When your assignment is complete, please answer the questions in this text file and upload it to I-Learn.

1. If you did not include your source code in your I-Learn submission, please provide the URL of your public GitHub repository.

2. Briefly describe your overall approach to the task and highlight the most difficult part of this assignment.

* As I worked with a CS major I was able to approach this problem with a better understanding of what a kNN algorithm was. The most difficult part was trying to implement a k-d tree.

3. Briefly describe your process for handling numeric data on different scales (i.e., normalizing).

* Everything was scaled to one.

4. Describe your results for the Iris data set. (For example, what level of accuracy did you see for different values of K?)

* When k = 3 my accuracy was 88.89%
* When k = 10 my accuracy was 91.11%
* When k = 5 my accuracy was 88.89%
* I would say that my accuracy for different values of k was very good.

5. How did your implementation compare to the existing implementation?

* Percentage of sklearn kNN correct was always very close to an accuracy of 100%
* Percentage of my HardCoded kNN correct was always between 80-100% so it was accurate but, not as good as the sklearn kNN algorithm.

6. Describe anything you did to go above and beyond the minimum standard requirements.

* I looked into cross validation because this area was a little hazy. I think I added a correct K-Folds Cross Validation plot. I split my data into k different subsets then I averaged the model against each of the subsets and plotted it giving the optimal number of neighbors.

7. Please select the category you feel best describes your assignment:

1 - Some attempt was made

2 - Developing, but significantly deficient

3 - Slightly deficient, but still mostly adequate

4 - Meets requirements

5 - Shows creativity and excels above and beyond requirements

8. Provide a brief justification (1-2 sentences) for selecting that category.

* I spent a lot of time looking into cross validation and trying to understand it while I implemented it in my code.