1 Introduction

2 Preprocessing

NaN Values NaN values need to be treated. In this way, two solutions are possibles, removing rows containing NaN values or replacing them. In order to avoid loosing too much rows, replacing the NaN values by the mean of the column is then considered. However, not to lose too much time, filling NaN values is applied only on the columns needed and not on the whole dataset.

Categorical features In this dataset, some features have been discretized as low, medium and high. Most of the machine learning algorithms do not accept such features. Then, a label encoding is needed. Features containing low, medium or high take, respectively, the values 1, 2 or 3.

Features selection After a quick look at the dataset, one can easily see that a lot of features are completely filled of zeros (also because NaN values are replaced by the mean of the columns, which gives 0). Then, those features do not give any additional information that would be useful to take a decision for the output. Then, they are dropped and the dataset is now containing about 25000 features.

To reduce the number of features, it is also possible to use the correlation between each features. In this case, this is impractical to compute the correlation for the whole dataset since it would require too much computational time. Then, one possible way to observe correlation between features is to sample per packet of 500, randomly the features with those it is now possible to compute the correlation. Then, by repeating a thousand times this procedure, it is possible to reduce the dataset size to ≈ 13000 features.[1]

Normalization To improve the performance of the classifiers that will be used, a normalization is performed, such that for each columns, the minmum value is 0 and the maximum value is 1.

3 Balanced classification rate (BCR)

As Figure 1, the training set is imbalanced. Then, to evaluate a classifier, it is better to use the "Balanced classification rate". To estimate the BCR on the test set, it is evaluated on the training set a bunch of time. A confidence interval of 95% is computed. In this way, the expected BCR on the test set should fall in this confidence interval and an estimation is then possible.

[1] It requires to setup a threshold (= 0.8), such that when the correlation between two features is above, the first feature is dropped. There is no additional computation done to choose one or the other feature to drop.

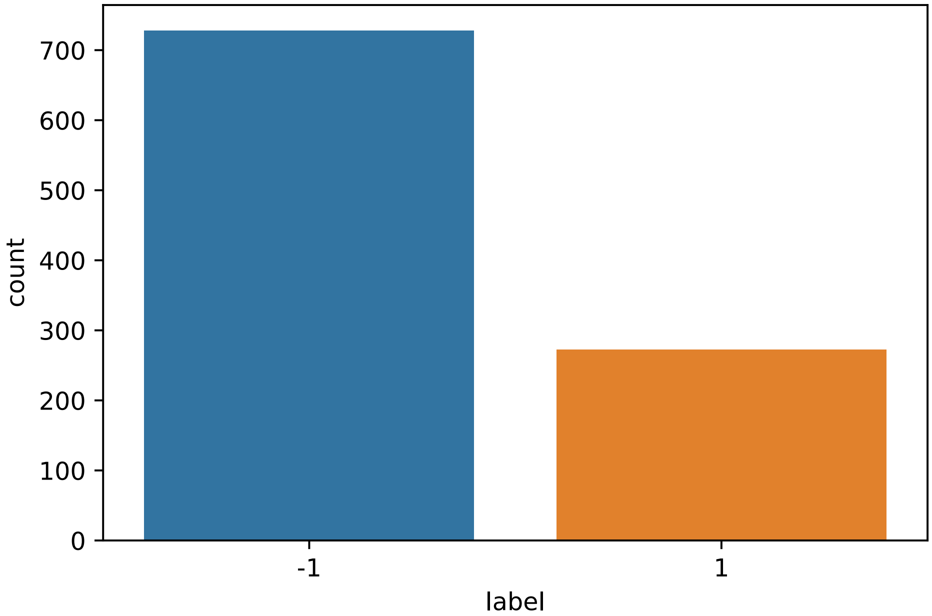

Figure 1: Dataset's classes

Figure 1 shows that the dataset is totally imbalanced. In this way, it is why the criterion used to evaluate the prediction of the classifier that will be used, is the Balanced classification rate.

4 Prediction on the test set

Creating balanced training set As seen before, the whole training set is imbalanced. Since the goal is to achieve the best performance in term of accuracy (for this section), it is better to train classifier on a balanced dataset. Then, a balanced training set is built from the original dataset by randomly picking 272 (number of tumor values) ”healthy” inputs.

Classification Now that balanced training set is built, it is possible to select a classifier to try to predict the labels of the test set. One can easily notice that the AdaBoostClassifier from sklearn is the one that perform the best on the training set.[2] AdaBoostClassifier returns the output of a set of DecisionTree. In this way, one can imagine that it would be better to exploit the amount of data that is neglected when the balanced dataset is built. Then, many training subset (balanced) are built and each of them serve a specific DecisionTreeClassifier. In this way, more data of the original training set is used and the output results on whether the sum of all the decision trees are positive or negative.

[2] It requires to train many classifiers on a balanced training set and evaluating the accuracy. It can be done several times, to get different datasets and then observing the robustness of the classifier.

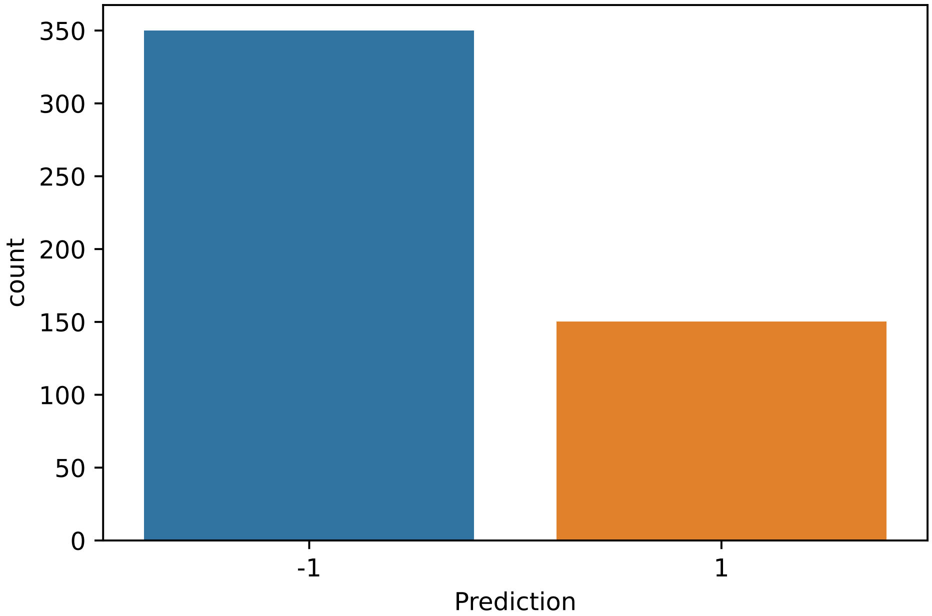

Figure 2: Prediction on the test set

5 Ways of improvements

Features selection Feature selection could be made differently. For example, K-Means could be applied on the dataset to select only ”K” features that would represent the centroids of the clusters. This becomes an unsupervised learning problem and the inputs are the features (the columns) and not the input values (rows) anymore.

Prediction

- It is possible to improve the prediction. For example, if the sum of the outputs of the decision trees is close of zero, additional trees could be needed to be more dividing.
- Another classifier can be used, even if, empirically, decision trees are one of the best performers on this set.