International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Diabetes Patients' Readmission Prediction

Mahalakshmi Nathan[1], Dr.Dasantila Sherifi[2], Dr. Veerabahu Muthusamy[3]

[1]MS, MBA
Doctoral candidate in Health Informatics
Rutgers University
[2]Assistant Professor and Health Information Management Program Director
Rutgers University
[3]MS(Ortho)
Orthopedic Surgeon, Department of Orthopedics,
Burjeel Hospital, Abu Dhabi, United Arab Emirates.

Abstract
Hospital readmission within 30 days of discharge is one of the most important quality indicators used to assess healthcare performance and financial efficiency. Such readmissions contribute significantly to avoidable healthcare costs and patient morbidity. Among chronic conditions, diabetes mellitus presents a particularly high risk, as patients with diabetes experience markedly greater 30-day readmission rates compared to the general inpatient population. This increased vulnerability highlights the urgent need for predictive modeling to identify high-risk individuals before discharge and implement targeted post-hospital interventions. In this study, we apply and compare multiple supervised machine-learning algorithms—including Logistic Regression, k-Nearest Neighbors, Decision Trees, Random Forests, Gradient Boosting, and Stochastic Gradient Descent classifiers—on a ten-year electronic health record (EHR) dataset comprising over 100,000 diabetes-related hospital encounters. We perform comprehensive data preprocessing that includes removal of irrelevant identifiers, imputation of missing values, feature normalization, and encoding of categorical variables. Feature engineering focuses on clinically relevant attributes such as prior inpatient visits, length of stay, number of diagnoses, number of medications prescribed, and patient demographics. To address severe class imbalance between readmitted and non-readmitted patients, techniques such as Synthetic Minority Oversampling (SMOTE) and class weighting are employed. Models are evaluated using key performance metrics including area under the receiver operating characteristic curve (AUC), precision, recall, and F1 score. Results demonstrate that ensemble methods, particularly Gradient Boosting, achieve superior discrimination and robustness following hyperparameter optimization. The findings suggest that systematic use of predictive analytics can help healthcare institutions identify high-risk diabetic patients, prioritize discharge planning, and optimize follow-up care. Finally, we present actionable business recommendations for clinical integration and discuss model limitations, ethical considerations, and future research directions to enhance predictive performance and real-world applicability.

Keywords: Diabetes; hospital readmission; machine learning; electronic health records; feature engineering; gradient boosting.

1. INTRODUCTION

Hospital readmission is a major indicator of healthcare quality and operational efficiency. Unplanned 30-day readmissions impose heavy financial burdens on healthcare systems and negatively impact patient outcomes. Among chronic diseases, Diabetes Mellitus (DM) is a leading contributor to readmission (Kukde, Chakraborty & Shah, 2024), given its complex metabolic nature and high rate of comorbidities. Studies report that 14.4–22.7% of hospitalized diabetic patients experience readmission within 30 days, compared to 8.5–13.5% in general inpatients (Ostling, Wyckoff, & Ciarkowski, 2017; Hsieh, 2019). These statistics highlight the urgent need for effective data-driven tools that can predict and mitigate readmissions. Diabetes Mellitus is a chronic metabolic disorder characterized by high blood glucose levels due to impaired insulin secretion or resistance. It affects approximately 9.3% of the U.S. population, with 28% undiagnosed (CDC, 2021). The disease's long-term complications—neuropathy, nephropathy, and cardiovascular events—further increase hospital utilization. Consequently, hospitals face penalties when readmission rates exceed federal thresholds (CMS, 2025), emphasizing the necessity for accurate prediction and preventive measures. The purpose of this research is to leverage machine learning algorithms to identify the critical factors influencing readmission. This predictive framework can guide healthcare providers to allocate resources efficiently, improve patient satisfaction, and enhance overall care quality.

1.1 Problem Statement and Impact

The aim is to determine drivers of elevated 30-day readmission among diabetic patients and to predict which patients are at high risk of readmission within 30 days of discharge. Successful prediction supports operational decisions including intensified discharge planning, early follow-up scheduling, medication reconciliation, and resource allocation, which can increase care quality and reduce penalties and costs associated with excessive readmissions. This has direct financial and clinical implications (Dhaliwal & Dang, 2025). Hospitals can prioritize high-impact interventions for small high-risk cohorts, improving patient outcomes while optimizing resources.

1.2 Impact On Business

Hospital readmission is a significant factor in overall medical costs and is a developing metric for care quality (Dhaliwal & Dang, 2025). Diabetes is linked to a higher risk of hospital readmission, just like other chronic medical illnesses. Hospital readmission is a high- priority health care quality measure and target for cost reduction, particularly within 30 days of discharge. Readmissions play a sizable part in the considerable, rising, and expensive diabetes burden that hospitalised patients bear. Diabetes patients with lower readmission rates have the potential to receive better care at lower costs to the health system. Our goal is to offer some understanding of the readmission risk factors.

2. METHODOLOGY

A. Source and Scope

For the purpose of this study, we used datasets made available by UC Irvine Machine Learning Repository. The analysis uses a curated subset of inpatient diabetes encounters spanning 1999–2008, drawn from US hospital records. The dataset comprises more than 100,000 diabetic patient encounters and about 50–55 variables covering patient demographics, admission/discharge metadata, clinical measures (e.g., A1c results, maximum glucose serum), medication indicators, and utilization history (e.g., number of outpatient, inpatient, and emergency visits). The readmission label classifies encounters of patients readmitted within

30 days, readmitted after 30 days, or not readmitted. For modelling purposes this was binarized to readmitted within 30 days (for the first value) and not readmitted within 30 days (for the second and third values).

B. Inclusion Criteria and Initial Data Filtering

- Length of stay: time_in_hospital
- Patient identifiers: encounter_id, patient_nbr
- Patient demographics: race, gender, age, weight ,payer_code
- Admission and Discharge details : admission_source_id ,admission_type_id, discharge_disposition_id
- Patient medical history: number_outpatient ,number_inpatient, number_emergency
- Patient Admission details: medical_speacialty ,diag_1, diag_2 and diag_3, time_in_hospital, number_diagnoses, num_lab_procedures, num_procedures, num_medications
- Clinical Results: max_glu_serum, A1c Result
- Medication Details: diabetesMed, change, 23 features for medications
- Readmission Indicator: readmitted
- Variable Types: Categorical:36, Numerical:13, Boolean:1

Additionally, each encounter used for analysis meets the following four essential criteria:

- A patient has been admitted;
- The length of stay ranged from 1 to 14 days;
- The inpatient was categorised as diabetic (at least one of the three initial diagnosis included);
- The inpatient received medication while it was a patient, as well as laboratory testing.

C. Missing Data and Cleaning

EHR data were heterogeneous with missing and inconsistent values. For example, many encounters lacked A1c or max glucose serum results. The following actions were taken to address such issues.

- Dropping or imputing rare/near-constant columns (Examide, Citoglipton had single values and were removed).
- Mapping categorical ID columns (admission_type_id, admission_source_id, discharge_disposition_id) to categorical types rather than treating them as ordinal numeric values.
- Removing direct identifiers (encounter_id, patient_nbr) from feature sets.
- Imputing missing numeric values with median or clinically informed defaults; one-hot encoding or target encoding for categorical variables with appropriate cardinality control.

Relationship between variables

To explore any potential relationship between variables and gain a basic comprehension of the data, an exploratory data analysis was carried out. Attempts were made to establish the relationships in order to employ fewer variables (given the large initial number of variables). Assessed how different variables affected readmission.

The data table produced by the query included some identifying columns, some numerical columns, and some categorical columns. The readmitted column, which indicates whether a patient was hospitalized within 30 days, more than 30 days, or not at all, is the most crucial one, given the purpose of the study. Discharge disposition id, a column that indicates the patient's next step after being admitted to the hospital, is another important column, as it is associated with status of the patient upon discharge. Looking at the IDs mapping.csv file provided by UCI, it can be noted that 11, 13, 14, 19, 20, and 21 are associated with hospice

or death. Since patients with this discharge disposition cannot be readmitted, they are eliminated from the predictive model.

Further examination of the columns reveals a mixture of categorical and numerical data. Key elements to note, include:

- encounter_id and patient_nbr: these are identifiers and not useful variables
- age and weight: are categorical in this data set
- admission_type_id,discharge_disposition_id,admission_source_id: are numerical data, but are IDs . They should be considered categorical.
- Examide and Citoglipton only have 1 value
- diag1, diag2, diag3 — are categorical and have a lot of values.
- medical_speciality — has many categorical variables, should be considered when making features.

D. Labelling and Class Balance

The original readmission label contained three classes; we created a binary target where the positive class is readmission within 30 days. The positive class was substantially smaller than the negative class, producing class imbalance that can degrade classifier performance. To address skewness we experimented with resampling (SMOTE, random undersampling), class-weighting in classifiers, and calibration strategies during model selection.

Variable Analyses

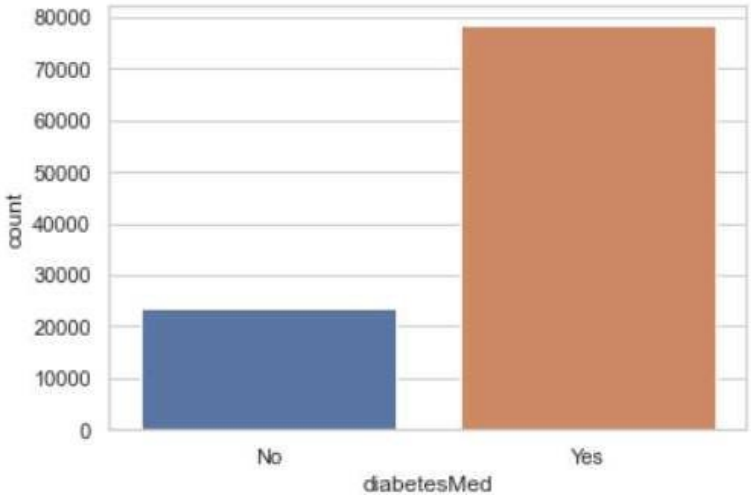

Figure 1. Diabetes Ratio

Figure 1 shows DiabetesMed. Diabetes medicines are given to the majority (77%) of the patients who are admitted in the hospital with a diagnosis of diabetes.

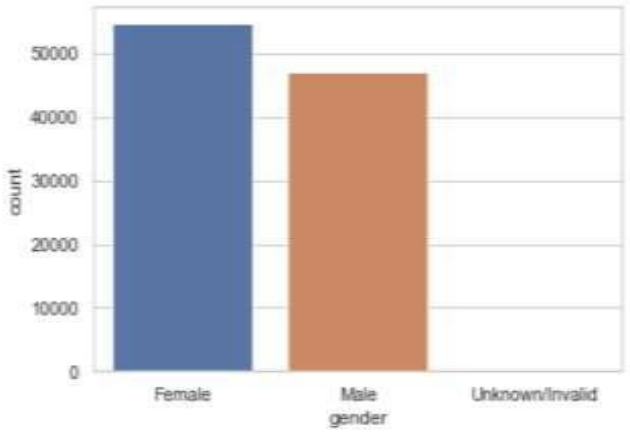

Figure 2. Proportion table

Figure 2 shows the patients' gender: There is almost an equal distribution of male and females that are admitted in the hospital, about 54% females and 46% males. Not a big difference with respect to gender.

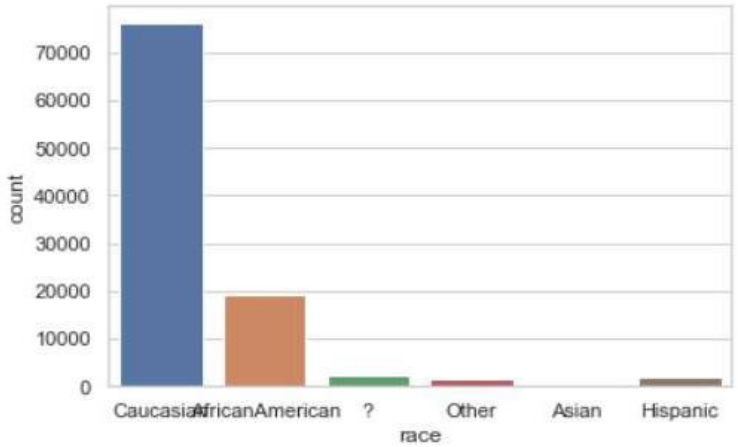

Figure 3. Race

Figure 3 shows the patients' race. There are 5 different categories in the race feature with the maximum (more than 70%) of them reporting the Caucasian race.

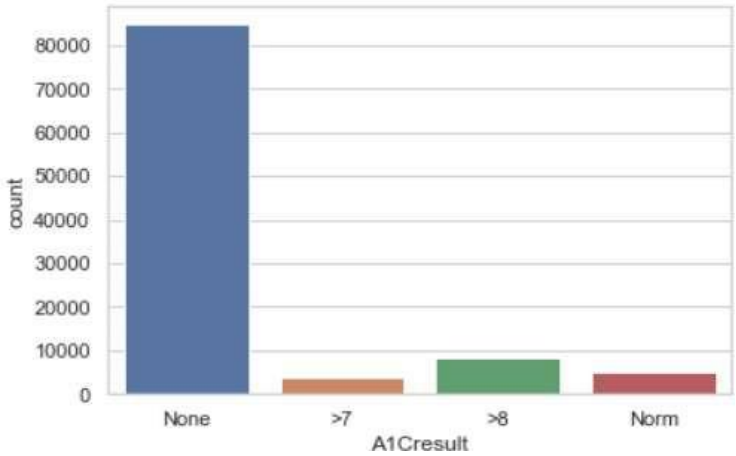

Figure 4. A1C Result

Figure 4 shows there 4 categories in A1Cresult which indicated that most of the patients do not have a reported A1C value.

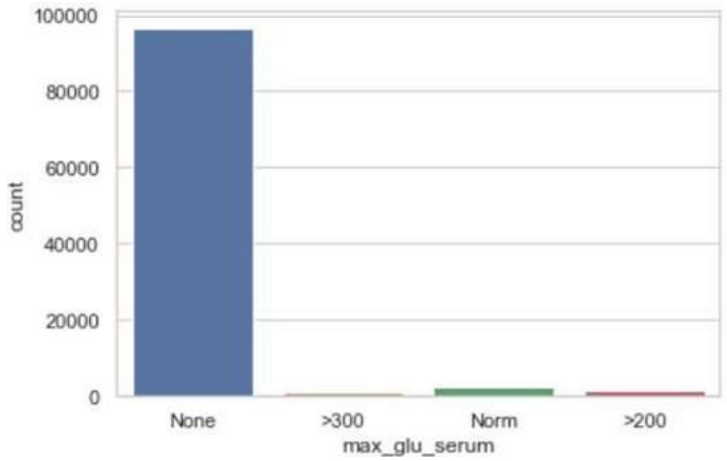

Figure 5. Serum

Figure 5 shows there are 4 categories in max_glu_serum which indicate the majority of the patients do not have a max_glu_serum value reported.

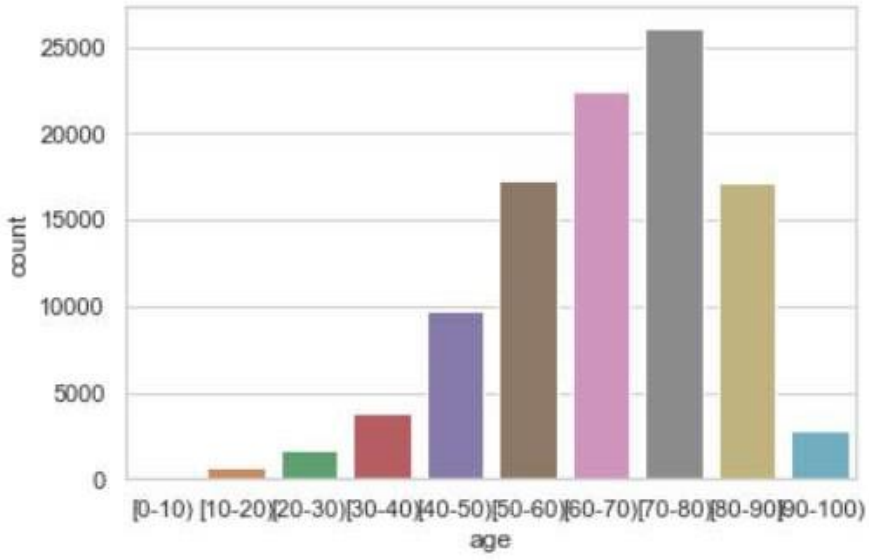

Figure 6. Serum

Figure 6 shows that the largest group of the patients admitted are between of 70 to 80 yrs.

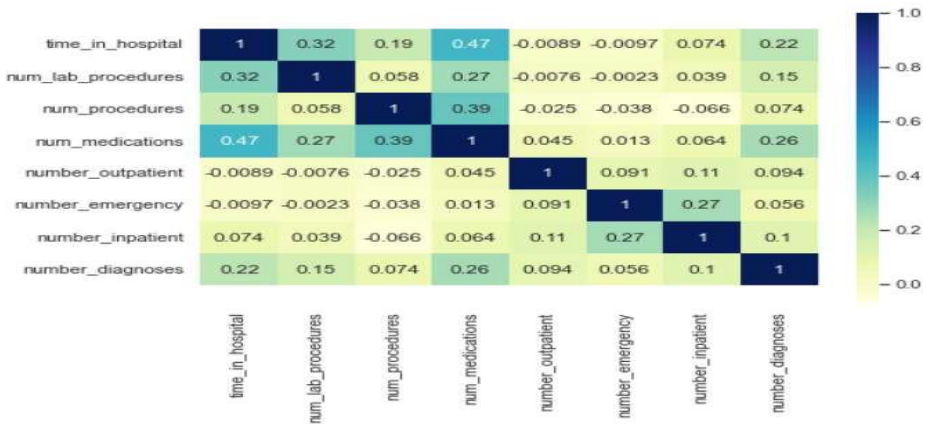

Figure 7. Heatmap Result

Figure 7 shows that there is low correlation between the independent variables. The highest r coefficient (0.47) exists between the number of medications given and time in hospital. This is a favorable findings as it suggests lack of multicollinearity for most of the independent variables.

Multicollinearity generally occurs when there is high correlation between two or more predictor variables. We can identify which variables are affected by multicollinearity and the strength of the correlation. There is a very simple test to assess multicollinearity in a model, the variance inflation factor (VIF). VIF identifies correlation between independent variables and the strength of that correlation.

Heatmap helps us to visualize the correlation between variables. It takes a rectangular data grid as input and then assigns a color intensity to each data cell based on the data value of the cell. This is a great way to get visual clues about the data.

Distribution of variables

Another step in any effort to analyse or model data should be to understand how the variables are distributed. Techniques for distribution visualization can provide quick answers to many important questions. What range do the observations cover? Are they heavily skewed in one direction? Are there significant outliers? When data is positively skewed, most frequent values are low and the tail is towards high values. If Mode< Median< Mean then the distribution is positively skewed. When data are negatively skewed, most frequent values are high and tail is towards low values. If Mode> Median> Mean then the distribution is negatively skewed.

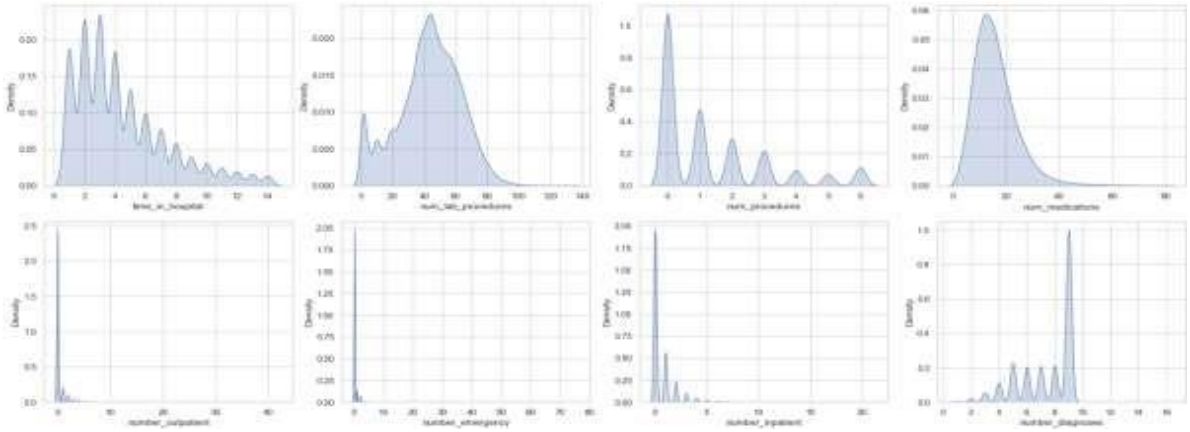

Figure 8. Matrix

Figure 8 shows visualization methods that display frequency, how data spread out over an interval or is grouped. Kernel density estimate (kde) is a useful tool for plotting the shape of a distribution. It helps with visualizing the distribution of data over a continuous interval or time period using a continuous probability density curve in one or more dimensions.

Presence of outliers

Outliers, being the most extreme observations, may include the sample maximum or sample minimum, or both, depending on whether they are extremely high or low. However, 12 the sample maximum and minimum are not always outliers because they may not be unusually far from other observations.

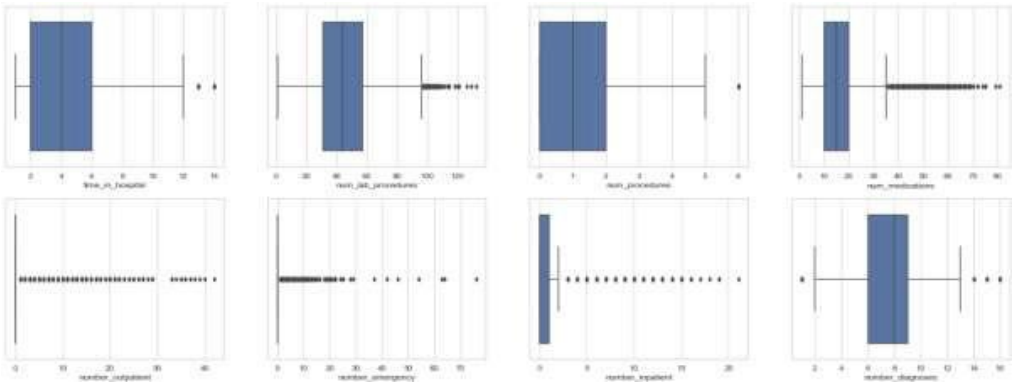

Figure 9. BoxPlot

Figure 9 shows the overall spread of the data while plotting a data point for outliers. This physical point allows their specific values to be easily identified and compared among samples. The box plots, used to identify outliers show some of the data points outside the range of the data.

3. MODELING AND EXPERIMENTAL SETUP

A. Models Evaluated

We evaluated a suite of supervised classifiers representative of linear, tree-based, and ensemble methods:

- Logistic Regression (with L2 regularization and class weighting)
- k-Nearest Neighbors (KNN)
- Decision Tree Classifier
- Random Forest
- Gradient Boosting (GBM/Gradient Boosting Classifier)
- Stochastic Gradient Descent (SGD) classifier (linear classifier with hinge/log loss)

B. Training, Validation, and Hyperparameter Tuning

The dataset was partitioned into train/validation/test sets with a typical 70/15/15 split to enable model selection and honest evaluation. For hyperparameter optimization we used grid/random search cross-validation with carefully chosen parameter grids (number of estimators, max_depth, learning_rate for boosting; n_neighbors for KNN; penalty and C for logistic regression), and 2- to 5-fold CV depending on compute constraints. Model selection prioritized AUC and recall at clinically sensible thresholds, since missing a high-risk patient (false negative) is often costlier than an occasional false positive in a targeted intervention workflow.

C. Evaluation Metrics

We report area under the receiver operating characteristic curve (AUC), precision, recall (sensitivity), F1 score, and calibration plots. Given class imbalance, precision-recall curves and recall at fixed precision thresholds were also examined to contextualize operational utility.

TRAINING, TESTING & VALIDATION:

Train/Test/Validation is a method to measure the accuracy of your model. It is called Train/Test/Validation because the data set is split into three sets: a training set, a testing set, and a validation set. In this study, we used 70% of the data for training, 15% for testing, and 15% for validation. The model was trained by using the training set and was tested by using the testing set. Training the model means to create/build the model. Testing the model means to test the accuracy/ performance of the model. Validating the model means to give an estimate of model skill while tuning model's hyperparameters.

PARAMETERS:

frac/test_size: It is the number that defines the size of the test set. It's very similar to train_size. Generally, either train_size or test_size are provided. If neither is given, then the default share of the dataset that will be used for testing is 0.25, or 25 percent.

random_state: Random state ensures that the splits that you generate are reproducible. With random_state=None , we get different train and test sets across different executions and the shuffling process is out of control. With random_state=0 , we get the same train and test sets across different executions.

```
df_test = df_valid_test.sample(frac = 0.5, random_state = 42)
df_valid = df_valid_test.drop(df_test.index)
```

Figure 10. Testing phase

Figure 10 shows the process of the test in the code structure.

MODELING

Logistic Regression

Logistic regression is an example of supervised learning. It is used to calculate or predict the probability of a binary (yes/no) event occurring. Logistic regression could be applied to machine learning to determine if a person is likely to be readmitted with Diabetes or not. Since we have two possible outcomes to this question - yes they are readmitted, or no they are not readmitted, this is called binary classification.

The probability of a person being infected with Diabetes Mellitus could be based on an autoimmune disorder where insulin producing cells are destroyed. Age, weight, high blood pressure or other indicators are considered to be independent variables along with family history, race/ethnicity etc. These variables are risk factors for Pre-Diabetes and Diabetes Mellitus.

Decision Tree

Decision trees are used as an approach in machine learning to structure the algorithm. A decision tree algorithm will be used to split dataset features through a cost function. The decision tree is grown before being optimized to remove branches that may use irrelevant features, a process called pruning. Parameters such as the depth of the decision tree can also be set, to lower the risk of overfitting or an overly complex tree.

The majority of decision trees in machine learning will be used for classification problems, to categorize objects against learned features. The approach can also be used for regression problems, or as a method of predicting continuous outcomes from unseen data. The main benefits of using a decision tree in machine learning is its simplicity, as the decision-making process is easy to visualise and understand. However, decision trees in machine learning can become overly complex by generating very granular branches, so pruning of the tree structure is often a necessity.

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better for classification problems.

In healthcare, Random Forest can be used to analyze a patient's medical history to identify diseases (Emir et al, 2015). Pharmaceutical scientists use Random Forest to identify the correct combination of components in a medication or predict drug sensitivity. Sometimes Random Forest is even used for computational biology and the study of genetics.

K-Nearest Neighbours

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slower as the size of that data in use grows.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

Some of the advantages of KNN are:

- Easy to implement: Given the algorithm's simplicity and accuracy, it is one of the first classifiers that a new data scientist will learn.
- Adapts easily: As new training samples are added, the algorithm adjusts to account for any new data since all training data is stored into memory.
- Few hyperparameters: KNN only requires a k value and a distance metric, which is low when compared to other machine learning algorithms.

Stochastic Gradient Descent

Gradient descent is an optimization algorithm which is commonly-used to train machine learning models and neural networks. Training data helps these models learn over time, and the cost function within gradient descent specifically acts as a barometer, gauging its accuracy with each iteration of parameter updates. Until the function is close to or equal to zero, the model will continue to adjust its parameters to yield the smallest possible error. Once machine learning models are optimized for accuracy, they can be powerful tools for artificial intelligence (AI) and computer science applications.

Stochastic gradient descent (SGD) runs a training epoch for each example within the dataset and it updates each training example's parameters one at a time. Since you only need to hold one training example, they are easier to store in memory. While these frequent updates can offer more detail and speed, it can result in losses in computational efficiency when compared to batch gradient descent. Its frequent updates can result in noisy gradients, but this can also be helpful in escaping the local minimum and finding the global one.

Gradient Boosting

Machine learning algorithms require more than just fitting models and making predictions to improve accuracy. Most winning models in the industry or in competitions have been using Ensemble Techniques or Feature Engineering to perform better.

Ensemble techniques in particular have gained popularity because of their ease of use compared to Feature Engineering. There are multiple ensemble methods that have proven to increase accuracy when used with advanced machine learning algorithms. One such method is Gradient Boosting.

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

4. EVALUATION OF THE MODEL

EVALUATION METRICS:

➢ ACCURACY represents the proportion of correctly classified observations. Accuracy for the matrix can be calculated by taking average of the values lying across the "main diagonal".

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

➢ PRECISION indicates how many of the correctly predicted cases actually turned out to be positive. Precision is a useful metric in cases where False Positive is a higher concern than False Negatives.

$$Precision = \frac{TP}{TP + FP}$$

➢ RECALL indicates how many of the actual positive cases we were able to predict correctly with our model. Recall is a useful metric in cases where False Negative trumps False Positive.

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

➢ SPECIFICITY (SP) also known as "True Negativity Rate" is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR). The best specificity is 1.0, whereas the worst is 0.0.

$$SP = \frac{TN}{TN + FP} = \frac{TN}{N}$$

➢ ROC_AUC SCORE is a graphical summary of the overall performance of the model, showing the proportion of true positives and false positives at all possible values of probability cut- off. The Area Under the Curve (AUC) summarizes the overall performance of the classifier.

EVALUATION OF DIFFERENT MODELS (BEFORE TUNING)

K-Nearest Neighbours		
Metrics	Train set results	Validation set results
Accuracy Score	60.30%	67.00%
Precision Score	63.30%	16.50%
Recall Score	49.10%	46.90%
AUC Score	65%	62.10%
Prevelance	50%	11.30%

Logistic Regression		
Metrics	Train set results	Validation set results
Accuracy Score	62.80%	66.20%
Precision Score	64.00%	18.00%
Recall Score	55.80%	55.80%
AUC Score	68%	66.10%
Prevelance	50%	11.30%

Stochastic Gradient Desent		
Metrics	Train set results	Validation set results
Accuracy Score	62.40%	66.40%
Precision Score	64.50%	18.00%
Recall Score	55.00%	55.30%
AUC Score	68%	66.10%
Prevelance	50%	11.30%

Decision Tree		
Metrics	Train set results	Validation set results
Accuracy Score	67.20%	63.60%
Precision Score	68.80%	17.00%
Recall score	62.90%	57.00%
AUC score	73.60%	62.30%
Prevelance	50%	11.30%

Random Forest		
Metrics	Train set results	Validation set results
Accuracy Score	64.30%	62.90%
Precision Score	68.80%	17.00%
Recall score	61.20%	60.40%
AUC score	69.60%	65.70%
Prevelance	50%	11.30%

Gradient Boosting Classifier		
Metrics	Train set results	Validation set results
Accuracy Score	69.50%	62.00%
Precision Score	16.40%	70.60%
Recall score	66.80%	57.40%
AUC score	77.20%	63.90%
Prevelance	50%	11.30%

Figure 11. Results

Figure 11 shows the overall accuracy scores of the six models.

HYPER PARAMETER TUNING

Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors.

1) Methods for tuning hyperparameters

● Grid Search:

Grid search is a sort of "brute force" hyperparameter tuning method. We create a grid of possible discrete hyperparameter values then fit the model with every possible combination. We record the model performance for each set then select the combination that has produced the best performance.

● Random Search:

The random search method (as its name implies) chooses values randomly rather than using a predefined set of values like the grid search method. Random search tries a random combination of hyperparameters in each iteration and records the model performance. After several iterations, it returns the mix that produced the best result.

● Bayesian Optimization:

The Bayesian optimization method takes a different approach. This method treats the search for the optimal hyperparameters as an optimization problem. When choosing the next hyperparameter combination, this method considers the previous evaluation results. It then applies a probabilistic function to select the combination that will probably yield the best results. This method discovers a fairly good hyperparameter combination in relatively few iterations.

1. RANDOM FOREST

Three important parameters of `RandomizedSearchCV` are

● **Scoring** = evaluation metric used to pick the best model

- **n_iter** = number of different combinations
- **cv** = number of cross-validation splits

Note that the number of variables and grid size also influences the runtime. Cross- validation is a technique for splitting the data multiple times to get a better estimate of the performance metric. For the purposes of this, we will restrict to 2 cv to reduce the time.

Baseline Random Forest		
Metrics	Train set results	Validation set results
AUC Score	0.696	0.657
Optimized Random Forest		
Metrics	Train set results	Validation set results
AUC Score	0.717	0.66

Figure 12. RandomForest

Figure 12 shows the scores of random forest.

2. GRADIENT BOOSTING

1. n_estimators

- The number of sequential trees to be modeled.
- Though GBM is fairly robust at higher number of trees but it can still overfit at a point. Hence, this should be tuned using CV for a particular learning rate.

2. max_depth

- The maximum depth of a tree.
- Used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.
- Should be tuned using CV.

3. learning_rate

- This determines the impact of each tree on the final outcome. GBM works by starting with an initial estimate which is updated using the output of each tree. The learning parameter controls the magnitude of this change in the estimates.
- Lower values are generally preferred as they make the model robust to the specific characteristics of the tree and thus allow it to generalize well.
- Lower values would require a higher number of trees to model all the relations and will be computationally expensive.

Baseline Gradient Boosting		
Metrics	Train set results	Validation set results
AUC Score	0.772	0.639
Optimized Gradient Boosting		
Metrics	Train set results	Validation set results
AUC Score	0.698	0.669

Figure 13. Gradient Boosting

Figure 13 shows the scores of Gradient Boosting.

3. STOCHASTIC GRADIENT DESCENT

- **scoring** = evaluation metric used to pick the best model.
- **n_iter** = number of different combinations.
- **max_iter** = maximum number of iteration.

Baseline Stochastic Gradient Descent		
Metrics	Train set results	Validation set results
AUC Score	0.676	0.661
Optimized Stochastic Gradient Descent		
Metrics	Train set results	Validation set results
AUC Score	0.676	0.661

Figure 14. Gradient Descent

Figure 14 shows the scores of the gradient descent.

FINAL MODEL SELECTION

As we can see from the above results, Gradient Boosting, Stochastic Gradient Descent and Random Forest give better results after hyperparameter tuning. However, the training and prediction AUC score of Gradient Boosting is far more than Stochastic Gradient Descent followed by Random Forest.

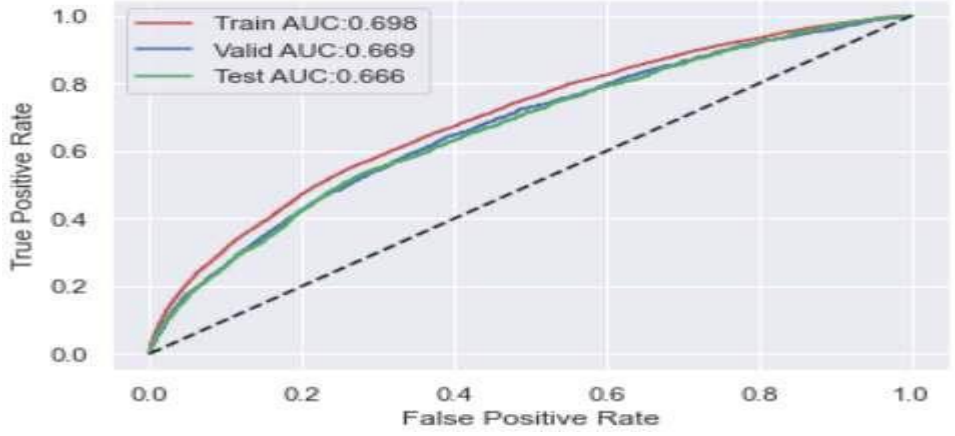

Figure 15. Gradient Boosting

Figure 15 shows the scores of the gradient boosting in a graph.

ROC Curve gives an idea about how well the model performs across all thresholds. Best threshold is selected for which the area under the ROC Curve is maximum, giving the best of Specificity i.e. maximum true positive rate while minimizing the false positive rate.
Following are the results of the final Model chosen to put in production.

Gradient Boosting			
Metrics	Train set results	Validation set results	Test set results
Accuracy Score	64.20%	66.10%	65.30%
Precision Score	66.00%	18.20%	18.50%
Recall score	58.40%	57.40%	57.60%
AUC score	69.80%	66.90%	66.60%
Specificity	69.90%	67.20%	66.40%
Prevelance	50%	11.30%	11.70%

Figure 16. Gradient Results
Figure 16 shows the scores of the gradient boosting.

5. FEATURE ENGINEERING

Exploratory analyses revealed key distributional patterns: the cohort skews older (many admissions in 70–80 age range), majority Caucasian, balanced gender distribution, and high prevalence of diabetes medication use (≈77% receiving diabetes meds). A1c and max_glu_serum were frequently missing; where present, they were strongly informative. Correlation analysis and VIF checks indicated limited multicollinearity (maximum observed correlation ≈ 0.47 between number of medications and time in hospital), allowing many features to remain in the model after careful feature selection.
Feature engineering produced a richer predictive space: categorical features were one-hot encoded (or aggregated by clinical groupings to limit dimensionality), and derived features included counts (total distinct diagnoses), recent utilization (number of admissions in prior year), medication counts, and flags for abnormal labs. After engineering, roughly 143 features (8 numerical, 133 categorical, plus two derived features) were available for modelling.
Features for the predictive model are built in this part. New variables are added to the dataframe for each segment, and it is then kept track of which dataframe columns will be used as features in the prediction model.
Numerical Features
Numerical features are the most user-friendly type of features. There is no need to change these functionalities.
Categorical Features
The following feature class that we wish to develop is categorical variables. Non-numerical data such as race and gender are examples of categorical variables. The simplest way to convert these non-numerical data into variables is to utilize a method called one-hot encoding.
The categorical features will be converted to integers via a process known as one-hot encoding. For each distinct value in a column, a new column is created using one-hot encoding. If the sample has that particular value, the column's value is 1; otherwise, it is 0.
Feature Engineering: Summary

143 characteristics were developed for the machine learning model through this procedure. The features are divided into
8 numerical features
133 categorical features
2 extra features

6. DISCUSSION

The goal of this research was to develop a predictive tool that may assist hospitals in lowering diabetes patients' readmission rates. The predictive tool developed identifies patients in high-risk percentiles using the methodology described above. Patients in high-risk percentiles are more likely to require a readmission; in fact, the model precision is much better for the higher percentiles, meaning that by using this tool, hospitals can focus special attention to such group. For example, the hospital can make better use of their resources by giving these patients highly effective diabetes medications like insulin, metformin, glipizide, and glyburide. When diabetic patients are initially admitted, medical facilities can take precautions by making an A1C and maximal glucose serum test mandatory and administering the appropriate care. At the time of discharge, a follow-up appointment to assess their progress could also be planned.

Use of such tools promises lower readmission rates and higher care quality for diabetic patients admitted to a hospital. The decrease in readmissions can assist hospitals in avoiding financial penalties associated with high readmission rates, resulting in hundreds or even thousands of dollars less spent on healthcare for each diabetic patient, while also enhancing patient outcomes and saving lives.

In summary, the following suggestions can be made:

- By making the A1C and maximal glucose serum tests mandatory, medical facilities will better understand the patient profile at their initial admission. This is because 80–90% of readmitted patients had not had these tests.
- The existing medication regimen for high-risk individuals should be evaluated more carefully. The predictors used in the logistic regression model included medications through variables such as diabetesMed, change, and 23 additional medication features. The model showed that patients prescribed [e.g., insulin, specific drug classes, medication count] had a higher likelihood of readmission. Therefore, optimizing medication plans for high-risk individuals may help reduce this risk.
- Predicted readmissions should be taken into consideration while planning the hospital's annual goals, finances, and infrastructure and inventory.
- Hospitals can make more informed decisions and resource allocation by identifying the high-risk patient population.

7. LIMITATIONS

This study has several limitations. Clinical databases contain valuable but heterogeneous and challenging data defined not only by the quantity of features but also by their complexity. There are missing values, incomplete or inconsistent records, along with high dimensionality. Since external datasets are not collected under controlled experimental conditions, the researcher has limited influence over data quality or collection methods. In comparison to data analysis from a carefully planned experiment or trial, the analysis of external data, such as those used in this study is more difficult, given that one has no control

over the data collection process. However, it is crucial to make use of these vast amounts of clinical data to discover new knowledge that may be unavailable elsewhere. Leveraging such large-scale real-world data remains critical for discovering patterns that may not emerge in smaller, controlled studies.
Another limitation lies in the age of the dataset. Since the data was collected several years ago, clinical protocols, medication standards, and documentation practices within electronic health records (EHRs) have improved considerably. As a result, the model's predictions may not fully reflect current healthcare workflows or readmission prevention strategies in the US. It could however be helpful in other part of the world. Additionally, the diagnosis codes are mapped to ICD-9-CM standards, which limits compatibility with more recent systems that now use ICD-10 coding.

The dataset exhibits class imbalance, as the proportion of patients readmitted within 30 days is substantially lower than other categories, which may influence prediction accuracy. Several important variables—such as access to care or social determinants of health—were not included, even though previous research suggests that access alone may account for up to 58% of variation in readmission rates (Nuckols, 2015).
Predictions could be inaccurate since the proportion of observations for readmission within 30 days is much smaller than for the other classifications. The dataset does not include many significant characteristics, such as access to care, patients location, or other social determinants of health, which has been demonstrated in one study to account for 58 percent of variation in readmission rates (Nuckols, 2015). Depending on the circumstances, there could be a wide range of other factors influencing readmissions.

8. CONCLUSIONS

Acute, unexpected readmissions to the hospital within a predetermined time frame following an initial hospitalization are referred to as readmissions. Hospital readmission is a significant factor in overall medical costs and is becoming a more reliable predictor of care quality. Patients experience disruption, and healthcare systems incur higher costs related to hospital readmissions. Some readmissions are preventable when the healthcare care organization has additional insights on patients' level of risk. With proper data modelling, groups of patients at high risk of readmission are identifiable. Readmission rates are a well-established indicator of health quality across the globe.

The goal of this research project was to create a predictive risk model by identifying factors that contribute to the readmission of diabetic patients. This was accomplished by comparing various classification models that predict readmission and determining the optimum model by utilizing machine learning techniques to analyze critical parameters that have an impact on the all-purpose readmission of a patient with diabetes within 30 days of discharge. The problem of predicting the risk of readmission was framed as a binary classification problem and several available prediction models were developed and evaluated. Using readmission for patients with a diabetes diagnosis as the background for the investigation, this study evaluated how different data preparation strategies, such as feature selection, missing value imputation, and class balancing techniques, may affect the outcomes of prediction modelling. After pre-processing, different predictive models, including Logistic Regression and Decision Tree, were applied to this upgraded dataset to determine the accuracy of readmission risk predictions. Different preprocessing options' effects on key performance metrics, including Area under Curve (AUC), Precision, Recall, and Accuracy, were evaluated. This study provides proof that, in terms of specific evaluation criteria, the majority of offered models with chosen pre-processing strategies significantly outperform the baseline methods (without any pre-processing).

We used multiple machine learning algorithms in this work to forecast readmissions of high-risk patients. By extending earlier research, we balanced classes while taking the skewness of the data into account. According to our findings, decision trees and logistic regression, which were widely used in the literature, are only marginally outperformed by Gradient Booster technique.

The number of visits during the previous year, the length of the stay, the quantity of medications, and the number of diagnoses were some of the major factors that influenced readmissions.

In order to increase the precision of readmission risk prediction, we intend to continue this research and look into the performance of classifiers. Further investigation into the pertinent feature space is also necessary, especially in light of the disparity in results among research papers. The latter is especially crucial because it can result in proactive procedures and regulations meant to address the elements that have a substantial impact on hospital readmissions. The accuracy and validity of prediction models continue to be an important, but elusive goal given the rising cost of hospital readmissions and the increased emphasis on healthcare quality.

9. FUTURE SCOPE

Geographical factors, such as whether a patient is from an urban or rural area, can also be included as features. This would make it possible to determine whether the readmission risk factors vary depending on the patient's geographic location or if the same characteristics are seen everywhere. Additionally, in evaluating the significance of age categorization, this would boost both urban and rural models.

Only diabetic patients were included in this investigation. For other important medical illnesses including heart disease, kidney disease, etc., a readmission prediction model would need to be created. Other significant information from the medical records must be gathered and reviewed, such as family history (to determine hereditary information), emotional status (depression), socioeconomic status, lifestyle habits (exercise), smoking status, and season of readmission. It will be valuable to conduct a more thorough analysis of the extra variables in the dataset and investigate their applicability to estimating the probability of readmission.

Living with diabetes is difficult and upsetting. The condition of a diabetic patient cannot be determined solely by looking at their medical records. To completely comprehend the frequency of readmission of diabetic patients, it is necessary to gather and analyze both subjective and objective patient data. Patients can be interviewed or surveyed to gather subjective data, which will deepen the understanding of patient scenarios. The doctor and patient interaction can also be recorded and analyzed, which could aid in the extraction of significant aspects related to the patient's attitude and willpower. Use of text-mining algorithms may enhance the ability of intelligent models to locate patients who are at high risk of readmission.

With additional time, we could test the AUC and recall using layered models and neural networks. In order to determine which specific categories of features are having an impact on the categorization, we can also delve deeper into the most crucial properties of the models. In order to evaluate if any models perform better, we may also experiment with changing the classification threshold for such models, particularly by lowering the number of false positives in order to raise the precision score.

REFERENCES

1. W. Ahmed, A. Rasool, A. R. Javed, N. Kumar, T. R. Gadekallu, Z. Jalil, and N. Kryvinska, "Security in Next-Generation Mobile Payment Systems: A Comprehensive Survey," IEEE Access, vol. 9, 2021, Art. no. 3105450, doi: 10.1109/ACCESS.2021.3105450.
2. CDC. (2021). National Diabetes Statistics Report. Prevalence of both diagnosed and undiagnosed diabetes. https://www.cdc.gov/diabetes/php/data-research/index.html#:~:text=38.4%20million%20people%20of%20all,or%20older%20(Table%201a).
3. Center for Medicare and Medicaid Services. (2025). Hospital Readmissions Reduction Program. https://www.cms.gov/medicare/payment/prospective-payment-systems/acute-inpatient-pps/hospital-readmissions-reduction-program-hrrp#:~:text=CMS%20calculates%20the%20payment%20reduction,payment%20adjustment%20factor%20of%200.97).
4. J.S. Dhaliwal, A.K. Dang. "Reducing Hospital Readmissions". [Updated 2024 Jun 7]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2025 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK606114/
5. B. Emir, E.T. Masters, J. Mardekian, A. Clair, M. Kuhn and S.L. Silverman. "Identification of a potential fibromyalgia diagnosis using random forest modeling applied to electronic medical records," Journal of Pain Research, 2015. 8, 277–288. https://doi.org/10.2147/jpr.s8256
6. C.J. Hsieh. "High Glucose Variability Increases 30-Day Readmission Rates in Patients with Type 2 Diabetes Hospitalized in Department of Surgery, " Sci Rep **9**, 14240 (2019). https://doi.org/10.1038/s41598-019-50751-7
7. N. Ivanov and Q. Yan, "System-Wide Security for Offline Payment Terminals," in Proc. 17th EAI Int. Conf. on Security and Privacy in Communication Networks (SecureComm), 2021, pp. 99–119.
8. B. U. I. Khan, A. Shah, K. W. Goh, R. R. Putra, A. R. Khan, and M. Chaimanee, "Decentralized Payment Framework for Low-Connectivity Areas Using Ethereum Blockchains," Eng. Technol. Appl. Sci. Res., vol. 14, no. 6, pp. 17798–17810, Dec. 2024.
9. R.D. Kukde, A. Chakraborty and J. Shah. "A Systematic Review of Recent Studies on Hospital Readmissions of Patients With Diabetes," Cureus. 2024 Aug 22;16(8):e67513. doi: 10.7759/cureus.67513. PMID: 39310630; PMCID: PMC11416148. https://pmc.ncbi.nlm.nih.gov/articles/PMC11416148/
10. Y. Lin, Z. Gao, Q. Wang, L. Rui, and Y. Yang, "BcMON: Blockchain Middleware for Offline Networks," arXiv preprint arXiv:2204.01964, 2022.
11. R. Li, Q. Wang, X. Zhang, Q. Wang, D. Galindo, and Y. Xiang, "An Offline Delegatable Cryptocurrency System (DelegaCoin)," arXiv preprint arXiv:2103.12905, 2021.
12. L. Mainetti, M. Aprile, E. Mele, and R. Vergallo, "A Sustainable Approach to Delivering Programmable Peer-to-Peer Offline Payments," Sensors, vol. 23, no. 3, Art. no. 1336, 2023.
13. T. K. Nuckols. "County-Level Variation in Readmission Rates: Implications for the Hospital Readmission Reduction Program's Potential to Succeed," Wiley Online Library, 2015. https://doi.org/10.1111/1475-6773.12268

14. S. Ostling, J. Wyckoff, S.L. Ciarkowski et al. “The relationship between diabetes mellitus and 30-day readmission rates,” Clin Diabetes Endocrinol 3, 3 (2017). https://doi.org/10.1186/s40842-016-0040-x
15. A. V. S. K. Reddy and G. Banda, “ElasticPay: Instant Peer-to-Peer Offline Extended Digital Payment System,” Sensors, vol. 24, no. 24, Art. no. 8034, 2024.
16. S. S. Sravan, S. Mandal, P. J. A. Alphonse, and P. L. Ramesh, “A Partial Offline Payment System for Connecting the Unconnected Using Internet of Things: A Survey,” ACM Comput. Surv., vol. 57, no. 13, Aug. 2024, doi: 10.1145/3687132.
17. UC Irvine Machine Learning Repository. (2025). Diabetes 130-US Hospitals for Years 1999-2008 - UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008
18. J. Yoon and Y. Kim, “Offline Payment of Central Bank Digital Currency Based on a Trusted Platform Module,” Journal of Cybersecurity and Privacy, vol. 5, no. 2, Art. no. 14, 2025.
19. J. Yi, J. Kim, and Y. K. Oh, “Uncovering the Quality Factors Driving the Success of Mobile Payment Apps,” J. Retail. Consum. Serv., vol. 77, Aug. 2024, Art. no. 103641, doi: 10.1016/j.jretconser.2023.103641.