kNN CPSC 6430 Report
Lake Summers

Problem Description

Given a data set representing capacitor quality control testing data, develop a kNN classification algorithm that will determine if a capacitor will fail or pass quality control.

Data Description

The initial data was a set of 118 examples showing capacitors failing and passing the quality control tests (Figure 1). Each record had three tab-separated entries. The first is a float representing the results of one test, the second being the result of another test. The third is either a 1.0 if the capacitor passed QC and a 0.0 if it failed QC. The data was then split up into two data sets: a training set of 85 examples, and a test set of 33 examples.

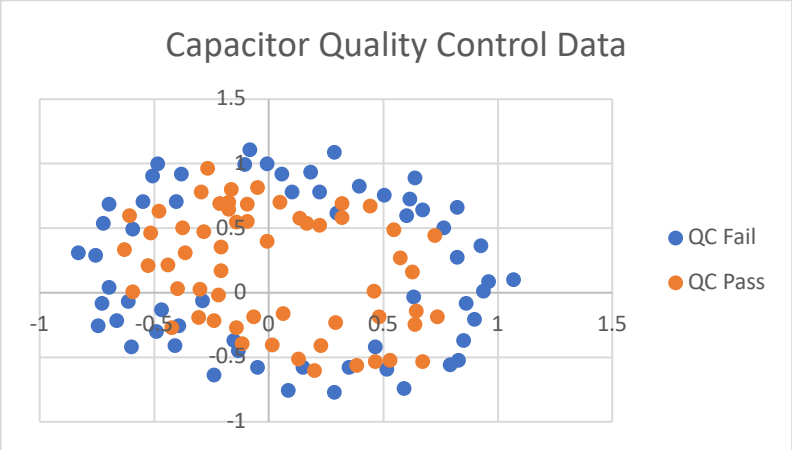

Figure 1

Training a kNN Algorithm

A k Nearest Neighbor algorithm was developed using 5-fold Cross Validation. The 85 examples of the training set were split into 5 folds of 17 records each. The five folds were used to create five smaller training sets of 4 folds (68 records) each, with the leftover fold (17 records) in each case used as the validation set. Each training set was then executed via kNN with odd values of k of 1-21. For each value of k, the number of misclassifications were recorded for all five training/validation set combinations (Figure 2)

k	1	3	5	7	9	11	13	15	17	19	21
Test 1 Error Count: Train 1234 and Test 5	8	5	8	10	10	9	7	7	8	9	10
Test 2 Error Count: Train 1235 and Test 4	9	6	3	2	5	5	6	6	8	7	11
Test 3 Error Count: Train 1245 and Test 3	7	3	3	4	4	7	5	5	7	8	7
Test 4 Error Count: Train 1345 and Test 2	7	2	3	3	3	3	3	5	7	7	7
Test 5 Error Count: Train 2345 and Test 1	8	6	4	4	5	5	6	6	6	6	8
Average:	7.8	4.4	4.2	4.6	5.4	5.8	5.4	5.8	7.2	7.4	8.6

Figure 2

From this data the cross-validated accuracy was plotted for each value of k (Figure 3). k=3 provided the best accuracy and was therefore chosen for kNN for the test set.

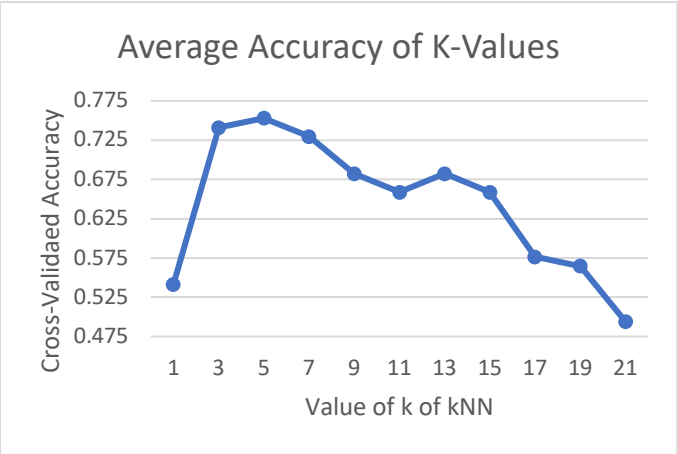

Figure 3

		Predicted QC Result	
		N	Y
Actual QC Result	N	TN = 10	FP= 6
	Y	FN = 6	TP = 11

Figure 4

Results

A confusion matrix for the results of the kNN algorithm with k=3 is shown in Figure 4.

As previously stated, the test set consisted of 33 examples. 16 were failed QC and 17 were passing QC scores. 22 of the 33 examples were correctly identified for an accuracy of 63.6%. Precision was equal to .6363, and recall was .647. The overall F1 score was .647. This lack of accuracy and lower overall F1 score is likely due to how the data is group, and kNN having a difficult time discerning between positive and negative along the border between the 2 classes of data as shown in figure 1.