KNIME Challenge

Diabetes classification and prediction

William Joseph Borruso, Luca Galli, Davide Ronchi

Università degli Studi Milano-Bicocca

February, 2023

Abstract

In this project, we used a dataset containing 17 variables about medical patients to build a classification model capable of predicting whether a patient is likely to have diabetes or not. The most fitting model resulted to be logistic regression made on a 80-20 partitioning through KNIME. The model showed an accuracy of ∼75% and a LogLoss of ∼0.5 while maintaining low computation times.

1. Introduction

Diabetes is a chronic medical condition in which the body is unable to produce or properly use insulin. Insulin is a hormone that regulates blood sugar levels. There are two main types of diabetes: type 1 diabetes and type 2 diabetes. In cases of type 1 diabetes, the body does not produce insulin, while in case of type 2 diabetes the body does not produce enough insulin or does not use insulin effectively. In both types of diabetes, high levels of glucose in the blood can lead to a range of serious health complications, including heart disease, stroke, kidney disease, eye problems, and nerve damage. Diabetes is a leading cause of death and disability worldwide, and its prevalence is increasing due to factors such as increasing rates of obesity and sedentary lifestyles.

1.1. Dataset

The dataset is provided by Kaggle, and consists in a cleaned version of the dataset provided by CDC's BRFSS 2015, which is a health-related annual telephone survey that collects responses from 400,000+ Americans about health-related risk behaviors, chronic health conditions, and the use of preventative services. The variables are either questions directly asked to participants, or calculated variables based on the responses of the partecipants.

2. Variables overview

The 17 variables made available by the dataset are the following:

- *Age*: 3-level age category:
  - Category "1": 18-24
  - Category "9": 60-64
  - Category "13": 80+
- *Sex*: 0 = female, 1 = male
- *HighChol*: 0 = no high cholesterol, 1 = high cholesterol
- *CholCheck*: 0 = no cholesterol check in 5 years, 1 = yes cholesterol check in 5 years
- *BMI*: Body Mass Index
- *Smoker*: Have you smoked at least 100 cigarettes (5 packs) in your entire life? 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 1 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*: 0 = no, 1 = yes

The target variable is called *Diabetes*: it assumes the value 1 if the patient is likely to have diabetes or the value 0 if the patient is unlikely to have diabetes.

3. Data preparation

In this first part, we analysed the dataset provided by Kaggle, observing its properties, its balancing and null values. Then, we proceeded by optimizing the data before proceeding with the classification.

3.1. Dataset overview and observations

The first main observation we made is that the Diabetes attribute, which is the class attribute, is very well-balanced, meaning that the amount of records assuming the value 1 and the records assuming the value 0 are present with about the same amount. This implies that random partitioning and random sampling will be about as effective as the stratified alternatives, since a random record selected from the dataset has ∼50% probability of having *Diabetes*=1 and ∼50% probability of having *Diabetes*=0.

The second main observation we made is that there are no null values, meaning that the data had already been cleaned before being published. Therefore it's not necessary to exclude incomplete records or manage null values.

Then, we made observations about the formats of the variables:

- Binary values are stored as integers, even though binary values are in fact categorical
- BMI is also represented as an integer, although it can assume decimal values
- The variable *Age* doesn't represent ages, but age categories, which are categorial variables and not integers (even if they are represented as numbers)

Finally, we computed a correlation matrix both with linear correlation and Spearman's rank correlation. Then we set a threshold of +0.5 and noticed that the attribute *PhysHlth* had a correlation coefficient that was higher than the threshold with respect to the attribute *GenHlth*. Therefore, we made sure that one of the two variables was excluded in the feature selection phase in order to avoid *multicollinearity*.

3.2. Pre-processing

3.2.1. Data formats

Binary data were converted from numerical to categorial variables to allow certain functions to use the frequencies of categories as inputs (frequencies of 1 and 0 categories in this case). The same conversion was done for the variable *Age*. BMI was finally converted from integer to double, to allow for decimal representations.

3.3. BMI binning

Numerical Binning is a way to group a number of continuous values into a smaller number of *bins*, to understand numerical data better, and can reduce noise and non-linearity. BMI, which is calculated from height and weight and can assume continuous values, has some standard ranges set from the *World Health Organization* to identify if the BMI of an individual falls into a healthy interval or not. We therefore used the WHO standard BMI ranges as bins for the BMI variable:

- Underweight: ≤ 18.5
- Normal: 18.5-25
- Overweight: 25-30
- Obese 1: 30-35
- Obese 2: 35-40
- Obese 3: ≥ 40

3.3.1. Feature selection

Feature selection is an important step to optimize the classification in terms of complexity, computation costs and performance, since it can be used to reduce overfitting, improve accuracy and avoid phenomena such as *multicollinearity*.

To perform feature selection while avoiding multicollinearity, we used a multivariate feature selection filter called *Correlation-based feature selection (CFS)*, which assesses the value of a subset of features based on both the individual predictive power of each feature and the degree of redundancy among them. Subsets of features that are highly correlated with the class attribute while having low intercorrelation are preferred.

The search procedure we selected in order to identify the optimal feature selection subset is called *Genetic Search*, that uses Goldberg's Simple Genetic Algorithm, with a population size of 20 and a number of generations of 20.

We therefore obtained an optimal subset of 9 features: *Age, HighChol, CholCheck, HeartDiseaseorAttack, HvyAlcoholConsump, GenHlth, DiffWalk, Hypertension, BMI*.

Then, we filtered these attributes in the preprocessing phase to evaluate the different models with the corrected training attributes.

3.3.2. Dataset partitioning

Initial holdout. To evaluate the performances of the selected models on the data, the dataset was initially split into two partitions:

1. Training set, containing 80% of the data: this partition will be used to train the models
2. Test set, containing 20% of the data: this partition will be used to test the generated predictors

We decided to use *stratified sampling* to form the partitions in the most balanced way possible, even though random sampling would probably still be effective, due to the strong balancing of the class attribute (*Diabetes* attribute).

K-fold cross validation. In the model selection phase only, after the initial holdout, the training set was also subjected to cross validation.

Cross validation involves dividing the original training data into k equally sized *"folds"*, where $k-1$ folds are used for training the model and the remaining fold is used for testing. This process is repeated k times, such that each fold gets a chance to be the testing set. The final performance score is the average of the individual evaluation scores for every k-th iteration.

This technique helps to mitigate the risk of overfitting, where the model performs well on the training data but poorly on new, unseen data, and limits the damage that strong outliers could cause to the predictor.

The value of k was selected by testing its most common values: 3, 5, 10. The final selected value for k was 10.

4. Classification

4.1. Models used

In our study we implemented various classification techniques, with the objective of choosing the optimal one for our problem. We worked with:

- Heuristic models: **J48** decision tree, implemented by Weka, which allows to classify even nominal data; **Random Forest**, a classifier made up of many decision trees, which can also handle categorical data;
- Regression models: **Logistic Regression**, a model in which the dependant variable is dichotomous;
- Probabilistic models: **Naïve Bayes** based on Bayes' theorem can also be used by combining numerical and categorical data
- Separation models: **Multilayer perceptrons**, a neural network;
- Supervised model: **k-nearest neighbor**, used to solve classification and regression.

Every model has been tested with different parameters and using different values of k in cross validation (k=3, k=5, k=10).

4.2. Performance measures

In our analysis, two criteria were mainly considered to evaluate the performance of our models: **Accuracy** and the value of the **LogLoss function**. However, we also observed Precision, Recall, F1 score and the AUC (Area Under the ROC Curve).

Accuracy indicates the percentage of positive and negative observations predicted correctly by the classification model and allows you to select the instance that guarantees the best performance on the records to predict. In particular:

Accuracy 4-1

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

- TP (True Positives) and TN (True Negatives) indicate the number of instances correctly classified as belonging respectively to the positive and negative class
- FP (False Positives) and FN (False Negatives) indicate the number of misclassified positive and negative instances

Log-loss is indicative of 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.

Log-loss 4-2

$$Logloss_i = -[y_i ln p_i + (1 - y_i) ln(1 - p_i)]$$
$$Logloss = \frac{1}{N} \sum_{i=1}^{N} logloss_i$$

Precision attempts to identify the proportion of positive identifications that were actually correct. It is defined as:

Precision 4-3

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

Recall attempts to identify the proportion of actual positives that were correctly identified. It is defined as:

Recall 4-4

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

Precision and Recall are the two building blocks of the F1 score. The goal of the F1 score is to combine the precision and recall metrics into a single metric. It is defined as:

F1 Score 4-5

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). AUC provides an aggregate measure of performance across all possible classification thresholds.

5. Results

5.1. Performance

The performances[1] of the selected models over 8022 records are represented in Table 1.

Classifier	Accuracy	Error	LogLoss	Correct	Wrong
ZeroR (*baseline*)	51.085%	48.915%	0.693	4098	3924
J48	73.298%	26.702%	0.556	5880	2142
Logistic Regression	74.857%	25.143%	0.513	6005	2017
Naïve Bayes	72.476%	27.524%	0.769	5814	2208
Random Forest	73.934%	26.066%	0.588	5931	2091
Multi-Layer Perceptron	74.857%	25.143%	0.528	6005	2017
k-Nearest Neighbors (*179*)	54.201%	45.799%	0.608	4348	3674

Table 1: Performance of the classifiers (see confusion matrixes at Appendix A)

Logistic Regression and *Multi-Layer Perceptron* showed the highest value of accuracy, which was 74.857% for both of them. However, *Logistic Regression* showed a lower LogLoss value.

Logistic Regression showed the highest AUC, with a value of *0.824*, as represented in Figure 1.

The *ZeroR* model was used as a baseline to compare the results of the selected models to a random model: models with an higher Accuracy and a lower LogLoss with respect to the ZeroR model are performing better than a random model.

The represented performance measures has to be considered inside of a confidence interval instead of fixed values. For instance, confidence intervals of the accuracies of Table 1 are represented in Table 2.

Model	Lower bound (Acc.)	Accuracy	Upper bound (Acc.)
J48	0.725	0.733	0.741
Logistic Regression	0.741	0.749	0.756
Naïve Bayes	0.716	0.725	0.733
Random Forest	0.731	0.739	0.747
Multi-Layer Perceptron	0.741	0.749	0.756
k-Nearest Neighbors	0.533	0.542	0.551

Table 2: Confidence intervals for Table 1 accuracies

The confidence intervals of the accuracies are computed using the following formulas:

Accuracy confidence interval CL 90% 5-1

$$\text{Lower bound} = \frac{acc + \frac{z^2}{2N} - z\sqrt{\frac{acc}{N} - \frac{acc^2}{N} + \frac{z^2}{4*N^2}}}{1 + \frac{z^2}{N}}$$

$$\text{Upper bound} = \frac{acc + \frac{z^2}{2N} + z\sqrt{\frac{acc}{N} - \frac{acc^2}{N} + \frac{z^2}{4*N^2}}}{1 + \frac{z^2}{N}}$$

5.2. Classifiers comparison

To compare the accuracies of the classifiers and choose the best one, we tried to identify statistical significant differences between models' accuracies.

In this comparison phase, the input records of the classifiers were partitioned iteratively with a *10-fold cross validation*, in order to obtain less biased accuracies and error rates.

Starting from the assumption that, for a sufficiently large k in cross validation, the difference d between the error rates of two models follows a normal distribution, we used the *Student T Distribution* to compute the *confidence interval* for the value of the true mean of d with a Confidence Level of 90% ($\alpha = 10\%$).

[1] Cross validation was not used for this evaluation phase, which represents one instance of every classifier. Therefore, these results might not be replicable in further applications of the classification models.

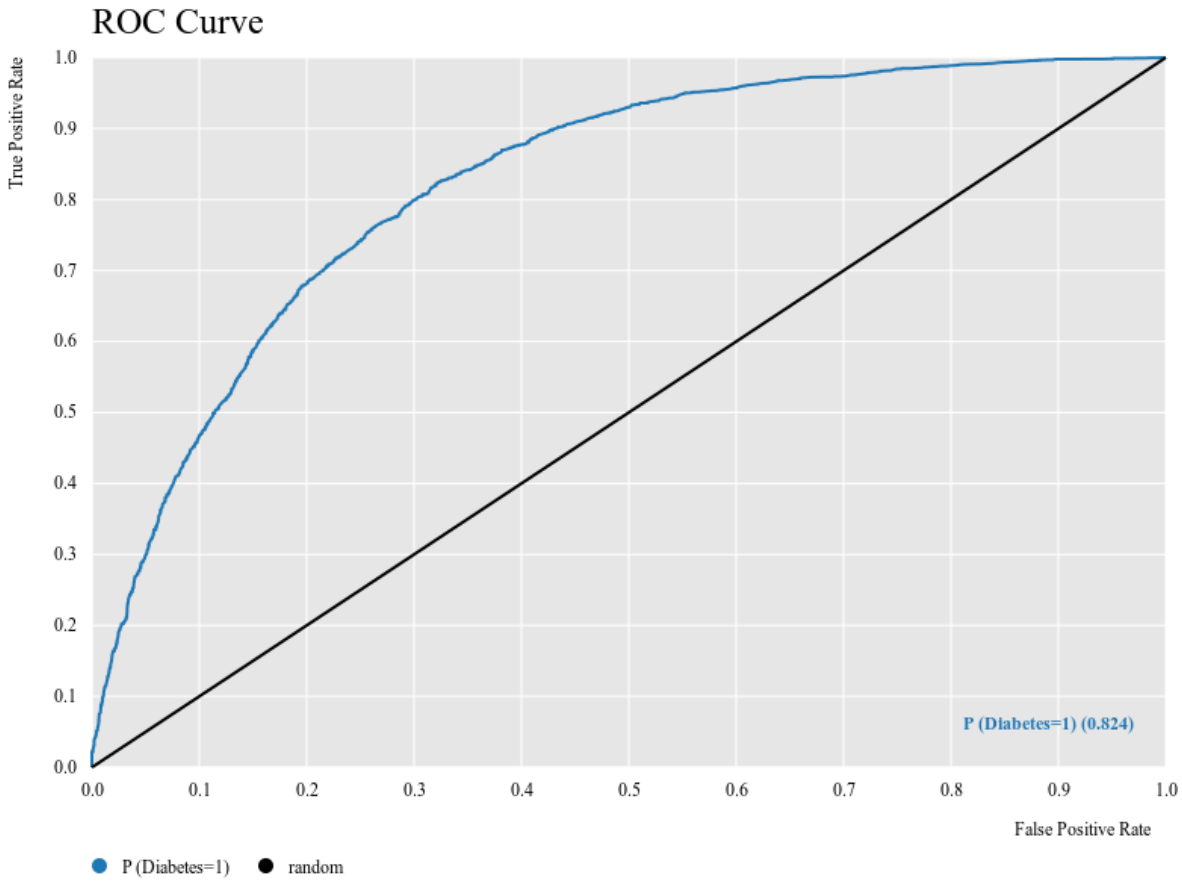

Figure 1: ROC curve for Logistic Regression (*blue*) compared to a Random Model (*black*)

t-student true mean interval CL 90% 5-2

$$(\bar{d} - t_{k-1,1-\frac{\alpha}{2}} \cdot \widehat{\sigma}_{\mathrm{d}^{\mathrm{cv}}}, \bar{d} + t_{k-1,1-\frac{\alpha}{2}} \cdot \widehat{\sigma}_{\mathrm{d}^{\mathrm{cv}}})$$

- $\bar{t}$ is the difference between the error rates of the classifiers M1 and M2
- $t_{n-1,\frac{\alpha}{2}}$ is the critical value of the t-student distribution with $k-1$ degrees of freedom and a confidence level of $1-\alpha$
- $\widehat{\sigma}_{\mathrm{d}^{\mathrm{cv}}}$ is the estimated value of the standard deviation of the accuracy

We performed a one-by-one comparison between all the selected models by computing the confidence interval of d for every couple, interpreted as follows: if a confidence interval calculated on the difference between error rates of the models M1 and M2 is completely negative, we can say that statistically, M1 performs significantly better than M2, while if the interval is completely positive, M2 performs better than M1. If the interval spans the value 0, the performance difference between the models is not statistically significant.

The best classification model for this dataset turned out to be a *Logistic Regression*, winning all the comparisons as represented in Figure 2. With cross validation, *Logistic Regression* showed the highest value of Accuracy and the lowest LogLoss, while maintaining low computation times.

6. Conclusion

In conclusion, our project aimed to identify the most accurate model for predicting the likelihood of diabetes in medical patients. To achieve this goal, we compared several different models, using various feature selection filters and search methods, and iterating the classification using k-fold cross validation with different values of k.

After thorough experimentation and analysis, our results showed that the Logistic Regression model with the CFS feature selection filter and Genetic Search method outperformed all other models in terms of accuracy and predictive power. This model achieved an accuracy rate of 74.857% and a LogLoss of 0.513. These can be considered as good results compared to the random model ZeroR baseline performances, which shows 51.085% accuracy and 0.693 LogLoss.

We also identified several key risk factors that were found to be strongly associated with diabetes, including age, cholesterol, heart problems, high alcohol consumption, BMI, hypertension and difficulties in walking and climbing stairs. These results can be used to inform early detection and prevention strategies for diabetes.

Appendix A. Confusion Matrixes

Observed\Predicted	1	0
1	4098	0
0	3924	0

Table A.3: ZeroR (*baseline*) Confusion Matrix

Observed\Predicted	1	0
1	2798	1126
0	1016	3082

Table A.4: J48 Confusion Matrix

Observed\Predicted	1	0
1	2846	1078
0	939	3159

Table A.5: Logistic Regression Confusion Matrix

Observed\Predicted	1	0
1	3000	924
0	1284	2814

Table A.6: Naïve Bayes Confusion Matrix

Observed\Predicted	1	0
1	2829	1095
0	996	3102

Table A.7: Random Forest Confusion Matrix

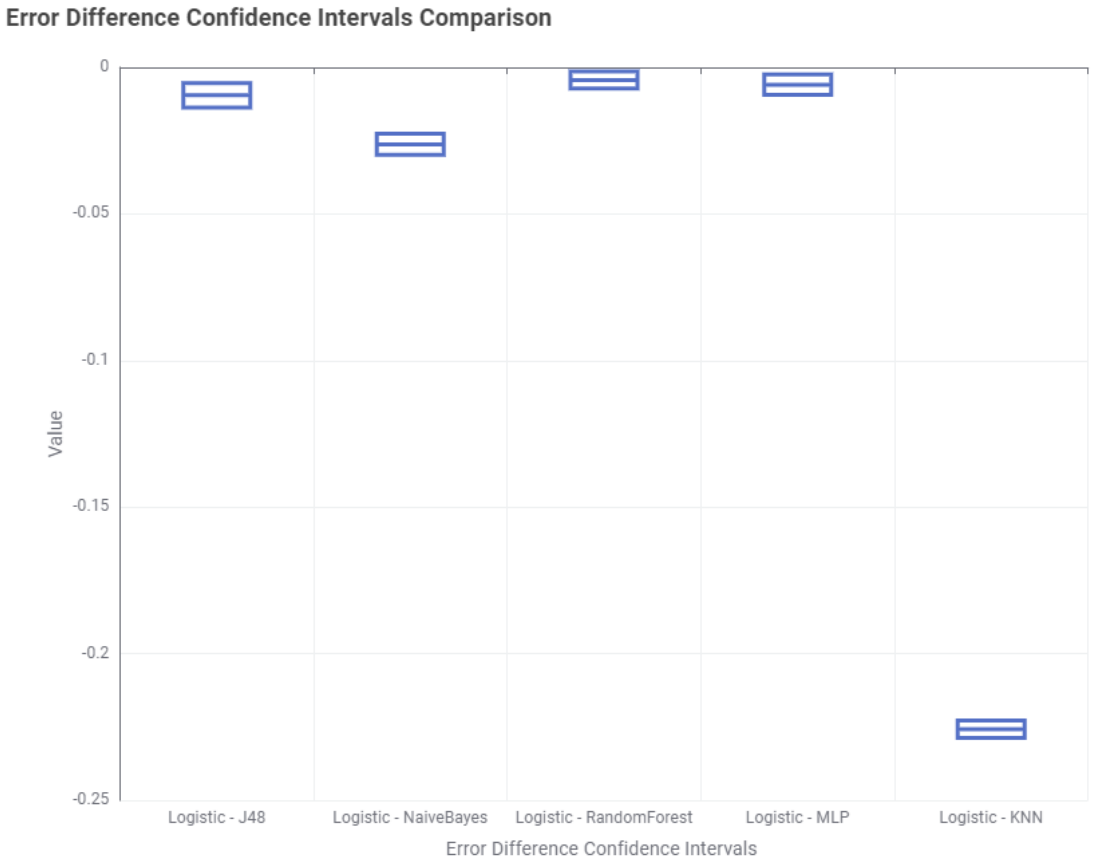

Figure 2: Confidence intervals of the differences between error rates of all models one-by-one

Observed\Predicted	1	0
1	2616	1308
0	709	3389

Table A.8: Multi-Layer Perceptron Confusion Matrix

Observed\Predicted	1	0
1	391	3533
0	141	3957

Table A.9: k-Nearest Neighbors (*179*) Confusion Matrix