Predicting Hospital Readmission in Diabetic Patients Using Machine Learning

Master Degree in Artificial Intelligence

Course: Machine Learning

Academic Year: 2024 - 2025

Author:
Fidanza Riccardo - VR516130

UNIVERSITÀ
di VERONA

University of Verona

Abstract

Diabetes mellitus is a chronic condition frequently associated with hospital readmissions due to challenges in effective disease management. In this study, a subset of the original *Diabetes 130-US hospitals for years 1999–2008* dataset is used to develop and compare machine learning (ML) methods for predicting readmission risk in diabetic patients. The prediction task is formulated as a binary classification to determine whether a patient will be readmitted or not. ML models are trained using the complete feature set as well as after applying feature selection. The goal is to identify effective ML approaches for the early detection of high-risk patients and to facilitate more efficient healthcare research.

Contents

1 Introduction

1.1 Context and Motivation

A large proportion of individuals with diabetes mellitus, a leading chronic non-communicable disease, experience recurrent hospital admissions due to insufficient control of their condition. The term "readmission" denotes the return of a patient to the same hospital department within a defined period for issues related to the same underlying disease. Such readmissions are often unplanned and may occur from various factors, such as misdiagnosis during the initial visit, disease recurrence, early discharge, or other clinical complications [2, 3]. These events not only affect patient health outcomes but also contribute significantly to the rising costs of healthcare systems. To mitigate these issues, it is crucial to identify patients at high risk of readmission before discharge, enabling timely interventions and improved care planning. Achieving this requires access to relevant, high-quality clinical data, which forms the basis for predictive modeling.

1.2 Data provenance

The provenance of the data used in this study derives from the *Diabetes 130-US hospitals for years 1999–2008* dataset, which is publicly available on the UCI Machine Learning Repository [1]. Each record corresponds to a single inpatient hospital encounter that meets the following criteria:

1. the encounter must be an inpatient admission
2. the diagnosis must include any form of diabetes
3. the length of stay must be between 1 and 14 days
4. at least one laboratory test must have been performed
5. at least one medication must have been administered during the encounter

1.3 State of the art

Hospital readmission among patients with diabetes mellitus is a well-recognised challenge in clinical practice, carrying substantial implications for both patient outcomes and healthcare costs. Consequently, this topic has been the subject of extensive research, with efforts directed towards identifying key risk factors and developing predictive models to anticipate and ultimately reduce preventable readmissions.
Using the same dataset employed in the present work, a previous study investigated the prediction of 30-day hospital readmission among diabetic patients using machine learning (ML) techniques. The authors applied several well-established algorithms, including logistic regression, decision trees, random forests, achieving a maximum area under the ROC curve (AUC) of approximately 0.68 [5].
Deep neural networks (DNNs) were also explored in this context, yielding a substantially higher AUC of 0.90 [4]. However, the black-box nature of these models, and the resulting lack of interpretability, may limit their clinical adoption.
In this study, we revisit the problem using traditional ML models, with the addition of data engineering and hyperparameter optimisation, to evaluate whether such approaches can match or surpass the performance of existing models while maintaining interpretability.

1.4 Objective

The primary objective of this study is to develop and compare machine learning (ML) methods for predicting the risk of hospital readmission in diabetic patients using the previously described dataset. Unlike prior research that has primarily focused on predicting 30-day readmissions, this work aims to classify whether a patient will be readmitted at any point in the future, without imposing a temporal constraint.
The proposed approach applies traditional ML models commonly employed in earlier studies, enhanced through the integration of hyperparameter optimization and feature selection techniques.

2 Experimental Process

2.1 Data Analysis and Preprocessing

In order to obtain valuable models for predicting hospital readmission, I need to perform a series of data analysis and preprocessing steps. These steps are crucial to ensure that the data are clean, well-structured, and suitable for machine learning algorithms.

2.1.1 Overview of the Dataset

The dataset used in this analysis comprises 101,766 individual observations, where each observation corresponds to a unique hospital encounter for a patient. It contains a total of 48 features, which can be logically grouped into several categories based on their nature and relevance to the analysis.

Firstly, the **demographic features** capture essential patient characteristics, such as race, gender, age, and weight. These variables provide a foundational understanding of the patient population and can be useful for identifying disparities or trends across different groups.

The second group consists of **hospitalization details**, which describe the circumstances and duration of the hospital stay. These include variables like the type of admission, discharge disposition, admission source, and the length of stay in days. Such features are critical for understanding the context of each hospital encounter and may also reflect the severity or urgency of the patient's condition.

Next, the dataset includes **administrative information**, which refers to aspects related to the healthcare provider and organization.

A particularly informative group is the **healthcare utilization** features, which quantify how intensively healthcare resources were used during the patient's interaction with the system. This includes the number of lab tests performed, procedures conducted, medications administered, and counts of outpatient, emergency, and inpatient visits.

Another important category is the **diagnosis information**, which captures diagnostic codes assigned during the hospital encounter. These codes are essential for identifying the clinical conditions that led to hospitalization and can offer predictive insights into patient outcomes.

Additionally, the dataset includes **laboratory results**, which provide numerical values for specific tests such as blood glucose levels and other relevant biomarkers. These features allow for a more direct measurement of the patient's physiological state.

The **medication details** group is particularly extensive and offers a detailed look into the pharmacological treatment received by each patient. This includes information on specific medications prescribed or administered during the hospital stay.

Complementing this, there are also **general medication indicators**, which include two features: one indicating whether there was a change in diabetic medication and another reflecting whether diabetes medication was prescribed at all. These high-level indicators can reveal treatment decisions that may be associated with readmission risk.

Finally, the **target variable** captures whether or not a patient was readmitted to the hospital after discharge. In the original dataset, this information is categorized based on the time to readmission (e.g., within 30 days, after 30 days, or no readmission). For the purpose of this study, the target was simplified into a binary variable, distinguishing between patients who were readmitted and those who were not. Analyzing the distribution of this variable (as shown in Figure 1) allows us to assess the class balance of the dataset, which is crucial for guiding the modeling strategy.

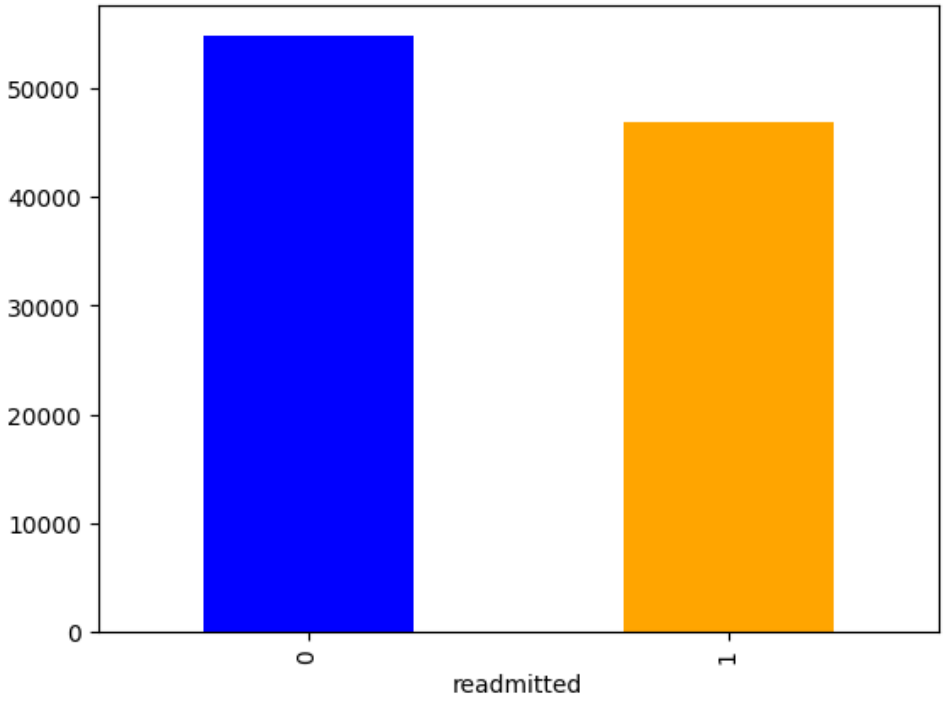

Figure 1: Distribution of the Target Variable

2.1.2 Dealing with Missing Values

After the initial Overview of the Dataset, we proceed to analyze and handle missing values.

Table 1: Percentage of Missing Values per Column

Column	Missing Values (%)
`race`	2.23%
`weight`	96.86%
`payer_code`	39.56%
`medical_specialty`	49.08%
`diag_1`	0.02%
`diag_2`	0.35%
`diag_3`	1.40%
`max_glu_serum`	94.75%
`A1Cresult`	83.28%

Columns such as `weight`, `max_glu_serum`, `A1Cresult`, `payer_code`, and `medical_specialty` present a high proportion of missing values, rendering them unsuitable for reliable analysis. As a result, these features are excluded from the dataset to preserve data integrity. For the remaining variables with lower levels of missingness, imputation is performed using the **mode**, taking into account the categorical nature and distributional characteristics of the data.

2.1.3 Unique values and One-hot Encoding

To prepare the dataset for machine learning models, it is essential to understand the categorical nature of various features. For this purpose, the number of unique values was computed for each categorical feature. Columns containing only a single unique value were subsequently removed, as they do not provide any discriminative information for the analysis.

After identifying the number of unique categories, I applied **One-Hot Encoding** to convert these categorical variables into a numerical format suitable for model training. This technique creates binary columns for each category, ensuring that the data is represented in a format that does not impose ordinal relationships among the categories, which is critical for preserving the integrity of nominal data.
As a result, the dataset is transformed into a format suitable for machine learning algorithms. The new shape of the dataset is (101766 rows $\times$ 103 columns), where the original 48 features have been expanded to 103 features due to the one-hot encoding of categorical variables.

2.2 Creation of a Subset of the Original Dataset

Considering the large size of the dataset, I created a subset to facilitate faster experimentation and model training. The subset is created by randomly selecting a portion of the original dataset, ensuring that it maintains the same distribution of the target variable.
The consequent subset contains 20353 observations and 103 features, which is a manageable size for initial model development and testing.

2.3 Data Splitting

The dataset is split once into two parts, with the aim of creating a training set and a test set. The split is performed using a stratified approach to ensure that the distribution of the target variable is preserved in both sets.

The split is done as follows:

- **Training set:** 80% of the data
- **Test set:** 20% of the data

The training set is used for 5-fold cross-validation and hyperparameter tuning. Specifically, the training set is divided into 5 equal folds. In each of the 5 iterations, 4 folds are used for training the model, and 1 fold is used for validation. This ensures that every fold serves as the validation set exactly once, providing a robust evaluation of model performance during training.
The test set is held out and used only for the final evaluation of the model after cross-validation and tuning are complete.

2.4 Evaluation Metrics

To assess the performance of the machine learning models, several evaluation metrics are used: accuracy, F1 score, recall, and precision. Each of these metrics captures different aspects of model performance and provides a more complete understanding of how well the model is performing:

- **Accuracy** In this case, that the dataset is not highly imbalanced, accuracy can be a useful metric as a general indicator of performance.
- **Recall** (also known as sensitivity) evaluates the model's ability to correctly identify positive cases. In the context of hospital readmission, this is crucial, as failing to detect patients who will be readmitted may have serious consequences.
- **Precision** High precision ensures that when the model predicts a readmission, it is likely to be correct, which is important to avoid unnecessary clinical interventions.
- **F1 Score** is the harmonic mean of precision and recall.

Together, these metrics offer a comprehensive evaluation framework that goes beyond simple accuracy, helping to better understand the trade-offs and effectiveness of each model.

2.5 Dimensionality Reduction through Feature Selection

To reduce the dimensionality of the dataset and retain only the most informative features, we applied **Recursive Feature Elimination with Cross-Validation (RFECV)**. This method works by recursively removing the least important features based on the rankings produced by a specified estimator, while evaluating model performance at each iteration using cross-validation. For this purpose, we selected a **Random Forest classifier** as the base estimator, owing to its robustness and its capability to provide meaningful feature importance scores. RFECV automatically determined the optimal number of features by maximizing the cross-validated `accuracy` score. As a result, the original feature set was reduced from 103 to 30 features, retaining those most relevant for predicting hospital readmission. Subsequently, we applied the same procedures of hyperparameter tuning, cross-validation, and model evaluation described earlier, now using the reduced feature set.

2.6 Hyperparameter Tuning, Cross-Validation, and test set evaluation

I performed hyperparameter tuning using a grid search strategy. This approach systematically explores a predefined set of hyperparameter combinations to identify the configuration that yields the best performance.

Once the optimal hyperparameters were determined, I trained a new model using these settings and evaluated it using cross-validation. During cross-validation, performance metrics were computed for each fold, and the mean and standard deviation of each metric were calculated across all folds.

Finally, to compare the models, I assessed their performance on the test set using the same evaluation metrics described previously, to see how well the models generalize to unseen data.

2.7 Machine Learning Models

A set of models was trained and evaluated. Among these, the Decision Tree Classifier and the Random Forest Classifier were included due to their simplicity and interpretability. These tree-based approaches are particularly advantageous because they can handle both numerical and categorical features without requiring feature scaling, making them robust and versatile across diverse datasets.
In contrast, the K-Nearest Neighbors (KNN) algorithm was employed as a non-parametric, instance-based learning method that does not rely on explicit assumptions about the underlying data distribution.
As a linear and interpretable model, Logistic Regression was incorporated to serve as a strong baseline, particularly effective in high-dimensional settings.
Bernoulli Naive Bayes was selected due to its computational efficiency and its appropriateness for binary-valued feature spaces, which aligns with the characteristics of our dataset.

3 Results

This section presents the results obtained from applying various machine learning models to the three different versions of the dataset: the full feature set, the PCA-reduced set, and the feature-selected set. It includes an analysis of performance metrics for each configuration.

3.1 Model Performance on Full Dataset

Cross-validated results, reported in Table 2, highlight the performance of the models when evaluated on the full dataset using repeated stratified k-fold cross-validation. Among all models, the **Random Forest** consistently achieved the best performance across all metrics, demonstrating robustness and generalization ability. The results obtained on the held-out test set, summarized in Table 3, confirm the trends observed during cross-validation. Again, the Random Forest classifier slightly outperformed the others, showing balanced precision and recall. While the initial results can be considered moderate—not particularly poor, but not outstanding either—I proceed by applying feature selection to investigate whether performance can be enhanced. These methods aim to reduce noise, remove redundant or irrelevant features, and highlight the most informative dimensions of the data. By simplifying the feature space, I hope to improve model generalization, reduce overfitting, and ultimately increase predictive accuracy and robustness on unseen data.

Table 2: Cross-validated performance on the full dataset

Model	**Accuracy**		**F1 Score**		**Recall**		**Precision**	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Decision Tree	0.6210	0.0069	0.6102	0.0065	0.6210	0.0069	0.6216	0.0079
Random Forest	**0.6316**	**0.0097**	**0.6182**	**0.0108**	**0.6316**	**0.0097**	**0.6352**	**0.0102**
KNN	0.5653	0.0061	0.5591	0.0064	0.5653	0.0061	0.5611	0.0065
Logistic Regression	0.6172	0.0122	0.5951	0.0135	0.6172	0.0122	0.6252	0.0146
Bernoulli Naive Bayes	0.6056	0.0095	0.5676	0.0116	0.6056	0.0095	0.6236	0.0126

Table 3: Test performance on the full dataset

Model	**Accuracy**	**F1 Score**	**Recall**	**Precision**
Decision Tree	0.6291	0.6188	0.6291	0.6303
Random Forest	**0.6318**	**0.6188**	**0.6318**	**0.6353**
KNN	0.5770	0.5696	0.5770	0.5732
Logistic Regression	0.6146	0.5943	0.6146	0.6203
Bernoulli Naive Bayes	0.6087	0.5725	0.6087	0.6267

3.2 Model Performance after Feature Selection

Table 4 presents the cross-validated results (mean and standard deviation across folds) for each classifier after feature selection. Once again, the Random Forest classifier achieved the highest performance across all metrics. However, its results show only very slight improvements compared to those obtained with the full feature set, suggesting that feature selection did not lead to significant gains. Table 5 reports the corresponding performance on the held-out test set. Consistent with the cross-validation results, Random Forest outperformed all other models, confirming its robustness and reliability. Overall, while feature selection led

to marginal improvements for some models, it was insufficient to yield a substantial boost in predictive performance. These findings further support the idea that the dataset may have intrinsic limitations in accurately predicting readmission outcomes.

Table 4: Cross-validated performance after feature selection

Model	**Accuracy**		**F1 Score**		**Recall**		**Precision**	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Decision Tree	0.5805	0.0066	0.5625	0.0073	0.5805	0.0066	0.5781	0.0076
Random Forest	**0.6327**	**0.0114**	**0.6210**	**0.0134**	**0.6327**	**0.0114**	**0.6352**	**0.0114**
KNN	0.5657	0.0092	0.5594	0.0091	0.5657	0.0092	0.5616	0.0096
Logistic Regression	0.6169	0.0144	0.5988	0.0151	0.6169	0.0144	0.6215	0.0169
Bernoulli Naive Bayes	0.6083	0.0092	0.6070	0.0095	0.6083	0.0092	0.6068	0.0095

Table 5: Test performance after feature selection

Model	**Accuracy**	**F1 Score**	**Recall**	**Precision**
Decision Tree	0.6276	0.6190	0.6276	0.6277
Random Forest	**0.6342**	**0.6232**	**0.6342**	**0.6366**
KNN	0.5763	0.5692	0.5763	0.5725
Logistic Regression	0.6139	0.5967	0.6139	0.6172
Bernoulli Naive Bayes	0.6146	0.6137	0.6146	0.6135

4 AUC Curve

To evaluate a model's ability to discriminate between classes, I used the Receiver Operating Characteristic (ROC) curve, which illustrates the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) across different decision thresholds. The Area Under the Curve (AUC) provides a single scalar value summarizing this performance: a value of 1 indicates perfect discrimination, 0.5 corresponds to random guessing, and values in between reflect the model's ability to distinguish positive from negative instances.

The results indicated in Figure 2 are related on the feature-selected dataset and show that the Random Forest model achieved the highest discriminative performance, while KNN and Bernoulli Naive Bayes exhibited the lowest. All models performed better than random, yet none reached near-perfect classification, suggesting room for improvement. The AUC metric is particularly valuable as it provides a threshold-independent assessment of model quality, facilitating fair comparisons across different classifiers. These results are in line with the state-of-the-art performance on this task, which reported a maximum AUC of approximately 0.68. Although artificial neural networks (ANNs) can achieve slightly higher AUC scores, they operate as black-box models, offering limited interpretability compared to the tree-based and linear classifiers used here.

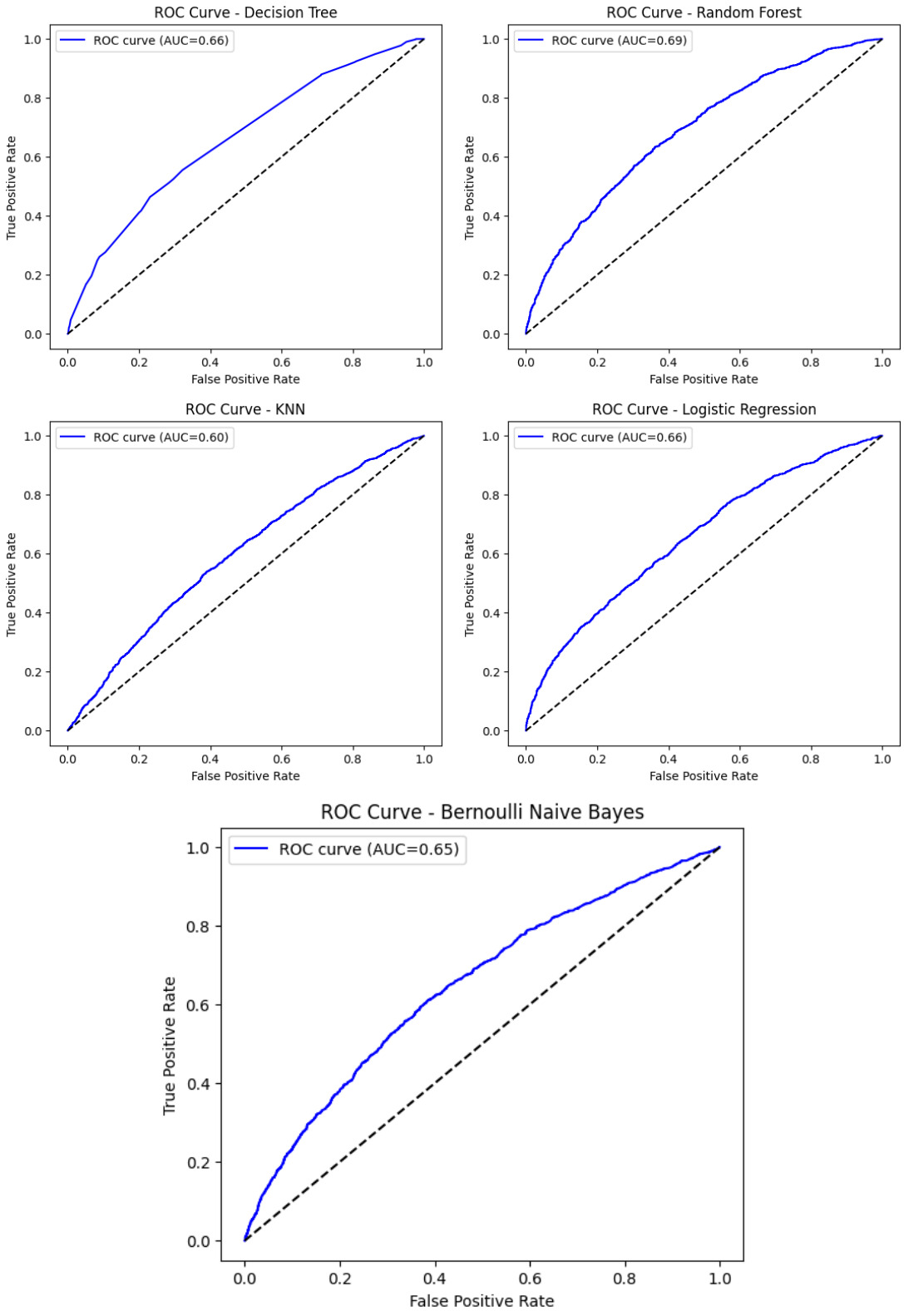

Figure 2: ROC curves of the models evaluated on the test set

5 Conclusion

In this study, we evaluated several machine learning models on a clinical dataset using both the full feature set and a feature-selected subset. Across all experiments, the Random Forest classifier consistently demonstrated the best performance, confirming its robustness and ability to generalize.

Feature selection led to only marginal improvements, indicating that most predictive information is already captured in the original features and that the dataset may have intrinsic limitations in accurately predicting the target outcomes. ROC and AUC analyses further confirmed these observations: Random Forest achieved the highest discriminative performance (AUC $\approx$ 0.69), which is in line with state-of-the-art results reporting a maximum AUC of approximately 0.68.

All tested models, including those from the state-of-the-art, achieved similar performance, pointing to intrinsic limitations in the dataset. Future work should focus on collecting a more informative dataset, as the current features may not be sufficient to substantially improve predictive performance. While the current results are reasonable, better-quality data may be required to achieve higher accuracy and discriminative power.

References

[1] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. 2017. URL: `http://archive.ics.uci.edu/ml`.

[2] Kathleen M Dungan. "The effect of diabetes on hospital readmissions". In: *Journal of Diabetes Science and Technology* 6.5 (2012), pp. 1045–1052.

[3] Elizabeth Eby et al. "Predictors of 30 day hospital readmission in patients with type 2 diabetes: a retrospective, case-control, database study". In: *Current Medical Research and Opinion* 31.1 (2014), pp. 107–114.

[4] Ahmad Hammoudeh et al. "Predicting hospital readmission among diabetics using deep learning". In: *Procedia Computer Science* 141 (2018), pp. 484–489.

[5] Yujuan Shang et al. "The 30-days hospital readmission risk in diabetic patients: predictive modeling with machine learning classifiers". In: *BMC medical informatics and decision making* 21.Suppl 2 (2021), p. 57.