1 Evaluation of KNN Model and Data Pre-processing

Aidan Garton - 02/12/2022 - CS158: Machine Learning

1. AveragePerceptron on "old" data:

   Split 0 accuracy: 73.45%
   Split 1 accuracy: 75.943%
   Split 2 accuracy: 51.07%
   Split 3 accuracy: 84.507%
   Split 4 accuracy: 83.098%
   Split 5 accuracy: 77.746%
   Split 6 accuracy: 78.605%
   Split 7 accuracy: 86.45%
   Split 8 accuracy: 68.676%
   Split 9 accuracy: 66.666%
   Average accuracy: 74.621%

2. AveragePerceptron on "new" data:

   Split 0 accuracy: 40.845%
   Split 1 accuracy: 59.154%
   Split 2 accuracy: 73.239%
   Split 3 accuracy: 58.464%
   Split 4 accuracy: 62.267%
   Split 5 accuracy: 60.577%
   Split 6 accuracy: 53.521%
   Split 7 accuracy: 57.788%
   Split 8 accuracy: 62.056%
   Split 9 accuracy: 62.666%
   Average accuracy: 59.058%

3. KNN on "old" data:

   Split 0 accuracy: 64.788%
   Split 1 accuracy: 77.464%
   Split 2 accuracy: 52.112%
   Split 3 accuracy: 74.647%
   Split 4 accuracy: 80.281%
   Split 5 accuracy: 77.464%
   Split 6 accuracy: 78.873%
   Split 7 accuracy: 73.239%
   Split 8 accuracy: 69.014%
   Split 9 accuracy: 68.0%
   Average accuracy: 71.588%

   KNN on "new" data:

   Split 0 accuracy: 67.605%
   Split 1 accuracy: 66.197%
   Split 2 accuracy: 71.83%
   Split 3 accuracy: 61.971%
   Split 4 accuracy: 63.38%
   Split 5 accuracy: 56.338%
   Split 6 accuracy: 56.338%
   Split 7 accuracy: 63.38%
   Split 8 accuracy: 74.647%
   Split 9 accuracy: 52.0%
   Average accuracy: 63.369%

4. KNNClassifier and PerceptronClassifier accuracies using pre-processing on real-valued data:

	KNN w/ length	KNN w/ feature	KNN w/ both	P w/ length	P w/ feature	P w/ both
fold 0	76.056%	53.521%	52.112%	40.845%	47.887%	47.887%
fold 1	73.239%	73.239%	71.83%	59.154%	57.746%	60.563%
fold 2	70.422%	83.098%	84.507%	74.647%	80.281%	74.647%
fold 3	66.197%	78.873%	74.647%	56.338%	78.873%	77.464%
fold 4	64.788%	78.873%	80.281%	36.619%	76.056%	70.422%
fold 5	57.746%	77.464%	77.464%	60.563%	76.056%	60.563%
fold 6	52.112%	83.098%	83.098%	50.704%	73.239%	71.83%
fold 7	73.239%	77.464%	77.464%	40.845%	63.38%	77.464%
fold 8	70.422%	74.647%	74.647%	63.38%	66.197%	61.971%
fold 9	53.333%	81.333%	76.0%	41.333%	68.0%	62.666%
Average of folds:	**65.755%**	**76.161%**	**75.205%**	**52.443%**	**68.771%**	**66.548%**

5. TTEST(knn w/ both, perceptron w/ length, 2, 1) = 0.0003053549841
   TTEST(knn w/ feature, perceptron w/ both, 2, 1) = 0.0007558546905
   TTEST(perceptron w/ both, perceptron w/ length, 2, 1) = 0.01327855754
   TTEST(knn w/ length, knn w/ feature, 2, 1) = 0.06030078592
   TTEST(knn w/ length, knn w/ both, 2, 1) = 0.08155964265
   TTEST(knn w/ length ,perceptron w/ both,2,1) = 0.859388113

6. From the t-test results we see that KNN with length normalization, perceptron with both types of normalization, and perceptron with just feature normalization all have high p values. This indicates that these models perform very similarly on all folds of the train and test splits since the probability that they're accuracies come from the same distriution is high. On the other hand, the knn with both types of normalization and the perceptron with length normalization have very different distributions for their accuracies.

   We also see that both types of models (Perceptron and KNN) perform best when the data has been pre-processed just using feature normalization. Prediction accuracy is similar and only a bit lower when both types of normalizaation are used. A large decrease in prediction accuracy occurs when just length normalization is used but it still provides much better accuracies then when no pre-processing is done.