

CS 065 Project Report – Famous Songs from the Decade for you.

By Barbara Vrhovac, Tino Chigwada, and Maala Membe

NOTE

The datasets we used for our project are different from the ones we mentioned in our proposal.

INTRODUCTION

The program will ask the user to input their preferences and then provide them with a list of 5 popular songs that align with their tastes from their chosen decade (1950s – 2010s). It then asks the user to rate the songs based on whether they like 5 recommendations or not. Based on their feedback; the program calculates its accuracy.

DATA DESCRIPTION

The program uses seven similar data sets (csv files) that were obtained from Kaggle. They were all uploaded by a Machine Learning Engineer at Polytechnic called, Nicholas Carbone. Each data set contains information on around a hundred popular songs from that decade (1950s to 2010s.) We believe that these data sets are reputable because they were collected from Spotify Organizer based on Spotify's playlists called, "*All out ...s*". Furthermore, Carbone's upload scored a 10/10 on usability by Kaggle (which means the data set is frequently updated and it has an overview, file, column, and license description.)

However, the 1950s data set contain 12 missing values in the top genre column.

Each row in the data set contains information on one popular song from that decade. All the data sets have 15 columns. Figure 1 provides a picture of two rows in the data sets, and Table 1 provides their names, description, type, and range. However, we only used 8 columns in our program: artist, top genre, year, bpm, dncc, val, live, and accous.

Figure 1 – Two rows in the data set

Number	title	artist	top genre	year	bpm	nrgy	dnce	dB	live	val	dur	acous	spch	pop
86	Five More	Deorro	big room	2017	128	88	70	-3	82	50	212	3	22	69
57	Time of O	Pitbull	dance pop	2014	124	80	72	-6	69	72	229	9	6	77

Table 1 – The data sets' column details

No.	Column Name	description	Type	Range
1	Number	Identity	Integer	1-105
2	title	Name of the song	String	-
3	artist	Singer's name	String	-
4	top genre	Genre of song	String	-
5	year	The year the song released	Integer	1950 - 2019
6	bpm	Beats per minute of the song	Integer	60 - 200
7	nrgy	Energy of a song, the higher the value the more energetic the song is.	Integer	0 - 100
8	dnce	The higher the value, the easier it is to dance to this song.	Integer	0 – 100
9	dB	The higher the value, the louder the song.	Integer	-25 – 0
10	live	The higher the value, the more likely the song is a live recording.	Integer	0 – 100
11	val	The higher the value, the more the positive feel for the song.	Integer	0 – 100
12	dur	The duration of the song.	Integer	115 - 512
13	acous	The higher the value the more acoustic the song is.	Integer	0 – 100
14	spch	The higher the value the more spoken words the song contains.	Integer	0 – 50
15	pop	The higher the value the more popular the song is.	Integer	0 – 100

ALGORITHM DESCRIPTION

1. Wrote a function that parses csv files into a table – a list
2. Wrote functions that get specific columns in a row
3. Wrote filter and grouping functions that edit a table by user specified categories
4. Wrote a main function which has:
 - a. An input statement gets the user's preferences
 - b. Conditionals that call prior functions to filter the table based on the user's input

- c. A for loop that prints a list of 5 randomly selected songs from the filtered list
- d. An input statement to get user's ratings of the 5 songs
- e. Statements that calculate the accuracy of the program

Note – The code for our project took inspiration from Titus H. Klinge's list expressions that he presented in his CS 065 (Fall 2020) at Drake University. We also adopted some of our assignment 8 code.

ANALYSIS AND OBJECTIONS

We found that the numerical values varied with the decades. For instance, the average tempo of the popular songs got faster over time.

We had to find new data sets to the ones we mentioned in our proposal because the observations did not match up and they did not have genre column in one csv. Our new data sets have about 700 songs in total whereas the original one had a lot more. When grouping our data for the preferences based on numerical values, we had to do some research on how to segregate the ranges.

INSTRUCTIONS FOR RUNNING CODE

The following are basic instructions on how to run the code:

1. Enter decade (choose from 1950, 1960, 1970, 1980, 1990, 2000, 2010)
2. Enter at least 4 categories you are interested in (Choose from Artist/Genre/Release year/Tempo/Danceability/Valence/Liveness/Acousticness)
3. Rate the songs from 1 to 5 based on your liking of the songs picked

As we did not have time to add error statements for invalid entries, below are two valid that can be tried to see how our program works

Figure 2 – First example

```
Choose a decade from 1950 to 2010:1950
Select by which categories you want to narrow your search by: Artist/Genre/Release year/Tempo/Danceability/Valence/Liveness/Acousticness:
year,valence,liveness,acousticness
Choose a year: 1957
Choose from Sad/Neutral/Happy/ valence: Sad
Choose from Not live/Live feel:Live feel
Choose from Low/Medium/High ,acousticness: Medium
```

Figure 3 – Second Example

```
Choose a decade from 1950 to 2010:2010
Select by which categories you want to narrow your search by: Artist/Genre/Release year/Tempo/Danceability/Valence/Liveness/Acousticness:
artist,genre,tempo,danceability
Choose an artist: Post Malone
Choose a genre: electropop
Choose from slow/medium/fast/ tempo: medium
Choose from Not danceable/Danceable/Very danceable: Very danceable
```

CITATIONS

- To find the bounds for our grouping functions we used -
<https://medium.com/@boplantinga/what-do-spotifys-audio-features-tell-us-about-this-year-s-eurovision-song-contest-66ad188e112a#:~:text=Liveness%3A%20Detects%20the%20presence%20of,averaged%20across%20the%20entire%20track>
- To find an implicit Boolean for of an empty list we used -
<https://stackoverflow.com/questions/53513/how-do-i-check-if-a-list-is-empty>