QUESTION 3 – REPORT

Martin Labenne

Answer

Methods

The task here is to propose an algorithm that detects mislabeled observations in a binary classification dataset. The issue with label noise is that it harms a lot of classification algorithms because introducing label noise make the model more complex and will more frequently fail to predict good labels. Ensemble classifiers respond to this problematic by combining the predictions of several base estimators built with a given learning algorithm to improve generalizability and robustness over a single estimator. Procedures that involve perturb-and-combine techniques are very interesting since it allows a variance reduction: diverse set of classifiers is created by introducing randomness in the classifier construction, the prediction of the ensemble is then given as the averaged prediction of the individual classifiers.

Because mislabeled samples are harder to correctly classify, I trained an ensemble of classifiers using a subset of training data, predict the labels of the rest of the data using them, and then the percentage of classifiers that failed to correctly predict a sample's given label is the probability that the sample is mislabeled. To do this, I used a Random Forest ensemble procedure retrieving the Out-Of-Bag decision function values to estimate the generalization score and deduce the mislabeled probability of a sample. From this step, I compared each sample mislabeled probability with a boundary value (hyperparameter) and conclude if the sample was initially mislabeled.

To test this algorithm, I generated a classification dataset with the *make_classification* function from scikit-learn, which introduces interdependence between the features and adds various types of further noise to the data. The data set is composed with 1000 samples and 20 features (10 informative, 5 redundant, 1 repeated and 2 random noise). Figures 1 shows the features mean and variance which Figure 2 shows the correlation matrix of this dataset.

Since we want to maximize true positives and minimizes false positives, we basically want to maximize Receiver Operating Characteristic (ROC) and therefore to maximize the Area under the ROC curve (AUC) across all possible boundary decision for each mislabeling proportion.

Results & Discussions

Run the algorithm on Simulated Dataset.

The dataset I simulated is interesting because of the redundant, repeated, and noisy features. I simulated the dataset once and repeated 5 times the following process to have an idea of the variability of the algorithm. First, I split the dataset into train and test splits with the 80/20 rule, then for each value of mislabeling proportion (cf. *Figure 3*) I mislabeled the data and selected the best hyperparameter which maximized the AUC score for the training set. I finally recorded the AUC score performed by the algorithm

with the best hyperparameter. You can find the results on *Figure 3*. Median values of the AUC score are really good, from 0.93 for a 5% mislabeling proportion and do not drop so far for 10% mislabeling: approximately 0.90.

I explored the dataset provided, it has 3901 samples, 54 features, no null values, and no constant features, it is full of standardized features and have some correlated features. I applied the algorithm one with an hyperparameter of 0.675 since the simulated data set had similar correlation force across features (cf. *Figure 4*) and 0.675 is a very good value for the hyperparameter since it provided good scores across all mislabeling proportions.

I could have estimated these results from bootstrap procedure and have a confidence interval, but it was too computably demanding for my computer, so I ran the algorithm once. The error rate was 3.5%.

The algorithm did not perform equally well for both classes, but still, the scores are pretty good. True positive rate of class 0 is 0.98 whereas class 1's is 0.95. It could be because the dataset is a bit unbalanced in favor of class 0 (2127 samples) against class 1 (1774 samples). Also, the algorithm is trained to maximize true positive rate and minimize false positive rate on class 0, this way, the algorithm could be better at predicting class 0 outcome.

Figures

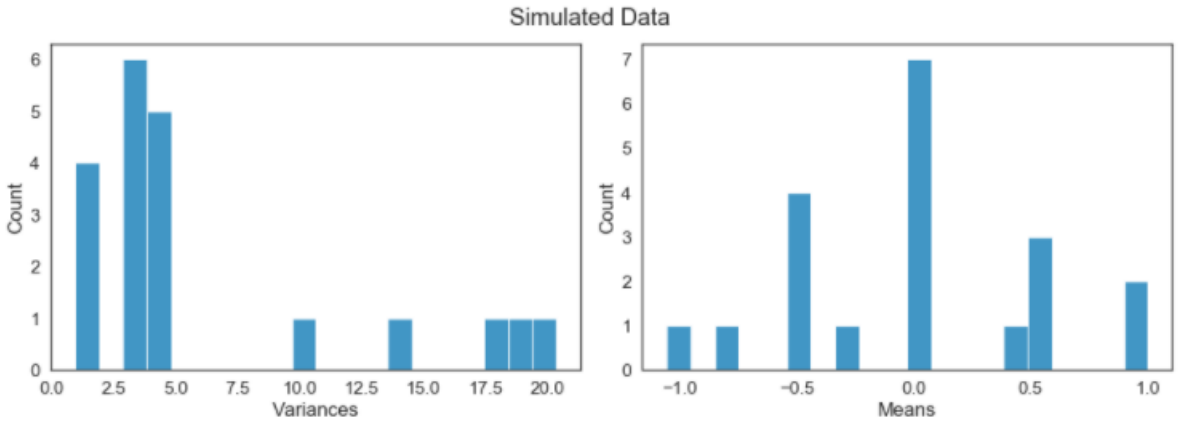

Figure 1: Mean and variances of the features from the simulated dataset.

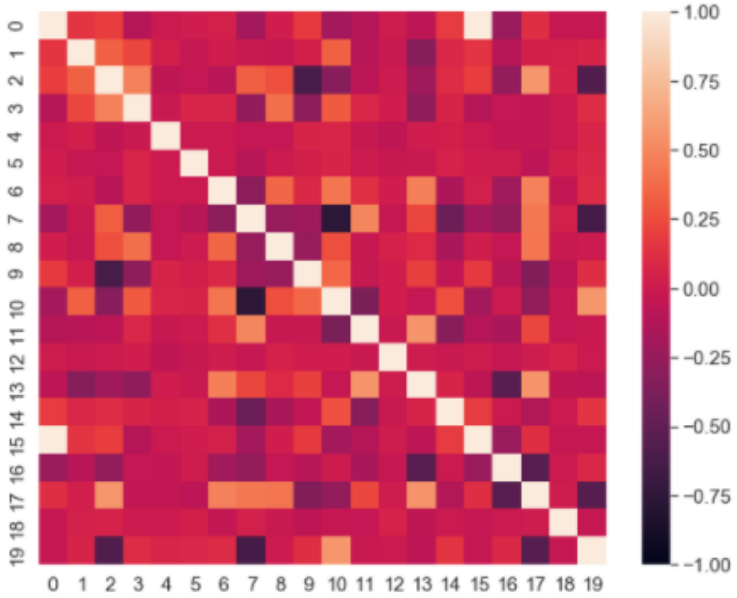

Figure 2: Simulated dataset correlation matrix

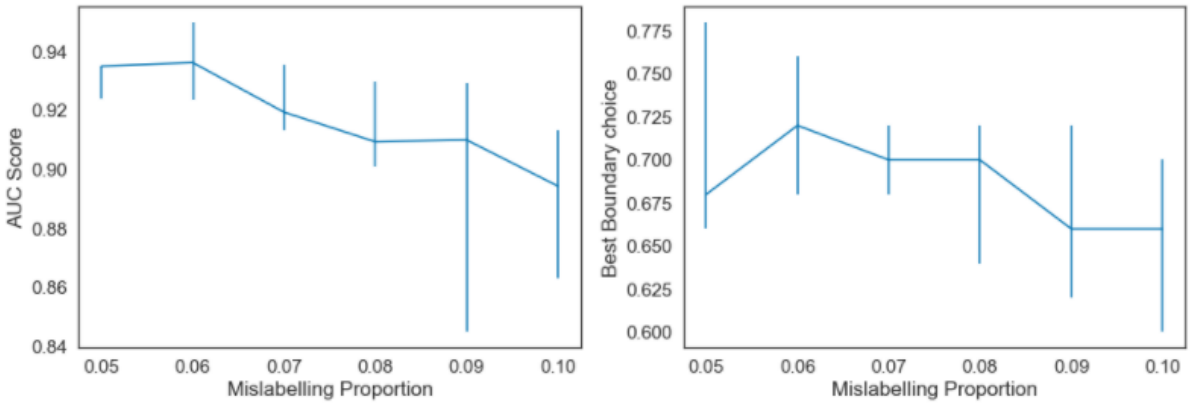

Figure 3: AUC Score and Best Boundary choice given the mislabelling proportion. The solid lines show the median value of the indices across the number of outer runs. The bars show the range of values achieved for that cluster count.

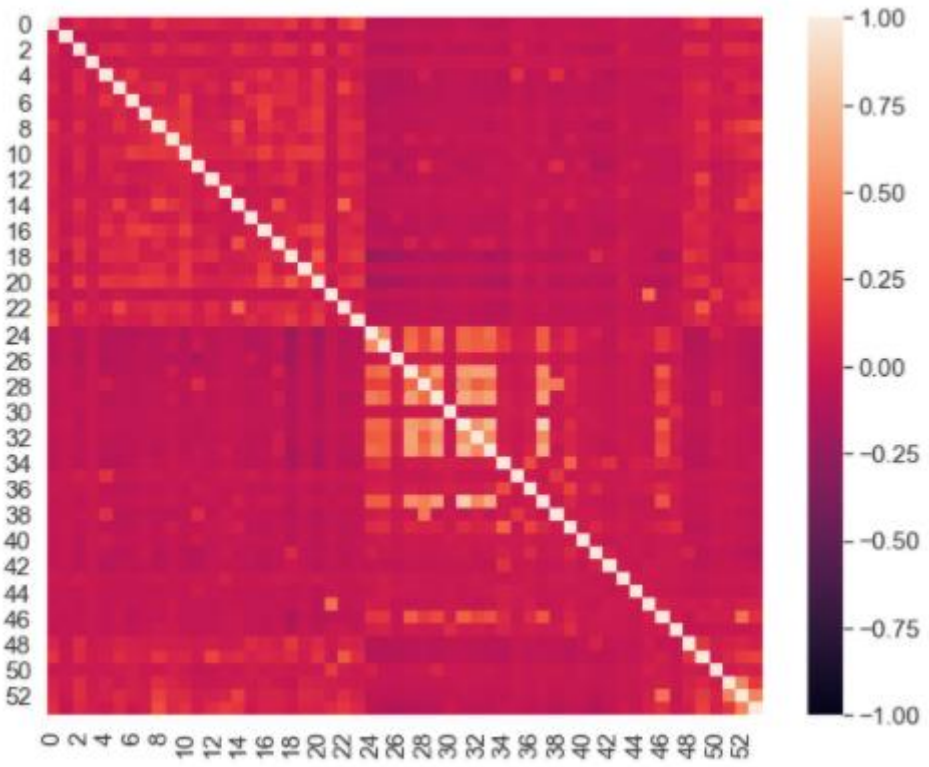

Figure 4: Correlation matrix of the provided dataset. Notice the group of correlated features in the center.