Predicting Readmission within 30 Days for Diabetic Patients

Tagore Kosireddy, Mihret Kemal, Michael Ngala, Feven Tefera

Michigan Technological University
Data Mining Course -5831
Final Project Report (2024)

Abstract

This project addresses the critical issue of readmission risk among diabetic patients within 30 days post-hospital discharge. The goal is to improve patient outcomes and optimize healthcare resource allocation. The study uses a classification approach to predict high-risk patients for readmission using the Diabetes 130-Hospitals Dataset. The workflow starts with data preprocessing and analysis, followed by the application of several machine learning algorithms, such as logistic regression, decision trees, random forests, XGBoost, and CATBoost. Performance evaluation metrics such as recall, precision, F1-score, and AUC score, are used to assess each predictive model in the study. By prioritizing recall and AUC score as primary metrics, Random Forest is obtained as the most suitable model in forecasting re-admissions, offering insights to healthcare decision-makers. Furthermore, feature importance analysis revealed that the number of procedures, number of medications, and time spent in the hospital significantly influence the predictive outcomes, which can help target interventions to reduce readmission risk effectively.

Introduction

Diabetes is a chronic illness that is widely prevalent and poses a significant challenge in the global healthcare sector. Its incidence is steadily increasing, and its management is becoming complex. The number of people affected by diabetes has doubled in the past 20 years (Zimmet et al. 2014). Moreover, the prevalence of diabetes in the US is affecting 30 million individuals and costing $327 billion annually (Stefan et al. 2013). Therefore, providing proper medical care for people with diabetes is crucial.

The majority of patients, especially those with diabetes mellitus require repeated hospitalization due to inadequate treatment (Shang et al. 2021). This repeated hospitalization leads to what is called readmission whereby patients are admitted back to the hospital after being discharged for a certain period. Cases of readmission can be a result of premature discharge, improper initial diagnosis, relapse, and others (Shang et al. 2021). Readmission not only leads to a financial burden on patients but it also leads to a waste of medical resources.

The 30-day readmission rate has become an important performance measure for hospitals used by Centers for Medicare and Medicaid Services (Shang et al. 2021). Having information about the features that can lead to readmission in electronic health records can be of great importance as patients who may be at risk of readmission can be catered to the most and this can lead to effective treatment considering the challenge of availability of resources in healthcare.

To identify patients who are at high risk of readmission, this study used a classification approach to evaluate medical information from the Diabetes 130-Hospitals Dataset. Various machine learning models such as Logistic Regression, Random Forests, XGBoost, and CATBoost were applied in the study. Through performance evaluation, we identified Random forests as the most effective model for accurately predicting readmission within 30 days. Additionally, we got an insight into the variables(number of lab procedures, number of medications, and time spent in hospital) that significantly influence the probability of readmission.

The report is structured into several sections. First, the Related Work section discusses previous studies and compares different methods and results. The Data section provides details on the dataset used, while the Methods section explains the techniques and models used to predict the risk of readmission. The Experiments and Results section covers the steps taken to preprocess the data, split, train and evaluate the models. Finally, the Conclusion section summarizes our findings and discusses any limitations of the study.

Related Work

In previous studies, various methodologies were employed to predict readmission rates among diabetic patients. (Hammoudeh et al. 2018) utilized Convolutional Neural Networks (CNNs) and addressed data imbalances using similar approach applied in this project, that is Synthetic Minority Oversampling Technique (SMOTE). The study demonstrated CNNs' superiority over the prediction outcome. On the other hand, (Bhuvan et al. 2016) employed a diverse set of classification algorithms such as Naive Bayes, Bayesian Networks, Random Forest, AdaBoost, and Neural Networks and highlighted Random Forest as the most accurate one.

Similarly, (Sushmita et al. 2016) explored different machine learning techniques for risk and cost prediction, employing methods like Support Vector Machine, Logistic Regression, Decision Trees, Random Forest, and Generalized Boosted Modeling and emphasized methods achieving high sensitivity/recall just like we did. (Bhuvan et al. 2016) specifically investigated readmission rates among diabetic patients, comparing classifiers like Naive Bayes, Bayesian Networks, Random Forest, AdaBoost, and Neural Networks, and highlighted Random Forest's accuracy.

In another study, (Shang et al. 2021) employed Random Forest, Naive Bayes, and decision tree ensemble, while (Mingle et al. 2017) utilized different ensemble models. Both (Shang et al. 2021; Mingle et al. 2017) studies evaluated their models performance using area-under-the-curve (AUC) metrics just like we did, achieving AUC values ranging from 0.64 to 0.79.

In summary, our study, along with others (Hammoudeh et al. 2018; Sushmita et al. 2016; Bhuvan et al. 2016; Shang et al. 2021; Mingle et al. 2017) utilized the Diabetes 130-Hospitals Dataset for predicting diabetic patient readmission, employing similar data preprocessing techniques albeit with some variations. While some studies (Sushmita et al. 2016; Shang et al. 2021; Mingle et al. 2017))) explored models similar to ours, (Hammoudeh et al. 2018; Sushmita et al. 2016; Bhuvan et al. 2016; Shang et al. 2021; Mingle et al. 2017)) used different algorithms such as Neural Networks, Support Vector Machine, and Naive Bayes. Like (Hammoudeh et al. 2018), we employed one hot-encoding to convert categorical data to binary and addressed data imbalance using SMOTE. Our approach involved employing Logistic Regression, Decision Tree, Random Forest, XGBoost, and CAT Boost models, achieving high recall and AUC score, especially with Random Forest.

Data

We are using the Diabetes 130-Hospitals Dataset from Fairlearn Datasets consisting of 10 years worth of clinical care data at 130 US hospitals and integrated delivery networks. The dataset contains 101,766 rows each describing a patient encounter and 25 features. The dataset is quite unbalanced between the two prediction classes (Figrue 3) with proportions of 11.16% for patients readmitted within 30 days and 88.84% for those not readmitted. Hence, it is important to consider this fact while performing any activities on the dataset. With this in mind, we used a class balancing scheme for the data set using SMOTE (Synthetic Minority Oversampling Technique) and tackled the class imbalance issue within our dataset. We performed essential preprocessing on our dataset like dropping columns and rows with missing and invalid values. Furthermore, one hot encoding is applied for the categorical variables to transform them into binary values for facilitating model training and interpretation. (explained in the experiments and results section in detail).

Methods

We implemented a basic machine learning pipeline to predict the readmission risk of diabetic patients, which is illustrated in Figure 1. This workflow starts with data preprocessing , which involves techniques for balancing classes, cleaning, filtering, and transforming data to make it suitable for analysis. This can involve scaling, normalizing or eliminating missing values.

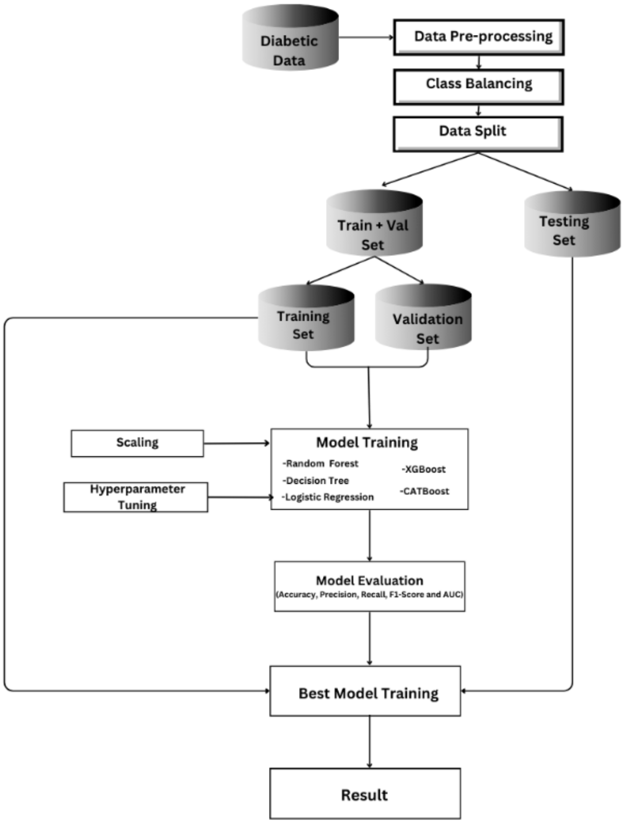

Figure 1: Block diagram for pipeline implementation

As our dataset is imbalanced in terms of target class we employed the SMOTE: synthetic minority over-sampling technique. SMOTE is a popular method used to address class imbalance problems in machine learning, especially in scenarios where the minority class is significantly underrepresented compared to the majority class. The way SMOTE works is by generating synthetic examples in a less application-specific manner. Unlike other methods, it operates in "feature space" rather than "data space". To oversample the minority class, SMOTE takes each minority class sample and creates synthetic examples along the line segments that join any or all of the k minority class's nearest neighbors (Chawla et al. 2002). The number of neighbors chosen randomly depends on the amount of oversampling required. we are using the default implementation which uses five nearest neighbors. The algorithm of how SMOTE works is clearly explained in original paper. By applying SMOTE, the class imbalance problem in our dataset is mitigated.

Next process in the pipeline is data splitting, in which the data is divided into various partitions of training, validation, and testing sets. subsequently, We are going to apply five machine learning models, namely Logistic regression(LR), Decision trees(DT), Random Forests(RF), eXtreme Gradient Boosting(XGBoost), and Category Boosting(CatBoost).

Logistic regression is described as a statistical model that uses the logit function which maps target variable as a sigmoid function of predictor variable (Cox 1958). The logit function returns only values between 0 and 1 for the dependent variable, irrespective of the values of the independent variable. It is used for binary classification tasks, where the target variable has two possible outcomes. It estimates the probability of an event occurring based on input variables and is widely used in various fields, especially in healthcare.

Decision tree is a versatile supervised learning algorithm used for both classification and regression tasks. The algorithm works by recursively splitting the data into subsets based on the most significant feature at each node of the tree (Wu et al. 2008). It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Decision tree use multiple algorithms namely ID3, C4.5,CART, CHAID, MARS to decide to split a node into two or more sub-nodes. The attribute selection is done by using criteria like Entropy, Information gain, Gini index, Gain Ratio, Reduction in Variance, Chi-Square. Overfitting in Decision tree is handled by Pruning them.

By employing the principles of stochastic modeling, random forests construct tree-based classifiers whose capacity can be arbitrarily expanded for increases in accuracy for both training and unseen data (Ho 1995). It is an ensemble learning method that works with a collaborative team of decision trees working together to provide a single output. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

Tree boosting is a highly effective and widely used machine learning method. Extreme Gradient Boosting(Xgboost) is a scalable end-to-end tree boosting system, a sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning (Chen and Guestrin 2016). It works by sequentially adding simple models to correct the errors made by previous models. It gives a prediction model in the form of an ensemble of weak prediction models. It is most commonly used for prediction problems.

CatBoost is a open-sourced gradient boosting library that handles categorical features and outperforms existing publicly available implementations of gradient boosting in terms of quality on a set of popular publicly available datasets(Dorogush, Ershov, and Gulin 2018). It improves on the original gradient boost method for a faster implementation. It uses a method called ordered encoding to encode categorical features. Unique characteristic of CatBoost is that it uses symmetric trees meaning at every depth level, all the decision nodes use the same split condition. It is faster than Xgboost.

Next part of workflow is model training which involves training the model on the dataset to help it understand the data patterns, and finally model evaluation comes in where the model is evaluated on unseen data to check its performance on various metrics.

Experiments and Results

Our experiment started by importing 130 Hospital dataset from fair learn and continued with the data preprocessing step.

Data Preprocessing

The dataset initially consisted of 25 columns, and from those we removed ”max_glu_serum” and ”A1C result” as they contain more than 80% missing data (Figure 2 shows missing values among all the columns). Next, invalid entries in the ”gender” column were removed. We also eliminated rows from the ”race” column that contained a small number of entries that fit into groups with few instances, such as Asian, Hispanic, others, and unknown. Note that we are not taking into account the entire population, which is one of our work’s limitation. Finally, categorical variables were transformed in to binary variables by one-hot encoding and SMOTE from the imbalanced-learn library was applied to solve the imbalance issue the target variable contains (shown in Figure 3).

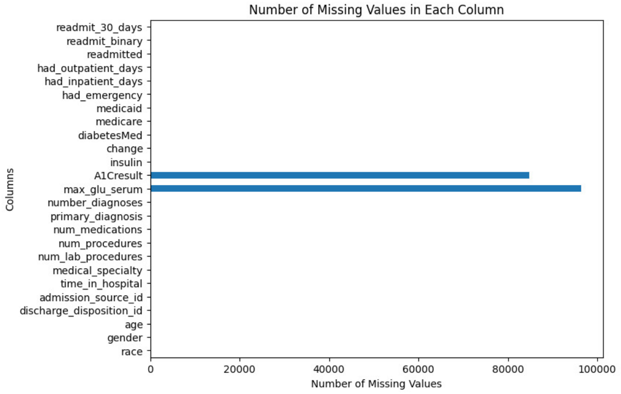

Figure 2: Missing values among columns

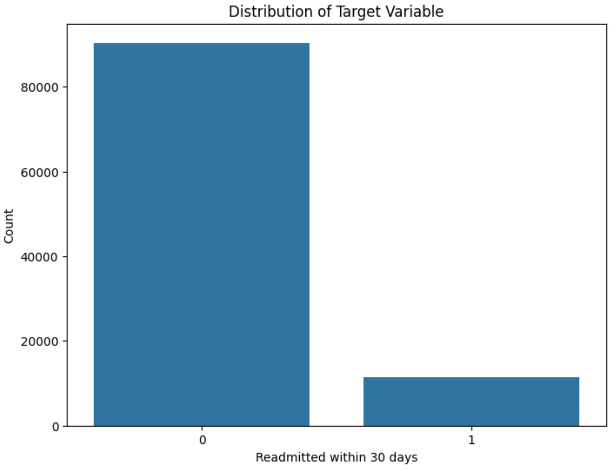

Figure 3: Distribution of target variable

Data Splitting

We divided the data into three parts with 60:20:20 ratio for training, validation and testing sets respectively with stratified split, and applied standard scaling.

Data Modeling

Armed with our preprocessed data, we continued with the process of training five distnict models namely logistic regression, decision tree, random forest, XGBoost and CATBoost.

To select the most suitable hyper-parameters, we conducted a 5-fold cross-validation across a pipeline encompassing all classifiers and their respective parameters. For logistic regression, we considered the regularizing parameter (C) with values of 0.001, 0.01, 0.1, 1, and 10, along with both L1 and L2 regularization techniques. In the case of decision trees, we explored tree depths of 5, 10, 20, minimum sample split of 2, 5, and 10 along minimum leaf samples of 1, 2, 4. For the random forest classifier, we varied the number of trees between 50, 100, and 200 with tree depths between 5, 10, and 20, along minimum samples split and minimum leaf samples of values 2, 5, 10 and 1, 2, 4 respectively. In XGBoost and CATBoost models, we adjusted the learning rate between 0.01, 0.1, and 0.2, maximum depth of 3, 5, and 7, and number of estimators of values 50, 100, and 200.

By defining a grid of hyper-parameters we trained and evaluated each ML model using f1-score as a metric for all hyper-parameter combinations, and we got a suitable hyper-parameter. As a result, Logistic Regression achieved its highest cross-validation f1-score of 0.921 with hyper-parameters C=0.01 and L1 regularization. Conversely, Decision Tree demonstrated its highest cross-validation f1-score of 0.890 with 4 min sample leaf and 2 min sample split. Random Forest exhibited a cross-validation f1-score of 0.930, leveraging a crucial hyper-parameter tweak: classifier n estimators: 200, 2 min sample split and 4 min samples leaf value. On the other hand, XGBoost and CatBoost demonstrated competitive cross-validation f1-scores of 0.922 and 0.921 respectively, employing optimal hyper-parameters: classifier learning rate: 0.1, classifier max depth: 7, and classifier n estimators: 200.

Model Evaluation

In our model evaluation process, we calculated a range of metrics to assess the performance of all the five machine learning models. These metrics included precision, recall, f1- score and AUC score, Table 1 illustrates performance metrics of each model. Additionally, we draw ROC AUC curves which is shown in Figure 4 as another comparison among each models with labels of AUC score.

We prioritize minimizing false negatives to avoid overlooking readmission risks, which pose significant health risks to patients. Simultaneously, we aim to minimize false positives to prevent unnecessary financial burdens and optimize resource allocation. With this in mind, we select recall and AUC score as the major evaluation metrics to choose a suitable model. Random forest model seemed to be the most suitable model for our problem having the highest recall and AUC score values of 0.88 and 0.94 respectively.

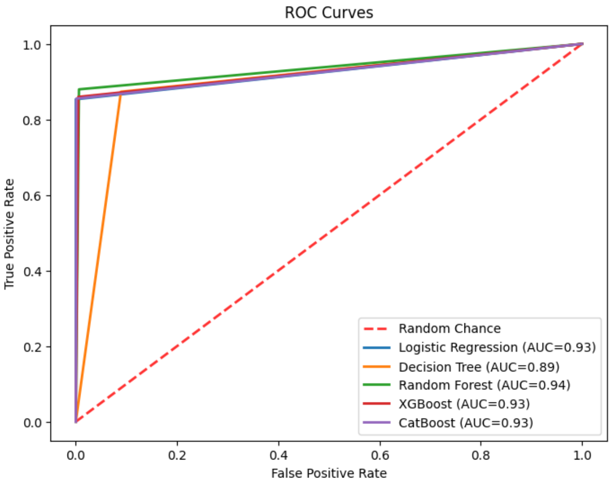

Figure 4: ROC Curve for each Model

After identifying the suitable model (Random forest), we trained it on the entire training dataset and predictions are obtained for the testing data. We also checked the confusion matrix to gauge the model's efficiency in discriminating between classes while minimizing false negatives and false positives, Table 2 shows the confusion matrix of the final Random forest model applied on the test dataset.

Another essential aspect is determining the feature importance of predictor variables. Figure 5 illustrates the feature importance of each predictor variable, revealing that the number of lab procedures, number of medications, and time spent in the hospital are the top three important features. These variables carry significant weight in predicting the risk of readmission within 30 days.

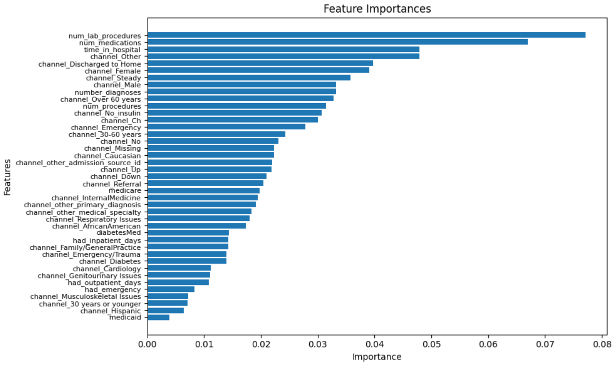

Figure 5: Feature importance

Conclusions

In conclusion, our study aimed to predict readmission risk among diabetic patients within 30 days post-hospital discharge, utilizing machine learning techniques applied to the

Model	Precision	Recall	F1-Score	ROC AUC
Logistic Regression	1.000000	0.853447	0.920929	0.926723
Decision Tree	0.907342	0.873416	0.890055	0.892113
Random Forest	0.991164	0.876541	0.930335	0.934364
XGBoost	0.994510	0.859698	0.922203	0.927476
CATBoost	0.998716	0.855299	0.921460	0.927100

Table 1: Performance Metrics evaluation for each Model

TP = 17170	**FP = 2077**
FN = 108	**TN = 15200**

Table 2: Confusion matrix for the Random Forest model

Diabetes 130-Hospitals Dataset. Despite the challenge of imbalanced class distribution, we were able to manage processing and evaluating the models, identifying Random Forest as the suitable model that will give us accurate results. The model also identified lab procedure, number of medication and time spent in hospital as the top features of importance to check for in the case of readmissions within 30 days. The major limitation of this work is the inability to include populations of minor races.

Moving forward, we aim to expand our scope by exploring different machine learning algorithms beyond those initially considered. For instance, investigating Support Vector Machines (SVM) or neural network (NN) models could provide valuable insights. We hope other models could improve or supplement Random Forests' prediction powers, ultimately maximizing efficiency of the outcomes.

References

Bhuvan, M. S.; Kumar, A.; Zafar, A.; and Kishore, V. 2016. Identifying diabetic patients with high risk of readmission. *arXiv preprint arXiv:1602.04257*.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cox, D. R. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2): 215–232.

Dorogush, A. V.; Ershov, V.; and Gulin, A. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Hammoudeh, A.; Al-Naymat, G.; Ghannam, I.; and Obied, N. 2018. Predicting hospital readmission among diabetics using deep learning. *Procedia Computer Science*, 141: 484–489.

Ho, T. K. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, 278–282. IEEE.

Mingle, D.; et al. 2017. Predicting diabetic readmission rates: moving beyond Hba1c. *Current Trends in Biomedical Engineering & Biosciences*, 7(3): 555707.

Shang, Y.; Jiang, K.; Wang, L.; Zhang, Z.; Zhou, S.; Liu, Y.; Dong, J.; and Wu, H. 2021. The 30-days hospital readmission risk in diabetic patients: predictive modeling with machine learning classifiers. *BMC medical informatics and decision making*, 21: 1–11.

Stefan, M. S.; Pekow, P. S.; Nsa, W.; Priya, A.; Miller, L. E.; Bratzler, D. W.; Rothberg, M. B.; Goldberg, R. J.; Baus, K.; and Lindenauer, P. K. 2013. Hospital performance measures and 30-day readmission rates. *Journal of general internal medicine*, 28: 377–385.

Sushmita, S.; Khulbe, G.; Hasan, A.; Newman, S.; Ravindra, P.; Roy, S. B.; De Cock, M.; and Teredesai, A. 2016. Predicting 30-day risk and cost of" all-cause" hospital readmissions. In *Workshops at the thirtieth AAAI conference on artificial intelligence*.

Wu, X.; Kumar, V.; Quinlan, J. R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G. J.; Ng, A.; Liu, B.; Philip, S. Y.; et al. 2008. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1): 1–37.

Zimmet, P. Z.; Magliano, D. J.; Herman, W. H.; and Shaw, J. E. 2014. Diabetes: a 21st century challenge. *The lancet Diabetes & endocrinology*, 2(1): 56–64.