Diabetes Classification using Machine Learning Algorithms

Rodrigo Santos, nº60552

19 May 2023

1 Introduction

Diabetes is a chronic disease that affects millions of people worldwide. Early detection and prediction of diabetes can help in preventing or delaying its onset, thereby reducing the risk of complications. Machine learning algorithms have shown promising results in predicting and classifying diabetes. This project aims to build and compare the performance of ML models for diabetes classification using a dataset that is publicly available.

2 Approach

In this project, I aimed to develop a classification model for diabetes using two datasets, one balanced dataset with 70,692 instances and another unbalanced dataset with 253,680 instances, from a survey done in 2015 from Behavioral Risk Factor Surveillance System.

2.1 Data Pre-processing

I began by performing some initial EDA (exploratory data analysis) to gain insights into the distribution and quality of the data. I checked for missing values which it didn't have, then I checked for duplicated data, whereas it contained duplicated instances and I decided to not drop it since it could be multiple respondants giving exactly the same answers as other respondant. Also, I did some visualization of the distribution of the features to understand better some relationships of the features and correlation to the factor that an individual has diabetes or not.

2.2 Data Splitting

I decided to split the data into training and test sets. I didn't have the need to split into validation set since I used GridSearchCV to tune the hyperparameters.

2.3 Model Selection

I experimented with seven different classification models: Decision Tree Classifier (DTC), AdaBoost Classifier (ABC), XGBoost Classifier (XGBC), Multi-Layer Perceptron Classifier (MLPC), Random Forest Classifier (RFC), Logistic Regression (LR) and Gaussian Naive Bayes (GNB).

2.4 Model Evaluation

To evaluate the performance of each model, I used several classification metrics such as accuracy, F1-score, precision, recall, and Matthews correlation coefficient.

2.5 Final Model Selection

I selected the best model based on its performance on the test set. I also examined the feature importance of the selected model to identify the most important predictors of diabetes.

3 Implementation

I started by loading the data and performing an EDA (exploratory data analysis) to understand better the data. Then, I started splitting the data into training and test sets with 90/10 ratio.

After the splitting, I defined seven pipelines for different machine learning models, namely DecisionTreeClassifier, AdaBoostClassifier, XGBClassifier, MLPClassifier, RandomForestClassifier, LogisticRegression and GaussianNB. I decided to use pipelines so I can define a StandardScaler for the following models: MLPClassifier and LogisticRegression. I then defined hyperparameters for each model using a dictionary and also defined metrics for GridSearchCV such as accuracy, f1 score, matthews correlation coefficient, precision, and recall.

Using GridSearchCV, I performed hyperparameter tuning on each model pipeline and evaluated the best performing model. I stored the best performing model and its corresponding hyperparameters for each model type.

Next, I started training each model with its best hyperparameters combination and predict using the test set to evaluate its performance using the same metrics I used on hyperparameter tuning.

Finally, with the best model trained, I examined the feature importance with its built-in feature importance method and using SHAP, to identify the most important features that determine if an individual has diabetes or not.

4 Results

4.1 Balanced Dataset

Model	Accuracy	F1	Matthews	Precision	Recall
DTC	0.7401	0.7445	0.4808	0.7284	0.7614
ABC	0.7540	0.7585	0.5088	0.7412	0.7767
XGBC	0.7582	0.7672	0.5187	0.7361	0.8012
MLPC	0.7551	0.7670	0.5138	0.7280	0.8104
RFC	0.7479	0.7556	0.4973	0.7297	0.7834
LR	0.7512	0.7547	0.5029	0.7405	0.7695
GNB	0.7239	0.7192	0.4479	0.7278	0.7109

Table 1 - Metrics results on the small dataset with 70 692 instances

4.2 Unbalanced Dataset

Model	Accuracy	F1	Matthews	Precision	Recall
DTC	0.8631	0.1621	0.1955	0.5894	0.0940
ABC	0.8641	0.1989	0.2206	0.5871	0.1197
XGBC	0.8656	0.2465	0.2531	0.5877	0.1560
MLPC	0.8652	0.2437	0.2495	0.5815	0.1541
RFC	0.8600	0.0141	0.0718	0.8793	0.0071
LR	0.8620	0.2421	0.2349	0.5354	0.1564
GNB	0.7710	0.4102	0.2983	0.3219	0.5652

Table 2 - Metrics results on the large dataset with 253 680 instances

4.3 Models Performance

When evaluating the performance of the models on both balanced and unbalanced datasets, I have to consider different metrics since that in the case of balanced datasets accuracy can be a reliable metric to evaluate the overall model performance. However, in unbalanced datasets accuracy alone can be misleading since that the majority class tends to dominate the accuracy calculation, leading to high accuracy values even if the model fails to effectively predict the minority class. So, its crucial to consider other metrics such as F1-score and Matthews Correlation Coefficient.

XGBoost Classifier (XGBC) and Multi-Layer Perceptron Classifier (MLPC) got similiar results on the performance but XGBoost got better results by little, so it is the best performing model on both datasets, altough both models performance decreases on the unbalanced dataset compared to the balanced one.

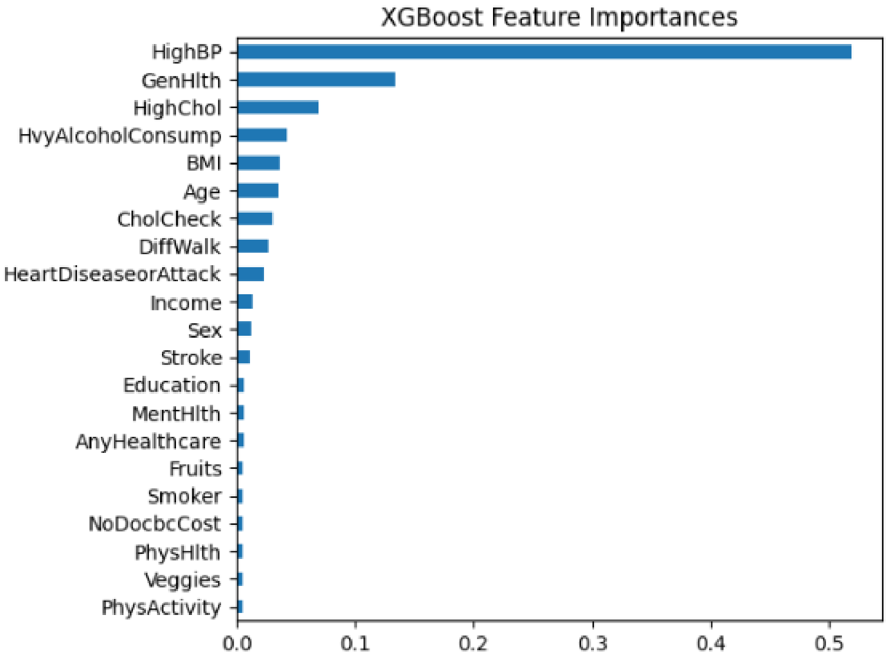
Decision Tree Classifier (DTC), AdaBoost Classifier (ABC), Random Forest Classifier (RFC) and Logistic Regression (LR) also show a drop in performance on the unbalanced dataset while having a good performance on the balanced dataset.

Gaussian Naive Bayes (GNB) shows relatively better performance on the unbalanced dataset, although its performance is still limited compared to other models.

In general, all models performed better on the balanced dataset compared to the unbalanced dataset due to the fact that the models faced challenges in predicting the minority class in the unbalanced dataset.

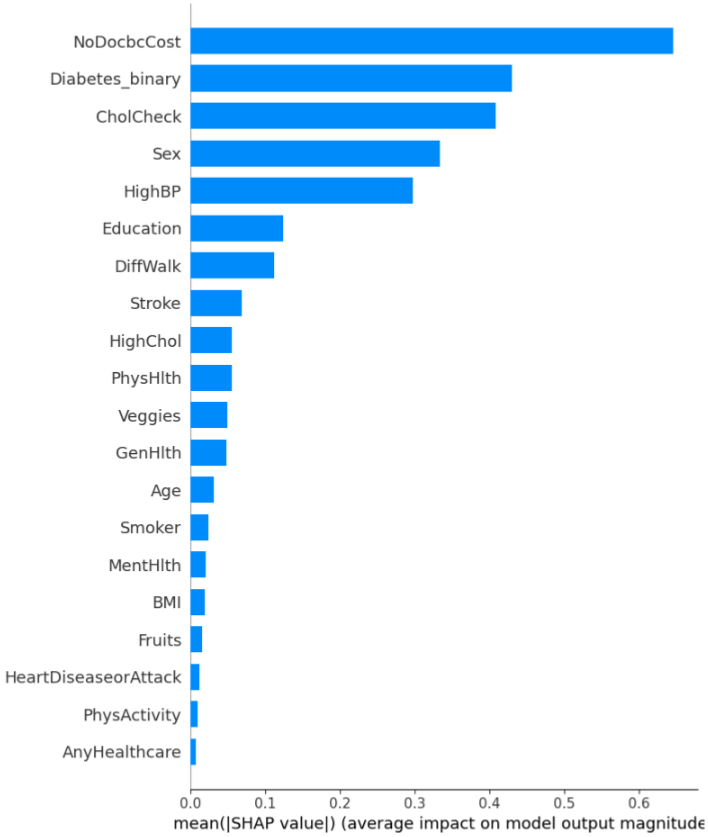
4.4 Feature Impotance

According to the plot from the feature importances on the best performing model (XGBoost), the following features had the most impact and influence on the model's output: "HighBP", "GenHlth" and "HighCol". All these features are health related features.



Graphic 1 - XGBoost built-in feature importance

Next, I used SHAP (SHapley Additive exPlanations) and plotted the summary plot using SHAP values. In the analysis, I found out that the features "NoDocbcCost", "CholCheck", "Sex" and "HighBP" displayed mean absolute SHAP values greater than 0.28, indicating their significant impact on the model's predictions. "CholCheck" and "HighBP" are health related features and the feature "NoDocbcCost" represents whether individuals experienced a situation in the past 12 months where they needed to see a doctor but were unable to due to financial constraints. According to SHAP, "NoDocbcCost" is the most influential feature, which it could be related to the fact that if a individual had seen the doctor before, it could have more chances of preventing the disease.



Graphic 2 - SHAP plot on the XGBoost

5 Final comments

In this project, one of the issues that I have encountered is the fact of taking so much time tuning the hyperparameters to find the best combination. If I had more computer power, I could have tested more on the hyperparameters tuning. Also, I wanted to use SVM (Support Vector Machine), but because of the time it was taking to do the GridSearchCV, I decided to not use the model. In the large dataset, I could have done oversampling or undersampling, but I decided to remain the unbalanced dataset as it is, so I could test the effects of an unbalanced dataset, and to do a comparison of both datasets.

6 Bibliography

1. kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset
2. https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

3. https://towardsdatascience.com/metrics-for-imbalanced-classification-41c71549bbb5