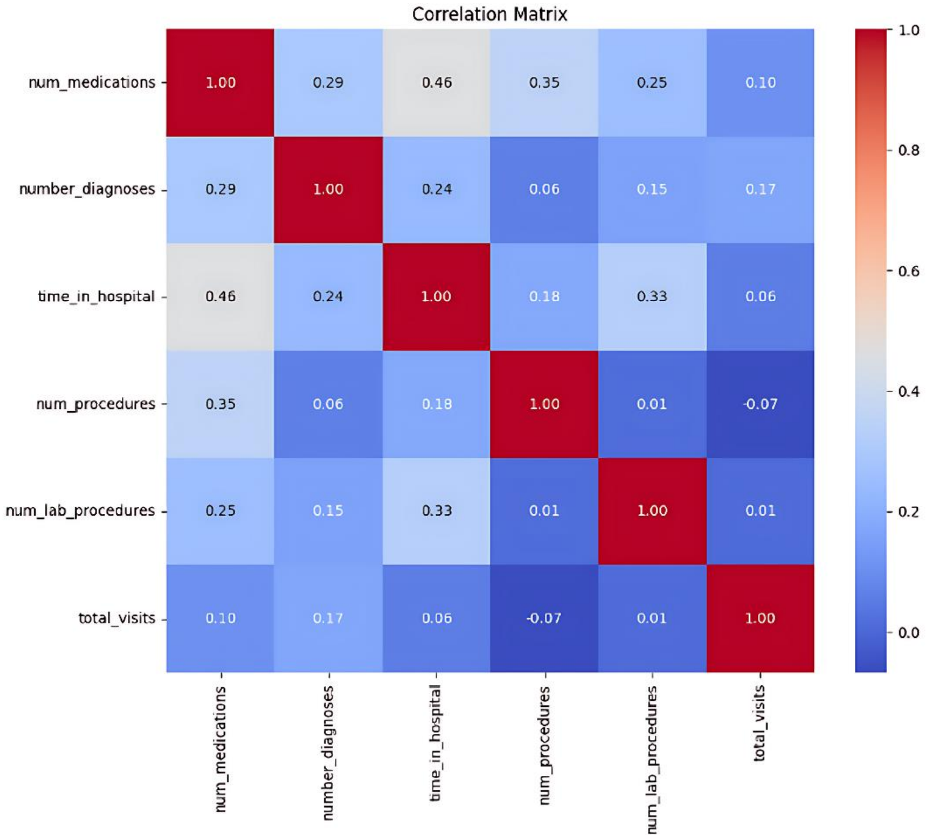


Figure 5.
Correlation Matrix for numerical features.

After the features were normalized, standardization was applied to put all features on a similar scale. Numerical features were scaled to have a mean of 0 and a standard deviation of 1 using the StandardScaler. Standardization was preferred over alternative scaling techniques primarily because it maintains the interpretability of the model's coefficients, particularly in linear models and regularization scenarios. By making sure that all features are on a similar scale, the model can converge more efficiently and perform gradient-based optimization more effectively, as no single feature dominates the learning process due to its scale. This approach is particularly beneficial for gradient-based optimization algorithms like Logistic Regression and Support Vector Machines, where the scale of the input data can significantly impact the convergence speed and stability of the model.

Lastly the same scaler was saved and applied to the test and validation datasets, to guarantee consistency across different datasets. After standardization results showed that all features were effectively centered and scaled, as they had means close to 0 and standard deviations close to 1.

3.3. Correlation Analysis

Evaluating and understanding how variables are related to each other, which can be useful for feature selection, hypothesis testing, and predictive modelling we made the following analysis:

A correlation analysis was conducted to ensure that the dataset's numerical features were properly independent of one another and the main goals were:

- To determine which feature pairings, revealed a strong linear relationship, as indicated by Pearson correlation coefficients.
- To detect potential multicollinearity issues, which could negatively impact model performance, particularly in linear models.

The correlation between each pair of numerical features was measured using the Pearson correlation coefficient, which was computed by the correlation matrix. After this, the Variance Inflation Factor (VIF) was calculated for each feature to quantify the degree of multicollinearity.

Overall, according to the correlation matrix in the figure 5, most features are weakly associated with each other. The color gradients in the heatmap suggests that the majority of the feature pairs have a correlation coefficient close to 0, indicating that the features are generally independent from each other. The results are useful for the model's performance, as it implies that each feature contributes unique information to the model. The strongest relationship was observed between time_in_hospital and num_medications, with a correlation coefficient of 0.47. This suggests that patients who stayed longer in the hospital tended to receive more medications. Additional moderate correlations were identified between num_medications and num_procedures (0.36) and time_in_hospital and num_lab_procedures (0.33). These correlations indicate a degree of association but are insufficient to suggest redundancy. Finally, most other pairs exhibited relatively low correlations, indicating that the majority of features were fairly independent of one another.

Regarding multicollinearity, all features had low VIF scores, with values close to 1. The feature num_medications had the highest VIF of 1.51, followed by time_in_hospital with a VIF of 1.37. These values suggest that none of the features exhibit significant multicollinearity.

After the above conclusions, we should take into account the strongest correlation and the clinical relevance. Based on the results of the correlation analysis and VIF calculations, all numerical features in the dataset were kept. The correlation coefficients did not reveal any notably strong relationships, and the VIF scores were well within acceptable limits, indicating no significant multicollinearity issues. Although the strongest relationship was observed between *time_in_hospital* and *num_medications*, it exhibited a correlation coefficient below 0.5, which is generally not considered an issue. The found association between *time_in_hospital* and *number_of_medications* were recognized as potentially clinically relevant. This relationship could provide valuable insights about patient care patterns.

Initial data split is a very important phase in ML and data analysis. To ensure the model is tested on unseen data, the training, validation, and testing datasets were prepared, these steps were taken prior any form of data manipulation. Data leakage occurs when the model is trained on information that won't be accessible knows in the real word. As a result, there is excessive prediction of the model's performance [21]. For the purpose of minimizing data leakage risks, we implemented an approach where the datasets were partitioned after being shuffled with a random state value of 123. The reasoning behind this is to ensure randomized in the splits.

Subsequently, 30% of the data was allocated to a combined validation/test set. The combined validation/test set was divided equally into distinct validation and test sets, with each set containing 15% of the total data. The remaining 70% was assigned to the training set. In some cases, just for comparisons outputs for different approaches the dataset has been successfully split into training and testing sets, with the training set containing 89% of non-readmitted cases and 11% of readmitted cases The prevalence of the target variable (readmitted) was maintained across all splits to for consistency. A random state was used so that the splits were reproducible.

3.4. Addressing Class Imbalance

Addressing class imbalance is an important step in predictive modelling, especially in healthcare contexts like predicting hospital readmissions. This imbalance in the dataset can make the model biased towards the majority class, in this case, patients who were not readmitted. This bias results in poor performance in the minority class-patients who were readmitted-undermining the model's capability to make accurate predictions where it matters most.

Initially, the dataset had a considerable imbalance, with 61,154 samples in the majority class (Not Readmitted) compared to only 7,831 samples in the minority class (Readmitted). The degree of this imbalance is visually represented in the Figure 6.

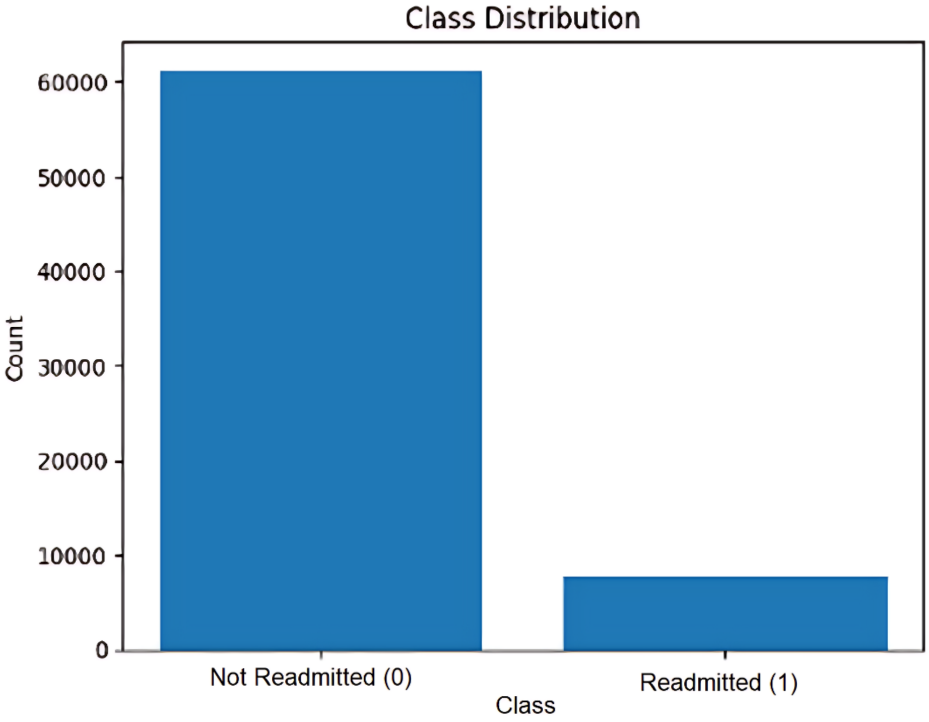


Figure 6.
Class distribution of readmitted feature.

3.5. Solving the Problem of Class Imbalance

Synthetic Minority Over-sampling Technique (SMOTE) was used. It is a popular method used in ML. It helps generate synthetic samples for the minority class by interpolating between existing minority instances. This technique works by choosing two or more similar instances from the minority class, identifying their nearest neighbors, and creating synthetic samples along the line segments that connect them. This way, we make the minority group more varied without copying the existing samples, which can help improve model generalization and reduce overfitting [22].

For this study, SMOTE was applied using the default parameters, which included:

k_neighbors=5: This parameter specifies that the synthetic samples should be generated using the five nearest neighbors of each minority class instance.

sampling_strategy='minority': This setting ensures that the minority class is oversampled to match the number of instances in the majority class.

After applying SMOTE, the dataset was reshuffled to prevent any ordering bias.

The SMOTE technique helped balance the dataset. It equalized the number of samples in both the majority and minority classes. After applying SMOTE, the class distribution for the readmitted variable appeared as:

- Not Readmitted (0): 61,154 samples
- Readmitted (1): 61,154 samples

Using SMOTE, the study ensured that the predictive model had an equal opportunity to learn from both classes. This approach lowers the chances of the model becoming biased towards the majority class and improves its ability to generalize well to unseen data. In healthcare applications, this balanced approach is essential for making accurate and reliable predictions, leading to better clinical decisions and improving patient care.

3.6. Model Development and Evaluation

In this section of the study, we have addressed these main issues: model selection, metrics, accuracy, confusion matrix, training process, AUC comparison across models and performance.

For predicting hospital readmissions, we used various ML models which included Logistic Regression, SGD Classifier, Decision Tree, Random Forest, Gradient Boosting, XGBoost, Support Vector Machine (SVM), Recurrent Neural Networks (RNN). Their high performance within binary classification models and their adequacy of large datasets were the reasons of our selection.

To make a selection of the above models, we of course considered previous studies and the most efficient supervised ML/DL models. Let us briefly explain each of them:

Logistic Regression (LR) model is preferred because of its simplicity of use its interpretability. This model is especially beneficial in the clinical field where one must be able to evaluate the importance of each feature towards a specific outcome. It demonstrates how every factor affects the likelihood of readmission adding great importance to the clinical point of view.

Scikit-Learn.org [23] - Strong capacity of managing large datasets combined with flexibility to different loss functions made SGD a valuable candidate for this study. Works well with large data that is very common in the medical field allowing faster training of the model and iteration.

Scikit-Learn.org [24] - The change in algorithm workflow enabled by the tree structure facilitates understanding of how a particular model arrived at a specific decision allowing for easier explanation of the predictions to a healthcare provider enhancing the interpretability of the model which was the major advantage of the model. But Decision Trees do tend to overfit, which can be resolved with various solutions including pruning or using ensemble methods like Random Forests.

Khanna [25] - Is especially useful on larger datasets and are less prone to overfitting as compared to single decision trees. The model was chosen because of its strength, accuracy, and ability to generalize in a variety of data types.

Gradient Boosting - Is very effective with structured data and can build machines that can exploit complex features and their interactions. However, it can be sensitive to overfitting if the parameters are not set properly and, because the algorithm is sequential in nature, it is also very intensive computationally [26].

XGBoost, or Extreme Gradient Boosting - Applies additional techniques such as L1 and L2 regularization to reduce overfitting. It also enables multi-threaded programming, which makes it quicker and more effective than traditional Gradient Boosting. It is known for its effectiveness and scalability, greatly exceeding other algorithms for performing tasks on structured data. This is the reason why Gradient Boosting models and especially XGBoost have proven to be very efficient in managing complex relationships within data and in making strong predictions [26].

Support Vector Machines (SVMs) [8] - Is a powerful supervised ML algorithm used for both the classification and regression problems. In particular useful when working with high dimensional spaces and when the number of dimensions is greater than the available number of samples. Robust to overfitting. The problems encountered are computational for large data sets, careful tuning of hyperparameters, and interpretation problems when comparing with other similar models.

Recurrent Neural Networks (RNNs) [8] act on sequential data. RNNs remember previous information, making them distinctive from the standard feedforward neural networks, and hence can be used for applications in time series data, natural language processing, and speech recognition. In this model the parameter sharing reduces complexity. But it also has some challenges like inefficient in computationally, and because of long sequences makes training difficult.

The evaluation of the models is based on several metrics. An overview of them and a concrete analysis are given in the following paragraphs.

Each model's initial performance was evaluated using a range of metrics, such as AUC, accuracy, recall, precision, specificity, and F1 score. In some cases, is used MSE and R2. These metrics were computed for the training and validation datasets to provide a complete view of each model's strengths and weaknesses.

AUC (Area Under the ROC Curve) [9] is calculated by plotting the true positive rate (Recall) against the false positive rate (1—specificity) at different threshold settings. It measures how well the model can differentiate between positive and negative classes. AUC ranges from 0 to 1, where 1 represents perfect classification and 0.5 suggests no discrimination (random guessing). It is particularly useful in evaluating the trade-off between sensitivity and specificity. Although we have operated with AUC as we have explained the reason why, we have also analyzed with MSE and R" to create an idea that the choice made is the right one.

Accuracy is the metric which refers to the number of true results set in proportion to the total number of results (true positives and true negatives) when evaluating the precision of the model. While useful, accuracy can be misleading in the presence of class imbalance as it does not explain the kind of false results provided. Formula: Accuracy = (TP+TN)/(TP+TN+FP+FN)

Recall is a metric used to evaluate how well the model identifies positive cases.

It estimates the model's ability to recognize all relevant instances (true positives) out of the actual positives. High recall is important in scenarios where missing positive cases (e.g., failing to identify a patient at risk of readmission) is costly or where the detection of positive cases is crucial.

Formula: Recall = TP/(TP+FN)

Precision measures the accuracy of the positive predictions. It represents the proportion of positive predictions made by the model that are truly positive. High precision is crucial when the cost of false positives is high [27]. Formula: Precision = TP/(TP+FP)

Specificity (True Negative Rate) measures how well a model correctly identifies and calculate its true negative rate, and so indicates how well a model avoids false positives for a condition. High specificity mitigates the chances of undue anxiety and any treatment for patients, who are not at risk, are not mistakenly identified as at risk, avoiding unnecessary stress and treatment.

Specificity = TN/(TN+FP)

The *F1 score* is effective as it gives an average of precision and recall, which makes it easier to deal with problems that have false positives and false negatives.

F1 = 2 *(precision * recall)/((precision + recall))

A Confusion Matrix is a table that enables the visualization of the performance of a classification model. It presents the true positives, true negatives, false positives, and false negatives and enables a more granular analysis of the model's performance and prediction and helps quantify other parameters and highlight where the model is misclassifying [27].

Confusion matrix in Table 1 is used to evaluate other models' evaluation metrics.

Table 1.
Confusion matrix.

	Predicted positive	**Predicted negative**
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

3.6.1. Training Process

The dataset was divided into X_train, y_train and X_valid, y_valid, which are datasets for training, validating and testing. Each model was developed and tested with several performance metrics: AUC, accuracy, recall, precision, specificity, and F1 score. However, model comparison was mainly based on AUC scores.

AUC scores showed that:

Logistic Regression displayed a balanced but low AUC performance, with 0.782 on the training set and 0.569 on the validation set.

SGD showed similar behavior to LR, slightly dropping from 0.692 in training to 0.585 in the validation set.

Decision Trees exhibited significant overfitting, with a high training AUC and a substantial drop in validation AUC.

Random Forest performed well on the training set (AUC of 0.848) but showed a notable drop to 0.457 on the validation set, though still better than DT, indicating some overfitting.

Gradient Boosting (GB) and XGBoost (XGB) had the highest AUC scores in training but considerably dropped, suggesting overfitting. Respectively - GB training set AUC 0.974 and validation 0.438 while XGB train 0.945 and validation 0.438.

The results, visualized in the figure 7, highlight the trade-off between model complexity and generalization ability. Models like GB and XGB, while powerful, need tuning to prevent overfitting, while simpler models like LR and SGD, though more stable, require enhancement to capture data details.

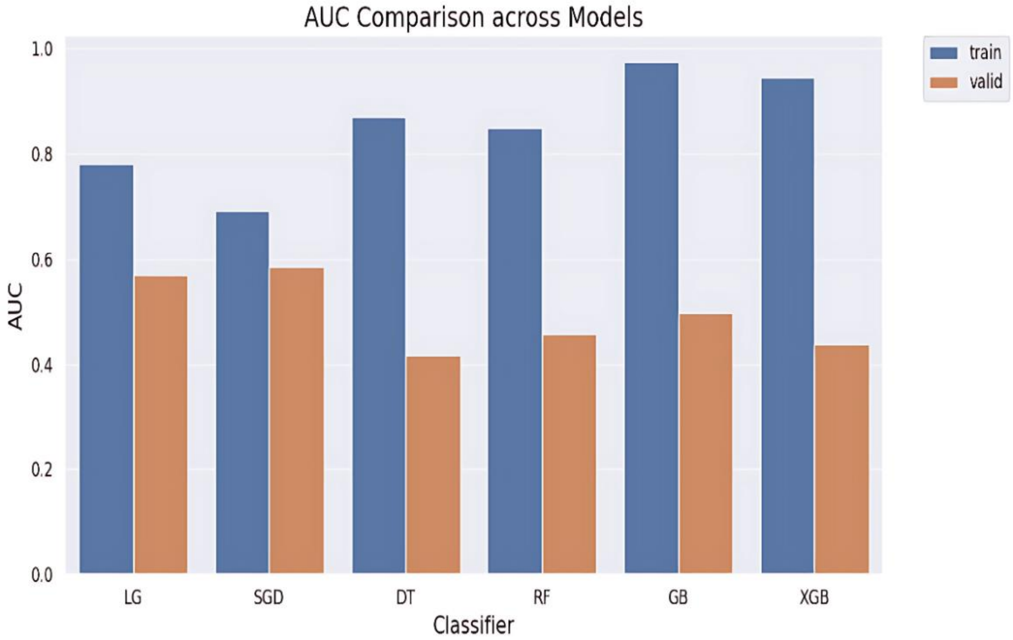


Figure 7.
AUC comparison of models.

Following these comparisons, in order not to overlook any other possibility, even there are several such, we also made an attempt outside of these parametric comparisons of the algorithms described above.

We followed the same procedure by applying SVM and RNN.

We create both a baseline and optimized SVM model to compare their AUC scores. Use the 'readmitted' column as our target variable.

The base SVM model achieved an AUC score of 0.51, while the optimized SVM model achieved an AUC score of 0.50. However, the optimization did not improve the AUC score, as the difference was negative (-0.003). This suggests that the optimization parameters may not have been effective or the dataset's size and features may limit model performance. Then we tried any other optimization parameter, a more extensive parameter grid with different kernels and a wider range of hyperparameters.

Performance Comparison:

- Base SVM AUC: 0.51
- Previous Optimized AUC: 0.50
- New Optimized AUC: 0.547

The top 3 parameter combinations all achieved the same mean CV score of 0.668, with slight variations in the parameters:

The key improvements came from:

- Using the sigmoid kernel instead of rbf
- Setting gamma to 0.01
- Using C=1 without class weights
- The degree parameter didn't significantly impact performance for the sigmoid kernel

Also, we tried based on this data the RNN optimized and not optimized to compare AUC scores.

The AUC scores for both the basic and optimized RNN models have been successfully calculated, showing a slight improvement in the optimized model's performance.
Basic RNN Model AUC Score: 0.66
Optimized RNN Model AUC Score: 0.665
Improvement: 0.6657 % (see Figure 8)

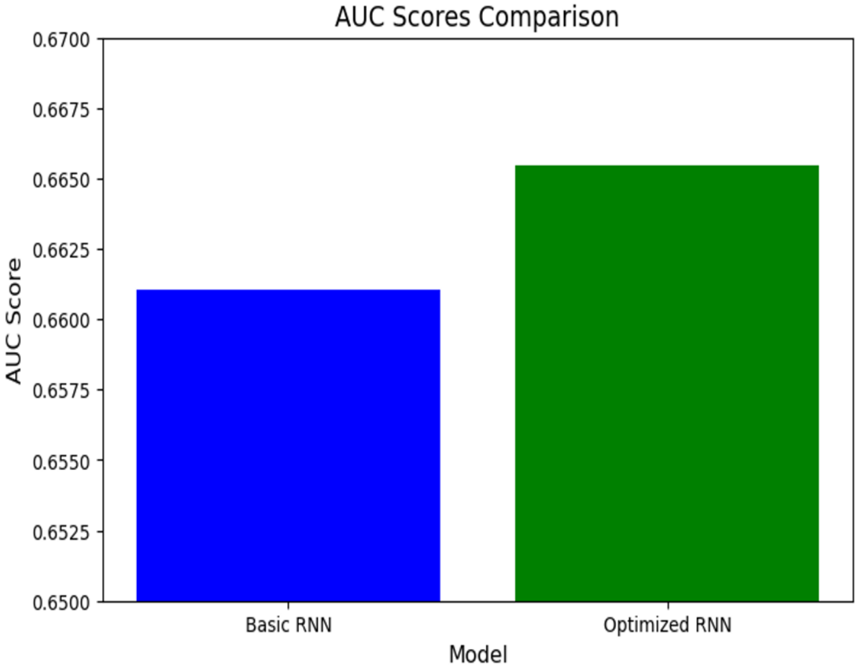


Figure 8.
AUC scores comparison for basic and optimized for RNN.

We explained the reason why we used the AUC score and the study referred to this method. In order not to overlook and to be sure of what we analyzed, we also operated based on the MSE and R2 score. All this to prove that our results based on AUC were well-argued. After this analysis we tried both the tasks classification and regression, predict the chance of 30-day readmissions, based on MSE and R2 scores (Figure 9).

The results show that the basic model overfits significantly, with a large gap between training and test R^2 scores, while the optimized model reduces overfitting and achieves more balanced performance. Here are the detailed results and visual comparisons:
Model Comparison Results:
Basic Model - Training R2: 0.7999 Test R2: -0.1248
- Training MSE: 0.0198 Test MSE: 0.1121
Overfitting Gap (R2): 0.9247
Optimized Model: - Training R2: 0.0730 Test R2: 0.0269
- Training MSE: 0.0918 Test MSE: 0.0970
Overfitting Gap (R2): 0.0461

DOI: 10.55214/25768484.v9i2.4929

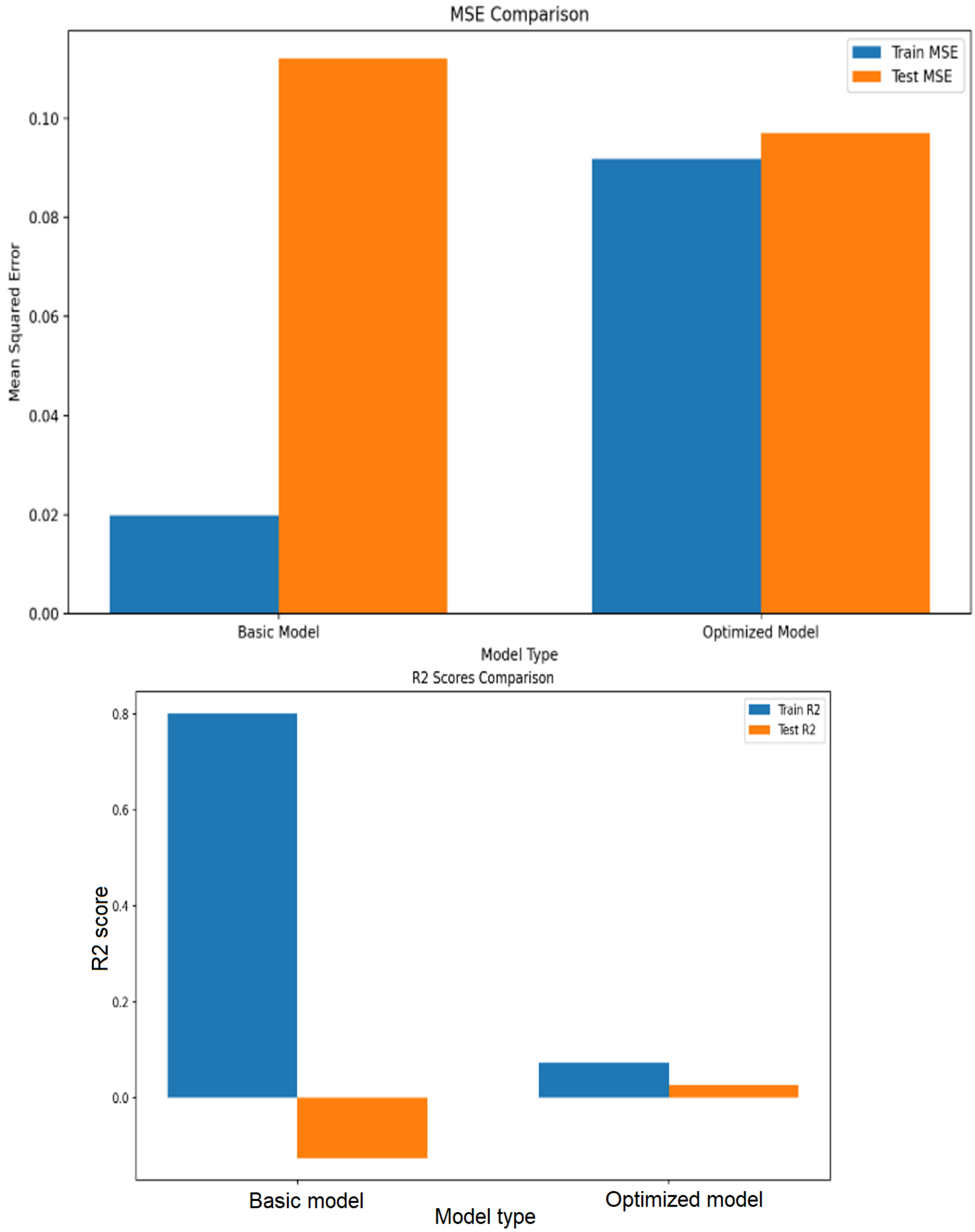


Figure 9.
Feature importance and performance comparisons based on MSE and R2 scores.

In the process of Selection for Optimization based on the baseline AUC scores, models such as SGD, GB, and XGB were selected for further optimization. These models were selected due to their strong performance and potential to balance complexity with generalization, making them suitable for optimizing the prediction of hospital readmissions. Logistic Regression was not chosen for further tuning because of its lower performance, suggesting it may not capture the data's complexity. Decision Trees were excluded from further tuning due to severe overfitting, as indicated by the validation AUC scores, making it less effective for this task without ensemble methods. SVM even optimized the AUC scores it was in range of 0.547. RNN basic (great overfitting 0.02 and 0.1) and optimized was in range of 0.66 AUC scores, and 0.09 ranges for MSE. While with R2 scores are 0.07 and 0.02 optimized and 0.8 and -0.1 for basic (great overfitting)

The hyperparameter tuning process was conducted using Randomized Search CV to optimize the Stochastic Gradient Descent, XGBoost, and Gradient Boosting Classifier models. This method was chosen due to its efficiency in searching through a wide range of hyperparameters without the computational expense of exhaustive grid search methods. A parameter grid was defined for each model, and Randomized Search CV executed multiple iterations to identify the optimal set of hyperparameters. The Area Under the Curve (AUC) score was the primary metric during hyperparameter tuning.

Comparison with baseline models it's important to consider the strengths and weaknesses of each approach.

In the figure 10 we can see that optimized models were evaluated against the baseline models, using AUC initially as a metric for comparison. After optimization, the SGD model showed a slight improvement in validation AUC, suggesting that the tuning process helped the model generalize better. The best parameters for SGD indicated a combination of elastic net regularization and optimal learning rate, which likely contributed to this improvement.

Referring to this comparison method, baseline comparison, and the conclusion obtained from RNN, we note that RNN has the best performance (0.665) or at least comparable to SDG. We should emphasize that unlike other methods, RNN has performance limitations due to the limitations of the computer architecture where the code is executed and the data is trained (the number of layers affects the accuracy of the output [28]).

The XGB model saw a minor increase in validation AUC from 0.438 to 0.473. Despite the adjustments to parameters like learning rate and subsample ratio, the model still exhibited signs of overfitting, possibly due to its inherent complexity.

GBC's performance declined post-optimization, with the validation AUC decreasing from 0.498 to 0.444. This decline suggests that the model might have become too complex during the tuning process or that the parameter grid did not effectively capture the optimal settings needed to combat overfitting. Even after applying early stopping and adjusting parameters, the model did not show any further improvement.

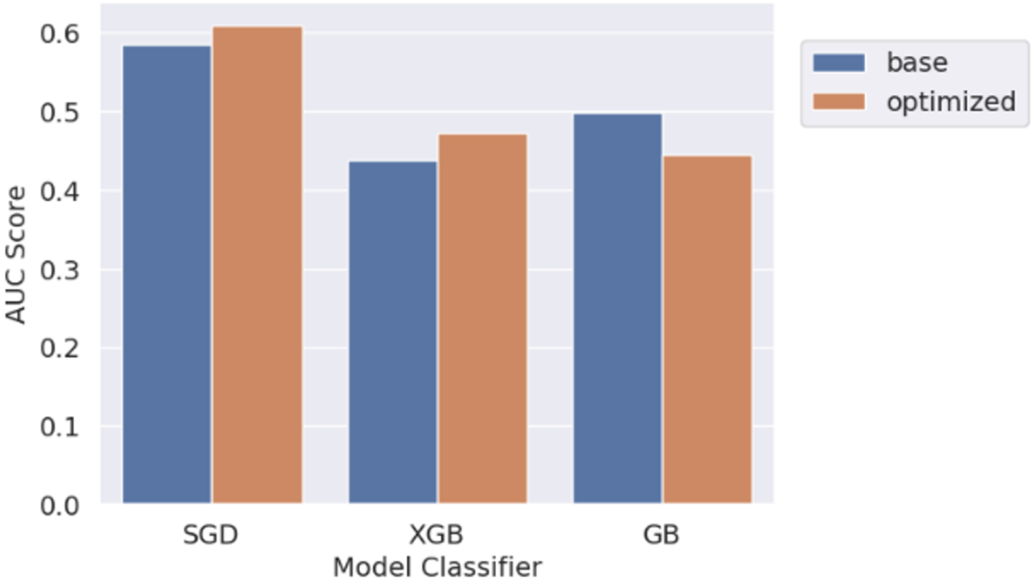


Figure 10.
Baseline models vs optimized models.

3.6.2. Model Evaluation

After completing the hyperparameter tuning process, the Stochastic Gradient Descent Classifier was the best-performing model based on AUC scores. The final evaluation was conducted on the test data using this optimized model. It is important for the model to be evaluated on test set as it provides an unbiased estimate of the model's generalization performance on unseen data.

Performance metrics such as AUC, accuracy, recall, precision, specificity, and the F1 score were measured on the test set during the evaluation of the SGDC model. The 0.7 threshold was set to balance recall and precision while maximizing AUC.

4. Results

This part presents the findings of predicting hospital readmissions and it focuses on the performance of various Machine Learning models, identifying key factors associated with readmissions, and comparing these models considering the following research questions:

What are the main factors that influence the hospital readmissions of diabetic patients?
Which ML model/algorithm performs best in readmission prediction, based on various metrics?

4.1. Main Risk Factors for Hospital Readmissions

To categorize the most significant risk factors for readmission, we have considered several models. We examine models like SGD Classifier (SGDC), Gradient Boosting Classifier (GBC), and XGBoost (XGB) to find out which features these models prioritized when predicting readmission among diabetic patients. Across all models, the most significant contributors to readmission risk were:

- Total Visits
- Number of Diagnoses
- Time in Hospital

Other notable factors included:

- Number of Procedures

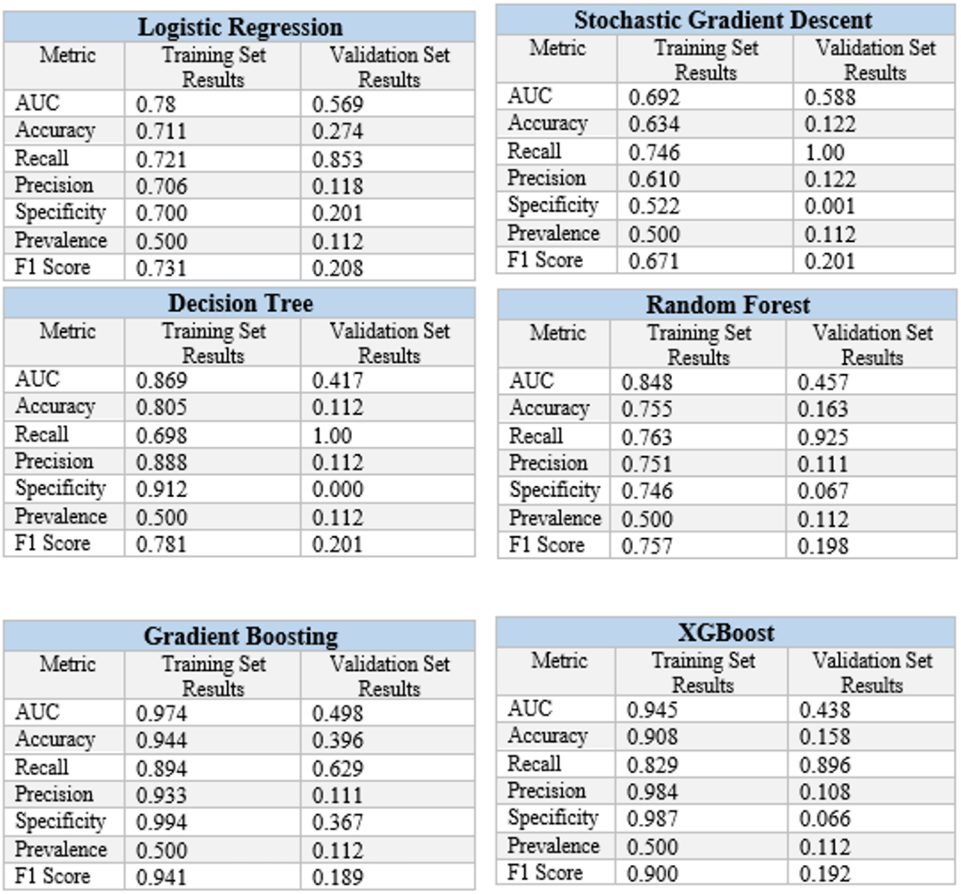
- Gender
- Use of medications like Glipizide.

To further understand the relationship between key features we explored LR model. Findings included:

- Glipizide use was linked to a 1.83 times higher likelihood of readmission.
- Total Visits increased the odds of readmission by 1.51 times for each additional visit.
- Insulin use was linked to a 34% reduction in the likelihood of readmission.

4.2. Evaluation of Machine Learning Algorithms

The algorithms were evaluated on different metrics. The results are shown in the Figure 11.



Logistic Regression

Metric	Training Set Results	Validation Set Results
AUC	0.78	0.569
Accuracy	0.711	0.274
Recall	0.721	0.853
Precision	0.706	0.118
Specificity	0.700	0.201
Prevalence	0.500	0.112
F1 Score	0.731	0.208

Stochastic Gradient Descent

Metric	Training Set Results	Validation Set Results
AUC	0.692	0.588
Accuracy	0.634	0.122
Recall	0.746	1.00
Precision	0.610	0.122
Specificity	0.522	0.001
Prevalence	0.500	0.112
F1 Score	0.671	0.201

Decision Tree

Metric	Training Set Results	Validation Set Results
AUC	0.869	0.417
Accuracy	0.805	0.112
Recall	0.698	1.00
Precision	0.888	0.112
Specificity	0.912	0.000
Prevalence	0.500	0.112
F1 Score	0.781	0.201

Random Forest

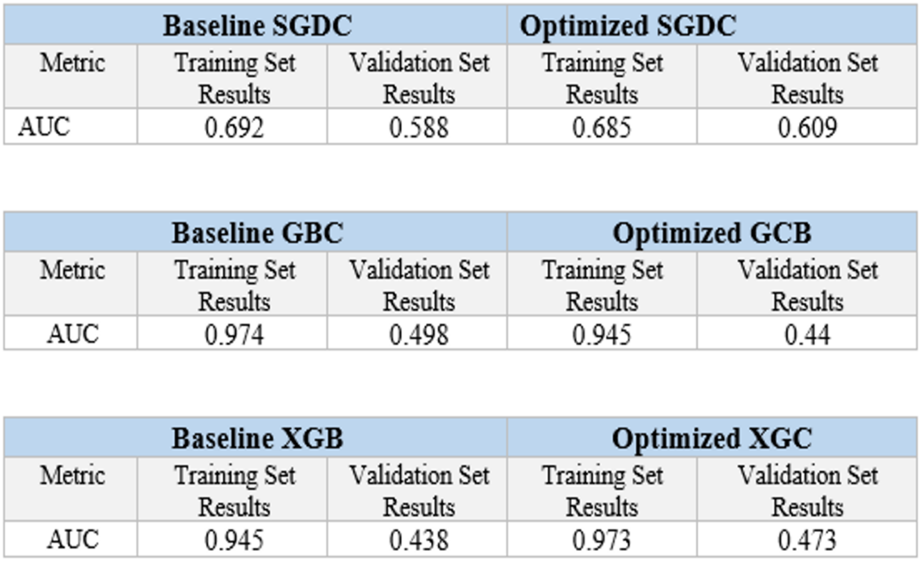
Metric	Training Set Results	Validation Set Results
AUC	0.848	0.457
Accuracy	0.755	0.163
Recall	0.763	0.925
Precision	0.751	0.111
Specificity	0.746	0.067
Prevalence	0.500	0.112
F1 Score	0.757	0.198

Gradient Boosting

Metric	Training Set Results	Validation Set Results
AUC	0.974	0.498
Accuracy	0.944	0.396
Recall	0.894	0.629
Precision	0.933	0.111
Specificity	0.994	0.367
Prevalence	0.500	0.112
F1 Score	0.941	0.189

XGBoost

Metric	Training Set Results	Validation Set Results
AUC	0.945	0.438
Accuracy	0.908	0.158
Recall	0.829	0.896
Precision	0.984	0.108
Specificity	0.987	0.066
Prevalence	0.500	0.112
F1 Score	0.900	0.192



Baseline SGDC			Optimized SGDC	
Metric	Training Set Results	Validation Set Results	Training Set Results	Validation Set Results
AUC	0.692	0.588	0.685	0.609

Baseline GBC			Optimized GCB	
Metric	Training Set Results	Validation Set Results	Training Set Results	Validation Set Results
AUC	0.974	0.498	0.945	0.44

Baseline XGB			Optimized XGC	
Metric	Training Set Results	Validation Set Results	Training Set Results	Validation Set Results
AUC	0.945	0.438	0.973	0.473

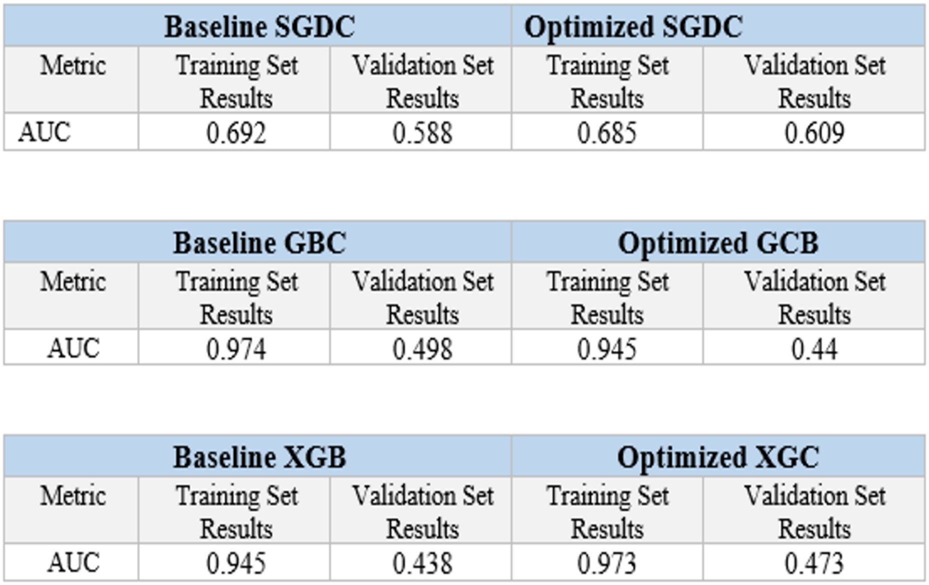
Figure 11.
Comparison of model's performance on all metrics.

Logistic Regression performed moderately well on the training phase (AUC 0.782, F1 0.731) but this did not translate well into validation as AUC dropped to 0.569 and F1 to 0.208. Stochastic Gradient Descent (SGD) was better at maintaining AUC in training and validation phases (0.692 to 0.588) but validated poorly with F1 0.201.

Decision Tree also suffered from severe overfitting, with its training AUC at 0.869 dropping to 0.417 on validation, and a validation specificity of 0. Similarly, Random Forest performed well in training (AUC 0.848) had a decline in validation AUC (0.457) and F1 score (0.198), with high recall but low precision and specificity, indicating a high rate of false positives. Gradient Boosting and XGBoost overfitted heavily, with validation AUCs around 0.498 and 0.438, respectively, despite high training AUCs.

4.3. Evaluation of Optimized Models

Based on the baseline AUC scores, models such as Stochastic Gradient Descent (SGD), Gradient Boosting (GB), and XGBoost (XGB) were selected for further optimization. These models demonstrated better initial performance than the other models.



Baseline SGDC			Optimized SGDC	
Metric	Training Set Results	Validation Set Results	Training Set Results	Validation Set Results
AUC	0.692	0.588	0.685	0.609

Baseline GBC			Optimized GCB	
Metric	Training Set Results	Validation Set Results	Training Set Results	Validation Set Results
AUC	0.974	0.498	0.945	0.44

Baseline XGB			Optimized XGC	
Metric	Training Set Results	Validation Set Results	Training Set Results	Validation Set Results
AUC	0.945	0.438	0.973	0.473

Figure 12.
Comparison of baseline models and optimized models.

Optimized models were evaluated based on their AUC score and compared with their baseline model.

It is seen in the figure 12 that after optimization, SGD got a slight improvement in validation AUC from 0.588 to 0.609, although its training AUC reduced from 0.692 to 0.685. Gradient Boosting achieved a drop in both training AUC from 0.974 to 0.945 and validation from 0.498 to 0.444. XGBoost yielded the best results as it improved from 0.438 to 0.473 and its training AUC rising from 0.945 to 0.973.

4.4. Performance of the Best Model

In this analysis, multiple models were deployed, including XGBoost, Gradient Boosting (GB), Random Forest, Decision Trees, Stochastic Gradient Descent (SGD), and Logistic Regression. It was noticed that models like Gradient Boosting and XGBoost achieved good results on the training data but suffered significant overfitting to the validation data. In comparison, simpler models such as Logistic Regression and Stochastic Gradient Descent did not capture all the features of the dataset but were more generalizable on validation data.

Table 2.
SGDC final performance.

Stochastic gradient descent (Optimized)			
Metric	**Training set results**	**Validation set results**	**Testing set results**
AUC	0.685	0.609	0.6
Accuracy	0.52	0.484	0.475
Recall	0.056	0.693	0.678
Precision	0.748	0.138	0.142
Specificity	0.984	0.458	0.448
Prevalence	0.5	0.112	0.119
F1 Score	0.105	0.23	0.235

Overall, simpler models, tend to generalize better in this dataset.

Gradient Descent Classifier (SGDC) was selected as the best-performing model due to its superior AUC score during validation. The final performance of the SGDC model was evaluated using the testing data to assess its predictive power on unseen data. A threshold of 0.7 was used when testing the model. The final performance of the SGD model is shown in Table 2.

4.5. Discussion

The study identified several factors influencing the likelihood of hospital readmission among diabetic patients. What emerged as fundamental predictors was the total amount of healthcare visits, the amount of healthcare diagnosis made and the length of stay in hospitals. These results are in agreement with other studies which looked in particular at the relationship between healthcare utilization complexity of the patient's condition in predicting readmissions [19].

Additionally, he use of glipizide was associated with a higher likelihood of readmission, while insulin use reduced the risk of readmission. This might suggest that patients on certain medications might be at higher risk of readmission, potentially due to underlying health conditions or complications associated with their treatment.

The study provides a comprehensive approach to data pre-processing of patients' healthcare readmission data. A systematic approach to dealing with missing values, outliers, and categorical variables ensured that the data was well prepared for modelling. Creating new features, like total visits and insulin treatment, provided valuable insights into the predictors of readmission. The use of SMOTE to address class imbalance was important in improving model performance. This technique ensured that the model was not biased towards the majority class and allowed for better generalization to minority class predictions.

However, the research has a number of limitations, such as data integrity, computational constrains, clinical application problems, overfitting etc.

The dataset contained inconsistencies in categorical features and significant missing data. Imputing missing values may introduce bias, while incomplete data may limit the model's ability to fully capture the factors driving readmissions.

Limited computational resources restricted the scope of hyperparameter tuning and model complexity. While Randomized Search CV was used, more exhaustive methods like Grid Search CV were not feasible. The application of SMOTE also increased computational demands. Access to better resources could allow for deeper optimization and more advanced models.

The models were evaluated on historical data but not tested in live clinical settings. Future work should aim to integrate these models into healthcare workflows, providing real-time decision support for healthcare providers.

The study lacked deep clinical expertise, especially concerning medication variables. While ML models identified important patterns, consulting with medical professionals could provide greater insight into clinical nuances, improving the model's relevance.

There appears to be a great overfitting in models like Gradient Boosting and the XGBoost. This means that the current feature set and hyperparameters have not been fully optimized. More work is needed on these models so that they do not overfit, and cross validate with the test set more effectively improving their performance on unseen data.

5. Conclusion

This research used different of ML algorithms like LR, Decision Tree, Stochastic Gradient Descent (SGD), Random Forest, Gradient boosting and XGBoost to predict hospital readmissions for populations of diabetic patients. The results show that, while models like Stochastic Gradient Descent were moderately performing, the more complex algorithms like Gradient Boosting and XGBoost had a tendency to overfit. Most of this work addressed data problems like class imbalance through SMOTE and creating engineering meaningful new features. Even though these efforts improved model performance, further refinement, and tuning are necessary to ensure the models generalize well on unseen data.

Despite data quality, computational resources, and overfitting limitations, ML approaches show promising results in improving discharge protocols and patient outcomes. Future research should not only focus on real-world clinical applications but also explore more advanced models, including deep learning, and apply these approaches to other chronic conditions like heart disease and lung disorders. Collaboration with healthcare professionals will be crucial in transforming these findings into practical tools for predicting readmissions and improving diabetic patient care.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] I. D. Federation, "IDF diabetes Atlas," Retrieved: https://diabetesatlas.org/, 2022.

[2] I. D. Federation, *Diabetes facts and figures.* International Diabetes Federation. https://idf.org/about-diabetes/diabetes-facts-figures, 2024.

[3] H. R. Solutions, "The hospital readmission reduction program: An overview," Retrieved: https://www.healthrecoverysolutions.com/blog/the-hospital-readmission-reduction-program-an-overview, 2023.

[4] C. M. Services, "Hospital readmissions reduction program (HRRP), Centers for Medicare & Medicaid Services," Retrieved: https://www.cms.gov/medicare/quality/value-based-programs/hospital-readmissions. [Accessed 10 September 2024], 2024.

[5] S. Ostling *et al.*, "The relationship between diabetes mellitus and 30-day readmission rates," *Clinical Diabetes and Endocrinology*, vol. 3, pp. 1-8, 2017. https://doi.org/10.1186/s40842-016-0040-x

[6] H. J. J. Audrey and J. Weiss, "Overview of clinical conditions with frequent and costly hospital readmissions by Payer, 2018," Retrieved: https://hcup-us.ahrq.gov/reports/statbriefs/sb278-Conditions-Frequent-Readmissions-By-Payer-2018.jsp. [Accessed 18 February 2025], 2021.

[7] D. J. Rubin, "Correction to: Hospital readmission of patients with diabetes," *Current Diabetes Reports*, vol. 18, no. 4, 2018. https://doi.org/10.1007/s11892-018-1035-z

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts: MIT Press. https://doi.org/10.1007/s10710-017-9314-z 2016.

[9] K. P. Murphy, *Probabilistic machine learning : An introduction*. Cambridge: MIT Press, 2022.

[10] I. University of California, *UCI machine learning repository. Diabetes 130-US hospitals for years 1999–2008 dataset.* UC Irvine Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008, 2014.

[11] A. Géron, *Hands-on machine learning with scikit-learn, keras, and tensorflow*. O'Reilly Media, Inc. https://doi.org/10.1007/s13246-020-00913-z, 2019.

[12] Z. Tafa, "Concurrent implementation of supervised learning algorithms in disease detection," *Journal of Advances in Information Technology*, vol. 7, no. 2, pp. 124–128, 2016. https://doi.org/10.12720/jait.7.2.124-128

[13] J. S. Weissman, J. Z. Ayanian, S. Chasan-Taber, M. J. Sherwood, C. Roth, and A. M. Epstein, "Hospital readmissions and quality of care," *Medical Care*, vol. 37, no. 5, pp. 490-501, 1999. https://doi.org/10.1097/00005650-199905000-00008

[14] A. D. Association, "Economic costs of diabetes in the US in 2017," *Diabetes Care*, vol. 41, no. 5, pp. 917-928, 2018. https://doi.org/10.2337/dci18-0007

[15] H. J. Jiang and M. Henschel, "Characteristics of 30-day all-cause hospital readmissions, 2016–2020. HCUP Statistical Brief, No. 304. Agency for Healthcare Research and Quality, Rockville, MD," Retrieved: https://hcup-us.ahrq.gov/reports/statbriefs/sb304-readmissions-2016-2020.jsp, 2023.

[16] A. Nurse, "Taking steps in the hospital to prevent diabetes-related readmissions, American Nurse," Retrieved: https://www.myamericannurse.com/taking-steps-in-the-hospital-to-prevent-diabetes-related-readmissions/, 2023.

[17] J. G. S. Soh, W. P. Wong, A. Mukhopadhyay, S. C. Quek, and B. C. Tai, "Predictors of 30-day unplanned hospital readmission among adult patients with diabetes mellitus: A systematic review with meta-analysis," *BMJ Open Diabetes Research and Care*, vol. 8, no. 1, p. e001227, 2020. https://doi.org/10.1136/bmjdrc-2020-001227

[18] X. Xiang, C. Liu, Y. Zhang, W. Xiang, and B. Fang, "Predictive modeling of 30-day readmission risk of diabetes patients by logistic regression, artificial neural network, and EasyEnsemble," *Asian Pacific Journal of Tropical Medicine*, vol. 14, no. 9, pp. 417-428, 2021. https://doi.org/10.4103/1995-7645.326254

[19] M. S. Bhuvan, A. Kumar, A. Zafar, and V. Kishore, "Identifying diabetic patients with high risk of readmission," *arXiv preprint arXiv:1602.04257*, 2016. https://arxiv.org/abs/1602.04257

[20] Y. Shang *et al.*, "The 30-days hospital readmission risk in diabetic patients: Predictive modeling with machine learning classifiers," *BMC Medical Informatics and Decision Making*, vol. 21, pp. 1-11, 2021. https://doi.org/10.1186/s12911-021-01423-y

[21] R. Narayanan, "Data leakage in machine learning: Detect and minimize risk, builtin.com," Retrieved: https://builtin.com/machine-learning/data-leakage, 2023.

[22] Globaldee, "SMOTE: A powerful technique for handling imbalanced data - the Content Farm Blog. The Content Farm," Retrieved: https://thecontentfarm.net/smote-for-handling-imbalanced-data/. [Accessed Jan. 19, 2025], 2023.

[23] Scikit-Learn.org, "Stochastic gradient descent - scikit-learn 0.23.2 documentation, scikit-learn.org," Retrieved: https://scikit-learn.org/stable/modules/sgd.html. [Accessed Sep. 9, 2020], 2020.

[24] Scikit-Learn.org, "Decision Trees - scikit-learn 0.22 documentation," Scikit-learn.org," Retrieved: https://scikit-learn.org/stable/modules/tree.html, 2009.

[25] V. Khanna, "Random forests in ML for advanced decision-making, Shelf," Retrieved: https://shelf.io/blog/random-forests-in-machine-learning/, 2024.

[26] D. Nelson, "Gradient boosting classifiers in python with scikit-learn, Stack Abuse," Retrieved: https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn, 2019.

[27] E. AI, "How to interpret a confusion matrix for a machine learning model. Evidently AI," Retrieved: https://www.evidentlyai.com, 2023.

[28] S. Meda and E. Domazet, "Advanced computer architecture optimization for machine learning/deep learning," *CIT Review Journal*, pp. 28–41, 2024. https://doi.org/10.59380/crj.vi5.5108