Name: Vishu Wadhawan
Student Number: 1007250252
Course: STA 2453

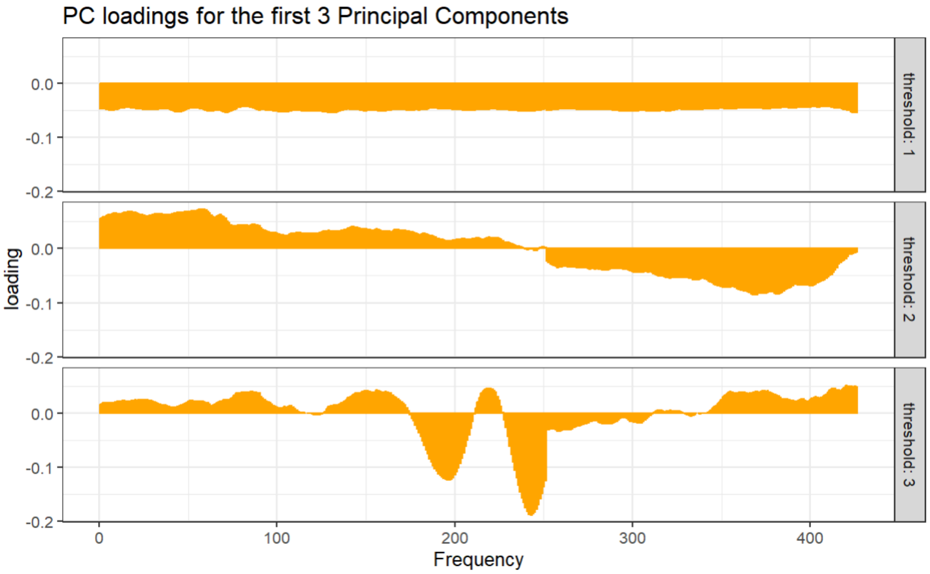
Progress Report

The goal for my project is to build a classification model that can identify a fish's species from its reflected sound pulses. The model needs to be able to detect patterns or significant associations between the emitted frequencies and specific species, thus, I began my project by examining the dimensions and different features of the data. There were 426 frequencies ranging from 45 Khz to 260 KHz, three separate fish species (Trout, whitefish, and Bass), and 14575 observations with each row corresponding to an individual fish. Moreover, I then examined how the length of each species varied with three different boxplots. From my findings, I discovered that there is very little variability within each species and thus, I decided to forego every feature in the dataset except for 'fishNum', 'species'. and the frequencies of course.

My next step was to visualize and perhaps have an estimate of what type of frequencies may be significant in predicting a fish' species. In order to accomplish this, I split the data into three different data frames: one for trout, one for whitefish, and one for bass. From here, I grouped each of the datasets by their fish number ('fishNum') and then calculated the average emitted sound for each of the 426 frequencies across all unique individuals. I then plotted these averages (y-axis) across all of the frequencies (50 KHz – 426 KHz) and compared the three plots side by side. My goal was to examine any differences between the plots and interestingly enough, there were a couple areas where the averages differed from one another. I found that the 90-95Hz range showed trout to have the lowest emitted sound strength with the bass having the second lowest and the whitefish the highest. In contrast, I saw a distinct pattern in the 170-197Hz range where trout had the highest average sound strength, while whitefish and bass had the second and third highest respectively. Of course, these findings were basic and preliminary in nature but nonetheless provided a sufficient visual representation of my data.

From here, I had thought that I was ready to start building my predictive model. I was initially inclined to build a simple neural network and thus began reading and researching the proper code and assumptions that I needed. Through my research, I discovered that there are a few limitations with this method, with the most important one being that it is hard to know which features are more predictive than others. Neural networks simply follow a ‘plug and chug’ process although, the nuisances such as ‘dropout’ and the amount of layers it requires can be played around with. After contemplating whether or not I should stick with this method or not, I talked to one of my classmates during the discussion prompts and they happened to be working on the same project as me. They discussed that they had used principal component analysis (PCA) to reduce the number of dimensions before modelling which ultimately inspired me to do the same. Hence, I preformed a simple PCA on the entire data set but only kept the frequencies as my variables. My results showed that the first 39 different principal components were capable of explaining over 80% of the variability data . I then visualized each of the 39 principal components in R with a line graph where the x-axis represented the frequencies, and y-axis the loadings. I hoped to find a loading that was much larger than the others and an obvious outlier, however, my visualizations did not show anything significant. Each PC principal component following the first one explained <5% of total variation within the data. In these graphs, there are peaks in miscellaneous places but the peaks in the first PC are more telling in my opinion because it explain 35% of the variance. Yet, the graph for the first principal components shows a complete uniform distribution of loadings across the 426 frequencies.

Nevertheless, I was not expecting big results from PCA, so I pivoted and began researching different dimensionality reduction techniques. There were two other techniques that I believed would be suitable for my project, t-SNE and UMAP. I implemented t-SNE on my model however the computations are taking extremely long to run. I even attempted to apply t-SNE to the PCA reduced data set and again, my computer cannot seem to handle it efficiently. This now made me wonder if running a neural network would even work on my computer, so, I have decided to build three classification models instead of just one. I plan to implement a classification tree, a k-means algorithm, and a simple neural network in order to predict a fish's species from the sonar data. My hypothesis is that the k-means and neural network algorithms will not work but the classification tree should be easily to implement. However, before I begin modelling, I want to implement UMAP on my data set and then run PCA and UMAP on a more reduced data set. I have an inkling that the variability in the sizes of the fish might be affecting my PCA analysis, so I want to reduce my data set for each of these fish so that there is less variability between individuals. I want to keep at least 1000 individuals for each group which should be sufficient enough for modelling.


PC loadings for the first 3 Principal Components
loading
threshold: 1
threshold: 2
threshold: 3
Frequency