Module 3: Clustering (Part 2)

10/1/2025

60/40 Points

Attempt 2

Review Feedback
9/29/2025

Attempt 2 Score:
60/40

View Feedback

Anonymous Grading: **no**

Unlimited Attempts Allowed

Details

Goal of the assignment:

In this assignment, you will analyze the clustering results of the Spotify dataset we looked at in class. Your goal is to study the relationship between genres and clusters built from song features. While doing so, you will **identify pairs of songs that belong to the same cluster, have minimum distance from each other, but are labeled with a different genre.** The point is to determine whether feature-based distance between songs and clustering are sometimes more meaningful than genre in identifying similar songs.

Steps:

1. Create a new Colab notebook. Name it **your-name-clustering-part2** (replacing your-name with *your actual name*, of course!!) Paste the code from my notebook `03-Clustering-Part2.ipynb` into yours, and add your code at the end. *Notice that to run the notebook you will need to load the spotify dataset in your own Google Drive folder.* Look at the *How to connect to Google Drive* video in the *Clustering* module on Canvas if you need further help, and let me know if you're stuck.
2. Make sure you understand what the code is doing, from beginning to end, at a high level (i.e., you don't need to understand every single command). Gemini in Colab can also help you understand the code, and I'm happy to answer questions, of course. You will see that the last cell builds a dictionary containing the Euclidean distance between all pairs of songs in the dataset. For convenience, the keys in the dictionary contain `trackName+artistName+genre+cluster_label` for each pair of songs in the dataset.
3. **Add a code cell at the bottom** that sorts the dictionary using the distance as sorting criterion, from smallest to largest, and prints the first twenty pairs of songs.
4. In a **text cell below your code, identify the first five pairs that are labeled with different genres but belong to the same cluster**. Are you surprised by these results? If you can, try to listen to some "unexpected" close matches of songs in different genres and write your impressions after actually hearing the songs.
5. Now, go back to the clustering results a few cells up. You will find a breadown of how many songs are in each genre per cluster. **Add a text cell at the end of the notebook** and answer the following question: which genres are more cohesive (i.e., less fragmented across clusters)? To answer this question, write down the top three genres that have the most songs in a single cluster.
6. Now answer the same question again with five clusters instead of three. To do this, simply change the number of desired clusters when cutting the tree, and running the cell with the breakdown of genre by cluster.

EXTRA-CREDIT OPPORTUNITY! (20 extra points). You will probably have noticed that building the dictionary of pairwise distances between songs takes a long time to run. The code I wrote (with two nested loops) is not the most efficient way to do this in Python. For extra-credit points, try to accomplish the same objective (getting pairwise distances between songs) in a more efficient way. *Hint: think about distance matrix instead!*

Deliverables:
A Colab notebook containing all the code and text. Please upload that to Canvas. That's all you need to upload.

Use of LLMs:

You are allowed (and even encouraged) to ask Gemini (available directly in Colab) or other LLMs for help if you don't know how to do something.

View Rubric

Select Grader

Sudha Pandey (TA)

Rubric for clustering assignment (part 2)

Criteria	Ratings					Pts
Code execution	**15 pts** **Excellent** The Colab notebook is properly named, and all steps are executed correctly. The Spotify dataset is loaded correctly from Google Drive and the notebook runs without errors, with correct output. The sorting of song pairs based on Euclidean distance is implemented correctly.	**10 pts** **Good** The notebook is properly named, and all required steps are executed, but there are minor issues in the code.	**7 pts** **Satisfactory** The notebook runs, but there are a few issues with the code.	**4 pts** **Needs improvement** The notebook has serious errors, or the steps are incomplete. The dataset is not loaded correctly, or the sorting of song pairs is incorrect.	**0 pts** **Incomplete** The notebook does not run, or the steps are not completed. Substantial parts of the assignment are missing.	15 / 15 pts
Interpretation	**15 pts** **Excellent** The first five pairs of songs from different genres but in the same cluster are identified accurately. The analysis of these pairs is insightful. The genre breakdown by clusters is correctly reported.	**10 pts** **Good** The first five pairs of different-genre songs in the same cluster are identified correctly, but the analysis lacks depth.	**7 pts** **Satisfactory** The first five pairs are identified, but the analysis is minimal. The student provides impressions on the songs but without meaningful insights. The task is completed in a basic form.	**4 pts** **Needs improvement** The analysis is incomplete, or the first five pairs are not correctly identified. Impressions of the songs are missing or very limited.	**0 pts** **Incomplete** The analysis of song pairs is missing or completely incorrect. The genre breakdown by cluster is not reported correctly or at all.	15 / 15 pts
Clarity and presentation	**10 pts** **Excellent** The notebook is well-organized, with code and text cells clearly organized. Explanations in text cells are detailed and well written.	**7 pts** **Good** The notebook is mostly well-organized, but some explanations could be clearer, or text cells are not as detailed as they could be. Formatting is acceptable, with minor	**5 pts** **Satisfactory** The notebook is functional but there are some issues with formatting, organization, or clarity.	**3 pts** **Needs improvement** The notebook is disorganized, with unclear explanations or poorly formatted text. The structure of code and text cells needs substantial improvement.	**0 pts** **Incomplete** The submission does not meet the basic requirements of the assignment.	10 / 10 pts