Name: Carson Ellsworth

Date: March 28, 2022

Deep Learning Homework 3

**Objective:** Implement both a Gaussian Naïve Bayes model and Logistic Regression model to evaluate and classify datasets.

**Dataset:**  The dataset I chose for this homework is from [UCI Machine Learning Repository: Data Sets](https://archive.ics.uci.edu/ml/datasets.php) website. It is a classification set on the Abalone sea creatures and eight attributes that were measured from them.

The dataset contains 8 different attributes (X’s)

1. Sex
2. Length
3. Diameter
4. Height
5. Whole Weight
6. Shucked Weight
7. Viscera Weight
8. Shell Weight

And a single output, Yk, consisting of the number of rings their shells contain (1 to 29)

**Preprocessing Decisions:** For this dataset there was very little required in terms of preprocessing. All that was needed was to assign an integer value to the possible sex values {Infant,Male,Female} -> {0,1,2} respectively.

**Gaussian** **Naive** **Bayes:** The GNBayes model allows for evaluating continuous data, unlike a regular Naïve Bayes, by assuming the data follows a normal distribution. This allows for a change to the probability function P(Xi|Yk) to be changed to a pdf. For this specific model I was unable to achieve a high accuracy unfortunately, I do not know if it was due to the dataset having around 4000 examples which just may not have been enough for a GNB model to split and still have adequate training examples or if it was due to a deeper issue with how the model implemented certain features. I went with the same approach as in project 1 which was using dictionaries to store the pdfs and dynamically build the distributions by creating function builders that return a specific distribution given a mean and standard deviation. After further evaluation of the model, it does appear that there is just not enough examples for the model to evaluate on. I came to this conclusion by looking at how the classifications were distributed in the dataset and saw that there are some classifications that only have a single example, this means that when the data is split between into training and testing it is possible to and actually likely that those with fewer examples are not included in the training set, which makes the model unable to see and train on those classifiers.

I could possibly improve this by removing the low example data from the dataset so that there is always a high number of examples to train on but unfortunately, I am already late on this assignment and need to turn it in.

The results of this model at its best is around a 15% accuracy, which given 29 possible classifications is better than statistical random. Though I will be choosing larger datasets in the future when using this model.

**Conclusion:** Even though the accuracy on this model is not great, it does show a better than statistical average of randomly picking the answer correctly (3%) where the GNB (15%).