KNIME Challenge: Diabetes Prediction

Marco Sallustio 906149 , Marco Guarisco 789244 , and Salvatore Rastelli 903949

Dipartimento di Informatica Sistemistica e Comunicazione, DISCO, Università degli Studi di Milano Bicocca

Piazza dell'Ateneo Nuovo, 1, 20126 Milano MI

Compiled February 17, 2023

The main goal of this project is to predict whether or not a person has diabetes based on the various factors using different machine learning concepts learned during the course. Beyond the main goal, described above, we will create, again using the Knime software[3], a data app that will allow both to carry out an exploratory analysis of the dataset through the visualization of histograms, pie charts, scatterplots, correlation matrix and will also allow graphically the results that our classification model will produce. The dataset, available on the Kaggle platform[1], contains 17 feature variables and 1 target variable - "Diabetes". This is a binary classification problem where 0 means no diabetes and 1 means diabetes.

1. INTRODUCTION

Diabetes is among critical diseases and lots of people are suffering from this disease. There are several factors that can be a cause of diabetes; among these there is certainly age, gender, cholesterol level, Cholesterol check frequency, BMI, smoker(yes or no), Heart disease or attack, physical activity, consumption of fruit and vegetables, alcohol consumption, general, physical and mental healt, difficulty climbing stairs and stroke(yes or no). The International Diabetes Federation (IDF), in 2021, calculated that, worldwide, 536.6 million people between the ages of 20 and 79 (9.2% of adults) are diabetic and that a further 1.2 million children and adolescents (0-19 years) have type 1 diabetes. The number of adults with diabetes is also projected to increase to over 642 million in 2030 and 783 million in 2045. In 2021, deaths attributable to diabetes in world, between 20 and 79 years, were 6.7 million, 32.6% of the total in subjects under 60 years of age. Furthermore, according to a study conducted, the generalized increase in body weight - in the ranges of overweight (body mass index between 25 and 29.9) and obesity (over 30) - is in fact considered the first cause of the epidemic of diabetes, today four times more widespread than in 1980 (there were 108 million).

2. DATASET

As already mentioned, the dataset contains 17 feature variables and 1 target variable - "Diabetes":

- age: 3-level age category 1 = 18-24, 9 = 60-64, 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 = yes 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: 0 = no, 1 = yes
- Diabetes: 0 = no diabetes, 1 = diabetes (Target variable)

3. CHALLENGE OBJECTIVES

The objectives of the Challenge were:

- Design and development of an interactive dashboard (a KNIME data app) for univariate and multivariate data visualization
- Train and deploy a supervised classification model to detect diabetes

4. CLASSIFICATION MODEL TRAINING

The classification model was learned using a stacked ensemble algorithm. Stacking, also called Super Learning or Stacked Regression, is a class of algorithms that involves training a second-level “metalearner” to find the optimal combination of the base learners[4]. In order to implement it we used the "AutoML learner" node provided by the H2O integration, with which we were able to create and train a stacked ensemble of 3 different learners: Random Forest, Generalized Linear Model, Gradient Boosting Machine. To store the model algorithm, we then used the "Model to MOJO" node and the "MOJO Writer" node, saving the algorithm as a MOJO (Model Object,Optimized).

5. PERFORMANCE MEASURE

In our analysis, as established by the Knime Challenge, a single measure capable of assessing performance was used: the Log-Loss Function. 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. The formula of Log-Loss is:

$$Logloss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(P(y_i)\right) + (1-y_i)\log\left(1-P(y_i)\right)\right]$$

where:

- N is the number of observations;
- y_i is the actual/true value;
- $P(y_i)$ is the prediction probability;

6. RESULT

To validate the learning algorithm,we used the Holdout method: we first divided the dataset into two partition using a random sampling method, as the diabetes attribute was equally distributed; the training partition was set to be 67% of the dataset size and the test partition was set to be 33%. We then applied the algorithm on the training subset and queried the classifier on the test subset and evaluated the LogLoss on the test set, that resulted be = 0.504.

7. FEATURE SELECTION AND FEATURE CREATION

After validating the model, we considered the use of feature selection or feature creation. As the algorithm was computationally highly demanding, the filter or wrapper method was rejected and we decided to try to apply PCA in order to reduce the number of attributes and, subsequently, the computational cost of the learning procedure. It turned out that the results were not appealing as we lost a lot of information in the process and the LogLoss evaluated on the transformed attributes was of = 0.528, much higher of the value obtained without using PCA, so we decided to not include this step in the Training workflow.

8. DEPLOYMENT

In order to implement a deployment workflow, we decided to use an interactive approach: we created a component "File Reader" that could get the Test dataset as an input from the user. We imported the classifier model with a "MOJO reader" node and queried it on the input dataset with the "MOJO predictor(classifier)" node linked to both the File and MOJO reader nodes. Lastly we created two other components that, respectively, display to the user the LogLoss value, the Preview of the table with the classifier prediction, filtering only the Diabetes, Prediction(Diabetes), P(Diabetes=0) and P(Diabetes=1) attributes, and a download link to save the complete table in a local directory choosed by the user itself.

9. DATA APP

This section will describe the entire implementation phase of the Data App. Data Apps provide a user interface to perform scalable and shareable data operations, such as visualization, data import, export, and preview. With developing a KNIME Data App[2], a workflow developer has complete control over the interactivity that will be available to the end-user and the complexity of the underlying workflow. Alongside accessing KNIME's visual programming environment, KNIME Data Apps also provide the possibility to reach out to any number of technologies that are integrated into KNIME's open ecosystem (scripting languages like Python, machine learning libraries like H2O, etc). Furthermore, through KNIME Server you can also share your Data App with end-users with a scope to monitor and tweak based on feedback. KNIME Data Apps are built using special nodes in KNIME Analytics Platform that allow the user to update the look and feel of each page, build in interaction, and combine multiple pages in the app.

A. Diabetes Dataset Visualization

For our Data App we decided to create a visualizations part for the various data present in the dataset, in order to carry out an exploratory analysis of the dataset. To do this, our idea was to create an app with several selectable filters in order to obtain different results based on them. We have therefore created, through the Range Slider Filter node, the filters relating to diabetes (0/1), age and gender (0/1); in addition to these filters, others have been created in which it is possible to choose which type of visualization you want to use (Histograms, Pie Charts or Scatter Plots) and finally it is possible to choose for which type of variable you want to obtain its visualization chosen in the previous filter. As for the histograms, you can choose between the following attributes:

- BMI
- General Health
- Age
- Mental Health
- Physical Health

As for the Pie Charts you can choose between the following attributes:

- Sex
- Difficult Walking
- Stroke
- High Cholesterol
- Smoker
- Physically Active

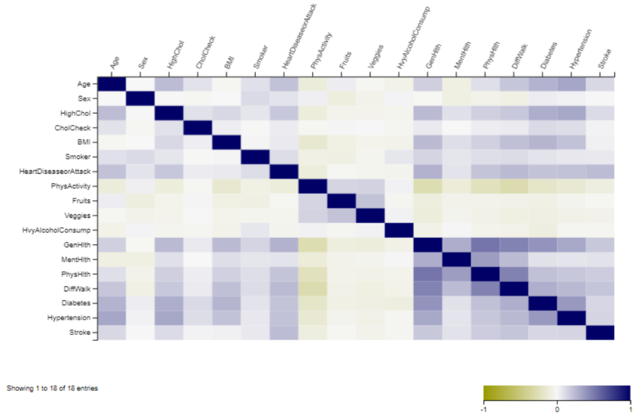
- Heavy Alcohol Consumption
- Diabetes

Finally, as far as Scatter Plots are concerned, it is possible to choose between the following pairs of attributes:

- Age-BMI
- Age-Mental Health
- Physical Health-Mental Health

B. Correlation Matrix

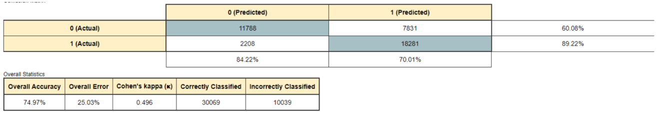
In addition to these views, within the same data app there is also the correlation matrix between all the attributes of the dataset. A correlation matrix consists of rows and columns that show the variables. Each cell in a table contains the correlation coefficient; it is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data.



C. Model Performance

C.1. Scorer View

Within this section the results of the classifier were evaluated. The following table has been added which summarizes the results obtained:



	0 (Predicted)	1 (Predicted)	
0 (Actual)	11788	7831	60.08%
1 (Actual)	2208	18281	89.22%
	84.22%	70.01%	

Overall Statistics

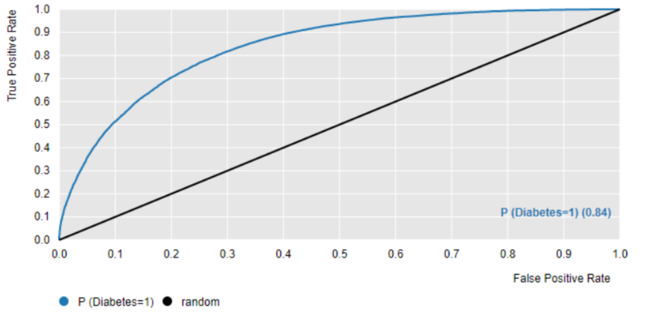
Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
74.97%	25.03%	0.496	30069	10039

C.2. ROC Curve and Lift Chart

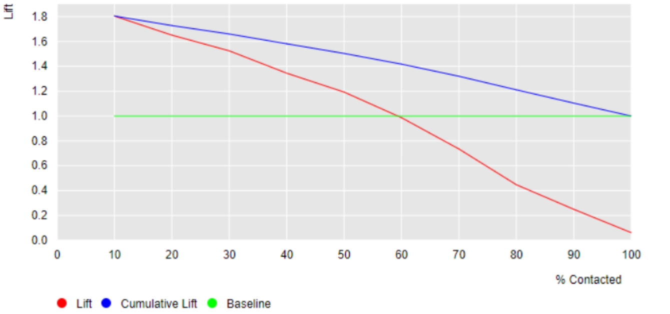
Finally, as the last section of the data app, two curves have been reported that allow us to evaluate our classifier: the ROC curve and the Lift Chart. The ROC curve (Receiver Operating Characteristic Curve) is the graphical technique for evaluating classification models:

- On the X-axis we have the percentage of false positive records (FPR);
- On the Y axis we have the percentage of true positive records (TPR).

Both values are expressed as a percentage of the total. The ROC curve represents the performance of a classifier without looking at the class distribution or the cost of error; it therefore serves to compare different classifiers to try to understand where a classifier is more or less effective. We have obtained this ROC Curve:



The Lift Chart, on the other hand, arises from the cumulative gains which has the percentage of the size of the subset considered on the X axis and the percentage of positive records on the Y axis. To obtain the Lift Chart, the percentage of positive records is divided by the percentage of the size of the subset considered and the corresponding Lift value is obtained. We have obtained this Lift Chart:



We have also reported the obtained AUC value under the two curves. The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes: the higher the AUC, the better the model's performance at distinguishing between the positive and negative classes. When AUC = 1, the classifier can correctly distinguish between all the Positive and the Negative class points. If, however, the AUC had been 0, then the classifier would predict all Negatives as Positives and all Positives as Negatives.When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative ones. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning that the classifier either predicts a random class or a constant class for all the data points. In our case we got a value of 0.839649138878574.

10. CONCLUSIONS

In conclusion, our machine learning project aimed to predict the presence or absence of diabetes in individuals using a stacked ensemble algorithm that incorporated Random Forest, Generalized Linear Model, and Gradient Boosting Machine learners. We used the Knime software and the H2O integration to create and train the model, optimizing the Log-Loss function to achieve a low value of $= 0.504$.

Our analysis was performed on a dataset that contained 17 feature variables and 1 target variable (Diabetes), and it was conducted using the Knime Challenge single measure of performance, the Log-Loss function. We decided not to perform feature selection, as we found that it did not lead to any significant improvement in the performance of our model. This decision was based on the fact that including all 17 feature variables in our analysis allowed us to achieve a lower Log-Loss value, indicating higher accuracy.

We developed a data app that allowed us to perform exploratory data analysis and visually display the predictions of our model. The data app included various visualization tools, including histograms, pie charts, scatter plots, and a correlation matrix, enabling us to easily identify patterns and relationships between the variables.

Overall, with our analysis we aimed to show the effectiveness of the stacked ensemble algorithm in predicting the presence or absence of diabetes in individuals, and our data app provided a user-friendly interface for exploring and interpreting the results of our model.

REFERENCES

1. https://www.kaggle.com/competitions/diabetes-prediction-competitiontfug-chd-nov-2022/overview
2. https://docs.knime.com/latest/data_apps_beginners_guide/#introduction
3. https://docs.knime.com/
4. https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html#id3