Assignment 2.1: Perceptron and kNN

For this assignment, you will conduct programming tasks in Jupyter Notebook and answer provided questions in a Google Doc or MS Word document. You will submit **2** documents to Blackboard:

1. A PDF file converted from your Jupyter Notebook.
2. A PDF file converted from your Assignment 2.1 Template Word Doc (document is linked in Blackboard assignment prompt).
   - This template has a copy of all short answer questions for this assignment.

Drum, Text, and Animal Data
You will work with three data sets for this assignment, as you did in assignment module 1.

Drum Data: These data are in drum_data.csv. These are the spectrum (frequencies) of sounds from a digital drum musical instrument. The task is to identify which type of drum produced the sound.

Text Data: These data are in count_hamilton.csv. These data are the word counts of the Federalist papers, important documents in American history which have disputed authorship. The task is to identify the author of these documents.

Animal Data: These data are from an animal shelter. These are records of animals coming in for treatment, with a few characteristics of the animals and information on the outcome. The task is to predict the outcome.

Written Questions
(short answer, 2-3 sentences):

For the animal shelter dataset, the classification results using k-nearest neighbor are not as good as the other datasets. Why do you think this is the case for these data? Explain in terms of inductive bias, and remember that you used OneHotEncoding, making the data more high-dimensional and sparse.
The reason for poor KNN perforance is what is termed as the “curse of dimensionality” – our training set, fully unraveled, has over 6000 rows for a very small sample size. At higher dimensions, distances begin to converge on a similar value with smaller variance between distances, which causes performance to drop given than KNN has an inductive bias towards assuming that all features are equally weighted and that points in close proximity to each other have the same class.

Which distance metric seemed to perform the best on the test dataset, Euclidean or Cosine distance? How large are the text feature vectors (how many features/words are in the dataset)? Do you think high-dimensional data like this are better suited for the Euclidean or Cosine distance, based on your understanding of the methods and the results you have obtained here?

Cosine distance outperformed Euclidean distance on the text data – there were over 8500 distinct columns for this dataset. Based on the performance of both the text and shelter sets, it seems that euclidean distance becomes less useful at high dimensions and can be substituted by cosine distance.

On the audio (Drum) dataset, you should have used nearest neighbor, perceptron, and your own handwritten implementation of the perceptron. Which classifier performs best, or are they all giving about the same results? If there is a clear winner, explain why this classifier might be a good match for the audio dataset.

All three perform roughly the same – KNN has about 96% accuracy, handwritten perceptron performs at 95.5% with 100 iterations, and the Sklearn perceptron performs at 100%. In terms of sheer magnitude, we can claim that the sklearn implementation is optimal compared to KNN because the audio dataset presents a binary class problem (two classes that can be expressed as +1/-1), and while KNN can handle multiclass problems, a perceptron is able to relax the assumption that weights for different features are differential.