Predicting Patient Health

Ayush Anand
Department of Computer Science, IIT Gandhinagar
ayush_anand@iitgn.ac.in

Bhoomika Mandloi
Department of Computer Science, IIT Gandhinagar
bhoomika.m@iitgn.ac.in

Ramireddy Lakshmi Nageswari
Department of Computer Science, IIT Gandhinagar
lakshmi.nr@iitgn.ac.in

Abstract

Diabetes is a chronic disease caused by abnormal levels of blood sugar which affects millions of people every year. Predicting whether a patient has diabetes based on features like age, weight, BMI, blood glucose levels, blood creatinine levels and so on is an important machine learning task. In this project, we have compared the performance of various machine learning models to predict whether a patient has diabetes or not. Recently methods like XGBoost and LightGBM have gained much popularity due to their high performance and efficiency. We compare these methods with more classic machine learning models like Random Forests and Adaboost and analyze how the results vary when the parameters are varied. Through a comparative analysis of these models, we gain important insights into the choice and performance of models used for predicting patient health tasks.

Keywords and phrases Diabetes Milletus, Random Forest, CatBoost, XGBoost, Gradient Boost, Adaboost

1 Introduction

Using machine learning models for predicting patient health has been a major area of research for decades now. Patient information stored in hospital databases consist of important features like age, BMI, certain chemical levels in blood and so on which might help in predicting whether a patient is suffering from a particular disease or not.

Diabetes Mellitus (DM)[8] is a significant public health problem that is approaching epidemic proportions.The risks of diabetes have increased further in recent years due to declining food habits and lifestyle. Diabetes can be caused due to several factors like obesity, consumption of unhealthy food, heredity and so on. WHO reports that as of 2015, almost 400 million people around the world had been diagnosed with diabetes, with the numbers rising every year. Diabetes is a major cause of blindness, chronic kidney failure, stroke and heart attacks. There are three kinds of diabetes. Type 1 Diabetes is caused when the pancreas fails to produce sufficient insulin. Type 2, and the most common type of diabetes is caused by excessive body weight and obesity. Third is gestational diabetes which occurs in pregnant women with no prior history of diabetes.

In this project, we have compared the performance of various machine learning models to determine whether a patient admitted to an ICU has been diagnosed with diabetes, specifically diabetes mellitus type 2. The dataset we have used is taken from MIT's GOSSIS `https://gossis.mit.edu/` (Global Open Source Severity of Illness Score) initiative.

2 Dataset

We have used the patient information dataset[6] from MIT's GOSSIS (Global Open Source Severity of Illness Score) `https://gossis.mit.edu/` initiative, made available through

Kaggle. The dataset consists of 130157 rows and 180 columns and contains information about various patients admitted to the ICU like age, BMI, height etc. which we will use to predict if the patient has diabetes.

2.1 Data preprocessing and cleaning

We start with an examination of how imbalanced the data is. We found out that the data is pretty imbalanced with 102006 data points belonging to the 0 class(not having diabetes) and the rest belonging to the 1 class. This is expected as the proportion of patients having diabetes should be much smaller than those not having diabetes.

The features consist of several indexing columns like patient id and hospital id, which do not really contribute much to the prediction we aim to make. We drop these columns straightaway. Next, we move on to an analysis of the missing values in the dataset. We note that certain features consist of many missing values, and these features effectively do not add much value to the information we have from the remaining features. So, we find out the ratios of the missing values in each feature column and drop those features which have more than 60 percent missing values altogether.

Next, we move on to analyze what are the categorical and numerical features we have. We find out that we have 89 numerical features and 6 categorical features. These categorical features need to be 1 hot encoded for our models to be run. Finally, we fill in the still missing values in a feature column as the average of all the other values of that feature.

Feature Selection: We have a total of 136 columns/features after the preprocessing and cleaning steps defined in the above sections. Training on these many features wouldn't be efficient as a lot of these features have much less importance in our prediction task as compared to the others. So, we fit a random forest classifier on the data and plot the relative feature importances. Figure 1 below shows the plot. Notice how only a few features carry a lot of importance as compared to the rest.

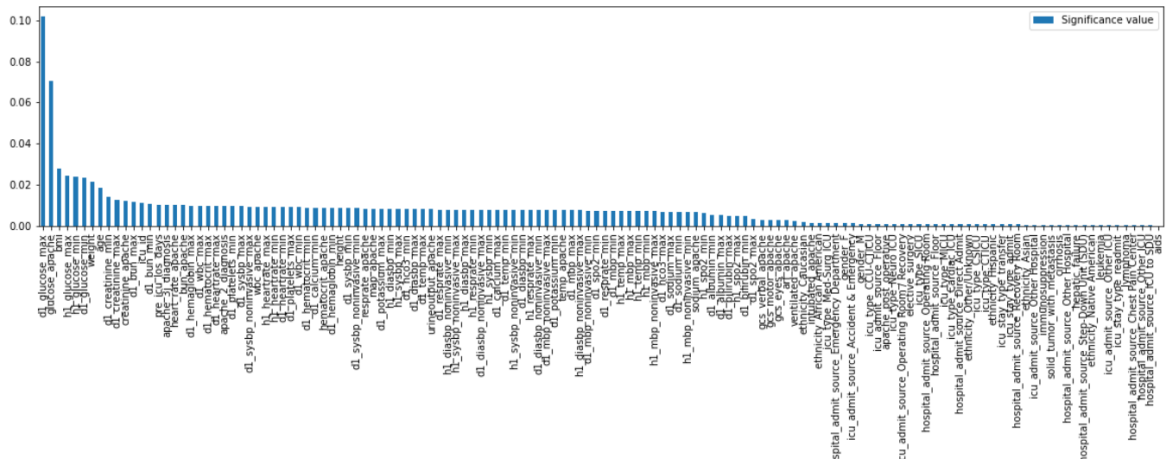

Figure 1 Relative Feature importances obtained after fitting random Forest Classifier on the data

Based on the above plot, we pick the top 10 most important features and train our remaining models on them.

3 Approach

Next, we give a brief description of the classification models used by us in the prediction task. We find the AUROC scores[9] for all of these models and provide a comparison of the results in the next section. The AUROC score was chosen because it provides an aggregate measure of performance across all possible classification thresholds.

3.1 Gradient Boosting

Each predictor in Gradient Boosting[5] aims to improve on the previous one by reducing the errors. It is a greedy algorithm and can overfit a training dataset instantly. It is an ensemble algorithm that minimizes an error gradient to fit boosted decision trees. Gradient Boosting's unique concept is that, rather than fitting a predictor on the data at each iteration, it fits a new predictor to the residual errors created by the preceding prediction. The method will get the log of the odds of the target feature in order to make preliminary predictions on the data. It is the number of True values divided by the number of False values. We use a logistic function to make predictions by converting the log(odds) value into a probability.

3.2 XGBoost

eXtreme Gradient Boosting[4] is abbreviated as XgBoost. XGBoost is a high-speed and high-performance implementation of gradient boosted decision trees. On classification and regression predictive modeling issues, XGBoost dominates structured or tabular datasets. Decision trees are created sequentially in this approach. In XGBoost, weights are very significant. Weights are allocated to all the independent variables , which are subsequently put into the decision tree, which predicts outcomes. The weight of variables that the tree predicted incorrectly is raised, and these variables are put into the second decision tree. These individual classifiers/predictors are then ensemble to create a strong and more precise model.

3.3 Adaboost

Adaptive Boosting classifier[2] combines multiple weak classifier algorithm to form strong classifier. We may achieve a decent overall classifier accuracy score by combining multiple classifiers with selection of training set at each iteration, and allocating the right amount of weight in final vote. AdaBoost combines the predictions from one-level decision trees, known as decision stumps. It helps in selecting the training set for each new classifier you train depending on the previous classifier's findings. When integrating the findings, it decides how much weight should be given to each classifier's recommended response.

3.4 CatBoost

Category Gradient Boosting[3] model can used in different ways not only as regressor or classifier. It can be used for ranking and recommendations and even as personal assistant. CatBoost provides necessary changes for the Gradient boost. In the gradient boosting we need to handle the categorical features separately by converting them to numerical values. That is a class is a given a number and the rest in the same way. While in the case of CatBoost it automatically deals with the categorical columns without any preprocessing. In case of Catboost in every level the number of features are one that is a binary decision tree which helps in decreasing the prediction time. In case of default libraries of Catboost the

default parameters provide better results which reduces the efforts in tuning of parameters. But the loop hole is the missing values have to be dealed or preprocessed before hand. Ordered boosting that is dealing with the categorical values makes it special.

3.5 Random Forest

Decision trees generally shows high variance and the outliers change the model drastically. So to deal with this ensemble of different decision trees are taken with few features missing in each decision tree. This can also used to find the importance of the features. After training the decision trees with different set of features we take the average of output in case of regressor. In the classifier we take the mode of the output from the decision trees as output. This helps in increasing the accuracy by dealing with the features which play no role and with the outliers.

4 Results

The average AUROC score obtained when we perform the XGBoost classification[1] using K-Stratified fold is 0.7143. Greater the AUROC greater the accuracy of the classifier. Here we took number of folds=5 and the below graph shows the AUROC over the folds and how it varies randomly.

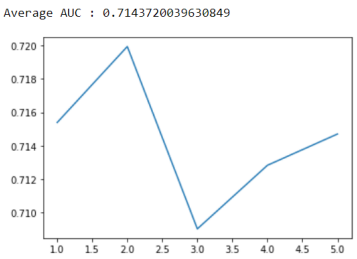

Figure 2 XGBoost AUROC scores over the 5 folds

Later we observe that when we perform the classification using Random Forest[7] with random parameters for the Sklearn model lead to the AUROC score of 0.81. Further the best parameters found for the Random Forest lead to 0.84 score.

Then the results of the AUROC score of Gradient Boost classifier is 0.82 and then we perform the ensemble of both Gradient boosting and random forest for the better performance.

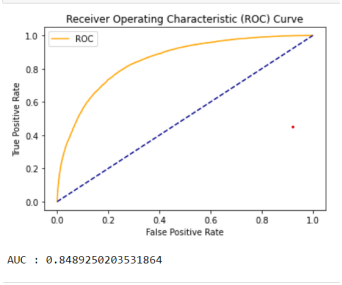

Figure 3 ROC of Random forest

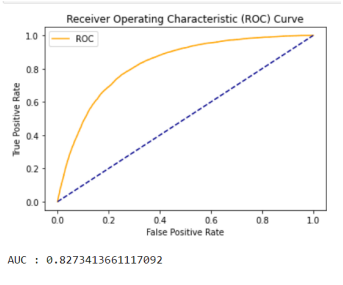

Figure 4 ROC of Gradient boost

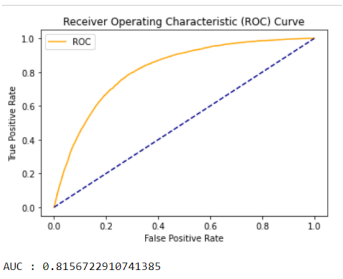

Figure 5 ROC of AdaBoost

Table 1 AUROC scores of various models

Model	AUROC Score
Random Forest with default parameters	0.8169
Random Forest with better parameters	0.8489
Gradient Boost classifier	0.8273
Ensemble of Random Forest and Gradient Boost	0.8399
AdaBoost classifier	0.8156
Ensemble of AdaBoost+RandomForest+Gradient Boost	0.8398
XGBoost average over 5 Stratified folds	0.7143
CatBoost classifier	0.8474

On performing the ensembling we get the AUROC to be 0.8399 approximately 0.84 which shows an increase with respect to gradient boosting. In the same way we perform the classification over the rest of the models and the AUROC of the models are tabulated above.

5 Discussions

We plotted the Receiver Operating Characteristic curves for each of the mentioned models and calculated the AUROC score. We see that the ROC curves are fairly above the 0.5 line, which means our classifiers have performed well. All the models give comparable performance with CatBoost slightly exceeding the others. The training time for CatBoost is also fairly low which leads us to believe that this could be one of the better models for the patient prediction task.

6 Conclusion

In this project we implemented and tested the performance of several machine learning models[10] on patient health data to predict if a patient has diabetes or not. We used the evaluation metric AUROC score for comparison as it gives us a metric for the model's ability of distinguishing between patients having a disease and no disease which is ideal for our scenario. We noticed that Random Forest, Adaboost and GradientBoost all give almost the same auroc score of around 0.82 which is pretty good. XGBoost slightly underperforms as compared to the other models with an AUROC score of around 0.71. CatBoost performs slightly better than the other models with an AUROC score of 0.84. Finally, we try to combine the predictions of some of these models and return their average as the final prediction. This also gives fairly good results.

7 Contribution

Equal contribution by all.

References

1 Ahanagangopadhyay. Ahanagangopadhyay/wids-datathon-2021-diabetes-prediction `https://github.com/ahanagangopadhyay/wids-datathon-2021-diabetes-prediction`.
2 D Akash. Understanding adaboost.
3 Apoorva. CatBoost The fastest algorithm and Medium.

4 J Brownlee. A gentle introduction to XGBoost for applied machine learning. Machine Learning Mastery.

5 B Jason. A gentle introduction to the gradient boosting algorithm for machine learning.

6 Kaggle. Wids Datathon 2021. Kaggle. (n.d.).

7 KanchanSatpute. Wids-Datathon-2021/WiDS-Datathon.ipynb at main · Kanchansatpute/Wids-Datathon-2021 https://github.com/KanchanSatpute/WiDS-Datathon-2021/blob/main/WiDS-Datathon.ipynb. 2022.

8 Maniruzzaman. Classification and prediction of diabetes disease using machine learning paradigm.

9 S Narkhede. Understanding AUC - roc curve. Medium.

10 Precillieo. WIDS-Datathon-2021 https://github.com/Precillieo/WIDS-Datathon-2021. 2022.