David D'Attile
Prof. Kauchak
CS158
13 February 2022

Assignment 04: KNN and Normalization Experiment Writeup

1. *Calculate the performance of the perceptron classifier on the 10-fold cross validation of the data (i.e. you should have 10 numbers) with the AveragePerceptronClassifier on the old binary data, i.e. "titanic-train.csv". Use a reasonable number of iterations based on your experience from the last assignment or from a small experiment:*

   Fold 0: 70.18%
   Fold 1: 75.86%
   Fold 2: 52.17%
   Fold 3: 87.32%
   Fold 4: 84.13%
   Fold 5: 76.13%
   Fold 6: 78.38%
   Fold 7: 73.92%
   Fold 8: 73.32%
   Fold 9: 87.29%
   Average: 75.87%

2. *Calculate the accuracy on the 10 folds on the new non-binary data, i.e. "titanic-train.real.csv". You should notice a pretty big difference here.*

   Fold 0: 40.85%
   Fold 1: 59.15%
   Fold 2: 73.24%
   Fold 3: 58.44%
   Fold 4: 62.20%
   Fold 5: 60.58%
   Fold 6: 53.52%
   Fold 7: 57.75%
   Fold 8: 62.08%
   Fold 9: 62.67%
   Average: 59.05%

3. *Repeat experiments 1 and 2 for your new k-NN classifier:*

Binary Titanic -	**Real-Valued Titanic -**
Fold 0: 66.20%	Fold 0: 67.61%
Fold 1: 61.97%	Fold 1: 66.20%
Fold 2: 52.11%	Fold 2: 71.83%
Fold 3: 73.24%	Fold 3: 61.97%
Fold 4: 83.10%	Fold 4: 63.38%
Fold 5: 77.46%	Fold 5: 56.34%
Fold 6: 74.65%	Fold 6: 56.34%
Fold 7: 80.28%	Fold 7: 63.38%
Fold 8: 70.42%	Fold 8: 74.65%
Fold 9: 68.00%	Fold 9: 52.00%
Average: 70.74%	Average: 63.37%

4. *Now, generate a table of scores with the individual 10-fold scores and the 10-fold average on the following algorithm variants:*
    - k-NN with length normalization
    - k-NN with feature normalization
    - k-NN with length and feature normalization
    - perceptron with length normalization
    - perceptron with feature normalization
    - perceptron with length and feature normalization

	Avg. Perceptron			k-NN		
	Length Norm	Feature Norm	Both	Length Norm	Feature Norm	Both
Fold 0	40.85%	64.70%	55.11%	76.06%	59.15%	42.66%
Fold 1	59.15%	76.62%	80.11%	73.24%	67.61%	76.06%
Fold 2	74.65%	81.61%	81.75%	70.42%	84.51%	81.73%
Fold 3	56.34%	80.21%	78.10%	66.20%	76.06%	76.07%
Fold 4	63.38%	78.20%	71.38%	64.79%	74.65%	64.89%
Fold 5	60.56%	78.90%	79.28%	57.75%	74.65%	76.24%
Fold 6	53.52%	84.38%	76.42%	52.11%	83.10%	82.89%
Fold 7	59.15%	82.69%	74.14%	73.24%	77.46%	73.52%
Fold 8	63.38%	80.59%	76.20%	70.42%	74.65%	67.80%
Fold 9	62.67%	84.78%	79.59%	53.33%	81.33%	77.33%
Average	59.37%	79.27%	75.21%	65.76%	75.32%	71.92%

5. *Pick a few (say 4-5) of these results (including the earlier results) and calculate their t-test score to figure out if the differences are significant. Pick a couple of the experimental results that are close and a couple where they're further apart:*

KNN Binary vs. Real-Valued T-Test:	0.1261434474
Perceptron vs. KNN (Length Normalized):	0.150956921
Perceptron vs. KNN (Feature Normalized):	0.003168984925
Perceptron Length vs. Feature Normalized:	0.000004584620964
KNN Length vs. Feature Normalized:	0.06491672726

6. *Write a short (3-4 sentence) paragraph summarizing your results:*

   Based on the results above, we can make a few interesting conclusions. With the first result, we can see that the KNN classifier does not perform much differently when classifying binary or real-valued data. Looking at the second and third results, we see that there *is* a significant performance difference between the Avg. Perceptron and KNN classifiers when the test/train data has been feature normalized (Avg. Perceptron performing better), whereas there is no significant performance between the same classifiers leveraging length normalization of the test/train data. Finally, with the fourth and fifth results, we see that utilizing a length-wise vs. feature-wise normalizer for train/test data for the Perceptron classifier yields significant evidence that feature-wise normalization is a characteristic of a more effective classifier, and contrastingly, length-wise vs. feature-wise normalization for the KNN classifier does not yield decidedly different performance.