Sean Toon

IT-460-18500-M01 Machine Learning 2024 C-5

10/27/24

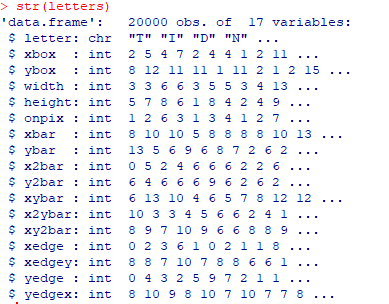
Module Seven Lab – Artificial Neural Networks

The dataset that I am going to be working with for this lab consists of 20000 examples of 26 English alphabet capital letters using 20 different randomly distorted and reshaped black-and-white fonts. First, I explored and prepared the data:

I started by importing the data with the following command:



And then viewed the data with the following command:



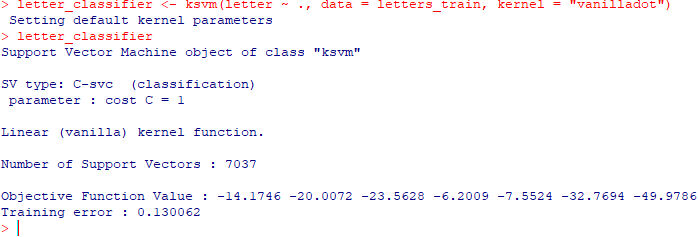
Because every feature is an integer, I did not need to prepare the data any further. I then created training and testing sets with the following commands. The training set uses 16000 of the 20000 entries:



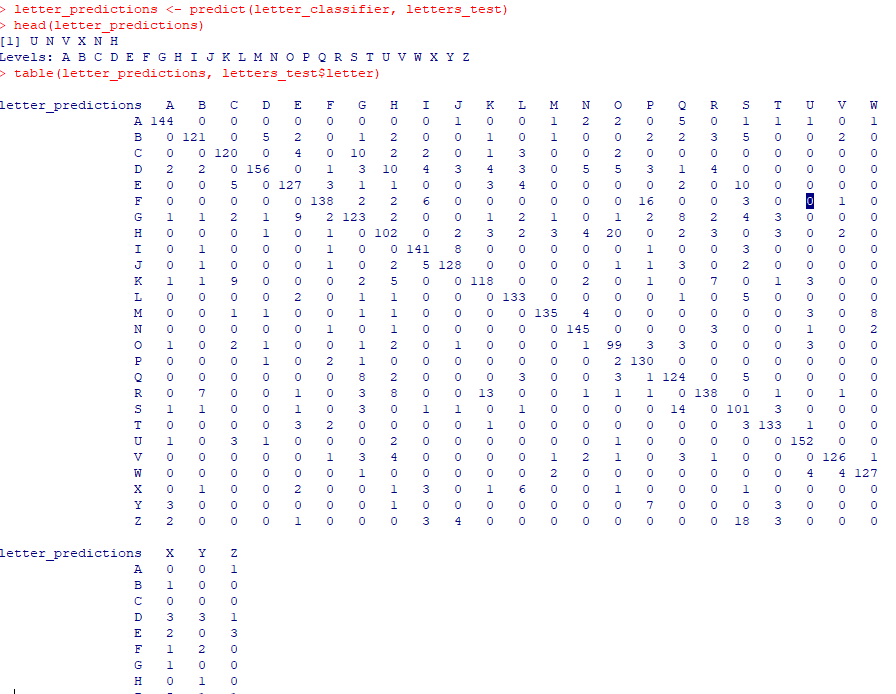
And the testing set consists of the remaining data:



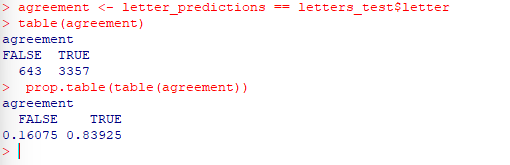
Next, I installed the kernlab package to use the ksvm() function to specify the linear kernel with the following code:



Next, I evaluated the model performance to get a better idea about how well the model would perform in the real world. I used the predict() function and was able to get the first six predicted letters U, N, V, X, N, and H. I then compared the predicted letter to the true letter using the data set while using the table() function to get a better visualization:



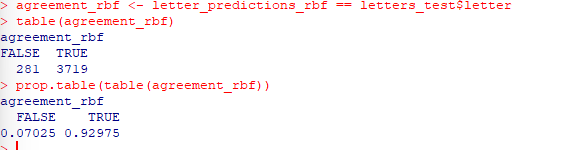
By returning vector values of true or false (whether the value matches the test data set) I can see that the classifier correctly identified 84 percent of the set:



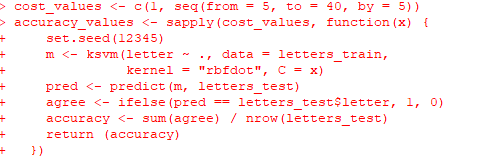
To attempt to improve the model performance, I first changed the SVM kernel function with the following code. I began with the Gaussian RBF kernel using the ksvm() function:

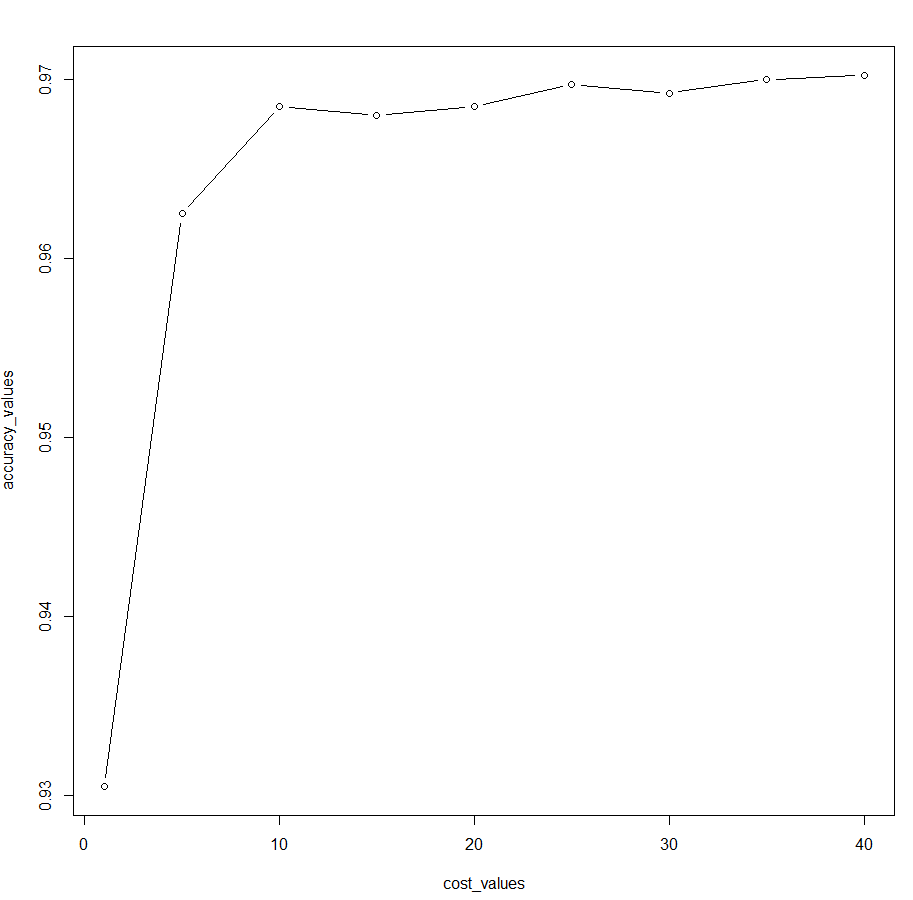


Then compared the accuracy with the linear SVM using the following code and I was able to see that by changing the kernel function I was able to increase the accuracy of the model from 84 percent to 93 percent:



Next, to try and increase the model performance even further, I identified the best SVM cost parameter. I used the sapply() function to apply a custom function to a vector of potential cost values. The seq() function to generate the vector as a sequence counting from to forty, by five and the plot() function to help visualize the result:





By identifying the best SVM cost parameter, I was able to increase the accuracy of the model to 97 percent!

References

Lantz, B. (2019). *Machine Learning with R* (3rd ed.). Packt Publishing.

https://mbsdirect.vitalsource.com/books/9781788291552