Sofía García del Barrio Cervera
EN.580.488 – FCBB
02/18/2022

P03-P1

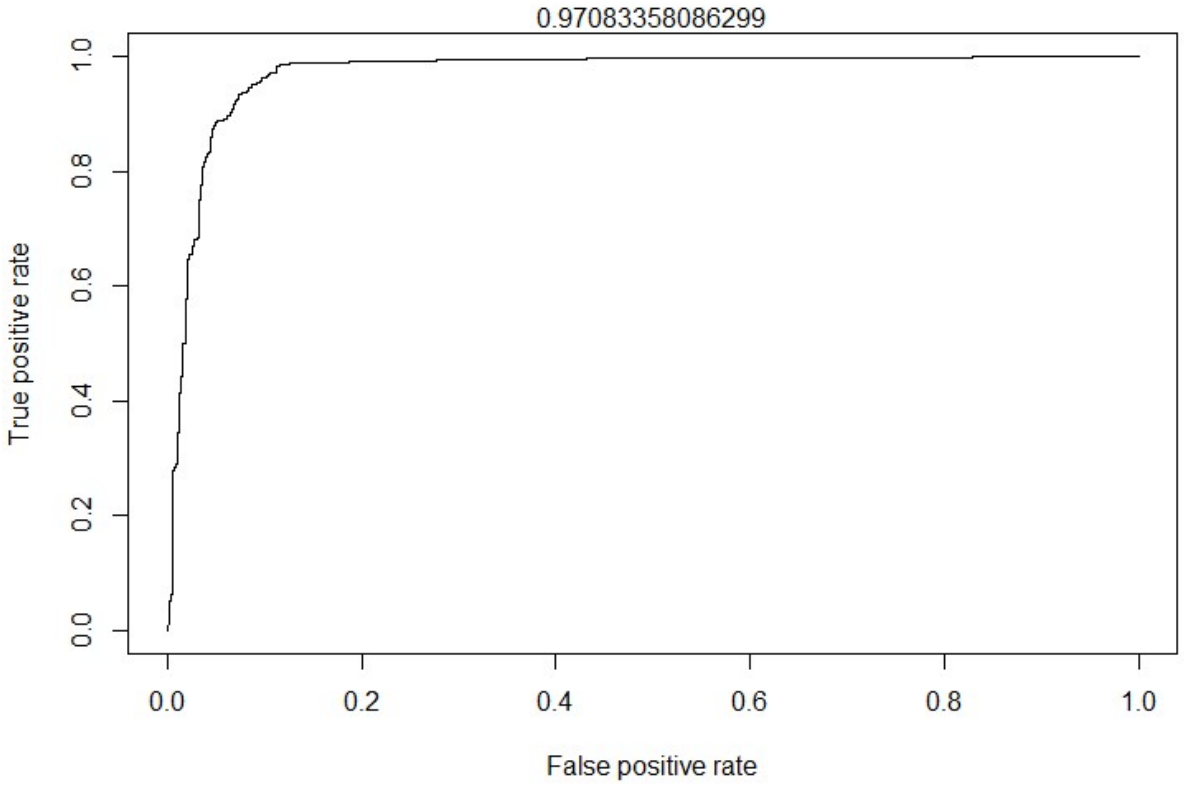

Figure 1: ROC curve of the SVM's performance and AUC value

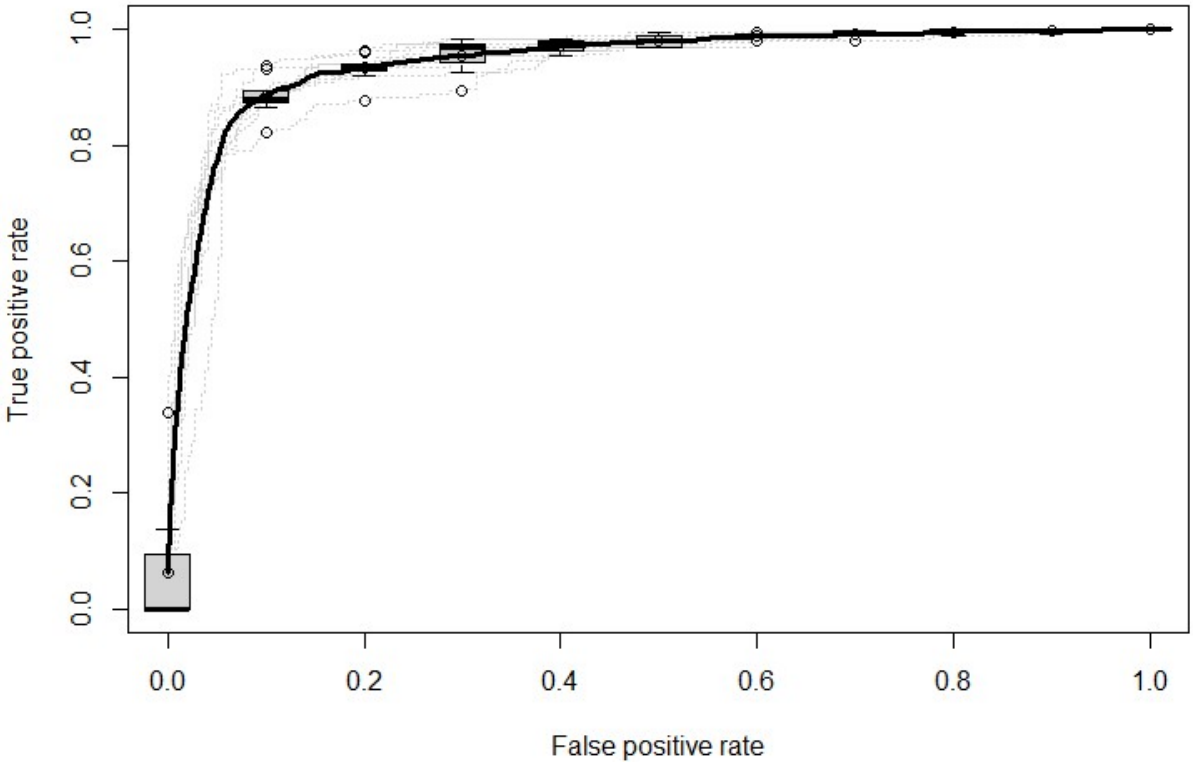

Figure 2: average ROC curve and BOX plot for 10-fold cross validation

P03-P2

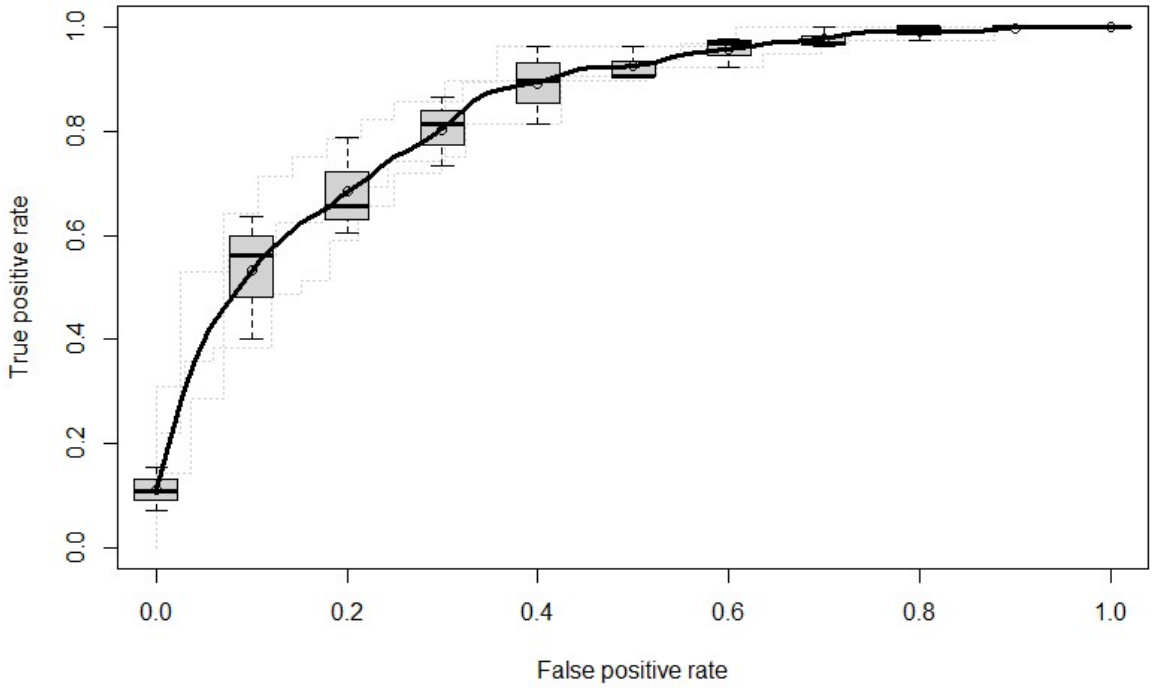

Figure 3: ROC curves of the SVM's performance for 3-fold cross validation

P03-P3

Both Perceptron and Naïve-Bayes achieve 0 error, so their accuracy for this specific dataset is optimal and equal.

Perceptron takes 6 epochs (iterations) to achieve zero error. From the graph, it can be observed that the code is learning, changing the weight factor in each iteration until it finally gets a perfect separation of the data. Once it finds the weight factor that work with no error, it does not change anymore.

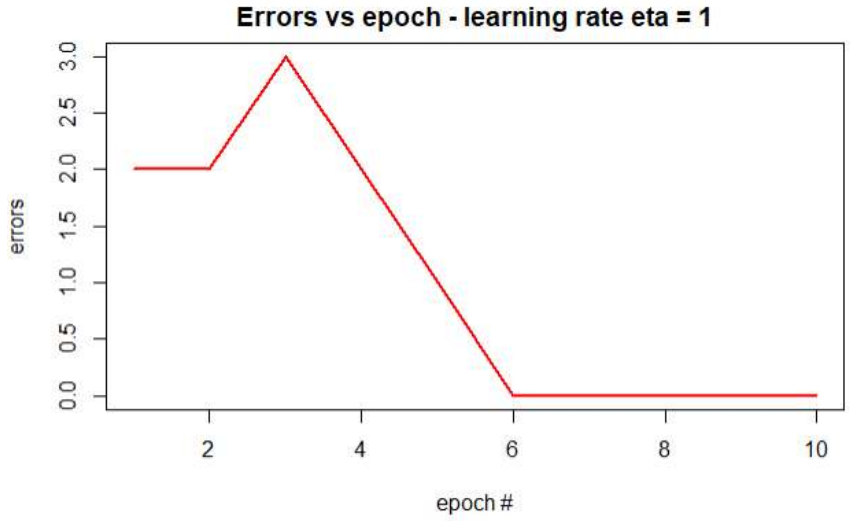

Figure 4: Errors vs epoch plot for Perceptron with eta=1

The results are from the perceptron function are the weight factor (first row) and errors in each iteration (second row). I get the following:

[1] -0.040 -0.068 0.182
[1] 2 2 3 2 1 0 0 0 0 0

Naïve-Bayes also achieves a perfect classification. We can see it in the table, shows zero false positive or negative results. We can also plot the ROCR curve, which has an AUC of 1. It means that all the predictions are true.

	setosa	versicolor
setosa	10	0
versicolor	0	10

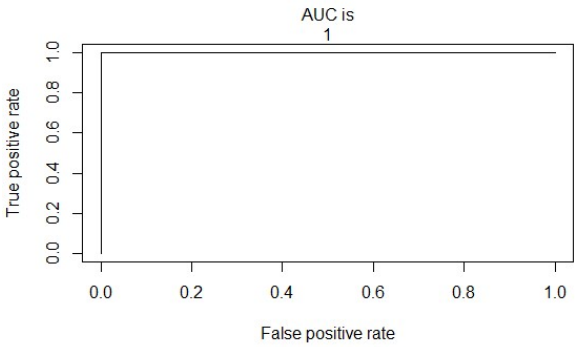

Figure 5: Naïve-Bayes results. Truth table (left) and ROCR curve with AUC value (right)

The dataset we were testing only had 20 samples, which is a low number. It is relatively simple for a method to achieve a perfect classification, so both Perceptron and Naïve-Bayes do so. However, if the dataset was larger, maybe Naïve-Bayes would not have worked as well. As Perceptron learns from each iteration and corrects the weight factor, we could predict that it might work better with large datasets.