CS 334 Machine Learning HW #1

1.Numerical Programming:

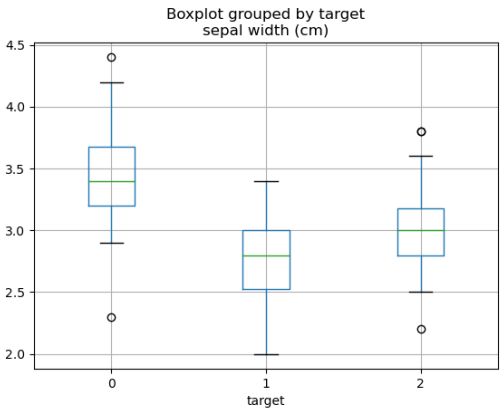
(d) Below is one instance of the results: It seems that the vectorized approach (on average) was about 90 times faster than the loop implementation.

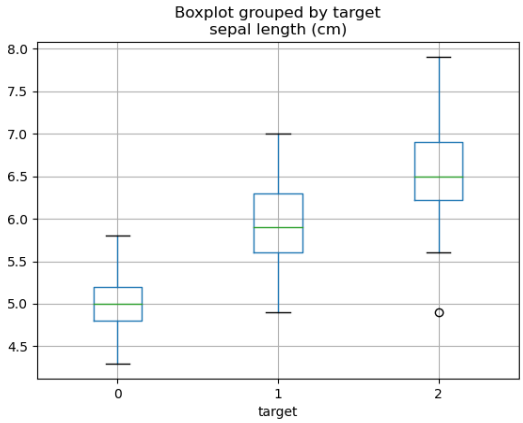
Time [sec] (for loop): 0.8902327319956385

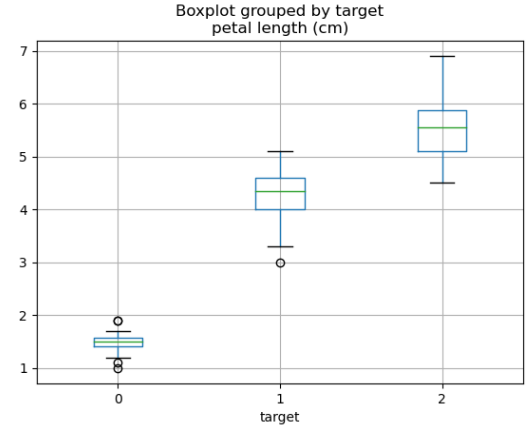
Time [sec] (np loop): 0.0016035569715313613

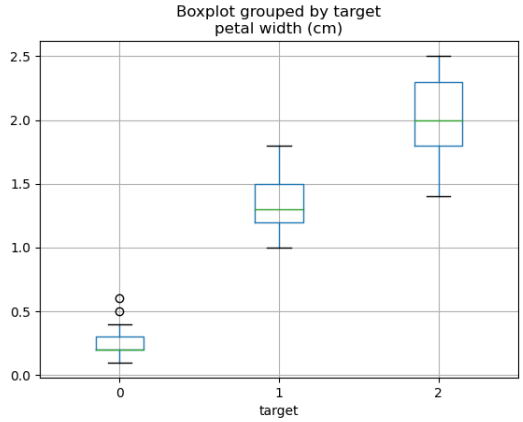
2.Visualization Exploration:

(b) Box Plots:

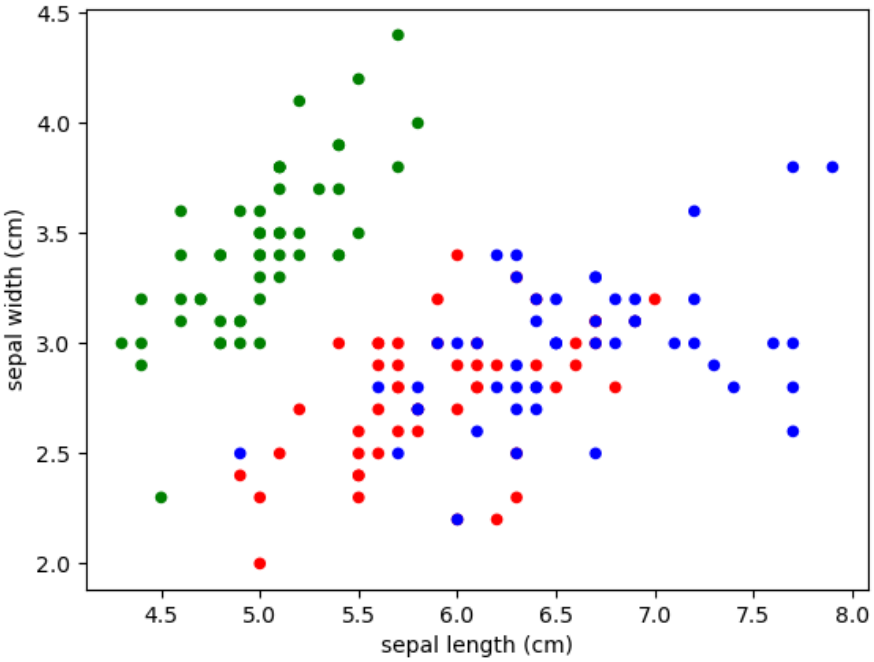


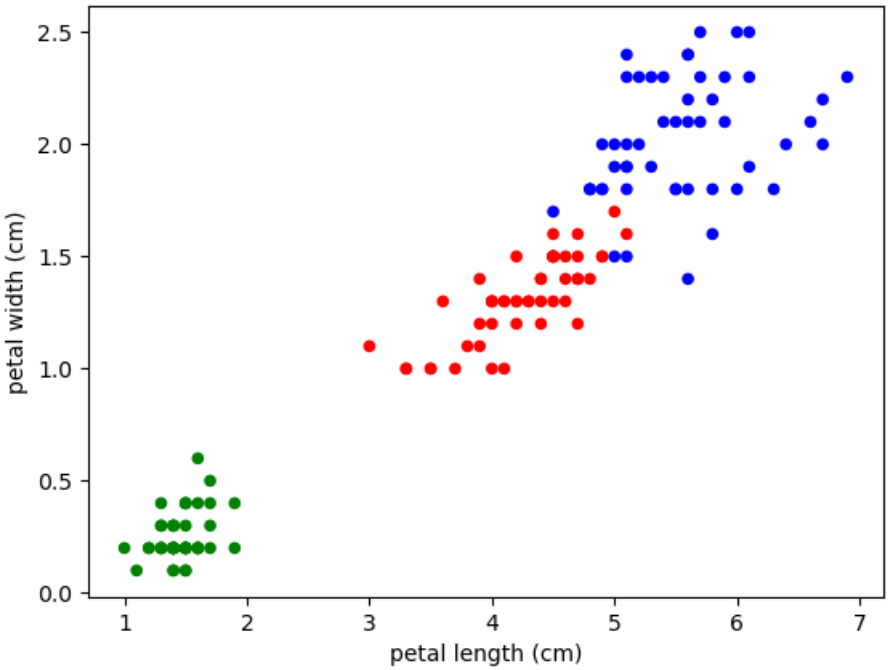






(c) Scatter Plots: (green = 0/setosa, red =1/versicolor, blue =2/virginica)

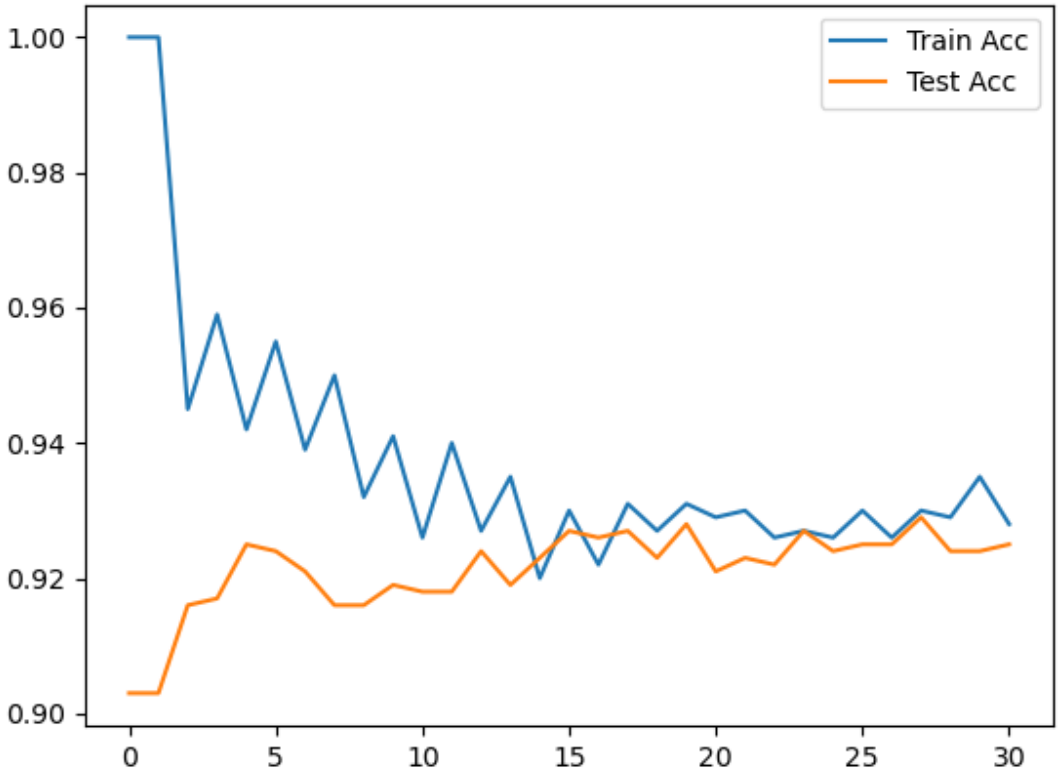




(d) Looking at the figures above, it seems that petal characteristics define each species much better than sepal. In general, Setosa (green) petals have a width in the range of 0.1 to 0.5 cm and a length in the range of about 1 to 2 cm. Versicolor (red) petals have a more varying width between 1 to 2 cm and a length between 3 to 5 cm. Virginica (blue) petals are much bigger with a width ranging from 1.5 to 2.5 cm and a length from 4 to 7 cm. Generally, Setosa sepals are also shorter and wider.

3.K-NN Implementation:

(d) Accuracy for different values of K with respect to test and training data:



Looking at the chart of my KNN implementation, it seems like the test accuracy and training accuracy starts stabilizing from around k = 5 at approximately 0.93.

(e) The computational complexity of my predict function for most cases is O(n log n). My predict function has two major loops:

1. The training data rows (n).
2. The rows of data used for prediction (z)

Both of these equates to O(z*n).

Additionally, for each z*n, an operation is done for each feature (d) == > O(z*d*n)

Lastly, list.sort is used to record the k shortest distances. Since the .sort function operates in O(n log n) time, our official final time complexity is **O(z*d*n + n log n)**.

However, since in most cases of KNN, we will assume that z (the length of test data rows) and d (the amount of features) would be much smaller than the training size (n), the time complexity can be simplified to O(n + n log n) or O(n log n).

4.K-NN Performance:

(d) Accuracy for different values of K with respect to different preprocessing techniques:

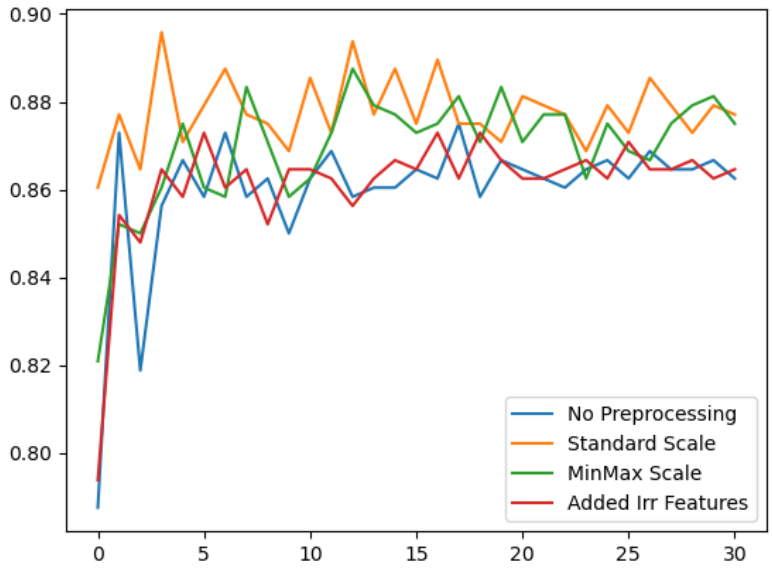


Figure {x = k, y = test accuracy}

Looking at the accuracy values, it seems like having irrelevant features and failing to process the data correlates to a lower test accuracy while either using a min max scale or a standard scaling method give a similar accuracy for stable values of k. These results seem to support our intuition that controlling feature weights by scaling a data set (therefore giving features similar weights) matters in situations where the features' flat values are disproportional from each other. The results also support our intuition in that random noise and irrelevant variables can lead to less accurate predictions.