Qualifying Round Task - Loop Q PRIZE

Overview of the solution

The given dataset is a collection of data taken from the scrumpulicious production chain (actually is not properly a chain since there are 60 parallel widgets overviewed by one oompa-loompa each). The final color of a candy is supposed to be dependent by the parameters achieved during production. Mathematically speaking, we are facing a probability distribution over the 60 variables, where we wish to estimate P[color=”GREEN” | x1,…,x60]. Obviously knowing the exact distribution we can predict whether the in-production candy will be GREEN or not, and how confident we are with the prediction. Unfortunately this distribution is unknown, and we are supposed to estimate it, and in these cases, common solutions are constituted by machine learning algorithms. The very big issue of this task is the fact that the dataset is very unbalanced: there are just 102 green candies on 15806 total. This leads on thinking about solutions that are less affected by unbalanced input data, and not less important, a score function that tells us how well we are proceeding. The choice is fallen on the xgboost classifier, and the f-score score function, that tells more information than a “simple” accuracy function: since the unbalance of the data, supposing to answer always that the output candy will be golden, we will obtain over 99% of accuracy, while we are not learning anything from data. The details about the solution are exposed in the next section.

Solution design criteria

The solution I adopted is a classical machine learning framework, that leads to estimating as best the probability distribution. The problem is expressed as an optimization problem, minimizing a cost function, or, in this case, maximizing the score function, which is the f-score function. Now I will explain the sequential tasks that compose my solution.

1. Reading the dataset, and consequently splitting into input variable and output variable (the last one). In my solution I had ignored the timestamp.

2. Splitting the dataset into train and test set. The first one is used to train my model and the second is used to estimate how well my solution goes on new data.

3. Train the xgboost classifier and estimate the best hyperparameters exploiting bayesian optimization, that given an input search space, it gives the parameters that give the best score when training the classifier with them. Actually it is just an heuristic searching algorithm, but it gives the best solutions with respect to grid search and random search that I have also tried before. The training is performed adopting 5-fold cross validation in a way that at each fold the proportion of label GREEN and NOT GREED are respected.

4. Finally, after trained the best classificator, I tested on test data to obtain an estimation on how well it performs on new data.

Strength of the solution

The final solution on test data gives an f-score of 0.7843, which is a good result with respect all the results I obtained in previous tests. The strength of my solution is that I obtained a model that predicts with an adequate confidence whether the candy will output green or golden, facing the issue of having just some examples of green candies. Other solutions may be alternative of mine, for example exploiting undersampling and/or oversampling, but in this case I discarded oversampling because there are very few green examples, so new estimated green candies may be affected by too noise.