Jessica Brennan

Machine Learning

Homework 3 Summary

I used the Iris dataset and evaluated the model based off of the petal length and petal width, as these features seemed to be the most telling for their classification. I chose the Iris dataset because it would not require much preprocessing, so I could focus more on the actual implementation, training (for one), and performance of the given model. I looked at both accuracy and f1-scores, but I based my evaluations on f1-scores.

The classifier that I used in part one and two was the logistic regression implementation in scikit-learn. This is because logistic regression is one of the most widely used models for classification in the industry and it personally seems easier to use and tune. In part one, I used the default classifier parameters of C=1.0 and random\_state=None. With these defaults, I received an f1-score of 0.87, which is not bad. For part two, I chose what parameters to tweak based off of the model’s scikit-learn documentation page, what the book showed as parameters, and different articles I read online. As my defaults suggest, I played around with C and random\_state. I mainly focused on C because it seemed to be the most popular parameter that people tuned. I started with very small C values (0.01) and worked my way up to larger values (1000.0). The f1-scores ranged from 0.52 to 1.0 and it seemed that C=10.0 produced 1.0 more consistently. I did not find any significant differences in f1-scores while tweaking random\_state, so I kept that at None.

Part three involved implementing a k-nearest neighbors algorithm. After writing this, I also double-checked scikit-learn’s implementation to ensure my scores were on point. From what I found, the default k value is 5, so I started off with this. While tuning the k value, I started at k=1 and worked my way up incrementally. As I assumed, lower k scores produced more inconsistent f1-scores, ranging from 0.87 to 1.0. I believe the f-1 scores are so high in this case because the size of the dataset is small. Higher k values, up until 9 or so, seemed to increase the f1-score more consistently. I stuck with k=6, as it seemed like a good balance between minimizing the error rate and only including nearby samples. The f1-score of k=6 often was 1.0.

While testing on the actual test set instead of the validation set, my best logistic regression classifier had a maximum f1-score of 0.95 and my best k-nearest neighbor classifier had a maximum f1-score of 1.0. While these scores slightly fluctuated, k-nearest neighbor generally seemed to stay ahead of logistic regression. Both are very good numbers, which has me skeptical of overfitting. However, my C for the logistic regression classifier was relatively low, which is said to prevent overfitting.