Diabetes prediction: a supervised learning approach

Enrico Mannarino, Simone Massardi, Giorgia Prina

February, 2023

Abstract

Diabetes is a chronic disease that depends on a multitude of factors. Based on a set of behavioral risk factors, Machine Learning models were built using the *Knime Analytics Platform* to solve a binary classification problem in order to predict whether a person has diabetes or not. These models were carefully evaluated and compared to select the best solution in terms of performance and reliability. The results obtained will serve as a basis for the construction of a useful tool to support medical diagnosis and prevention activities.

Contents

1. Introduction

Diabetes is a chronic disease characterized by high levels of glucose in the blood (hyperglycemia) and caused by an altered amount or function of insulin. Insulin is the hormone produced by the pancreas that allows glucose to enter cells and its subsequent use as an energy source. When this mechanism is altered, glucose accumulates in the bloodstream. According to recent estimates, about 10% of the world's population has diabetes and the trend is increasing due to the aging of the population and the obesity rate. It is well known in medical literature that some of the main risk factors for diabetes are obesity, lack of physical activity, family history and poor diet.

Based on the data available, a binary classification model is to be implemented with the aim of predicting whether a person has diabetes.

Below we will present the description of the dataset, the preprocessing phase, the analysis and comparison of the different classification methods and techniques that have been implemented to select the best model, and finally the presentation of the results.

2. Dataset description

The dataset consists of 40108 observations and 18 features:

- *Age*: 13-level age category from 1 (18-24) to 13 (80 or older).
- *Sex*: 0 = female, 1 = male.
- *HighChol*: 0 = no high cholesterol, 1 = high cholesterol.
- *CholCheck*: 0 = no cholesterol check in 5 years, 1 = cholesterol check in 5 years.
- *BMI*: Body Mass Index.
- *Smoker*: have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes]; 0 = no, 1 = yes.
- *HeartDiseaseorAttack*: coronary heart disease (CHD) or myocardial infarction (MI); 0 = no, 1 = yes.
- *PhysActivity*: physical activity in past 30 days - not including job; 0 = no, 1 = yes.
- *Fruits*: consume fruit one or more times per day; 0 = no, 1 = yes.

- *Veggies*: consume vegetables 1 or more times per day; 0 = no, 1 = yes.
- *HvyAlcoholConsump*: Adult male: more than 14 drinks per week. Adult female: more than 7 drinks per week. 0 = no, 1 = yes.
- *GenHlth*: would you say that in general your health is: (scale 1-5) 1 = excellent, 2 = very good, 3 = good, 4 = fair, 5 = poor.
- *MentHlth*: days of poor mental health scale 1-30 days.
- *PhysHlth*: physical illness or injury days in past 30 days scale 1-30.
- *DiffWalk*: do you have serious difficulty walking or climbing stairs? 0 = no, 1 = yes.
- *Hypertension*: 0 = no hypertension, 1 = hypertension.
- *Stroke*: ever had a stroke? 0 = no, 1 = yes.
- *Diabetes*: 0 = no diabetes, 1 = diabetes (target variable).

3. Preprocessing

We performed the following operations on the dataset in order to make it more suitable for implementing the models.

3.1. Correlation, class balance, missing values, outliers

First, we checked for possible multicollinearity between the variables (excluding the class attribute). Since they were categorical variables, we performed a rank correlation test using the Spearman coefficient and set a threshold value of 0.8. Since no pair of variables exceeded this threshold, we did not exclude any for redundancy.

We then checked the balance of the class attribute and found that it is a well-balanced class (51% Diabetes = 1, 49% Diabetes = 0).

None of the variables provided any missing value.

Finally, we noticed that the BMI variable has some unusually high values (> 90) that remain possible although unlikely. Since we have no further information on how the data was collected, we decided to keep these observations in the dataset.

3.2. Feature transformation

We discretized three variables: *BMI*, *MentHlth*, and *PhysHlth*. The discretization of the BMI index was performed by assigning each value to the corresponding weight class (Figure 1), referring to the partition indicated by the World Health Organization [1] as shown in Table 1.

On the other hand, the discretization of the two variables *MentHlth* and *PhysHlth* was unsupervised and performed by partitioning the value range into six equal-width bins. This step was performed after the training-test partitioning to remain consistent even if the evaluation is carried out with a different test set. In order to do so we used the knime nodes *Auto-Binner* and *Auto-Binner(Apply)*.

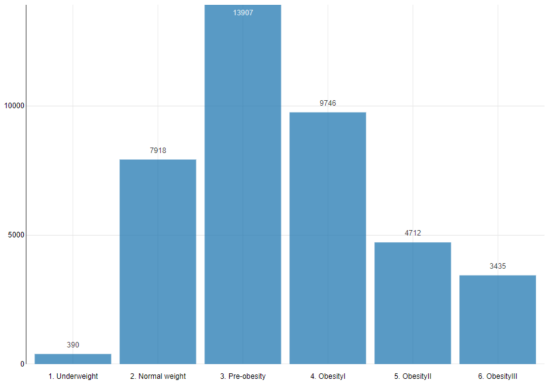

Figure 1: Distribution of BMI categories

BMI	Nutritional status
Below 18.5	Underweight
18.5-24.9	Normal weight
25.0-29.9	Pre-obesity
30.0-34.9	Obesity class 1
35.0-39.9	Obesity class 2
Above 40	Obesity class 3

Table 1: WHO Nutritional status

4. Models

The models implemented to approach the classification problem are various and can be divided into the following categories according with their functions:

- **Heuristic models**: these are models that are based on the search for an approximate solution, using a gradual approach of error correction rather than overly rigorous algorithms. They are therefore less demanding

in terms of assumptions and computationally. In our case we implemented the *Random Forest* classifier and two evolutions of this such as *Gradient Boosted* and *XGBoost*.

- **Regression models**: these are models based on statistical regressions. In particular we implemented *Logistic Regression*.
- **Separation models**: these are models aimed at spotting the mathematical functions that best separate the attributes space. From this set of classifiers we selected *Multi-Layer Perceptron* (MLP), a neural network that exploits backpropagation error to classify instances.
- **Probabilistic models**: they are models that rely on Bayes' Theorem. Out of these, we adopted the *Bayesian Network*.

4.1. Evaluation metrics

In our analysis the main metric used to evaluate the performance of the classifiers is *Log Loss*. This measure captures how close the prediction probability is to the corresponding actual/true value (0 or 1 in case of binary classification). The more the predicted probability diverges from the actual value, the higher is the log loss value [2].

$$LogLoss_i = -[y_i \cdot ln(p_i) + (1-y_i) \cdot ln(1-p_i)], \quad (1)$$

where i is the given observation, y is the actual value and p is the prediction probability. In order to evaluate a model and summarize its skill, log loss score of the classification model is reported as average of log losses of all the observations/predictions:

$$LogLoss = \frac{1}{N} \sum_{i=1}^{N} LogLoss_i. \quad (2)$$

In addition to log loss, other metrics have been calculated in order to compare the selected classifiers such as *Accuracy*, *Precision*, *Recall*, *Specificity*, F_1*-measure* and AUC.

- **Accuracy** indicates the fraction of correctly classified records:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

  where TP (True Positives) and TN (True Negatives) are the positive and negative records correctly classified while FP (False Positives) and FN (False Negatives) are the positive and negative records incorrectly classified.

- **Precision** indicates the fraction of positive values correctly predicted over the number of positive predictions:

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

- **Recall** indicates the fraction of positive values correctly predicted over the number of actual positive values (*True Positive Rate*):

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

- **Specificity** indicates the ability of the classifier to correctly predict negative values (*True Negative Rate*):

$$Specificity = \frac{TN}{TN + FP} \quad (6)$$

- **F_1-measure** is a composite indicator that provides an overall evaluation of the classifier worth. It is calculated as the harmonic mean between *Recall* (R) and *Precision* (P):

$$F_1 - measure = \frac{2 \cdot R \cdot P}{R + P} \quad (7)$$

- **AUC** indicates the area underneath the ROC curve (*Receiver Operating Characteristic*), a curve that highlights the ability of the classifier to make better (or worse) predictions compared to a random classifier. On the y axis we have Recall while on the x axis is the False Positive Rate. AUC is a synthetic indicator that referres to the curve and represents an accuracy measure that varies between 0 and 1, where 1 indicates a perfect classification.

As anticipated, despite the performance evaluation is conducted through different indicators, because of the nature of the phenomena, priority is given to *Log Loss* and *Accuracy*.

5. Classification and results

In this section, the results obtained by implementing different classification methods and approaches are presented and compared to identify the best model.

First, the dataset is partitioned using proportional stratified sampling, with respect to the class attribute *Diabetes*, according to the 80%-20% training/test set proportions. Then the two main approaches that we considered are a K-folds Cross Validation approach with the totality of attributes and a Feature Selection using *Wrapper* method.

5.1. K-folds Cross Validation

Cross validation has been implemented using Knime *X-partitioner* and *X-aggregator* nodes with $k = 10$ folds. This was done in order to have a clearer understanding of the model performances and to better compare the performances of all models. We tuned the parameters of MLP, Gradient Boosted and XGBoost as follows: the number of hidden layers in the MLP classifier has been set equal to 2 while Gradient Boosted and XGBoost parameter optimization was implemented through *Parameter Optimization Loop* nodes, using Brute Force as the parameter search strategy. It tests all possible combinations of parameters by varying them within a pre-established range and considers the configuration that returns the minimum value of the Log Loss target function. Regarding the Gradient Boosted method, we optimized the following parameters: *tree depth* (range 3 to 8, stepsize = 1), *number of models* (range 100 to 500, step size = 50) and *learning rate* (range 0.05 to 0.2, stepsize = 0.05). As a result the best configuration is tree depth = 3, number of models = 450 and learning rate = 0.05. On the other hand for the XGBoost method we optimized the *Boosting rounds* (range 100 to 1200, step size = 100) parameter which optimal value turned out to be 100. The classifier inside the parameter optimization loop was trained on a sub-partition of the training set (60% of the whole data set) and evalueted on a validation set (20% of the whole data set). It would have been optimal to implement KFCV inside the parameter optimization loops in order to obtain a more solid result but this procedure would have been too computationally demanding. Finally, for these two classifiers, cross validation was then performed using the optimized parameters. Table 2 reports the results in terms of Log Loss and the main metrics obtained evaluating the trained methods on the test set.

Classifier	Log Loss	Accuracy	Recall	Precision	Specificity	F_1 – measure	AUC
Gradient Boosted	0.509	0.75	0.789	0.739	0.71	0.763	0.827
Logistic	0.513	0.75	0.776	0.745	0.722	0.76	0.824
MLP	0.518	0.745	0.821	0.72	0.666	0.767	0.823
XGBoost	0.523	0.743	0.778	0.734	0.706	0.756	0.818
Bayes Net	0.659	0.741	0.733	0.753	0.749	0.743	0.816
Random Forest	0.902	0.743	0.759	0.743	0.726	0.751	0.806

Table 2: Results after training with KFCV

5.2. Feature selection

The chosen approach is *wrapper* type. For the classifiers provided by WEKA, feature selection is carried out through the *AttributeSelectedClassifier* node with the *WrapperSubsetEval* method and *BestFirst* selection strategy aimed at maximizing accuracy. *WrapperSubsetEval* performs KFCV with an adjustable number of folds that we set equal to 10 and a classifier of choice which we set to be the same as the one that we want to train. This guarantees that the attributes selected are the best fit for that specific classifier. For computational cost reasons we set the node of the MLP classifier to use the *ClassifierSubsetEval* instead of *WrapperSubsetEval*. This one doesn't perform KFCV but evaluates the attributes on a validation set.

As for the non-WEKA methods (i.e. Gradient Boosted and XGBoost), *Feature Selection Loop* nodes were used with *Backward Feature Elimination* selection strategy. It starts with having all features selected and in each iteration the feature that has on its removal the largest impact on the magnitude of the Log Loss function is removed. Although the most correct procedure would have been to use cross validation again, performing feature selection in this way would have been excessively computationally expensive. Therefore, feature selection was only carried out on a validation set (20% of the whole data set). At the end of the loop, the attribute configuration that minimizes the Log Loss is considered. What follows is the list of attributes that performed best for each classifier:

- **Gradient Boosted**: *all attributes*
- **Logistic**: *Age, Sex, HighChol, CholCheck, BMI, HeartDiseasorAttack, PhysActivity, Fruits, HvyAlcoholConsump, GenHlth, DiffWalk, Hypertension, Stroke.*
- **Random Forest**: *HighChol, CholCheck, BMI, HeartDiseaseorAttack, HvyAlcoholConsump, GenHlth, DiffWalk, Hypertension.*
- **MLP**: *Age, Sex, HighChol, CholCheck, BMI, HeartDiseasorAttack, PhysActivity, Fruits, HvyAlcoholConsump, GenHlth, Hypertension, Stroke.*
- **Bayes Net**: *Age, Sex, HighChol, CholCheck, BMI, HeartDiseasorAttack, PhysActivity, HvyAlcoholConsump, GenHlth, Hypertension, Stroke.*
- **XGBoost**: *Age, Sex, HighChol, CholCheck, BMI, HeartDiseasorAttack, Veggies, HvyAlcoholConsump, GenHlth, DiffWalk, Hypertension, Stroke, MentHlth, PhysHlth.*

Table 3 shows the results obtained on the test set.

Classifier	Log Loss	Accuracy	Recall	Precision	Specificity	F_1 – measure	AUC
Gradient Boosted	0.509	0.749	0.788	0.738	0.707	0.762	0.827
Logistic	0.514	0.748	0.774	0.743	0.721	0.758	0.824
XGBoost	0.522	0.742	0.776	0.735	0.707	0.755	0.821
MLP	0.528	0.746	0.793	0.732	0.697	0.761	0.82
Bayes Net	0.581	0.745	0.764	0.744	0.726	0.754	0.819
Random Forest	0.655	0.738	0.785	0.725	0.69	0.754	0.806

Table 3: Results after training with feature selection

5.3. Performance evaluation

To evaluate the results we used the metrics mentioned in the previous chapter. Firstly, with regard to precision and recall, the aim of each model is to find the best trade-off for our purpose since the more one increases the more the other decreases and vice versa. While the F_1-measure is higher the more precision and recall are high. In our case the goal is to decrease the Log Loss.

If we compare Table 2 and Table 3 we notice that the model with the best result in terms of recall is the MLP, but as could be expected it has a low precision as it rebalances the result. After the feature selection the results improve but the same pattern remains. In terms of precision, on the contrary, the best is the Bayesian Network which, as for the MLP model, rebalances this result with a higher recall. The F_1-measure therefore summarizes these two metrics in general and shows us how, taking into account only precision and recall, the method that optimizes them before the feature selection is precisely the MLP while after the feature selection became the Gradient Boosted. Just like precision and recall, specificity and recall are also inversely related: as recall increases, specificity tends to decrease, and vice versa. If we evaluate the results of the specificity measure, we notice how this decreases and therefore worsens for all models after making the feature selection. The models are less able to assess how many selected negative items are really negative. However, the best of these is always the Bayesian Network.

With regard to accuracy, the best models both before and after feature selection are Gradient Boosted and Logistic Regression. The first one slight improvement after the feature selection, a thousandth better than Logistic.

All these measures are summarised by the ROC curve, which is a graph that relates recall and specificity of a diagnostic test to the variation of the cut-off value. As we can see, with both approaches, the best results in terms of the considered metrics were achieved with the *Gradient Boosted* and *Logistic* models, which perform similarly as can also be observed from the combined ROC curve in Figure 2.

The result does not show any particular differences between the two models, as we expected given the values extrapolated from the table. Applying feature selection results in minimal improvements. Significant improvements are only observed for MLP and Bayesian Network methods, while the other methods show almost overlapping results, especially in terms of Log Loss. This is evidenced by the fact that the two best methods, Gradient Boosted and Logistic, perform best with the following configurations: Gradient Boosted achieves the best result (in terms of Log Loss) when trained on all attributes, while Logistic achieves a similar result to the approach without feature selection by only removing three attributes. Figure 3 explains how Log Loss improve as the number of variables considered increases. For this reason, we believe that there are no significant advantages in terms of performance improvement or dimensionality reduction.

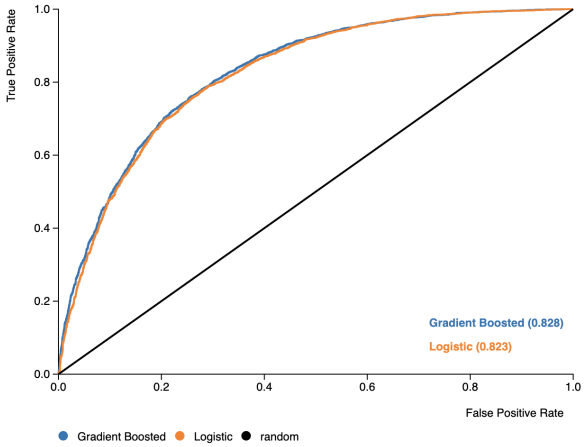

Figure 2: ROC curve comparison between Gradient Boosted and Logistic

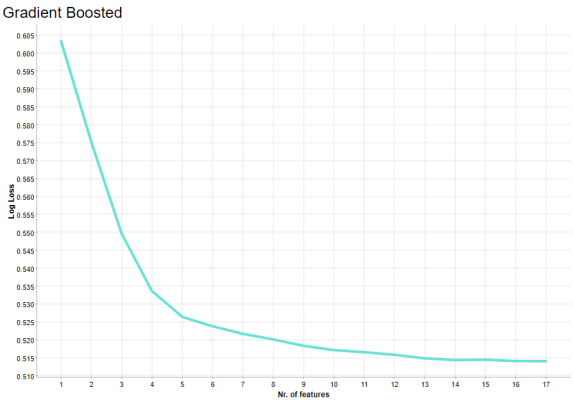

Figure 3: Log Loss VS Number of features (Gradient Boosted)

5.4. Feature importance

The aim of the work is not only to provide accurate predictions regarding the diagnosis of diabetes in a person. It is clear that the predictions provided by a classification model cannot replace expert opinion, which is why obtaining results that are not too accurate does not detract from the value of the work carried out. An equally important aspect that could contribute to the medical field in this case is the identification of the main risk factors, which thus lead to an increased likelihood of

contracting the disease. To this end, the weight that each variable has on the probability of having diabetes was determined, using the *Logistic Regression*. This method is based on the calculation and analysis of the *Odds Ratio*, obtained through the exponential of the coefficients estimated by the model. A value of the *Odds Ratio* between 0 and 1 indicates a negative association, i.e., a variable whose increase/change from baseline leads to a reduction in the probability of having diabetes, while a value greater than 1 indicates a positive association (increased risk) and thus has an opposite interpretation. It is important to note that all variables are significant at a significance level set at 5%. In particular, it is evident that the variables *GenHlth*, *Age* and *BMI* have a great weight in favour of the diagnosis of diabetes, as opposed to variables such as *Fruits*, *Veggies* and *PhysActivity* that reduce it. Comparing the results with the World Health Organization's statements on the main risk factors [3], a concordance of what was obtained and the experts' opinion emerges.

6. Conclusion

After implementing the methods and carefully comparing the results, we can draw the following conclusions. Clearly, the results obtained in terms of performance could be significantly improved. During the work, we encountered some structural limitations that provided insights for further improvements. We highlighted how limited computational capacity can have a negative impact, particularly with regards to parameter optimization and KFCV. Nonetheless, we believe that the compromise we reached by using a single validation set still allowed us to obtain a sufficiently robust result.

The other fundamental aspect we observed is related to the intrinsic nature of the available data. Since they concern behavioral risk factors, they naturally do not include some indications that are very relevant in diagnosing the disease. For example, it is well known how important the genetic component and blood glucose monitoring are [3], which are absent in this dataset. In order to improve predictive performance, it would therefore be useful to integrate the data under the supervision of experts in the field. However, we believe that the developed model can be considered a valid starting point for the development of a tool that facilitates the prevention and early diagnosis of diabetes.

References

[1] https://www.who.int/europe/news-room/fact-sheets/item/a-healthy-lifestyle---who-recommendations.

[2] https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a.

[3] https://www.who.int/news-room/fact-sheets/detail/diabetes.