Sean McLean

ALY 6020

Module 1

Midweek Project

Introduction/Analysis

The goal of the assignment is to look at the accuracies of three Iris colors using a dataset supplied in Python. The overall accuracy is computed by looking at the prediction of each data point and comparing them to the actual result. These numbers are computed from algorithms and coding provided in the lessons of the module this week. After importing the necessary packages, the dataset was uploaded and then divided into training and testing sets. These steps were important toward understanding what is being predicted among the variables.

The three colors of the Iris flowers in the dataset were identified as red, green, and yellow. They were then classified by values using the iclass function, with red being 0, green being 1, and yellow being 2. A visualization of the data points of all the Iris flowers were then executed to show their relationships (See Appendix A). Looking at the graph the three Iris flowers all seemed to be in clusters which makes it easier to see the differences between them. Numerous functions like get_neighbors and vote were created to randomly select data points to calculate the accuracies of each type of Iris flower (See Appendix B).

Questions

1. What was the overall accuracy of the model?

   There were 12 data points that were selected overall and 11 of the 12 points that were predicted were the same as the actual result. Of the 12 data points selected, there were two points from the red flower, six points from the green flower, and four points from the yellow flower. The calculation of the accuracy overall is 92 percent, indicating that the KNN model is strong. The module lessons indicated that a 65-70 percent success rate is where you want the models to be at the minimum.

2. What was the accuracy of each type of iris?

   The red flower had two points chosen that were both predicted correctly so it had a perfect 100 percent accuracy rating. The yellow flower had four points selected and were all predicted correctly so it also had a perfect 100 percent accuracy rating. Of the six data points containing a green flower, five were predicted correctly, the other being predicted as one, but the actual data point was from the yellow Iris flower. The accuracy rating overall for the green flower was 83 percent with one incorrect prediction. Despite this, all three Iris flowers had very high accuracy ratings which suggests that the model works very well at its predictions.

3. Would you classify the model as a good model or not?

Considering all the factors with the dataset this is a good model in several ways. There is a nice balance of the three Iris flower clusters, so the data is consistent, and there does not seem to be an issue with overfitting from the data visualization that was created. Because there is no baseline accuracy to use, I am unsure if the accuracy of the data is good, but a 92 percent prediction rate is high regardless. The next step would be to compare this with other models to clarify how good the model is.

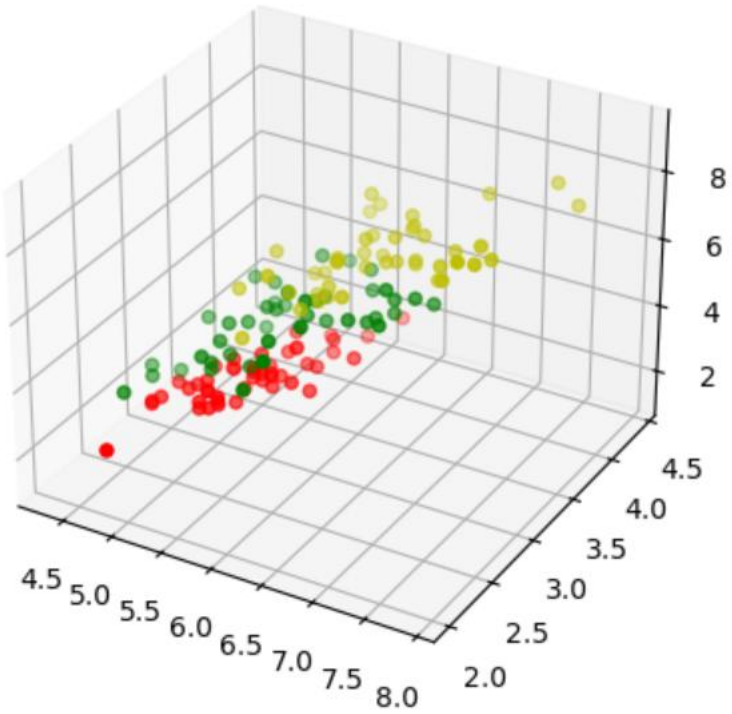
Conclusion

The analysis revealed that the KNN model achieved an impressive overall accuracy of 92 percent in predicting the colors of Iris flowers, far surpassing the module's recommended minimum success rate of 65-70 percent. Both the red and yellow flowers achieved perfect accuracy ratings of 100 percent, while the green flower had an accuracy of 83 percent. Data

visualization displayed distinct clusters for each flower type, indicating clear separability and no overfitting. Although there was no baseline accuracy for direct comparison, the high accuracy rate demonstrates the model's robust predictive ability. Future comparisons with other models will help further validate its performance.

Appendix

Appendix A



Appendix B

```
[39]: for i in range(n_training_samples):
          neighbors = get_neighbors(trainset_data,
                                    trainset_labels,
                                    testset_data[i],
                                    3,
                                    distance=distance)
          print("index: ", i,
                ", result of vote: ", vote(neighbors),
                ", label: ", testset_labels[i],
                ", data: ", testset_data[i])
```

```
index:  0 , result of vote:  1 , label:  1 , data:  [5.7 2.8 4.1 1.3]
index:  1 , result of vote:  2 , label:  2 , data:  [6.5 3.  5.5 1.8]
index:  2 , result of vote:  1 , label:  1 , data:  [6.3 2.3 4.4 1.3]
index:  3 , result of vote:  1 , label:  1 , data:  [6.4 2.9 4.3 1.3]
index:  4 , result of vote:  2 , label:  2 , data:  [5.6 2.8 4.9 2. ]
index:  5 , result of vote:  2 , label:  2 , data:  [5.9 3.  5.1 1.8]
index:  6 , result of vote:  0 , label:  0 , data:  [5.4 3.4 1.7 0.2]
index:  7 , result of vote:  1 , label:  1 , data:  [6.1 2.8 4.  1.3]
index:  8 , result of vote:  1 , label:  2 , data:  [4.9 2.5 4.5 1.7]
index:  9 , result of vote:  0 , label:  0 , data:  [5.8 4.  1.2 0.2]
index:  10 , result of vote:  1 , label:  1 , data:  [5.8 2.6 4.  1.2]
index:  11 , result of vote:  2 , label:  2 , data:  [7.1 3.  5.9 2.1]
```