DIABETES PREDICTION COMPETITION
Classification - Diabetes or not?

Team_09: Federico Ferretti, Ilaria Salvatori, Marta Privitera

Abstract

Diabetes is a chronic health condition that affects how your body turns food into energy. Body's tissues and organs can be seriously damaged and some complications can be life-threatening over time. What are the causes of Diabetes? Is it possible to predict with classification models whether a person is likely or unlikely to have Diabetes? The answers to these questions are at the center of the "Diabetes Prediction Competition" launched by the Kaggle Community Competitions hosted by ML GDEs or TFUGs, sponsored by Google Developers. Per doctors, the most common forms of diabetes are: being overweight, obesity, physical inactivity, insuline resistance, genes and family history, genetic mutations and hormonal diseases. The aim of the present work is the realization of a model of classification capable of correctly cataloguing whether a person suffers of Diabetes or not on the basis of a few physical quantities in the dataset provided.

Contents

1 Introduction

Diabetes is a chronic, metabolic disease characterized by elevated levels of blood glucose. The chronic disease starts from the process of digestion: when you eat carbohydrates, your body breaks this down into glucose. Once glucose is in the bloodstream, it needs help (a “key") to get into its final destination where it is used for providing energy to the body. The consequence is that glucose stays into the bloodstream and its level rises. Over time, this condition leads to serious damage to the heart, blood vessels, eyes, kidneys and nerves. Moreover, chronicle high level of sugar remaining in the bloodstream results in mental health issues like dementia or depression.
Even if there is no cure for Diabetes some specific strategies may be applied to control the effects of this disease on many patients: following a Mediterranean diet, exercising regularly, achieving a healthy weight, taking medication according to recommendations and monitoring the blood glucose and pressure levels. Predictive models make early diagnosis possible and can lead to effective lifestyle changes which are important to prevent the development of the illness.

About 422 million people worldwide have diabetes, the majority living in low-and middle-income countries, and 1.5 million deaths are directly attributed to diabetes each year. Both the number of cases and the prevalence of diabetes have been steadily increasing over the past few decades. So, it is clear that diabetes is a widespread and dangerous disease and for this reason, there is a globally agreed target to halt the rise in diabetes and obesity by 2025.

The primary objective of this work is to construct the best classification model capable of predicting whether a person suffers from Diabetes or not.

2 Dataset

2.1 Description

The dataset is composed of 17 feature variables and one target variable. Moreover, it has been synthetically cured from the original data source using CTGAN.

- *Age*: 3-level age category (_AGEG5YR see codebook). 1 = 18-24, 9 = 60-64, 13 = 80 or older;

- *Sex*: 0 = female, 1 = male;

- *HighCol*: 0 = no high cholesterol, 1 = high cholesterol;

- *ColCheck*: 0 = no cholesterol check in 5 years, 1 = yes cholesterol check in 5 years;

- *BMI*: Body Mass Index;

- *Smoker*: Have you smoked al 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 fruits one or more times per day. 0 = no, 1 = yes;

- *Veggies*: Consume vegetables one 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 scale 1-30 days;

- *PhysHlth*: Physical Physical illness or injury days in past 30 days scale 1-30;

- *DiffWalk*: Do you have serious difficultly walking or climbing stairs? 0 = no, 1 = yes;

- *Hypertension*: 0 = no hypertension, 1 = hypertension;

- *Stroke*: 0 = no, 1 = yes;

- *Diabetes*: 0 = no diabetes, 1 = diabetes (Target variable);

3 Performance measure

The evaluation metric for this competition is the LogLoss function: the lower the LogLoss value is, the better the model performance is. The Logloss function is:

$$H(p,q) = -\sum_i p_i \log q_i = -y \log \hat{y} - (1-y)\log(1-\hat{y})$$

where y_n is the vector which represents the true class lable and $\hat{y}_n$ is the predicted probability of the n^{th} class. This function, also known as the cross-entropy loss, measures the difference between the predicted probability distribution of the model and the true probability distribution of the target class.
Even if the Challenge requested to use Logloss, we considered analyzing the performance of our trained models in more depth, using additional measurements: Accuracy, Recall, Specificity and Precision; in particular, the first three show results concerning the single class, instead the Accuracy is referred to the overall model. Accuracy indicates the percentage of observations correctly predicted positive and negative:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}; \tag{1}$$

Where TP and TN indicate the number of instances correctly classified as belonging respectively to the positive (Diabetes=1) and negative (Diabetes=0) class; FP and FN indicate the number of positive and negative instances wrongly classified. In general, models with higher accuracy are rated as better. However, accuracy is not sufficient and for this reason other criteria are used to obtain a more complete evaluation.

Recall represents the portion of positive records correctly classified by the model.

$$Recall = \frac{TP}{TP+FN}; \tag{2}$$

A high recall value indicates that few records positive were classified incorrectly.

Precision describes the fraction of records that are actually positive among all those predicted as such.

$$Precision = \frac{TP}{TP+FP}; \tag{3}$$

A high precision value determines a lower number of false positives values.

Specificity indicate the portion of negative records correctly identified by the model.

$$Specificity = \frac{TN}{TN+FP}; \tag{4}$$

A further method used to evaluate the classification model is the ROC curve, which shows on the ordinate axis the percentages of True Positives and on the ascisse axis the percentage of false positives. We will see the graphic representation of the roc curve on our optimal model in the final section “Results"

4 Knime Workflow

With the aim of finding the best classification model for the diabetes variable, we built a workflow that would compare different performance measures. By comparing different preprocessing strategies, different classification techniques and performance evaluations, we were able to determine the best classification model associated to the lowest Logloss.

4.1 Preprocessing

After importing the dataset, an exploratory analysis was carried out to investigate the type of the variables and the presence of missing values. It emerged that the dataset does not contain missing values and the majority of variables is binary. Moreover, the class attribute is balanced so there will not be problems related to the ’zero rule’ (see fig. 1)

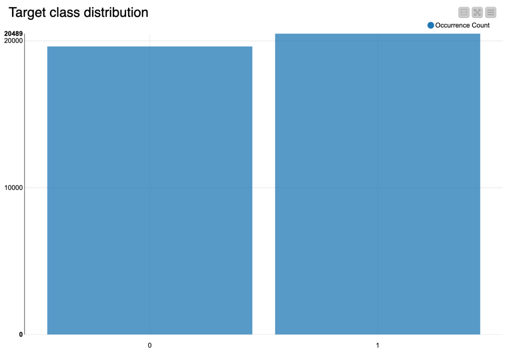


Figure 1

Our preprocessing includes four techniques:

1. **Stratified Sampling**: since the histograms showed that classes belonging to individual attributes are distributed differently, it was decided to adopt a layered approach to maintain proportions.

2. **Discretization**: we focused on the numerical attributes. With this technique we want to overcome the problem related to the presence of outliers. Two different discretization approaches were applied:
   - *MentHlt* and *PhysHlt* have been discretized according to their distributions: in particular, where the variance is the highest, we put the split point. In this way we got two intervals for both *hysHlt* and *MentHlt*. Concerning BMI, it has been discretized according to the Italian Ministery of Health division; in addition to this, each of the seven ranges has been associated with a number from one to seven. From this point onwards, we will refer to this type of discretization as “First Discretization".
   - Here, the split points for *MentHlt* and *PhysHlt* partitions are placed with respect to the percentiles. Concerning the *BMI*, we divided it in the same way as described above.

3. **Binarization**: it tooks *PhysHlt*, *MentHlt* and *BMI* from the First Discretization. The two intervals for the values of the first two varibles are associated to the values 0 and 1. Moreover, seven columns of binary attributes have been created to represent each of the seven intervals of the *BMI* variable.

4. **Correlation**: the correlation between the variables has been computed and the attributes have been filtered according to the value of their correlations to the others; in particular, if the correlation was major than 0.5, the attributes were discarded for the future consideration, otherwise were maintained.

These preprocessing methods have been adopted and tested with different classification models in order to have a large choice of results among which we selected the best performing model. The process which gave the lowest LogLoss has been finally selected as the best one. So, at the end of the process, considering the 4 best LogLoss obtained, we can say that the two types of preprocessing found the best are the Correlation and the First Discretization.
From now on the 4 types of preprocessing will be quoted with the corresponding description numbers (e.g. stratified sampling =1)

5 Modelling methodologies

5.1 Feature Selection

The advantages of feature selection are improved model performance, reduced model complexity, faster training times, and increased interpretability of the model.

5.1.1 Filter Method

For this process we use the KNIME node "Attribute Selected Classifier" which combines Feature selection and classification into a single step. It selects a subset of the most relevant features using a filter method, and then it applies a classification algorithm on the selected features.

5.1.2 Backward Feature Selection

At the beginning is trained a model using all the initial variables, then the attributes with the most irrelevant impact of the model are removed. This process is iterated until the subset of variables that give the best performance is identified.

5.1.3 Forward Feature Selection

This process begins with an empty set of variables and iterative adds the variables that have the highest impact on the model’s performance. The limitations of these methods are that the best subset of variables can not be identified with certainty.

In our case, the correlation matrix in fig. 2 does not show strong correlation between the variables; this suggests that these are independent variables.

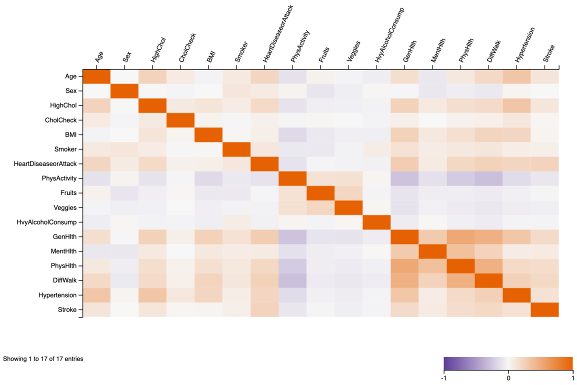


Figure 2: Heatmap shows the correlation between the dataset variables.

However, to confirm our initial hypothesis, a targeted feature selection analysis was conducted. Firstly, we used a Filter node in which we have selected 'GainRatio' as objective function. We chose the most refined metrics compared to other functions such as the 'InformativeGain' (not suitable for highly diversified classes). As 'Search' method, we selected the Ranker, in order to speed up the procedure of the Filter. At the end of the process, the following emerged: as we can see in the fig. 3 when the number of attributes increases, the performance improves, according to the independence observed by the correlation matrix.

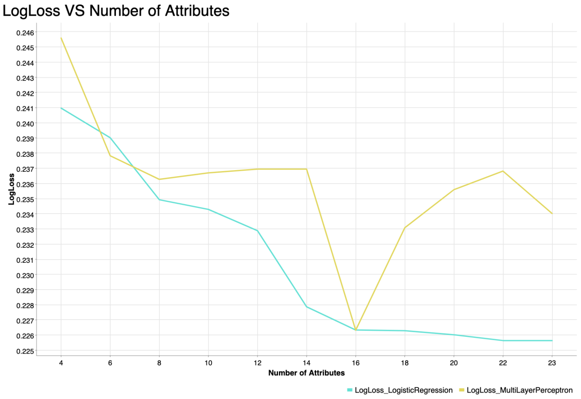


Figure 3: LogLoss Vs Number of Attributes with Filter approach

We report only the results related to the models with the best performance with the Univariate Filter technique. The previous procedure does not aim to determine redundant but irrelevant attributes because is Univariate. The assessment of a possible reduction in the number of redundant attributes has been carried out through the use of the Wrapper approach, which includes a series of techniques using a classification model to select which are the redundant attributes among all those available. In particular, the technique implemented by the KNIME *Forward Feature Elimination* node was chosen, which, by means of a loop and a classification model trained and validated on the training sets subsequently used (K training sets), provides a performance estimate with a reduced set of variables. The Backward technique has also been tested but has not produced more significant results than the Forward in terms of performance. In such a scenario, the choice was made to maintain the set of attributes that minimizes LogLoss. The results do not recommend reducing the number of attributes.

5.2 Models used

In this work, different classification techniques were applied with the aim of identifying the most suitable technique on the basis of the available data:

1. Probability models: among the most used methods are those based on probabilistic hypotheses such as the **Naive Bayes** classifier which allows, starting from the high quantity of data, to obtain accurate results under hypothesis of independence between attributes.

2. Heuristic models: although they do not guarantee optimal results, they provide approximate and reasonable solutions. These methods refer to decision trees, that are developed from a subset of initial data (training set) for which the output class is known. Classification is based on the majority class within the final node. These models focused on the **Random Forest** classifier, the **J48 Regression Tree**, and the **Naive Bayes Tree**.

3. Regression models: based on **Logistic Regression**, their advantage is the possibility of considering any type of input, resulting in extremely flexible. After identifying the response, Diabetes and input attributes, the method intends to measure the effect of each input on the output and the response will be identified about the value assumed by the inputs that influence it the most.

4. Artificial Neural Network: they are based on the collection of units or nodes connected together and called artificial neurons, which once received and processed the signal send it to the neurons connected to them. Among the classifiers belonging to this category of algorithms, we can find the **Multilayer Perceptron**, which uses the Backward Propagation Error technique to classify instances.

We tested eight classifiers in order to find the one that best predicted the phenomenon and therefore had a lower LogLoss. In particular, we have tested: Naive Bayes, Naive Bayes Learner, Decision Tree, Random Forest, NB Tree, J48, MLP, Logistic Regression. Finally, we can anticipate that the best among these turned out to be the Logistic Regression in each case, as we will demonstrate later.

6 Result Benchmark

6.1 Performance Evaluation using Holdout

In this first phase, the Holdout method was used (at random seed fixed), which is based on the partition of the dataset into two subsets using a stratified sampling procedure in which the stratification variable is Diabetes. In this way, the training set (67% of instances) and the test set (33% of instances) were obtained. The trainers created using the 8 models were trained with the training set and validated through the test set. As you can see from the table 1 the best classifiers in terms of LogLoss are the Logistic Regression and the NBTree. We show only the results related to the most performing preprocessing techniques (the First Discretization and the filter of the raw dataset with Correlation). Both techniques were applied to the dataset after stratified sampling. The minimum value is 0.223.

Model	Preprocessing	LogLoss
Log.Regression	2,3	0.223
Log.Regression	4	0.225
NBTree	4	0.241
J48	2,3	0.241
J48	4	0.241

Table 1: LogLoss comparison for different classification models using Holdout method

Model	FS	Preproc.	LogLoss
Log.Reg.	FW	4	0.225
Log.Reg.	BW	4	0.226
Log.Reg.	FW	2,3,4	0.226
Log.Reg.	BW	4	0.227
MLPerceptron	FW	4	0.231

Table 3: Logloss comparison for different classification models using CrossValidation and Feature Selection

6.2 Performance Evaluation using CrossValidation

In order to compare the performance of different classifiers and define their statistical relevance, a cross-validation approach is used. The original dataset is then divided into K subsets of equal nonintersecting extent. Each model is then trained using K-1 of these subsets and tested on the remaining one. The procedure is iterated K times until the end of the possible combinations. For each metric, this approach generates K different performance values, one for each process, and its average provides a more accurate estimate of the real performance of the classifier compared to what would be with only one partitioning (Holdout). In order to avoid overfitting, we chose a number of K not exceeding 10: by partitioning with a K number greater than 5 and testing we noticed that the optimal value for LogLoss was reached at 5. The results in table 2 show that the lowest value of LogLoss is given by 0.224. However, even if the LogLoss value is higher than the one observed through the Holdout procedure, we consider it more reliable because it is calculated as an average of the values obtained in the K iterations of cross validation. By keeping track of the performance of each individual iteration, it is possible to calculate the average of the performance metric being studied.

Model	Preprocessing	LogLoss
Log.Regression	2,3,4	0.224
Log.Regression	4	0.225
MultiLayerPercep.	2,3,4	0.233
MultiLayerPercep.	4	0.233
J48	4	0.24

Table 2: LogLoss comparison for different classification models using CrossValidation method

6.3 Performance Evaluation using Feature Selection and Cross Validation

After obtaining subset attributes from the two methods, Filter and Wrapper, we trained and tested the classifiers on these new subsets using Cross Validation. We then compared the results in terms of performance. As you can see from the table, the most performing classifier is once again the Logistic Regression. We observe that the best results are related to Correlation preprocessing in which one attribute has been discarded: *PhysHlth*. In fact this latter is the only one which has a correlation value higher than 0.5 with *GenHlth* attribute. As we expect, the Backward and Forward techniques gave the best values for the LogLoss. This is probably due to the fact that a classifier was used for selecting attributes instead of a target function as in the Filter method. After testing several classifiers among the best models found, we chose the Logistic Regression for the wrapper selection. Moreover, we used backward and forward feature elimination as methodology. Once the best possible accuracy has been reached, the cycle is interrupted and the columns relevant for the prediction have been selected using a column filter.

7 Results and Conclusion

7.1 The Best Workflow

From the table4 we can decide which model and which technique has turned out more performing than the others. This table has been made up by putting together the results of the previous tables.

Model	technique	Preproc.	LogLoss
Log.Reg	Holdout	1,2,3,4	0.223
Log.Reg.	CV	2,3	0.224
Log.Reg.	Holdout	1,4	0.225
Log.Reg.	CV	4	0.225

Table 4: LogLoss final comparison

Our considerations for the choice of the best predictive model are the following:

- Considering the LogLoss values, as we see through the tables above, the lowest are the ones related to the holdout with discretization and stratified sampling and to the cross validation with discretization (0.223 and 0.224). Then we have the values obtained with the Holdout classification (0.225) with correlation and stratified sampling techniques and then there is the one obtained with the cross validation together with correlation (0.225). Actually, even the Forward technique returns a logloss value of 0.225 but it was chosen to keep among the best four the procedure associated with the Correlation technique because from the point of view of accuracy brings better results.

The choice fell on the Cross Validation technique rather than on the Holdout, although the LogLoss results from the first method is slightly higher. The explanation of this action lies in the greater reliability that the Cross Validation method presents, which for this reason entails a lower risk of overfitting than the Holdout method. This is possible due to the very nature of this method, which allows to have a more accurate estimate of the performance thanks to the K iterations executable.

- The flow considered the best one, used the application of First Discretization as pre preprocessing technique; subsequently, as a performance evaluation technique, Cross Validation was adopted with the use of wrappers. Moreover, as already anticipated, the applied optimal classifier has been the Logistic Regression. This is a very efficient classificator and in particular, is very good for binary classification problem, like the one in our case.
- The preprocessing part influence the value of the Logloss obtained.In support of our decision we can see from Tables 1, 2 and 3 the fourth type of preprocessing is quite always associated with the lowest values of the Logloss. For this reason we select the model with this type of preprocessing.

7.1.1 KNIME training workflow

In our KNIME training workflow we insert all the nodes required to build our best predictive model.The best model we select starts with the fourth type of preprocessing in which numeric columns are transformed into bins columns and then into binary column. The correlation filter removes the *PhysHlth*column because it was high correlated (> 0,5) with other variables.
Then we create the Cross Validation metanode in which we find the partitioner node, the classificator (Logistic Regression) node, the weka predictor node and the aggregator node. In this metanode the output of the predictor is connected also with the "accuracy" metanode, in which is calculated the value of the accuracy of the model and other performance measures
The output of the "Cross Validation" metanode is the input of the "LogLoss" metanode in which through the "Math formula" and other nodes, the performance Evaluation of the model is calculated and shown in a table. As we know, this value is equal to 0.223.

7.1.2 KNIME deployment workflow

The KNIME deployment workflow contains two node for two input dataset, the training and the test set. For this reason the classification method we have inserted in the deployment workflow contains, instead of the metanode of the "Cross Validation", the Holdout method. In fact, in this context a random partition to obtain the test set by CrossValidation does not meet the goal of using a specific test set to evaluate performance, which is possible using the Holdout method.
The training dataset goes through the preprocessing, it passes through the "Holdout" metanode and then it ends with the "LogLoss" metanode.
The test set passes through another "Preprocessing" metanode that is the same as the one used for the training dataset. Then it enters in the second input port of the "Holdout metanode" and it is connected to the "Weka Predictor" node, in this way the prediction are made on the test set.
The final step is the "Logloss" metanode whose output is the Logloss value obtained training the model on the test set.

7.2 Beyond the LogLoss

As we can see in the table 5 and the table 6, observing the values of the performance measurements calculated on the prediction Diabetes=1, we can say that the results are satisfactory. Starting from the summary observations of the numbers associated with the values of the confusion matrix, we can see how the false positives and the false negatives, i.e the missclassified records, are much lower than the number of elements correctly classified.

TruePositives	15858
FalsePostives	5503
TrueNegatives	14116
FalseNegatives	4631

Table 5: Values from Confusion Matrix

Moreover, looking specifically at some performance measures, we have evidence of what has just been said.

Recall	0.774
Precision	0.742
Accuracy	0.747

Table 6: Additional Performance Measures

Noting the value of Recall, for example, it is evident that 77% of positive records have been correctly ranked; Precision also shows that the portion of positive records among those predicted as such is 74%. Finally, we can see that the level of Accuracy also has a rather high value for prediction: it turns out therefore that 75% of the positive and negative records have been correctly predicted.

7.2.1 ROC curve

As mentioned above, the ROC curve can be considered an additional method of evaluating performance of the classifiers. The interesting peculiarity of this graphic technique lies in not taking into account the distribution of the response variable. Below, we illustrate the graph inherent in the model considered the best: Logistic Regression.

Looking at the representation (see 4), we can say that the model is good because the area underlying the curve is greater than the area underlying the line that represents the model "Zero Rule". In addition, as a proof of the model's goodness, we note that at 50% of True Positive, the Logistic model returns about 11% of False positive.

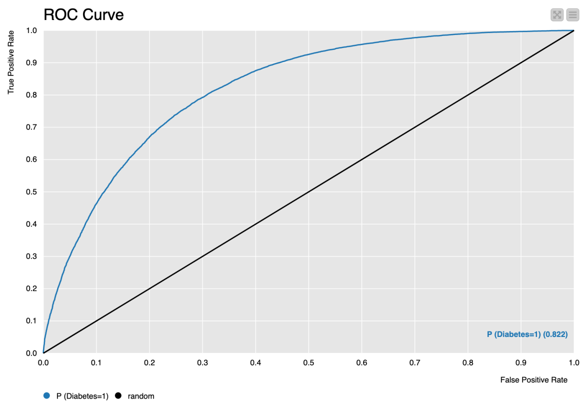


Figure 4: ROC curve of the best predictive model

8 Realization of the Data App

We created one dashboard for Univariate & Multivariate Analysis and for the Performance Evaluation Analysis. We added filters to let users interact with our Data Visualization.

8.1 Univariate & Multivariate Analysis

The first visualization is about the target class variable of fig. 5. We made a comparison between the two different distributions of the target class: the first one shows the total number of records corresponding to the diabetes 0 and diabetes 1 before training the model, the second one shows the total records corresponding to the class Diabetes after training procedure. It is possible to choose to visualize the records when the Diabetes values is equal to 0 or 1 or both.

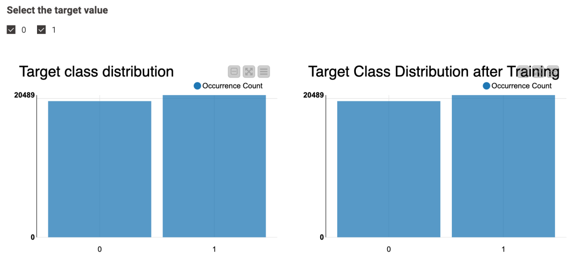


Figure 5: Diabetes occurences count

Below, in fig.6 we can see the barchart relating to the average distributions of the binary variables associated with the presence or not of diabetes in the subjects; the representation refers on the one hand to the observations in the original dataset, the second shows the same data observed on the basis of the actual prediction. The means of the variables are lower when Diabetes value is equal to 0 and higher when it is equal to 1. This can indicate that these variables have an influence on diabetes.

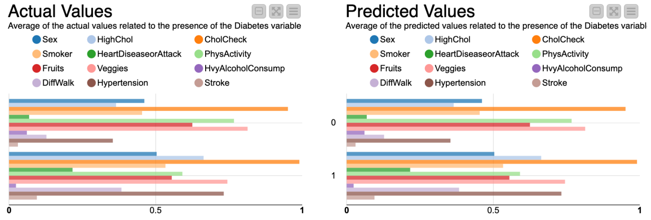


Figure 6: Average distribution of binary variables

The third visualization is related to the value of the correlation for each couple of variables. Colours close to red indicate a high positive correlation, while colours close to blue indicate a low negative correlation in fig. 2.

Then we used two density plot for the variables *BMI* and *Age* and two histograms for the other two numeric variables. Below we can see two examples of both of them (fig. 7, 8).

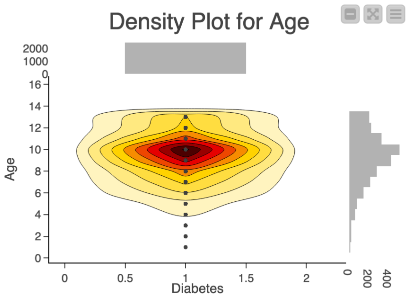


Figure 7: 2D Density Plot example of the DataApp

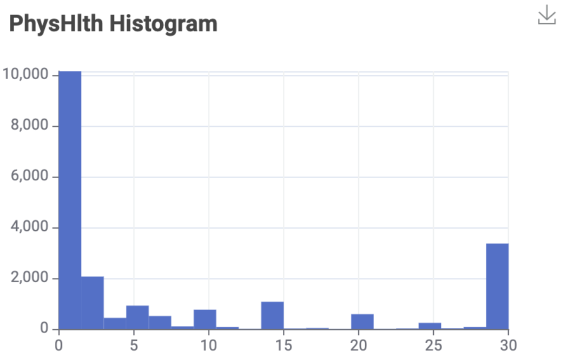


Figure 8: Histogram example of the DataApp.

The choice of histograms is due to the fact that it was not easy to spot the distribution of these two variables using the 2D representation density plot. We visualize also the four boxplots for all these variables. Below we can see an example of a boxplot is shown in fig. 9.

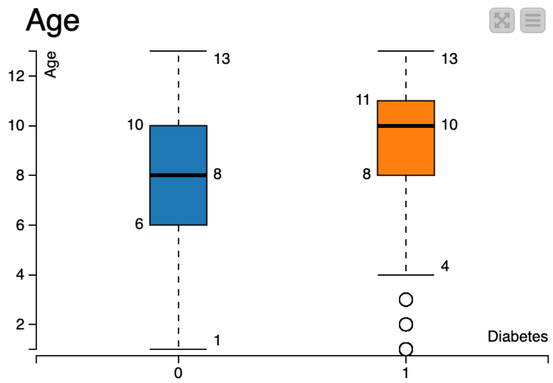


Figure 9: Boxplot example of the DataApp

8.2 Performance Measures Analysis

The first graph of this section represents the Cumulative Gain Chart, thanks to which we can see the performance achieved by the classification model adopted with respect to the No model (see fig.10).

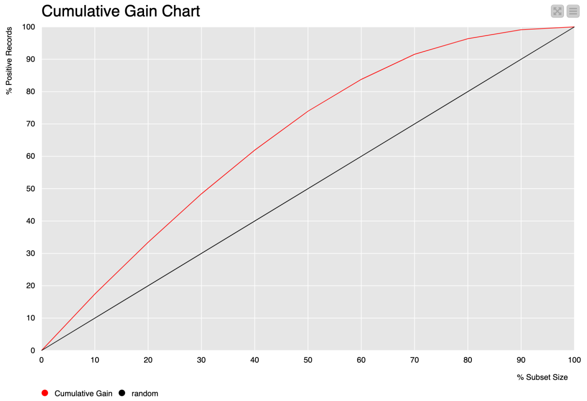


Figure 10: Cumulative Gain Chart

Observing the representation, it is evident that increasing the subset size, the classifier losses efficiency: graphically, moving to the right, the probability that an individual is affected by Diabetes is lower.

The Lift Chart in fig. 11 shows in blue the ratio between the percentage of the true positive of our model and a random prediction of the true positive with respect to the percentage of the population. The green line refers to a prediction based on the random model.

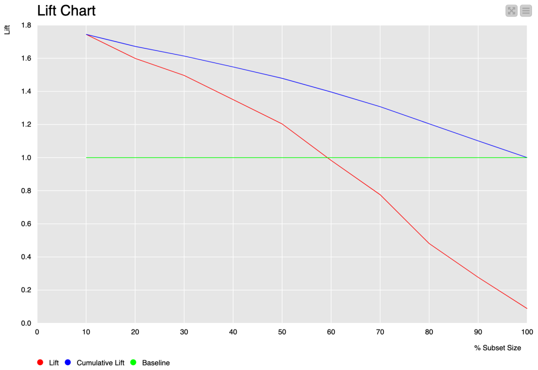


Figure 11: Lift Chart

Then we show the ROC curve that has been already described in the previous paragraphs. Through the node Binary Classification Inspector it is possible to interact with the dashboard varying the threshold using a cursor that shows the predicted probability from the model. In this way you can see how AUC, Accuracy, Sensitivity, Precision change. When the probability changes, a blue square on the ROC curve moves to indicate the corresponding probability.

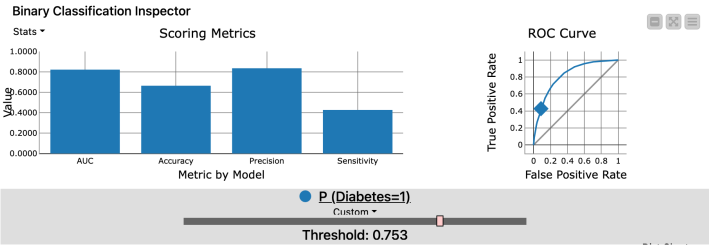
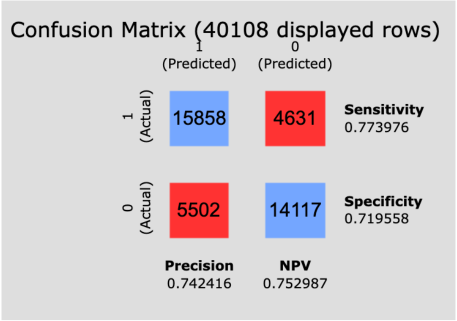


Figure 12: Binary Classification Inspector

Changing the probability value also changes the values of the confusion matrix in 13 associated with our model: TN, FP, TP, TN and associated performance measures.



Confusion Matrix (40108 displayed rows)

	1 (Predicted)	0 (Predicted)	
1 (Actual)	15858	4631	**Sensitivity** 0.773976
0 (Actual)	5502	14117	**Specificity** 0.719558
	Precision 0.742416	**NPV** 0.752987	

Figure 13: Confusion Matrix

As we can see from the last graph in fig.14, looking at the class that presents Diabetes = 1, we notice that the probability associated with the phenomenon increases with the number of records. Instead, in reference to the class that presents the variable Diabetes = 0, it is evident that to a greater number of records corresponds a lower probability associated to the phenomenon, that is not to have the diabetes.

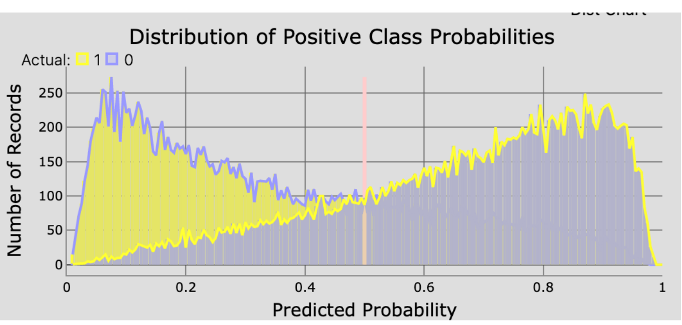


Figure 14: Distribution of class Probabilities