BINARY DIABETES

Note: You will find how the document is very similar to the comments from the jupyter notebook, but we considered that were necessary to properly follow the notebook process and operations.

INTRODUCTION

The main objective of our project is to apply a classifier on a dataset of 22 feature variables. This dataset contains information of people that may or may not have diabetes. Our goal is to use the diabetes column as target and classify between; 0 (non-diabetic patient) and 1 (diabetic). We have chosen this dataset about diabetes because it's a really frequent disease that still has no cure, although there are treatments to keep living with it, we believe that more people should know about it.

EXPLORATION OF THE DATA

First of all, before we process our classifier, we have to visualize and explore the data. The initial step is to know the shape and the legend of the dataset. From this information, we can observe that the data is in a scale instead of its actual value, this way the privacy of the patients to create this dataset is preserved. Then we realize that not all the scales are in the same range and for some it has an explanation. Take for example all the categories that are either yes or no obviously it can just have 2 values 0 or 1 (binary variables). But then there are some features that the range is 1-30 like Physical Activity maybe for each day of the month the subject exercices? we did not find any reasonable explanation and there are other more subjective examples like General Health that goes 1 to 5 why that range? Again we did not find any logical solution to that. Moreover, it is necessary to separate binary variables from the other ones because later we will only scale the non-binary variables in order to have all the scaled variables in the same range (we mentioned that some ranges are from 1-30, others from 1-5 and so on). After this separation is done it is time to search if there are any missing or duplicated values. We didn't find NA values so we will not have to deal with them, but we did find 24206 duplicated observations so we removed them.

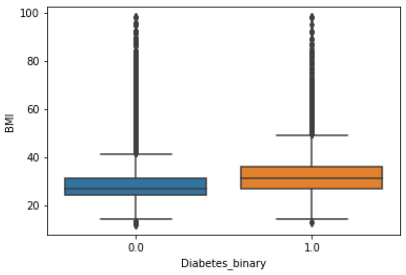
Moreover, to check if we had some redundant features we made the sns heatmap correlation plot of all the variables. The result turned out that all of our features were informative, which means that feature extraction was not needed in our dataset. We demonstrated that all the features were explicative by observing that none of the variables has a correlation between them higher than 0.6.

<u>Exploration of some features</u>

Before starting with the classifier models, we looked into some features alone and we discovered that;

- Less than 20% of the patients suffer from diabetes, IMBALANCED SAMPLE.
- The education variable does not create biases in the output of the classifier.
- We have more non-smokers than smokers patients.

- The information from the dataset is extracted from more female than male patients.
- More than 40% of the patients have high cholesterol levels.

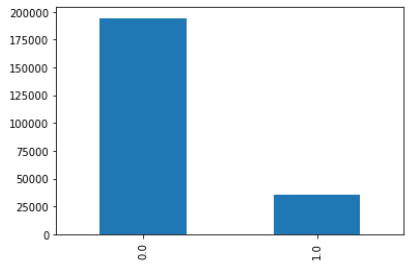


- Patients with more BMI (mass body index) have more risk of suffering from diabetes. Class 1 represents diabetic patients. The BMI mean for diabetic patients is higher than from non-diabteic.

DIFFERENT CLASSIFIERS MODELS TRAINED WITH THE BEST HYPERPARAMETERS

Our goal in this part is to understand and apply all the classification models explained in class and observe and compare the different results obtained from each classifier.

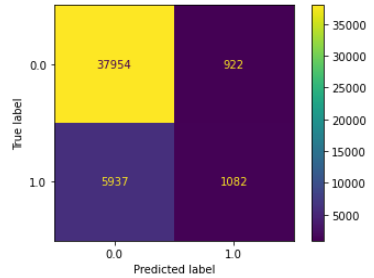
Preprocessing data before classifiers



As we can see, the minority class is class 1 (Diabetic). The data is highly unbalanced because as we have seen before in the analysis of the features, more than 80% of our patients are from class 0 (non-diabetic). So, we expect to have problems by classifying the minority class (diabetic class) as the algorithm will learn that it is more probable to be non-diabetic. Going further, we expect to have a great number of False Negative patients, diabetic (class 1) patients misclassified as non-diabetic (class 0) patients. This great number of False Negative patients will affect Recall (TP / (TP + FN)) causing its lower score and also will affect in a minor way to Precision (TP / TP + FP) because we will obtain a low number of True Positive patients (Diabetic patients well classified as Diabetic).

Firstly we are not going to modify the dataset and see the results we get, then if the results are unsatisfactory we will modify the training set as needed maintaining fidelity to the original data. We must start by scaling (only the non-binary features) and dividing the data into various sets.

Classifiers



The first classifier is Logistic Regression and gives these results: Accuracy: 0.85, Precision: 0.53, Sensitivity (Recall):0.15, Specificity: 0.97 and F1 score: 0.23. Now we can conclude that our first suppositions about the imbalanced data were true as Specificity is very high due to the good classification done for the class 0 (non-diabetic,

which is the bulk of the dataset) and the low number of False Positive patients (patients from class 0 misclassified as class 1). Also, Accuracy has a good score because as the majority of the observations are from class 0 and this class is well classified, the model in total assumes that both classes will classify well. Moreover, we can confirm as expected that the Recall score is very low and that Precision is low because of the small number of initial data we have for class 1 (imbalanced data). To solve this there are many methods, one of them is balancing the data. Even so, first, we will try to fit more complex classifiers with the Grid Search setting as the scoring parameter of the F1-SCORE (a metric measure that in its formula involves Precision and Recall). By this way, we will try to get higher Precision and Recall scores in our models. We first prefer to perform this technique before starting to balance the data because we think that it is better to maintain the real training data to provide a more realistic model. If we see that with this F1 maximization technique any of our classifiers work well, we will perform the balance of the dataset.

Testing the following algorithms, gave a low Recall and therefore f1 score which did not surpass 0.3 even though using Grid Search for finding the best hyperparameters to increase the f1 score as much as possible: Logistic Regression, K-Nearest-Neighbours, Decision Tree classifier.

Then we tested out ensemble learning methods such as soft and hard Voting classifiers, Bagging and Random Forest also applying grid search to increase the f1 score. The only noticeable finding was the soft Voting system which gave an f1 of 0.33 and the rest were overall better than other classifiers but it was not enough.

We tried with Boosting classifiers AdaBoost and Gradient Boosting again with grid search to increase f1, but the recall is still low and the f1 score < 0.3 which is really low. Even so, the other metrics were quite good.

Continuing with the Support Vector Machine classifier, which needed too much computational power and we were unable to see the result as our dataset has more than 200000 training observations, even leaving the computer running for hours and hours.

While testing different classifiers we encountered that KNN, voting and bagging were overfitted to the training set as the metric with training data was almost perfect but did not work well with new data.

UNDER-SAMPLING AND SELECTION OF THE MODEL

We have seen how applying Grid Search with F1 as scoring, trying to maximize our Precision and Recall, has increased the metrics scores in the majority of the models but not as much as we need. So, as we continue having poor results, it is time to resample our data. In this case, the technique used will be UNDER SAMPLING to reduce our majority class (class 0 = non-diabetic patients). But the idea is not to balance our classes and take the same observations from both (50-50 balance); the idea is to reduce a few observations of the majority class to obtain better Recall and Precision results in our classifier models and to continue with a realistic data in which class 0 is the majority class by a big difference. It is also very important to apply the under-sampling only in the training data. The data is modified to improve the training of the model to then obtain better results. So, we cannot modify the test sample because it simulates a real set of patients in which we need to predict its value (0 or 1, non-diabetic or diabetic) to classify. So, we apply the under-sampling to the majority class (non-diabetic, class 0) in the training data and we reduce the training samples from 155501 to 73889 while we maintain the minority class (diabetic, class 1) training samples at 28078.

The next step is to decide which is the best classifier model trained with the Grid Search to then apply the under-sampling. To do it we have fixed some requirements:

1- The first requirement is to have an F1 score higher than 0.25 before under-sampling (logistic regression, decision tree, hard voting and random forest discarded)

2- The second requirement is to select a model without overfitting (KNN, soft voting and bagging discarded)

3- The third requirement is that when we maximize F1, Accuracy and Specificity scores are not highly decreased (NB Gaussian discarded)

After analyzing the requirements and the classifiers we have fitted, we conclude that the only valid option is to pick Adaboost or Gradient Boosting (the unique classifiers that follow the three requirements) as the final classifier to apply the under-sample then. As we are trying to maximize the Recall and F1 score (reduce the True Negative patients) and Adaboost has better Recall and F1 than Gradient Boosting, we will pick Adaboost.

Once we have selected the model, we fit the Adaboost classifier with the under-sampling, with the best parameters found in the Grid Search and we evaluate the model to compare it with the Adaboost fitted without under-sampling.

```
Evaluation of the Confusion Matrix with the under sample applied:

Accuracy :  0.8266042052511167
Precision :  0.4365454791187998
Sensitivity (Recall) :  0.46017951275110414
Specificity :  0.8927616009877559
F1 score :  0.4480510473019836

Evaluation of the Confusion Matrix without the under sample:

Accuracy :  0.8542760649308203
Precision :  0.5753299954483386
Sensitivity (Recall) :  0.18008263285368287
Specificity :  0.9760006173474637
F1 score :  0.2743055555555556
```

We have been so aggressive in the under-sampling because our main objective was to reduce the number of False Negative patients. False Negatives are patients that we are diagnosing as non-diabetic when they are suffering from diabetes. So, we cannot permit a lot of failures here because the error can cause death. Moreover, we have tried to not reduce the other good quality metrics we already have such as Specificity and Accuracy. Furthermore, we have tried to avoid getting a worse classification of the majority class 0 (we know the classification of the majority class will be worse but we have tried to control it as much as possible) when we improve the classification of the minority class 1. Finally, we also want to maintain the big difference in observations between class 0 and class 1 shown in the training data to maintain the fidelity of the dataset. In the results we can observe how all our objectives have been fulfilled; the recall has been improved from 0.18 to 0.46, the error between class 0 and class 1 has been balanced (similar number of False Negatives and False Positives), Accuracy and Specificity have not been highly decreased and we still have a big proportion of class 0 observations predicted in comparison to class 1 observations predicted.

CONCLUSIONS

We have been following a process to obtain more realistic and better results by proving all the classifiers trying to maximize the F1 score in the Grid Search and doing a controlled (not balancing 50-50) under sample ONLY in the training data. But, in order to confirm all our suppositions, let's prove what will happen if we balance the data 50-50. We will use the Adaboost classifier which is the classifier model that better fits our problem. These are the metrics we have obtained:

```
Accuracy :  0.7035189018411592
Precision :  0.3120721131903241
Sensitivity (Recall) :  0.7793132924918079
Specificity :  0.6898343450972322
F1 score :  0.44567564264472237
```

We can observe how it is true that the Recall has increased a lot (because the number of True Negative patients has been highly reduced) and now the model does not misclassify patients from class 1 as badly as it does without a 50-50 balance. Even so, other metrics have worsened. For example, Accuracy Specificity and the most significant, Precision, has been highly reduced, which means that now the model does not classify well patients from class 0 (the number of False Positive patients has increased a lot). The Accuracy is lower because the majority class (0) does not classify well. So, we conclude that yes, we know this is an option if the only thing we want is to maximize the Recall and improve the class 1 classification, but we think that it is not a good model because of the cost we have to pay with the Specificity, Accuracy and Precision is very high. What we have tried to compute during our projects is to maximize the lower value metrics without minimizing a lot of the good quality metrics we already have such as Accuracy or Specificity.

Another mistake we found by balancing 50-50, is that the predictions do not show that there is a majority class (0) with a big difference in observations from the minority class (1). In the 50-50 balanced data predictions, we have more predicted observations from class 0 than from class 1, but the big difference shown in the training data is not represented, so we are not following the fidelity of the training data. This means we will not have a realistic model that explains well what the training data represents if we balance 50-50. Otherwise, with the under-sampling predictions, we will continue having imbalanced data as in training and as in reality, but with an improved classification of class 1. The barplots show what we have explained.