Project1 Report: KNN Binary Classification on TigerFish Dataset

Simeon Babatunde

Problem Description

Given a TigerFish dataset containing the body length, dorsal fin length and species of fish, we want to create a K-Nearest Neighbor binary classification algorithm that, given the body length and dorsal fin length of a fish, will predict if the fish is **TigerFish1** or **TigerFish0**.

Data Description

The first line of the dataset contains a single integer indicating how many sets of labelled data we have to work with. Each line after that contains three tab-separated entries. The first is a float representing the body length in centimeters, followed by a float representing the dorsal fin length in centimeters, then an integer identifying the fish as either TigerFish0 (with a 0) or TigerFish1 (with a 1).

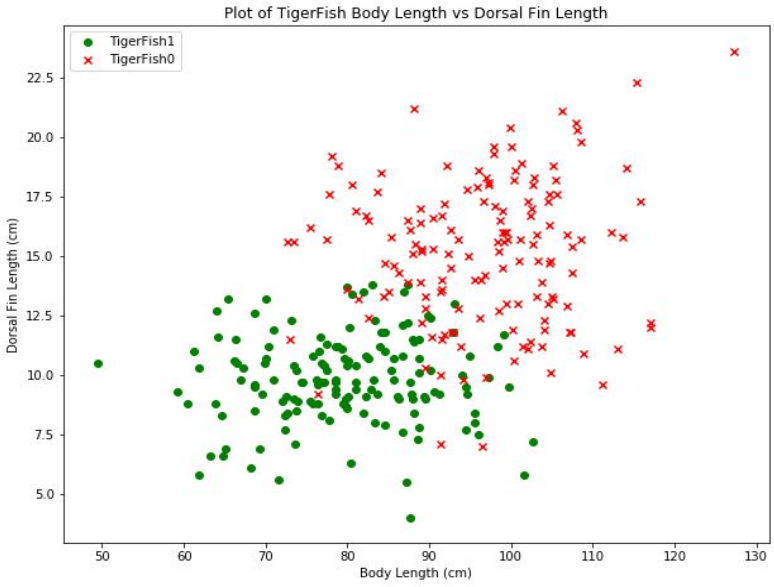

Figure 1. The Initial Dataset

Training the Binary KNN Algorithm

In order to develop the KNN algorithm, we use the 5 fold cross validation approach. The data was first randomized, followed by normalization. The data was normalized using $(x_i - x_{min})/(x_{max} - x_{min})$. The normalization ensures that both features have the same scales (0 to 1). The normalized data was then divided into training (240) and test (60). The training portion was further divided into 5 folds of 48 records each. This gives a smaller training set of 192 records and 48 records for test. For each fold, the training set was executed using KNN with odd values of k from 1 to 21. The misclassification(error) for each was recorded (Figure 2). Furthermore, the cross-validated accuracy **(1 - (errors per k / 240))** for each k was plotted, and k = 11 provided the best accuracy (Figure 3). So k = 11 was chosen for the KNN for the test set.

K	1	3	5	7	9	11	13	15	17	19	21
Test1 Error	6	7	2	2	2	1	2	3	3	3	3
Test2 Error	3	2	2	2	2	2	2	2	2	2	2
Test3 Error	3	5	5	4	4	3	3	3	4	5	5
Test4 Error	8	4	5	4	5	4	4	4	5	5	5
Test5 Error	7	5	4	4	3	4	5	5	5	5	5
Total	**27**	**23**	**18**	**16**	**16**	**14**	**16**	**17**	**19**	**20**	**20**
Accuracy %	88.8	90.4	92.5	93.3	93.3	94.2	93.3	92.9	92.1	91.7	91.7

Figure 2. Misclassifications for different values of k on the five training sets.

The KNN algorithm will determine if a fish is TigerFish1 (positive case) or TigerFish0.

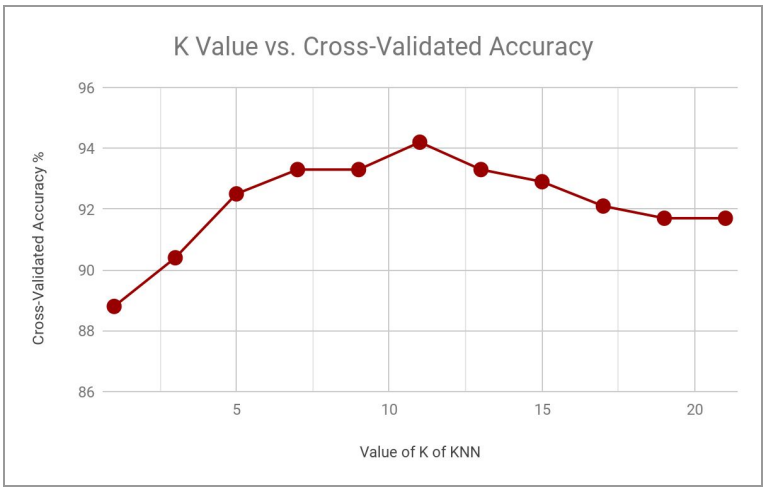

Figure 3. Average Accuracy for different value of k

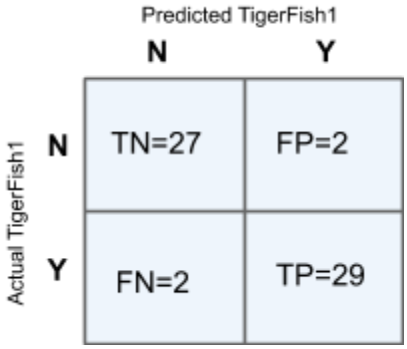

Figure 4. Confusion Matrix

Results

Figure 4 shows the confusion matrix of the Nearest Neighbor algorithm with k = 11. The test set consisted of 31 TigerFish1 and 29 TigerFish0 records. 56 of 60 records were correctly predicted for an **accuracy** of 0.93. **Precision** was to 0.94 i.e., every time the algorithm predicted that a fish was TigerFish1 it was correct. **Recall** was 0.94 i.e., percentage of TigerFish1 that are correctly classified. The overall **F1** score was 0.94.

Since all the major performance metrics (accuracy, precision, recall and F1 score) show good results, we can then agree that the selected number of Nearest Neighbors (**k = 11**) seem appropriate.