Predicting Diabetes in ICU Patients using Machine Learning

Salma El Shahawy

Mael Illien

The City University of New York

School of Professional Studies

Supervisor

Dr. Nasrin Khansari

Table of Content **2**

Abstract

[COMPLETE LATER]

Introduction

Diabetes mellitus is suspected based on symptoms. It is crucial to diagnose the patient within the first 24 hours of admission to an Intensive Care Unit (ICU) to save the patient's life. However, current diagnosis tests commonly require a long time to determine the patient's diabetes condition. Therefore, the goal of this study was to build machine learning models that utilize ensemble algorithms to predict whether a patient suffers from diabetes mellitus. The study is based on patient information from a dataset compiled by MIT and GOSSIS which are part of a growing global effort and consortium spanning Argentina, Australia, New Zealand, Sri Lanka, Brazil, and more than 200 hospitals in the United States. Specifically, the datasets include more than 130,000 ICU visits from patients, spanning a one-year timeframe. The dataset consists of patients' health conditions and symptoms that will be used to train the ML model. Informed by this dataset, the ensemble algorithm will be able to classify whether a new patient is suspected to suffer from diabetes, which ultimately reduces diagnosis time.

Diabetes Mellitus is a metabolic disturbance disease that is characterized by hyperglycemia and a relative lack, or complete absence of insulin, which by virtue may affect all organ systems in the body [1]. Prevention, by rapid diagnosis, and treatment are important in patients that suffer from this disease. Many of the complications associated with diabetes, such as nephropathy, retinopathy, neuropathy, cardiovascular disease, stroke, and death, can be delayed or prevented by early diagnosis [8,9].

However, finding a way to diagnose the mellitus type within the first 24 hr of admission to the ICU unit is pretty challenging due to delays in the test results. With the advent of machine learning applications in the healthcare industry, early diagnosis of mellitus diabetes can be predicted by developing a machine learning algorithm that utilizes electronic patient's records. Information such as patients' age, BMI, previous blood test outcomes, and previous urine test outcomes will inform the algorithm to classify the diagnosis based on statistical methods.

Literature Review

There are a growing number of new statistical procedures Leo Breiman has called ”algorithmic” [6]. Coming from work primarily in statistics, applied mathematics, and computer science, these techniques are sometimes linked to "data mining," "machine learning," and "statistical learning." Among the variety of algorithmic approaches, there are some that depend on the combination of fitted values from a number of fitting attempts in which individual models are said to be “combined” or “bundled” [14].

One example could be the combination of fitted values from several regression analyses with differing nuisance parameters. Another example would be to average the fitted values from nonparametric regression applied to a large number of single-subject experimental trials [11]. The term “ensemble methods" is commonly reserved for bundled fits produced by a stochastic algorithm, the output of which is some combination of a large number of passes through the data. Such methods are loosely related to iterative procedures on the one hand and to bootstrap procedures on the other. An example is the average of a large number of kernel smoothes of a given variable, each based on a bootstrap sample from the same data set.

Ensemble learning is a powerful machine learning paradigm which has exhibited apparent advantages in many applications. An ensemble in the context of machine learning can be broadly defined as a machine learning system that is constructed with a set of individual models working in parallel and whose outputs are combined with a decision fusion strategy to produce a single answer for a given problem. The idea behind ensemble models is that a “weak” learner or procedure can be strengthened given an opportunity to operate “by committee.” Ensemble methods often perform extremely well and in many cases, can be shown to have desirable statistical properties [5,7,12,16].

Most of the related literature [9,10,13,19,20,21,22] on diabetes classification is based on the Pima Indians Diabetes Database (PIDD) data from the UCI machine learning repository. The approaches include Logistic Regression, Naives Bayes, Support Vector Machines but also tree-based methods like Decision Trees and Random Forests. Several authors have applied ensemble methods like bagging and boosting techniques to improve performance over simple trees. Hasan et al. in [13] places particular importance on data pre-processing steps and uses an ensemble model to achieve better classification performance than similar works.

In summary, our goal is to build a machine learning model that is capable of predicting diabetes. While logistic regression is the most common classification approach in the medical setting, the performance of this technique in other studies using the PIMA dataset was inferior to other methods. We include logistic regression in our study in order to have a ground truth model to reference and compare. While the dataset under study is structurally different, the improved performance in [13] justifies our approach in considering ensemble models built from the aggregation of weak learners with different parameters to improve performance.

Data and Methods

Data Description

The data is made available by MIT’s GOSSIS (Global Open Source Severity of Illness Score) initiative which can be found online on Kaggle. This sample is a small proportion of the target population. While the number of diabetes patients in ICUs is unknown, the WHO estimates that 422 million people suffer from the disease [insert source].

The dataset under study is composed of 130,157 observations with 183 variables. The target variable of interest is *diabetes_mellitus*, which is binary, categorical, and indicates whether a patient has been diagnosed with diabetes. Since we know the labels that the target variable can take on, this is a supervised classification exercise.

The variables are broken down into a number of categories, namely: identifier, demographic, APACHE covariate, vitals, labs, lab_blood_gas, APACHE comorbidity and the target variables. Examples of variables within these categories are given below:

- Identifier (2 variables):
    - encounter_id, hospital_id (mostly provided for indexing purposes)
- Demographic (15 variables):
    - age, bmi, ethnicity, gender (demographic and physical description of the patient)
    - hospital_admit_source, icu_id, icu_type (mostly describing where the patient came from before being admitted to the ICU)
- APACHE covariate (28 variables):
    - albumin_apache, apache_2_diagnosis, bilirubin_apache, gcs_eyes_apache, heart_rate_apache, sodium_apache (variables based on the apache scale, elaborate on this)
- Vitals (52 variables)
    - diastolic and systolic blood pressure, heart rate, respiratory rate, core temperature (maximum and minimum recorded values are included for all variables in this category)
- Labs (60 variables)
    - concentrations of albumin, bilirubin, blood urea nitrogen, calcium, glucose, bicarbonate, hemoglobin, potassium, sodium, counts of platelets, white blood cells, volume proportion of red blood cells (maximum and minimum recorded values are also included for all variables in this category)
- Labs, Blood, Gas (16 variables)
    - arterial pressure of carbon dioxide and oxygen, arterial PH, highest fraction of inspired oxygen (maximum and minimum recorded values are also included for all variables in this category)
- APACHE comorbidity (7 variables)
    - aids, cirrhosis, immunosuppression, leukemia (important comorbidities which may affect the body and disguise the presence of diabetes)
- Target variable (1 variable)
    - the binary (0,1) variable indicating whether a patient suffers from diabetes

APACHE stands for Acute Physiology and Chronic Health Evaluation and is an ICU severity of disease classification and scoring system. APACHE scores are calculated within the first 24hrs of a patient's admission to an ICU. A higher score corresponds to more severe disease and a higher likelihood of death. The variables from the dataset with the 'apache' prefixes and suffixes are used in the computation of the APACHE score.

Most of the variables in the dataset are prefixed with either 'h1' or 'd1' indicating that the reading was taken during the first hour or first day in the ICU stay respectively.

Data Pre-processing

With the initial number of variables and observations, the dataset holds over 23 million data points. The majority of the variables have no predictive power and provide no informational gains to the modeling exercise. It is desirable to reduce the dimensionality of the dataset in order to use only the most explainable features. Determining which variables are safe to drop required analysis of each variable. The main analysis and processing steps are described below.

Distributions and data validation

Some variables were severely skewed. Skew can introduce unintentional bias to the models and variable transformations can be considered in order to mitigate it. Other predictors distributions displayed large irregularities and discontinuities which were suspected to lead to instability. Such variables were either dropped or binned as explained below. Irregularities were discovered using histograms which revealed unusual concentrations of data points at the beginning and ends of some variable ranges. Another important consideration is data validation which requires combing out nonsensical values such as negative ages or weights, or percentages outside the 0-100 range.

Correlation

By analyzing the relationships between each variable and the target variable via correlation and setting up a variance threshold, we can distinguish sets of variables whose distributions are similar and might be problematic in generating a reliable model. Such variables provide no additional information and often obscure models with nonsensical coefficient estimates.

Constants

A number of variables contained values that hardly differed across the entirety of the dataset and were essentially constants. Constants provide no value in modeling and were dropped entirely from the dataset. Quasi-constants which add more noise to the model were also dropped.

Missing values

Datasets often contain a number of missing values which is often referred to as a plague of machine learning. Appropriately dealing with missing values requires careful use of imputation techniques in order to avoid introducing bias. The variable type, whether categorical or numeric, is considered in determining the imputation method. Regression methods were used to impute the numeric variables and KNN was used for categorical variables. Variables with more than 30% of missing values were considered noise and were removed entirely.

Feature engineering

From the original set of predictors, we looked for ways to combine variables in order to reduce the dimensionality. We produced aggregates such as 'danger_level' from the sum of selected features. We also reformatted predictors using techniques such as binning which takes a numerical range and reduces it to a smaller set of discrete values.

Feature selection

Following from the pre-processing steps described above, we proceeded to drop predictors that were determined not to add value and ended up with a subset of selected features, original and engineered to be used in building the classification models.

Methodology

The data processing and modeling work was realized in the Python programming and the associated machine learning package *sklearn*. The methodology is broken down into steps which are individually explained in detail below.

Train Test Split

In order to evaluate the predictions of a model, it is necessary to know the actual value of the target variable of a given observation for comparison. The standard methodology for evaluation is to split the dataset into a training set used for model building, and a testing set used for evaluation. Common split proportions fall in the 70-80% range for training data. In this paper, a train-test split of 80%/20% was used.

Cross Validation

Different models make different assumptions about the underlying distribution of the data. Assumptions about the functional form of the model translate into the number of parameters that need to be estimated. Finding the optimal parameter for a model requires a tuning process. The process involves evaluating model performance while varying the parameter values over a predetermined range. This evaluation is performed on training data using the method of cross-validation in order to preserve the unseen testing data for final prediction evaluation.

The cross-validation method splits the training data into k folds, and builds models from the training data of k-1 of folds in order to evaluate it on the k-th fold. The process is repeated by varying the training and testing folds.

Performance Evaluation

The performance of classification models are evaluated using a confusion matrix which is a tabular summary of the predicted classifications in relation to the true values as shown in Table 1. A number of performance metrics which are derived from the confusion matrix are formulated below.

	Actual class = 1	Actual class = 0
Predicted class = 1	TP (true positive)	FP (false positive)
Predicted class = 0	FN (false negative)	TN (true negative)

Table 1. Confusion matrix

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

$$Specificity,\ Selectivity,\ True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$Sensitivity,\ Recall,\ True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$Precision,\ Positive\ Predictive\ Value = \frac{TP}{TP + FP}$$

Different classification applications will place varying importance on the mistakes made by classifiers and the metrics above will be quoted accordingly. For example, we can consider in the context of this study that it is more costly not to treat a patient who truly suffers from diabetes than to treat a patient for diabetes when they do not actually suffer from the disease.

An alternate metric which provides an overall measure of classification performance is the AUC which stands for the area under the ROC (receiver operating characteristics) curve. The ROC curve provides a visual representation that allows for the comparison of multiple classifiers on the same plot where the best classifier hugs the upper left corner when sensitivity (TPR) is plotted against 1 - specificity.

Modeling

We considered a number of classification models for this exercise, each with different assumptions, in order to find the model with the best prediction performance. We used Logistic Regression (LR) which is a popular model in the medical field as the ground truth model as the basis for comparison.

Logistic Regression (LR)

Instead of modeling the response Y, logistic regression models the probability Y belongs to a particular category. In order to bound the probability between 0 and 1, logistic regression makes use of the logistic function (1) which can be rewritten to take the form of log odds (2) and is linear in X. The model is fit using maximum likelihood which seeks coefficient estimates such that the predicted probability of an observation belonging to a class is as close as possible to the response value. For practical reasons, we instead minimize the negative log likelihood L as formulated below (3).

$$p(Y = 1 \mid X) = p(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}} \text{ .... (1)}$$

$$log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X_1 +...+ \beta_p X_p \text{ .... (2)}$$

$$min\left\{L = - \ log(\Pi_{i:y_i=1} \, p(X_i) \, \Pi_{i':y'\_i=0} \ (1 - p(X_{i'})))\right\} \text{ .... (3)}$$

Regularized Logistic (RL)

A proposed improvement over the basic logistic regression model involves the use of regularization or shrinkage methods. Shrinkage methods minimize a slightly different quantity in order to reduce variance at the cost of introducing a little bias. A penalty term is added to the loss function that we seek to minimize which places a constraint on how large the coefficients are allowed to grow (4). The two common types of penalties used in regularization are L1 (Ridge) and L2 (Lasso). We limit our scope to the latter which provides the added functionality of variable selection by forcing some of the coefficients to be equal to zero.

$$min\left\{L + \lambda \sum_{j=1}^{p} \left|\beta_j\right|\right\} \text{ .... (4)}$$

Naive Bayes (maybe)

[MIGHT NOT BE USED]

Support Vector Machines (SVM)

Support Vector Machines are a generalization of maximal margin classifiers and linear decision boundaries that make use of feature expansion into higher dimensional spaces using kernels where linearity is preserved. We seek a separating hyperplane and a soft margin *M* supported by a few data points (support vectors) that is controlled by the slack variables ε and the non-negative tuning parameter C as formulated below (5). Points are classified based on which side of the hyperplane they lie in.

$$max_M\left\{y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} +...+ \beta_p x_{ip}) > M(1 - \epsilon_i)\right\} \text{ .... (5)}$$

$$\text{Subject to } \sum_{j=1}^{p} \beta_j^{\,2} = 1, \quad \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C$$

When performing linear classification, we simply refer to the method as a support vector classifier. When the support vector classifier is combined with a non-linear kernel, the resulting classifier is known as a support vector machine. The general solution takes the form of (6) where *S* is the collection of indices of the support points and *K* is a kernel function. Examples of kernels include the linear (inner product), polynomial and radial kernels. The latter two are used when a more flexible, non-linear boundary is required to better fit the data.

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i) \ldots\ldots\ldots\ldots\ldots\ldots \text{(6)}$$

Simple Decision Tree (DT)

Tree-based methods involve stratification or segmentation of the predictor space into a number of simple regions represented by tree nodes. For a classification tree, we predict that an observation belongs to the most commonly occurring class of observations in the region to which it belongs. More technically, we classify the observations in node *m* representing region *Rm* to the majority class of node *m* in which the proportion of class k observations is maximized as formulated in (7).

$$max_k \left\{ \hat{p}_{mk} = \frac{1}{N_m} \sum_{x_{i \in R_m}} I(y_i = k) \right\} \ldots\ldots\ldots\ldots \text{(7)}$$

While simple decision trees models can easily be visually represented and intuitively interpreted, large decision trees suffer from a variance problem and are likely to overfit the training data which leads to poor test prediction performance. Cost-complexity pruning is a method used to reduce the size of a tree which might lead to lower variance and better interpretation at the cost of a little bias. Misclassification error, Gini index or entropy/deviance are different measures of node impurity which can be used to control the size of the tree.

Ensemble Methods

In order to improve on the prediction performance of a single model which may suffer from generalization and variance issues, often referred to as weak learners, ensemble methods make use of multiple learning algorithms with the aim of obtaining better results than any constituent model. The methods that follow are examples of ensemble models which can be applied to tree-based models but also extended to other methods.

Bagging

Bagging is short for Bootstrap Aggregation which is a general-purpose parallel procedure for reducing the variance of a statistical learning method. This involves generating B different bootstrapped training data sets and training a model on the bth bootstrapped training set in order to get $\hat{f}^{*b}(x)$ and finally average all the predictions to obtain:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \ldots\ldots\ldots\ldots\ldots\ldots (8)$$

In the classification setting we record the class predicted by each of the B trees and take a majority vote to determine the class. The number of models B is not a critical parameter with bagging and there is no risk of overfitting.

Random Forests (RF)

Random Forests is a tree-based method that offers an improvement over bagging by way of a small tweak that decorates the data which has the effect of making the constituent models more dissimilar. Like bagging, individual models are trained on bootstrapped samples, but a random sample of *m* predictors is chosen from the full set of predictors . When *m=p*, we simply get bagging.

Boosting

Boosting in another general-purpose procedure that can be applied to a multitude of models. It is an additive method where models are learned in series. Boosting does not involve bootstrap sampling, but instead models are fit on a modified version of the original data set. Unlike the parallel bagging procedure, boosted models are built in a sequential way using residual output of previous models which. In some cases, the residuals are weighted to form pseudo-residuals which are used to place more emphasis on previously misclassified observations. The boosting method is designed to learn slowly and controlled by a shrinkage parameter to mitigate the risks of overfitting the training data. The most popular boosting procedure is called AdaBoost.M1 [insert future reference].

Limitations

While the results of this investigation could potentially be generalized to the diabetic population as a whole, the dataset and study are limited to patients in ICUs. We cannot say how the prediction results would extend to diabetic patients outside of ICUs since a number of variables from the current dataset would not be collected. Furthermore, the varying availability and quality of healthcare around the world places a limitation on how this study can be extended.

While the GOSSIS Consortium pools data from Oceania, Southeast Asia, South America and North America, the dataset lacks representation of Europe, Africa, and the rest of Asia where the prevalence of diabetes might differ due to nutritional and environmental factors.

Finally, While there is plenty of diabetes classification literature using the Pima Indians dataset, there is no published research using data from ICUs to serve as a comparison for our work.

Results

[COMPLETE LATER]

Discussion

[COMPLETE LATER]

Conclusion

[COMPLETE LATER]

References

[1] A. S. Alanazi and M. A. Mezher, "Using Machine Learning Algorithms For Prediction Of Diabetes Mellitus," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-3, doi: 10.1109/ICCIT-144147971.2020.9213708.

[2] R. A. Berk, “An Introduction to Ensemble Methods for Data Analysis,” Sociological Methods & Research, vol. 34, no. 3, pp. 263–295, 2006.

[3] R. A. Berk, “Data Mining within a Regression Framework,” Data Mining and Knowledge Discovery Handbook, pp. 231–255, 2004.

[4] R. A. Berk, Y. He, and S. B. Sorenson, “Developing a Practical Forecasting Screener for Domestic Violence Incidents,” Evaluation Review, vol. 29, no. 4, pp. 358–383, 2005.

[5] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 3, pp. 261–277, 2001.

[6] L. Breiman, “Statistical Modeling: The Two Cultures,” Statistical Science, vol. 16, no. 3, 2001.

[7] P. Bühlmann and B. Yu, “Analyzing bagging,” The Annals of Statistics, vol. 30, no. 4, 2002.

[8] Diagnosis and Classification of Diabetes Mellitus, American Diabetes Association, Diabetes Care Jan 2010, 33 (Supplement 1) S62-S69; DOI: 10.2337/dc10-S062

[9] K. Driss, W. Boulila, A. Batool and J. Ahmad, "A Novel Approach for Classifying Diabetes’ Patients Based on Imputation and Machine Learning," 2020 International Conference on UK-China Emerging Technologies (UCET), Glasgow, UK, 2020, pp. 1-4, doi: 10.1109/UCET51115.2020.9205378.

[10] D. Dutta, D. Paul and P. Ghosh, "Analysing Feature Importances for Diabetes Prediction using Machine Learning," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2018, pp. 924-928, doi: 10.1109/IEMCON.2018.8614871.

[11] J. Faraway, “Human Animation Using Nonparametric Regression,” Journal of Computational and Graphical Statistics, vol. 13, no. 3, pp. 537–553, 2004.

[12] Y. Grandvalet, “Bagging Equalizes Influence,” Machine Learning, vol. 55, no. 3, pp. 251–270, 2004.

[13] M. K. Hasan, M. A. Alam, D. Das, E. Hossain and M. Hasan, "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers," in IEEE Access, vol. 8, pp. 76516-76531, 2020, doi: 10.1109/ACCESS.2020.2989857.

[14] T. Hothorn, “Bundling Predictors in R,” DSC 2003 Working Papers, 2003. [Online]. Available: https://www.r-project.org/conferences/DSC-2003/Drafts/Hothorn.pdf.

[15] G. Luo, Automatically explaining machine learning prediction results: a demonstration on type 2 diabetes risk prediction. Health Inf Sci Syst 4, 2 (2016). https://doi.org/10.1186/s13755-016-0015-4

[16] S. Mannor, R. Meir, and T. Zhang, “The Consistency of Greedy Algorithms for Classification,” Lecture Notes in Computer Science, pp. 319–333, 2002.

[17] N. Nai-arun, R. Moungmai, Comparison of Classifiers for the Risk of Diabetes Prediction, Procedia Computer Science, Volume 69, 2015, Pages 132-142, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.10.014.

[18] S. Perveen, M. Shahbaz, A. Guergachi, K. Keshavjee, Performance Analysis of Data Mining Classification Techniques to Predict Diabetes, Procedia Computer Science, Volume 82, 2016, Pages 115-121, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2016.04.016.

[19] A. M. Posonia, S. Vigneshwari and D. J. Rani, "Machine Learning based Diabetes Prediction using Decision Tree J48," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 2020, pp. 498-502, doi: 10.1109/ICISS49785.2020.9316001.

[20] D. Sisodia, D. S. Sisodia, Prediction of Diabetes using Classification Algorithms, Procedia Computer Science, Volume 132, 2018, Pages 1578-1585, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.122.

[21] P. Sonar and K. JayaMalini, "Diabetes Prediction Using Different Machine Learning Approaches," 2019 3rd International Conference on Computing Methodologies and

Communication (ICCMC), Erode, India, 2019, pp. 367-371, doi: 10.1109/ICCMC.2019.8819841.

[22] V. V. Vijayan and C. Anjali, "Prediction and diagnosis of diabetes mellitus — A machine learning approach," 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Trivandrum, India, 2015, pp. 122-127, doi: 10.1109/RAICS.2015.7488400.