

# Assignment report,

Machine Learning

MSc Applied Bioinformatics,

Marie Schmit

# Abstract

Data exploratory and analysis

1. Enose samples are separated by their sensory scores. Sample 10F9 can be considered as an outlier. HPLC samples are less clearly clustered by samples, but three close groups still emerged. There are some potential outliers like 0F12 or 5F6.

Table PCA scatter plots of enose and HPLC grouped by sensors class

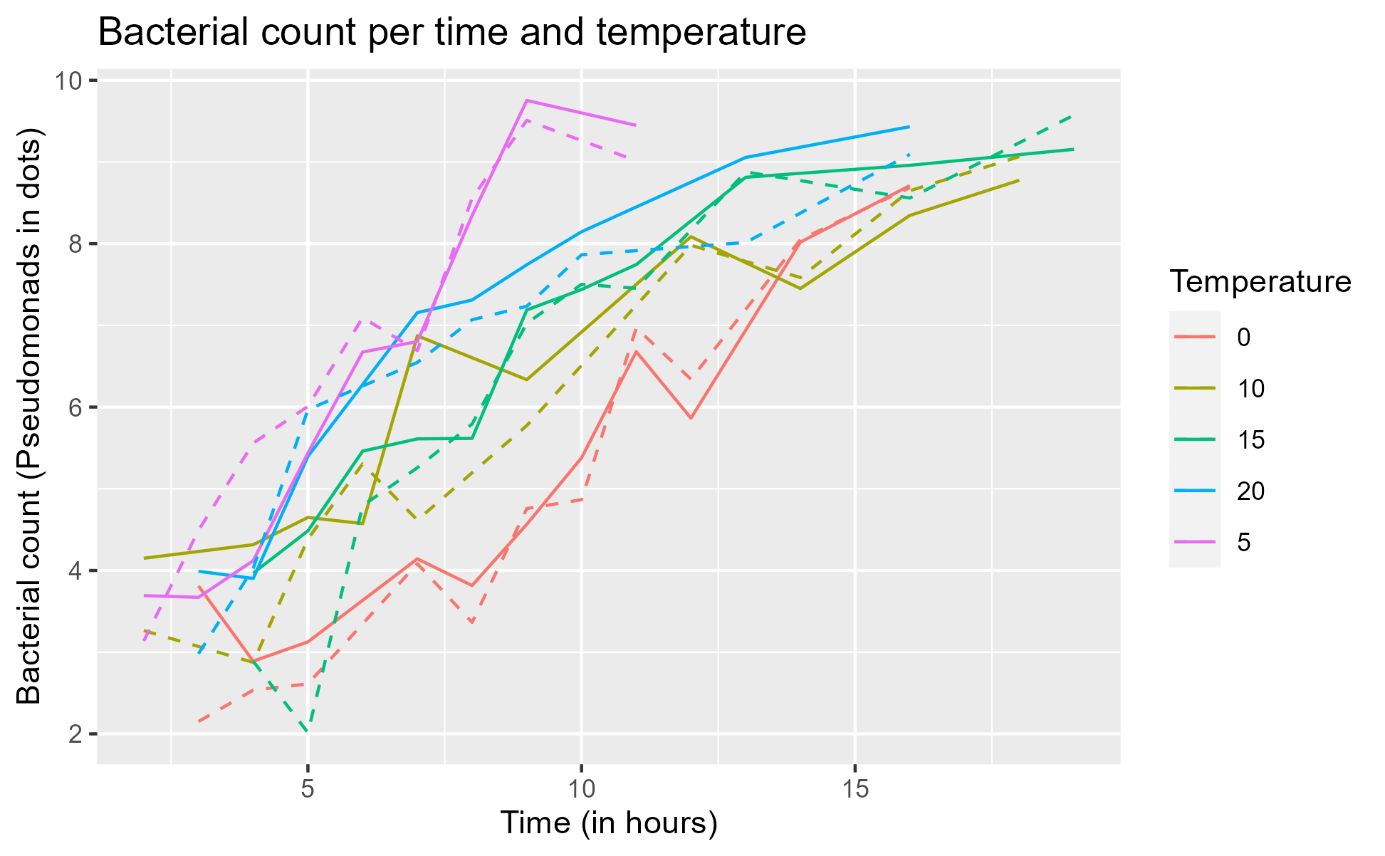
|  |  |
| --- | --- |
|  |  |

We try other analytical method to find better separation. First, we display the PCA plot with other methods (in 3D, in a biplot). We also use HCA. With this method, three clusters emerge, but they do not necessarily correspond to their sensory value. The previous outlier for enose (10F9) is no longer an outlier: it is now in the same cluster as F1a, which does not correspond to its sensory value. The same goes for HPLC, for which the clusters 2 and 3 are often mixed. This analysis is not the best.

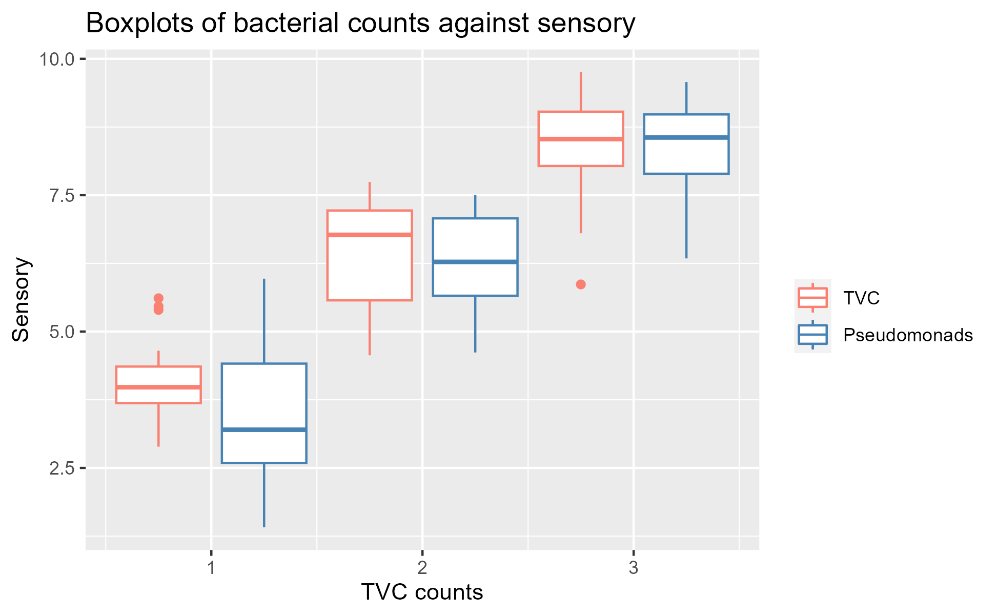
|  |  |
| --- | --- |
|  |  |

We decide that the data that seem to stand out are not sufficiently so to be considered as outliers. We choose not to remove them to avoid the risk of introducing errors if they are significant.

1. The number of bacteria, for both type of bacteria, increases with time. There are more bacteria for higher temperatures. The count of bacteria is abnormally high for 5 degrees. We also have less samples for this temperature. The number of bacteria seems to grow with time and temperatures increase.



1. The number of bacteria is for both types TVC and Pseudomonias higher when the sensory score is higher. Rotted meat has thus more bacteria than fresh one. TVC bacteria are slightly more numerous than pseudomonas. Also, TVC has three outliers.



Classification

We partition the data using the function createDataPartition, that ensure a balanced representation of our train and test sets. We make the partition reproducible by using the function “seed” and use the created index to create the datasets.

We first train the model using k nearest neighbour methods. We want to find the best fit and k parameter. To find the best model, we test different methods of scaling (center, auto scale and range scale), for different values of k (from 1 to 20). We evaluate the accuracy of the model for each of those scales and k values and we select the parameters leading to the best accuracy.

For HPLC data, the results are the following with one partition. Without scaling, the best accuracy is 0.9 for k =7. With scaling, the best combination is no scaling with various k values (different k give the same accuracy, we choose k = 7). Scaling was not necessary here. Our tunned model is thus: no scaling, k=7.

Table Cross table for HPLC data, 1 iteration, k = 7, no scaling

Calendar

Description automatically generated with low confidence

We use the previous model on 100 iterations, and save the accuracies values in a dedicated list. Each of those values is used to calculate the cumulative mean accuracy.

The process is repeated for enose data. The best accuracy is 1 for non scale data, k=3. This indicates and overfitting. However, all the accuracies values are not equal to 1. We reduce the number of train by changing the partition from 7:3 to 5:5. The accuracy is now 0.83 for k=6, without scaling and 0.88 for k=8 with center scaling. We save the cumulative mean accuracy of the partition with 7:3. We train the model on 100 iterations with k=3. The mean accuracy for 100 iterations is 0.791. The best accuracy of 1 is maybe due to the little dataset with which we work.

We apply the svm-rd classification method to both data sets, trying different kernels: 'vanilladot', 'polydot', 'rbfdot', 'tanhdot'. For HPLC, the best accuracy is 0.7, for the kernel rbfdot. We chose this model for the training with 100 iterations. The mean accuracy for all the iteration is 0.789. Concerning enose data, the best accuracy value is 0.9 for the same kernel rbfdot. This accuracy is again very high. For 100 iterations, the cumulative mean accuracy is 0.789, too.

For random forest classification, we first define the taskwith which we work, with the predictor “sensory. We inspect the frequency of the class and set the learner to random forest. The hypoparameters that be optimise are the following: "ntree", "mtry", "replace", “classwt", etc. We chose to tune ntree which is the number of trees in the forest; mtry, the number of features to sample at each node; maxnodes, the maximal number of leaves; nodesize, the minimum number of cases authorised in a terminal node (leaf). We define the lower and upper values ofr the hyperparameters in a dedicated space (with the lrn function). The number of features randomly sample at every nodes is lesser than the number of features but still large so trends of the data are analysed. The nodesize has to be small enough to avoid having a tree that is too big, but large enough to avoid underfitting. Our methods for evaluation are resampling strategy cross validation and performance measure classifier accuracy. Since we cannot evaluate every hyperparameter, we choose to stop tuning after 20 evaluations. We perform the tunning with grid search, that evaluates each combination of hyper parameters.

For HPLC, the best random forest parameters are 200 trees, 2 feature sample at each node, 20 nodes per leave maximum, 2 allowed cases in a leaf minimum, with an accuracy of 0.84.

Table Confusion matrix for HPLC tuned model

Text

Description automatically generated with medium confidence

We test this model for one iteration. The accuracy is 0.7. One sample of sensory 1 and one of sensory 2 are misclassified.

Chart, bar chart

Description automatically generated

We train the model for 100 iterations, the mean accuracy for all iterations is 0.813.

We repeat the exact same steps for enose data set. We set the maximal number of features to sample for each node to 8 since enose has 8 sensors. The best model has 200 trees, 2 features to sample per node, 20 leaves maximum, 2 cases allowed in terminal node. The accuracy for this model is 0.731.

Table Confusion matrix for one iteration for tunned model, enose data

Text

Description automatically generated

Two elements are misclassified: one for sample one, the other for sample two.

Chart, bar chart

Description automatically generated

We train and test the model with those hyperparameters, for 100 iterations. The cumulative mean of all iterations is 0.745.

1. We calculated cumulative mean accuracies for every dataset and classification method and create a plot for each dataset (enose of HPLC).

Chart, histogram

Description automatically generated

**For enose data, knn has the higher cumulative mean accuracy**. It is very high for a few iterations, then drops and get higher again. Svm has a very similar pattern. Random forest has a very poor mean accuracy at the beginning, but it gets higher with more iterations.

Graphical user interface, chart

Description automatically generated

**For HPLC data, random forest has the higher cumulative mean**. Knn is very high for a few iterations, but it drops rapidly to approximatively 0.7 accuracy.

1. For each algorithm, we calculate the number of misclassied (false positive or false negative number) for each iteration and sum this number up. Each of those sums is stored into a vector containing three element (one sum per class). We also calculate the sum of the proportions for 100 iterations and store the results in a similar vector (see all the plots in folder “Plots”.

For enose, rf and knn have the smallest proportions of misclassification, with almost no misclassifications (2.5 in total proportion) for the class 1 for knn (the “true” class, corresponding to fresh meat). For HPLC, it is rf that has the smallest numbers for all the classes (with maximum 10% of missclassifications for class 1), but svm has the smallest number of misclassifications sum of proportion (approximatively 2).

|  |  |
| --- | --- |
| Chart, bar chart  Description automatically generated | Chart, bar chart  Description automatically generated |