Machine Learning project

Davide Ghilardi, Lorenzo Lobosco, Fabio Salerno

Abstract

Diabetes is a chronic medical condition that affects a large portion of the global population. According to the International Diabetes Federation, there were an estimated 463 million adults living with diabetes in 2019, a figure that is projected to reach 700 million by 2045. Early detection and accurate prediction of diabetes can significantly improve patient outcomes and reduce healthcare costs. Machine learning models have shown promising results in predicting diabetes, but there is a need for further research to improve the accuracy and interpretability of these models. The aim of this study is to develop a Light Boosting model for diabetes prediction that will contribute as a valuable tool for healthcare providers to identify patients at risk for diabetes and enable earlier intervention and prevention strategies. The proposed model has the potential to reduce the number of missed diagnoses and unnecessary testing, ultimately improving patient outcomes and reducing healthcare costs.

Keywords: Machine Learning; Classification; Diabetes; Gradient Boosting

Contents

1. Introduction

Diabetes is a major health problem worldwide, affecting millions of people. According to the International Diabetes Federation, in 2021, approximately 537 million adults aged 20-79 years were living with diabetes globally. This number is projected to rise to 642 million by 2040, which is a 20% increase [4]. Diabetes is a chronic metabolic disorder characterized by high levels of glucose (sugar) in the blood. The condition occurs when the body either does not produce enough insulin or is unable to use insulin effectively. Insulin is a hormone produced by the pancreas that helps to regulate the level of glucose in the blood [1]. There are three main types of diabetes: type 1, type 2, and gestational diabetes.

Type 1 diabetes is an autoimmune disease in which the immune system attacks and destroys the insulin-producing cells in the pancreas, leading to a deficiency of insulin. This type of diabetes typically develops in childhood or adolescence, and people with type 1 diabetes require lifelong insulin therapy.

Type 2 diabetes is the most common type of diabetes, accounting for about 90% of all cases. This type of diabetes occurs when the body becomes resistant to the effects of insulin, or when the pancreas does not produce enough insulin to meet the body's needs. Type 2 diabetes can develop at any age, but it is more common in people who are overweight or obese, have a family history of diabetes, or lead a sedentary lifestyle.

Gestational diabetes occurs during pregnancy and typically resolves after delivery [1]. Diabetes is a leading cause of death, disability, and reduced quality of life. It is associated with an increased risk of cardiovascular disease, blindness, kidney failure, amputations, and other serious health problems. The burden of diabetes is particularly high in low- and middle-income countries, where access to diabetes prevention and management services is limited. The increasing prevalence of diabetes is driven by several factors, including aging populations, unhealthy diets, physical inactivity, and obesity. These trends are expected to continue in the future, highlighting the urgent need for effective prevention and management strategies [7]. This project was born with the goal of investigating the potential risk factors that can impact on the diabetes diagnosis. The aim of this project is to develop a diabetes prediction model that can help healthcare professionals identify patients who are at high risk of developing the disease, and to provide targeted interventions and prevention strategies to these individuals.

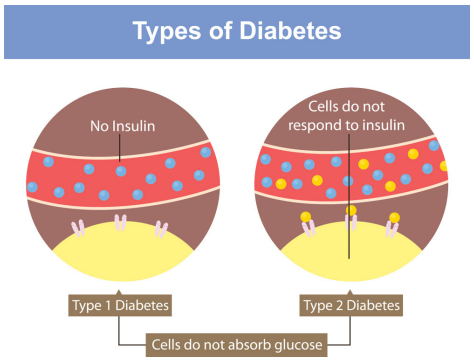

Figure 1 Graphical representation of type 1 and type 2 diabetes.

2. Dataset

The dataset used for our analyzes contains a list of risk factors that can lead to a diabetes diagnosis. It is composed of 40108 records, each referring to a different patient, and 18 variables, including the response variable. The dataset's variables are the following:

- **Age**: (AGEG5YR code) age categories; 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).

The dataset does not present any missing values and the target variable is balanced.

3. Preprocessing

The goal of data preprocessing is to transform raw data into a format that can be easily and effectively analyzed by machine learning algorithms. Considering the project goal, the target model and the data available, there the following operations were implemented:

- Feature creation
- Normalization
- Train-test split

A. Feature creation

The following feature where created and added:

- **CholRisk**: it's the product of CholCheck and HighChol and takes their place; this choice was taken since HighChol can be assessed only with a previous CholCheck
- **Comorbidity**: it means that several pathologies coexist in the same individual so it's defined as the total number of diseases, and in particular

$$\textit{Comorbidity} = \textit{HeartDiseaseorAttack} + \textit{CholRisk} + \textit{Hypertension} + \textit{Stroke}$$

- **Negatives**: a variable which includes all the negative habits of a person; it's defined as

$$\textit{Negatives} = \textit{HeartDiseaseorAttack} + \textit{CholRisk} + \textit{Hypertension} + \textit{Stroke} + \textit{Smoker} + \textit{HvyAlcoholConsump} + \textit{DiffWalk}$$

- **Positives**: a variable which includes all the positive habits of a person; it's defined as

$$\textit{Positives} = \textit{Fruits} + \textit{Veggies} + \textit{PhysActivity}$$

- **BMI_label**: since BMI attribute indicated the exact value of BMI, it was decided to categorise it into the five classes defined by the World Health Organization (WHO): underweight (BMI $<$ 18.5); normal weight (18.5 $\leq$ BMI $<$ 25); overweight (25 $\leq$ BMI $<$ 30); obesity class I (30 $\leq$ BMI $<$ 35); obesity class II (35 $\leq$ BMI) [6].

Finally, the scale of the variable GenHlth, which took values from 1 (excellent) to 5 (poor), was inverted to range from 0 (poor) to 4 (excellent).

$$\textit{GenHlth} = 5 - \textit{GenHlth}$$

B. Normalization

After the feature creation step, all the quantitative variables were normalized. Normalization is a data preprocessing technique that is used to transform numerical data to a standard scale, without changing the shape of its distribution. The main advantages of normalization are: easier interpretation of model coefficients, faster convergence of optimization algorithms and gain robustness to outliers so in general normalization could improve the performance and accuracy of machine learning models.

C. Train test split

To assure that predictions weren't biased by training data, causing overfitting, it was decided to split the dataset into train and test partitions. In particular, the first included 70% of the data, while the second the remaining 30%. Then, the classifiers were trained with specific methods only on the train set. Finally, the best optimized version of the model was chosen to be evaluated on the test set partition.

4. Modelling

A. Gradient Boosting

Gradient Boosting is a popular machine learning technique for both regression and classification problems. It is an ensemble method which involves combining the predictions of multiple weak models, called learners, to form a stronger model. The gradient boosting algorithm consists of iteratively training a new model to predict the errors of the previous one, gradually improving the overall performance. In particular, at each iteration, the algorithm calculates the negative gradient of the loss function with respect to the current model's predictions and uses this information to train a new model that predicts the residual errors. The new learner is then added to the ensemble and the process is repeated until the error is minimized or a stopping criterion is met (see Friedman, J. H., 2001 [3] for more details) . Gradient boosting is a powerful and widely used machine learning technique that can be highly efficient and effective for solving complex prediction problems. Some of its popular implementations include XGBoost [2] and LightGBM [5].

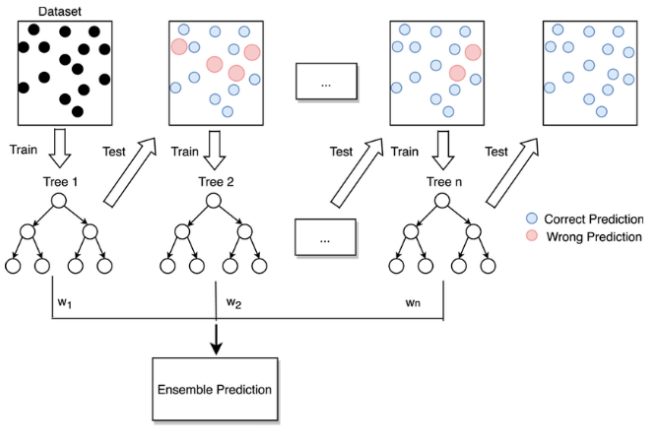

Figure 2 Gradient boosting workflow schema

B. Light Boosting

The light boosting model is a type of gradient boosting model. It's implemented in LightGBM (Light Gradient Boosting Machine), an open-source gradient boosting framework that uses tree-based learning algorithms. The key idea behind the LightGBM algorithm is that it uses a novel technique called "leaf-wise" tree growth to build a tree. This approach is different from the traditional "level-wise" tree growth because it focuses on growing the tree by expanding the leaf node that will yield the maximum gain. As a result, the LightGBM algorithm can converge much faster and generate better results than other gradient boosting algorithms. LightGBM can also handle large datasets efficiently by splitting the data by feature rather than by row, reducing memory usage and making it faster to train.

C. Feature selection

Before sending data to models, it was decided to implement a wrapper feature selection. This choice has been taken to reduce the redundancy introduced by data creation and to obtain faster results in the modelling phase. The wrapper feature selection algorithm can take a forward or a backward approach. It trains a model called evaluator on a subset of features S, which at the beginning contains all the attributes (in the backward approach) or none of them (in the forward approach). Then, at each step, it chooses the best feature to add (forward) or remove (backward) based on the cross-validation score of the evaluator. In the project, the evaluator is a light boosting model and, with a 5-fold CV, the following features were selected: 'Age', 'Sex', 'PhysActivity', 'HvyAlcoholConsump', 'PhysHlth', 'GenHlth', 'Comorbidity', 'Negatives', 'BMI_label'.

D. Grid Search

Grid search CV (Cross-Validation) is a technique used in machine learning to find the optimal hyperparameters of a model. They are parameters set prior to training a model that can significantly affect its performance. Grid search CV consists of two steps: the definition of a dictionary containing a grid of possible hyperparameter values, and the search through the grid to find the combination of hyperparameters that results in the best performance. The latter is done by training and evaluating the desired model for each combination of hyperparameters using k-fold cross-validation. In this project the following hyperparameters were tuned with a 3-fold CV:

- 'num_leaves': sets the maximum number of leaves in each tree.
- 'learning_rate': sets the step size used in updating the weights of the model; smaller values can result in a more accurate model but require more iterations to converge.
- 'n_estimators': sets the number of boosting rounds to perform; a large value can improve accuracy but also increases the training time.
- 'max_depth': sets the maximum depth of each tree;
- 'feature_fraction': sets the fraction of features to use in each tree; a value less than one introduces randomness in the selection of the subset of features, helping to prevent overfitting and improve the generalization of the model.
- 'min_data_in_leaf': sets the minimum number of samples required to be in a leaf node of each tree.

After all hyperparameter combinations have been evaluated, the one with the highest performance, measured in log-loss, is selected and passed to the final model. A drawback of Grid search CV is that can be computationally expensive since complexity grows exponentially with the increase in the number of hyperparameters that have to be tuned.

E. Cross validation

After the grid search step, the best model has been trained on the training set with a 15-fold cross validation. This means that, for 15 times, the training set has been split into two random partitions which contain respectively, 80% and 20% of the observations. Then, the model has been trained on the first partition and the metrics are evaluated on the second, which the model has never seen. It is very important to perform CV when training a machine learning model since it can spot overfitting and test the generalization capability of the model.

F. Performance metrics

For the analysis, three criteria were used to assess performance:

- *Log-loss*: also called binary cross-entropy, is a loss function used in classification problems. It measures the difference between the predicted probabilities of a model and the

actual target values and is defined as the negative logarithm of the likelihood function of the predicted probabilities

$$\text{Log-loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \ln(\hat{p}_i) + (1-y_i)\ln(1-\hat{p}_i)\right] \quad (1)$$

- *Accuracy*: computed from the confusion matrix, it's the ratio between the sum of correct predictions (TP + FN) and the total number of predictions;
- *AUC*: it's the area under the Receiver Operating Characteristic (ROC) curve, which relates the percentage of false positives (FPR, false positives rate) with the percentage of true positives (TPR, true positives rate) for various values of the decision threshold.

All three metrics have to be compared with the performance of a random classifier, so one that makes predictions at random, according to the class-attribute distribution in the train set. This approach leads to different baselines for each metric, in particular:

- Log-loss should be evaluated with: $p \ln \frac{p}{1-p} + \ln(1-p)$;
- Accuracy should be evaluated with: $2p(1-p)$;
- AUC should be evaluated with 0.5.

where p is the probability of the positive class in the training set. It's also important to note that while accuracy weighs all the predictions the same (correct and not), the log-loss function penalizes incorrect predictions more strongly than correct predictions.

5. Results

The following table summarizes the results of the cross validation performed on the training set; the first row refers to the model trained on all the features, and the second to the one trained with feature selection.

Model	Log-loss	Accuracy	AUC
FS	0.513 (0.005)	0.748 (0.005)	0.823 (0.005)
No FS	0.511 (0.005)	0.748 (0.005)	0.824 (0.006)

Table 1 Mean performance metrics and standard deviations for the models trained with 15-fold CV.

It can be seen that even though non feature selection model shows better performances, they aren't statistically different from feature selection model's.
The table below shows the performances of the two models, trained on the complete training set, on the test set.

Model	Log-loss	Accuracy	AUC
FS	0.517	0.745	0.820
No FS	0.515	0.748	0.821

Table 2 Performance metrics for the models on the test set.

The results are the same, non feature selection model presents slightly better values for all the metrics. Since for the deployment, the model with better performance has to be chosen, the selected one is the non feature selection. Moreover, it's important to say that neither FS, nor No FS shows signs of overfitting, with their performances worsening alway less than 1%.

6. KNIME Dashboard

In KNIME, a dashboard is a visual display of the most important and relevant information from the data analysis process. It can include customisable interactive charts, graphs, and other visualizations that allow to quickly and easily see the results of the data pipeline. Dashboards in KNIME can be used to monitor ongoing processes, to provide a quick overview of the status of a project, or to provide a summary of the results of a complex analysis. Finally, dashboards in KNIME can be updated in real-time, giving the stakeholder instant access to the latest data and insights, which can be customized to meet his specific requirements.

A. Explorative analysis

The primary objective of the initial data app component was to develop an exploratory data analysis dashboard that emphasizes the correlation between the key explanatory variables and the target. The two primary explanatory variables that are considered in this regard are the age bins and the BMI. These variables are critical in analyzing the correlation between a patient's age and weight and the occurrence of diabetes. Additionally, to gain a better understanding of how habits and diseases are distributed among healthy patients and those affected by diabetes, two barcharts have been implemented. The patients have been classified into two categories: those who are healthy (0) and those who have been diagnosed with diabetes (1). By analyzing the data obtained from these barcharts, for example it is possible to gain insights about how various diseases affect patients with or without diabetes. The barcharts have been created using the sum of patients in each category. For instance, by analyzing the data from the barcharts, it is possible to observe how the occurrence of a specific disease or habit is affected by having diabetes or not. This information can then be utilized to identify potential risk factors for patients, as well as for creating more targeted interventions to mitigate the risk of developing diabetes.

B. Modelling results

The second component of the data app aimed to showcase the accuracy and effectiveness of the machine learning model. To achieve this, various graphical representations were used to visualize different aspects of the model's performance. First the ROC curve graph was developed to illustrate the ability of the best model to predict outcomes with good probability. This graph provided a clear understanding of how the model performed in comparison to the baseline. Taking into account the two models, NO feature selection and feature selection, for each of them a scatter plot and a line plot were implemented. The scatter plot has the goal to compare the probability values against the corresponding Log Loss values. Furthermore the line plot has the goal to identify the average change in Log Loss based on the variation of model parameters.

7. Conclusions

This machine learning project aimed to develop an accurate model for detecting diabetes by using patient information such as age, type of diet, habits, weight, and health exploiting potential risk factors. The project used the Light GBM algorithm to

train and test the model, and compared the performance of the model with and without feature selection and additionally to find the optimal hyperparameters a grid search was performed. The evaluation results showed that the model achieved a satisfying log-loss value. Additionally, the project investigated the impact of feature selection on model performance. The results showed that reducing the number of features maintained approximately the same performance. The idea behind the project has significant implications for the medical field, providing reliable methods for detecting diabetes cases based on patient information can provide many benefits such as efficient resources allocation, early detection and personalized treatments. However, further research is necessary to evaluate the model's performance on different populations and to determine its generalizability to other clinical settings. Additionally, the inclusion of additional risk factors and larger datasets could further enhance the accuracy of the model. There are several potential directions for further development of this machine learning project, some of them are: expansion of the dataset, this project used a specific dataset for developing and testing the model. However, the inclusion of larger and more diverse datasets could enhance the accuracy and generalizability of the model. This could include data from different populations, countries, and clinical settings. Integration with electronic health records (EHRs), the integration of machine learning models with EHRs could enable real-time detection and monitoring of diabetes in clinical settings. This could improve the diagnosis and treatment of diabetes by providing clinicians with timely and accurate information. Incorporation of more risk factors, while this project used several risk factors, including age, BMI, health problems the inclusion of additional information, such as genetics, glucose level or environmental factors, could further enhance the accuracy of the model. Comparison with other machine learning algorithms, while this project used the Light GBM algorithm, other machine learning algorithms could be evaluated for their performance in detecting diabetes using risk factors. This could provide insights into the strengths and limitations of different algorithms and inform future research in this field. Overall, there are several potential directions for further development of this machine learning project, which could have significant implications for the diagnosis and treatment of diabetes. In conclusion this project tries to demonstrate the potential of machine learning for detecting and diagnosing diseases and highlights the importance of data-driven approaches in healthcare and it has to be said that machine learning models can assist experts in disease detection, and thus become an important asset for the medical industry.

References

[1] Centers for Disease Control and Prevention. *Diabetes*. https://www.cdc.gov/diabetes/basics/diabetes.html. accessed September 30, 2021.

[2] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[3] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

[4] *International Diabetes Federation Diabetes Atlas*. https://www.diabetesatlas.org/.

[5] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[6] World Health Organization. *Obesity*. https://www.who.int/health-topics/obesity#tab=tab_1.

[7] World Health Organization. *Global Report on Diabetes*. Geneva, Switzerland: World Health Organization, 2016. URL: https://www.who.int/publications/i/item/9789241565257.