**Data Science Project Report**

**1. Principal Investigator**

Lauren Szabo szabol1@newpaltz.edu

**1) Individual Contribution Breakdown (list the percentage)**

Since this was a solo project all work was done by me.

**2. Title of Project**

Music Matcher

**3. Introduction**

The motivation for my project was to create my own music recommendation system that could potentially be more accurate than Spotifys nearest neighbour system. I used liked as well as disliked song features to build my decision tree whereas Spotify only uses liked songs to recommend music. My objective was to originally be able to predict whether a user will like a song with 80% accuracy.

**4. Background/History of the Study**

Since this topic is mostly informal and not an in depth study on any particular topic there is not much background or history on music recommendation systems. The idea of a recommendation system really appealed to me and I thought it would be interesting because it can be applied to a lot of different topics.

**5. Approach and Implementation**

I used Hadoop map reduce to get my thresholds and bin my data. I calculated the mean of each column and then used standard deviation to create a threshold range for what counts as in range (liked/preferred). To calculate the threshold range I only used liked songs since these are the values we want the songs to have. So after sorting through and binning both the liked and disliked column values I assigned their class values afterwards in python. The rest of the code was done in python and involved me building the decision tree. My implementation was simplified a lot because everything was already binned so there was no need to do any calculations for the decision tree. In the end it was successful in predicting if the song was inRange (liked) or outOfRange (disliked) based on the built decision tree and each individual songs audio values.

**6. Experiment Results and Discussion**

When I was writing my decision tree I had some issues mainly with my misunderstandings on how decision trees work. I for some reason was under the impression that you split on the attributes information gain rather than the split itself. After clarifying writing the decision tree was simple and I was able to mess around with the means and what deviation I would be using. I started out with having results in the low 70s and with some tinkering was able to get it to 80%. My end goal however is to be able to predict with 95% accuracy. The main thing getting in my way currently is that Spotifys API has some issues in getting the genre information for a track, and instead you have to rely on the artists genre information. I ran into issues with call limits and have no found a good way around this for large bulks of data. This is something I would like to solve in order to increase the accuracy of my tree.

**7. Conclusion**

Overall, I’m happy with the end result of this project. Learning how to use Hadoop has also been really interesting and a valuable experience. I hope to be able to take the project further by creating a front end for it, as well as being able to get all of the genre information for each song in order to increase accuracy.

**8. References**

**Due to the non factual or research based nature of the project topic there were not many references used**

“Web Api.” Web API | Spotify for Developers, developer.spotify.com/documentation/web-api. Accessed 8 Dec. 2023.

Spotify Engineering. “Introducing Voyager: Spotify’s New Nearest-Neighbor Search Library.” Spotify Engineering, 14 Nov. 2023, engineering.atspotify.com/2023/10/introducing-voyager-spotifys-new-nearest-neighbor-search-library/.