Ryan Laurents

Data Mining – CSE 4334

P2

11/15/2020

CLASSIFIERS

This document serves as a readme to explain my process for selecting the classifier and parameters used by said classifier.

**INITIAL TESTING/CHOOSING A CLASSIFIER:**

The sklearn package contains many different classifiers, although we only covered 4 in the introduction to this assignment. Before I chose one to create my model I wanted to make sure I was choosing the best option. Below is a list of classifiers I tested prior to making my decision.

1. KNeighborsClassifier()
2. GaussianNB()
3. SVC()
4. LinearSVC()
5. DecisionTreeClassifier()
6. RandomForestClassifier()[Unresolved bugs, moved on after initial bugfixes failed]
7. MLPClassifier()

In the end, I went with the **MLPClassifier()** as it was outperforming other models by

10%-20% on average. There are two more classifiers offered in sklearn that I did not try, AdaBoostClassifier() and QuadraticDiscriminantAnalysis(). The only reason I did not try these is because I am previously familiar with using a neural net model such as MLPClassifier and it has already outperformed other options.

**REFINING:**

**FINAL CLASSIFIER: MLPClassifier(activation = 'logistic', solver = 'lbfgs', alpha = 1.0, tol = 0.01, max\_iter=10000)**

The MLPClassifier() has many parameters to tweak to increase accuracy. To compare these variations, I went through each parameter and tried every available option. To ensure I wasn’t getting any outliers, I created a loop to average out the accuracy of 50 iterations for each variation. Based on the best results, I would continue forward with the “winner”, moving through all the parameters.

There is one specific parameter we use which happens to nullify most of the parameter options for MLPClassifier. This is solver = 'lbfgs'. The other options available for this are ‘adam’ and ‘sgd’. The reason we chose ‘lbfgs’ is because it is optimized for small datasets(sub 1000 samples), while ‘adam’ and ‘sgd’ work better for larger datasets. As mentioned before, many parameters only apply to the ‘adam’ and ‘sgd’ solvers, so we are not able to use them.

As for the other parameters, I chose each one based on testing outcomes. Activation

describes the activation function for the hidden layer. Logistic (the logistic sigmoid function) returns f(x) = 1/(1 + exp(-x)). The alpha parameter is the L2 penalty (regularization term) parameter. Default is .0001 but we found better results with 1. Tol is the tolerance for optimization. When the loss score is not improving by at least tol = 0.01, convergence is considered reached and training is stopped. When testing with smaller tolerances, convergence is not reached in a reasonable amount of iterations. Which brings us to our last chosen parameter, max\_iter = 10000. This is simply to ensure we don’t take too much time for calculation. Most of our iterations are in the 500-1000 range when tol stops learning, but I wanted to make sure we account for the outliers.

**FINAL THOUGHTS:**

An average accuracy of 60-65% is okay for this size of dataset, but there are probably several ways to increase that. As mentioned in the tips section of the assignment, there could be some unimportant attributes in the data that could be removed such as Age. I chose not to tweak the data at all in my testing, as that could be a very deep rabbit-hole. If I were to continue tweaking, I think I would consider which attributes could be removed from the data, and which attributes could be normalized [such as time played] so it is less impactful on the results while still being considered.