CS432/532: Final Project Report

Project Title: Symphony of Data: A NoSQL Approach to Spotify's Song Analytics

Team Member(s): Deshpande Abhishek, Mane Akash, Purandare Chinmay

I. PROBLEM

We are addressing the challenge of analyzing the 'Top 10000 Songs on Spotify 1960 - Now' dataset using a combination of Tkinter for the development of a robust graphical user interface (GUI) and MongoDB as our NoSQL database. This dataset encompasses a wide range of attributes such as track duration, popularity, danceability, and acousticness, among others. Our primary objective is to tackle four non-trivial data analysis tasks, leveraging these attributes. These tasks involve exploring music trends through data visualization, employing MongoDB's querying capabilities to delve into artist behavior, and utilizing advanced statistical techniques to assess the impact of factors like genre and label on track characteristics. Our project aims to deliver an accessible and insightful solution for users to interact with and gain valuable insights from this rich music dataset.

- **Dataset Link:** https://www.kaggle.com/datasets/joebeachcapital/top-10000-spotify-songs-1960-now

Task 1 - Trend Analysis by Popularity, Danceability, and Energy: We explored the potential correlation between the popularity of tracks and their danceability and energy levels. This involved grouping tracks by popularity ranges and analyzing average danceability and energy within each range. The goal was to see if popular tracks share common characteristics.

Task 2 - Comparison of Major and Independent Labels: We investigated whether tracks from major record labels are inherently more danceable and energetic than those from independent labels, and how this might affect their popularity. This task required aggregating tracks by label and calculating average values for the relevant musical features.

Task 3 - Impact of Release Timing: This analysis centered on whether the release timing of tracks, specifically on Fridays or weekends, had any influence on their tempo, energy, and overall popularity. We examined trends based on the day of the week tracks were released to uncover any patterns.

Task 4 - Correlation Between Track Duration and Popularity Over Time: We looked into the relationship between track duration and its popularity across different years. This involved grouping tracks by year of release and analyzing how the average popularity and duration trends have evolved over time.

II. SOFTWARE DESIGN AND IMPLEMENTATION

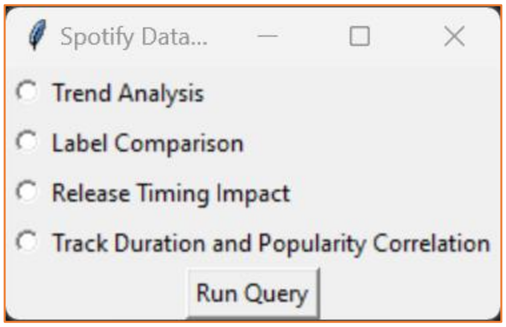
A. Software Design, NoSQL Database, and Tools Used

- **Tkinter for GUI**: We implemented Tkinter, a standard Python interface to the Tk GUI toolkit, to create a user-friendly graphical user interface (GUI). This GUI allowed users to interact with our software through visually appealing and intuitive graphical components like buttons, labels, and windows.
- **MongoDB as NoSQL Database:** MongoDB, a NoSQL database, was chosen for its scalability, flexibility, and powerful querying capabilities. It enabled efficient handling of large datasets, like the 'Top 10000 Songs on Spotify' dataset, and allowed complex queries and aggregations necessary for our analysis.
- **Matplotlib for Data Visualization:** We used Matplotlib, a comprehensive library for creating static, interactive, and animated visualizations in Python. This tool was crucial for plotting and visualizing data trends and correlations derived from our analysis.
- **Python Requests and PIL:** The Python Requests library was used to fetch images from URLs for album covers, and PIL (Python Imaging Library) was employed to process and display these images in the GUI.
- **Pymongo for Database Connection:** Pymongo, a Python distribution containing tools for working with MongoDB, was used to establish a connection between our Python application and the MongoDB database.

B. Implemented Parts:

- **Database Connection and Data Aggregation:**
    1. **MongoDB Connection**: Implemented a function (mongo_connection) to establish a connection with MongoDB, accessing the "Spotify" database and specifically the "Songs" collection.
    2. **Aggregation Pipelines:** For each analysis task, created MongoDB aggregation pipelines to process and analyze the data. Grouping songs based on specific criteria (like popularity, key, energy or label). Calculating average values, specifying weights, fetching data year-by-year, etc. for attributes such as danceability, energy, and popularity. Sorting results to prepare data for analysis and visualization.
- **Data Analysis Tasks:**
    1. **Task 1 - Trend Analysis:** Developed a query (trend_relationship) to examine the relationship between a song's popularity and its danceability and energy levels. The focus was on understanding how these characteristics correlate with track popularity.
    2. **Task 2 - Label Comparison:** Created a query (label_comparison_enhanced) to analyze how tracks from major vs. independent labels differ in danceability, energy, and popularity, highlighting the influence of label type on these attributes.
    3. **Task 3 - Release Timing Impact:** Designed a query (release_timing_impact) to investigate if the day of the week a track is released affects its tempo, energy, and popularity. This task explored the strategic aspects of music release timing.
    4. **Task 4 - Duration and Popularity Correlation:** Formulated a query (duration_popularity_correlation) to explore the potential correlation between track duration and popularity over different years, aiming to identify trends over time.
- **Data Visualization:**
    1. **Graphical Representation:** For each task, developed specific plotting functions (like plot_trend_relationship) using Matplotlib. These functions generated bar charts, line graphs, and scatter plots to illustrate data findings.
    2. **Customization and Clarity:** Tailored each visualization to the respective data analysis task, ensuring that the graphs were not only informative but also easy to understand. This involved thoughtful choices in color coding, labeling, and layout.
- **GUI Development:**
    1. **Tkinter GUI:** Built a comprehensive GUI using Tkinter that allowed users to interact with the software and choose among different analysis tasks.
    2. **Matplotlib Integration:** Seamlessly integrated Matplotlib graphs into the GUI, enabling users to view the results in a separate window for each analysis task.
    3. **Interactive Features:** Enhanced the GUI with interactive elements such as radio buttons for task selection and canvas widgets for displaying results, improving the overall user experience.

III. Project Outcome

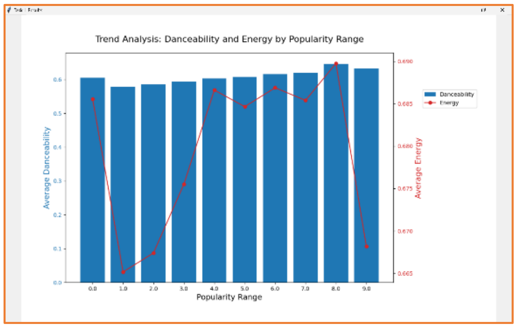


Analysis:

a. The UI snapshot showcases a simple and intuitive selection panel where users can choose one of four data analysis tasks related to Spotify's track data—namely, Trend Analysis, Label Comparison, Release Timing Impact, and Track Duration and Popularity Correlation.

b. Execution Command: Below the task options, there is a "Run Query" button designed to execute the selected data analysis task, indicating an immediate action can be taken once a task is chosen.

1. Task 1 - Trend Analysis by Popularity, Danceability, and Energy:



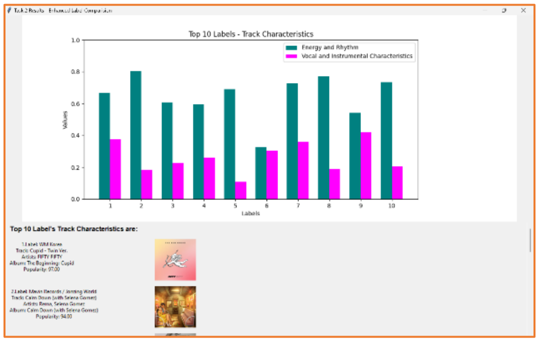
Analysis:

a. Tracks with higher popularity scores tend to exhibit higher danceability and energy levels.

b. The trend suggests a correlation where more popular tracks are likely to be more energetic and suitable for dancing.

c. This pattern indicates that tracks which engage listeners physically through dance might have an edge in gaining popularity.

2. Task 2 - Label Comparison: Danceability and Energy:


Task 2 Results - Enhanced Label Comparision
Top 10 Labels - Track Characteristics
Energy and Rhythm
Vocal and Instrumental Characteristics
Values
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8
9
10
Labels

Top 10 Label's Track Characteristics are:

1.Label: WM Korea
Track: Cupid - Twin Ver.
Artists: FIFTY FIFTY
Album: The Beginning: Cupid
Popularity: 97.00

2.Label: Mavin Records / Jonzing World
Track: Calm Down (with Selena Gomez)
Artists: Rema, Selena Gomez
Album: Calm Down (with Selena Gomez)
Popularity: 94.00

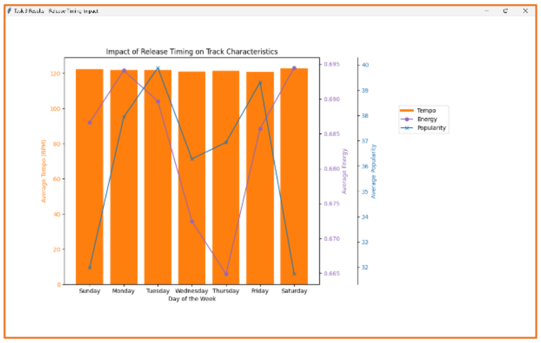
Analysis:

a. Major labels' tracks have a tendency to be more energetic and danceable compared to those from independent labels.

b. The high energy and rhythm scores among the top labels suggest that these characteristics may play a role in a track's popularity.

c. The variability in vocal and instrumental characteristics across labels points to diverse strategies and audience preferences.

3. Task 3 - Release Timing Impact:



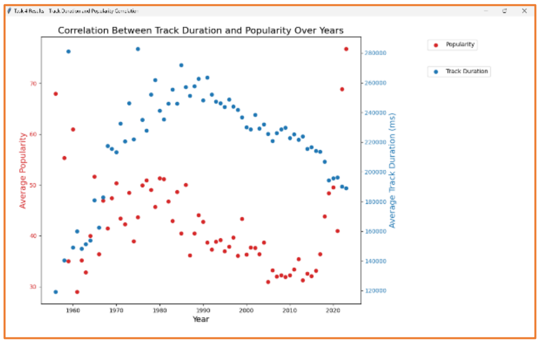
Analysis:

a. There is a variation in track characteristics like tempo, energy, and popularity depending on the day of the week they are released.

b. However, the data does not show a consistent pattern that could suggest an optimal release day for maximizing a track's popularity.

c. This implies that while release timing may have some effect, other factors are likely more influential in determining a track's success.

4. Task 4 - Duration and Popularity Correlation:



Analysis:

a. The data analysis reveals no consistent correlation between the duration of a track and its popularity over the years.

b. Trends in track duration and popularity do not seem to influence each other significantly.

c. This suggests that the length of a track is not a major factor in determining its success, and listeners' preferences for track duration may have evolved or varied over time.

REFERENCES

[1] MongoDB, Inc., "MongoDB Documentation," https://docs.mongodb.com/.
[2] Python Software Foundation, "Tkinter — Python interface to Tcl/Tk," https://docs.python.org/3/library/tkinter.html.
[3] Kaggle, "Top 10000 Spotify Songs 1960 - Now," https://www.kaggle.com/datasets/joebeachcapital/top-10000-spotify-songs-1960-now.
[4] M. Waskom, "Seaborn: statistical data visualization," https://seaborn.pydata.org.
[5] S. Chodorow, "MongoDB: The Definitive Guide," O'Reilly Media, Inc., 3rd edition, ISBN: 978-1491954461, 2019.
[6] M. Lutz, "Learning Python," O'Reilly Media, Inc., 5th edition, ISBN: 978-1449355739, 2013.
[7] A. Clark et al., "Pillow (PIL Fork) Documentation," Read the Docs, https://pillow.readthedocs.io/en/stable/index.html.