Music Analysis and Recommendation from Spotify

Walid Harkous and Eric Argenio
School of Computing and Data Science
Wentworth Institute of Technology
Boston, MA 02115, USA
{harkousw, argenioe}@wit.edu

***Abstract*— Music is one of the few things that all people can connect over. Whether music is being played socially or in private, it's important that people always have fresh music to enjoy listening to. Why spend hours looking for new music when you can have songs recommended based on what you actively enjoy? This research was conducted to analyze the listening habits of people by examining real data of playlists created by real people.**

***Keywords*— Spotify, Music, Big Data, Analysis**

INTRODUCTION

The focus of this research project is to dive deep into playlists created by real Spotify users and the tracks within and look at any trends and patterns. We want to answer basic questions such as which artists were a big hit and during which year? Or what was the hottest track during a certain period of time? We also want to categorize the playlists based on the genres and description of the playlists.

The results of this research project could be useful in many different areas. By looking at the trends of what songs are popular and recognizing patterns that make a popular song popular, we could begin to predict whether or not a given song would have been popular in a certain year or even begin to make prediction models on what the next popular genre and sub-genre would be.

The results of this analysis will also provide a new way to recommend more songs based on a given playlist of predetermined songs. A small part of this project is to be able to use some of the data and results as a supplement to a recommendation algorithm that provides more songs to “complete the playlist”.

The goal of this research project is to use our previous knowledge as well as apply all of the concepts learned in the COMP3800 Big Data Course. Using Hadoop, PySpark, Pandas, and Jupyter Notebook, we will analyze the public dataset of playlists provided by Spotify.

RELEVENT WORK

Music analysis is not anything new. Producers and record companies are always aware of who the top artists and hits are at any given moment. But there is not a lot of research on what makes up a playlist and why people will group up

songs together, or which songs would go well together in a playlist, for example.

This research project was partially inspired by the challenge put out by Spotify's R&D team. The challenge was simple: "given a seed playlist title and/or initial set of tracks in a playlist, to predict the subsequent tracks in that playlist".

Many submissions have already been posted and there are many notebooks and discussions out there that analyze the data in their own way. We drew some inspiration from each, but none of them used Spark to analyze the data and answer the questions we wanted answered. Most of the analysis out there used python and Pandas to display the results visually in the form of graphs and plots- which we do as well.

DESCRIPTION OF THE PROJECT

The implementation of this project was straight-forward. Set up the project environment and notebook, import the data, normalize the data, and then answer some questions while analyzing the data using PySpark.

The first part of the project included making sure that Jupyter Notebook was running and set up with PySpark, and then hosting the project on GitHub for collaboration. This was the same thing that we have done for every assignment and lab, so it was not very difficult to get set up and ready to start analyzing some data.

The second part of the project was getting the data into the notebook. The dataset we used consisted of about 1000 files, each filled with playlists, information regarding the playlists, as well as the tracks inside the playlists and information regarding the tracks. In total, we have about 30Gb of data that we are working with.

```
+--------------------+--------------------+
|                info|           playlists|
+--------------------+--------------------+
|{2017-12-03 08:41...|[{false, null, 11...|
+--------------------+--------------------+

root
 |-- info: struct (nullable = true)
 |    |-- generated_on: string (nullable = true)
 |    |-- slice: string (nullable = true)
 |    |-- version: string (nullable = true)
 |-- playlists: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- collaborative: string (nullable = true)
 |    |    |-- description: string (nullable = true)
 |    |    |-- duration_ms: long (nullable = true)
 |    |    |-- modified_at: long (nullable = true)
 |    |    |-- name: string (nullable = true)
 |    |    |-- num_albums: long (nullable = true)
 |    |    |-- num_artists: long (nullable = true)
 |    |    |-- num_edits: long (nullable = true)
 |    |    |-- num_followers: long (nullable = true)
 |    |    |-- num_tracks: long (nullable = true)
 |    |    |-- pid: long (nullable = true)
 |    |    |-- tracks: array (nullable = true)
 |    |    |    |-- element: struct (containsNull = true)
 |    |    |    |    |-- album_name: string (nullable = true)
 |    |    |    |    |-- album_uri: string (nullable = true)
 |    |    |    |    |-- artist_name: string (nullable = true)
 |    |    |    |    |-- artist_uri: string (nullable = true)
 |    |    |    |    |-- duration_ms: long (nullable = true)
 |    |    |    |    |-- pos: long (nullable = true)
 |    |    |    |    |-- track_name: string (nullable = true)
 |    |    |    |    |-- track_uri: string (nullable = true)
```

Figure 1. Main Dataframe Schema

There was a lot of useless information that was not necessary for our analysis and would only slow operation times. The playlist column was also deeply nested with its own sub-columns that were not immediately visible when displaying this Dataframe (*fig. 1*). So, some flattening and normalization of the data was needed.

z

```
+-------------+-----------+-----------+-----------+----------------+
---+---+--------------------+
|collaborative|description|duration_ms|modified_at|            name|
cks|pid|              tracks|
+-------------+-----------+-----------+-----------+----------------+
---+---+--------------------+
|        false|       null|   11532414| 1493424000|      Throwbacks|
52|  0|[{The Cookbook, s...|
|        false|       null|   11656470| 1506556800|Awesome Playlist|
39|  1|[{Eye Of The Tige...|
|        false|       null|   14039958| 1505692800|         korean |
64|  2|[{On And On, spot...|
+-------------+-----------+-----------+-----------+----------------+
---+---+--------------------+
only showing top 3 rows

root
```

Figure 2. Playlist Dataframe top 3 rows

```
root
 |-- collaborative: string (nullable = true)
 |-- description: string (nullable = true)
 |-- duration_ms: long (nullable = true)
 |-- modified_at: long (nullable = true)
 |-- name: string (nullable = true)
 |-- num_albums: long (nullable = true)
 |-- num_artists: long (nullable = true)
 |-- num_edits: long (nullable = true)
 |-- num_followers: long (nullable = true)
 |-- num_tracks: long (nullable = true)
 |-- pid: long (nullable = true)
 |-- tracks: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- album_name: string (nullable = true)
 |    |    |-- album_uri: string (nullable = true)
 |    |    |-- artist_name: string (nullable = true)
 |    |    |-- artist_uri: string (nullable = true)
 |    |    |-- duration_ms: long (nullable = true)
 |    |    |-- pos: long (nullable = true)
 |    |    |-- track_name: string (nullable = true)
 |    |    |-- track_uri: string (nullable = true)
```

Figure 3. Playlist Dataframe Schema

The result after removing the info column and exploding each sub-column in the playlists into its own column can be seen in (*fig. 2).* But we still have another nested array of tracks within each playlist (*fig. 3).* We do the same thing we did to the playlists column and explode each track into its own row, creating a whole new Dataframe of tracks (*fig. 4).* This gives us two different Dataframes that we can analyze- The playlists and the tracks.

```
+--------------------+-----------------+--------------------+-----------+
|          album_name|      artist_name|          track_name|duration_ms|
+--------------------+-----------------+--------------------+-----------+
|        The Cookbook|     Missy Elliott|Lose Control (fea...|     226863|
|         In The Zone|    Britney Spears|               Toxic|     198800|
|Dangerously In Lo...|          Beyoncé|       Crazy In Love|     235933|
|           Justified| Justin Timberlake|      Rock Your Body|     267266|
|            Hot Shot|           Shaggy|        It Wasn't Me|     227600|
|         Confessions|            Usher|               Yeah!|     250373|
|         Confessions|            Usher|             My Boo|     223440|
|                 PCD|The Pussycat Dolls|             Buttons|     225560|
|The Writing's On ...|   Destiny's Child|         Say My Name|     271333|
|Speakerboxxx/The ...|          OutKast|Hey Ya! - Radio M...|     235213|
+--------------------+-----------------+--------------------+-----------+
only showing top 10 rows
```

Figure 4. Tracks Dataframe top 10 rows

The third part to this project was to begin answering the questions we had and analyze the data.

I. Track Analysis

a. Who were the most popular artists in 2010-2017 in users' playlists?

```
+-----------------+-----+
|      artist_name|count|
+-----------------+-----+
|            Drake|  939|
|       Kanye West|  415|
|   Kendrick Lamar|  385|
|          Rihanna|  350|
|           Eminem|  332|
|       The Weeknd|  296|
|     Lil Uzi Vert|  292|
|       Ed Sheeran|  285|
|           Future|  265|
|      Chris Brown|  259|
|    Justin Bieber|  251|
|        Lil Wayne|  242|
|          Beyoncé|  234|
| The Chainsmokers|  232|
|Twenty One Pilots|  226|
|          Big Sean|  222|
|      Post Malone|  221|
|           J. Cole|  219|
|    Kenny Chesney|  204|
|         Maroon 5|  203|
+-----------------+-----+
only showing top 20 rows
```

Figure 5. Top Artists 2010-2017

b. What were the most popular tracks in 2010-2017 in users' playlists?

track_name	artist_name	count
One Dance	Drake	55
HUMBLE.	Kendrick Lamar	52
Broccoli (feat. L...	DRAM	50
Closer	The Chainsmokers	46
Congratulations	Post Malone	44
Don't Let Me Down	The Chainsmokers	42
Roses	The Chainsmokers	39
Bounce Back	Big Sean	39
iSpy (feat. Lil Y...	KYLE	39
Jumpman	Drake	39
Mask Off	Future	38
Bad and Boujee (f...	Migos	38
XO TOUR Llif3	Lil Uzi Vert	37
White Iverson	Post Malone	36
Panda	Desiigner	36

only showing top 15 rows

Figure 6. Top Tracks 2010-2017

c. What is the average track duration? What is the longest track? Shortest?

Avg Track Duration	Max Track Duration	Min Track Durat
3.89 mins	40.4 mins	0.5 m

Figure 7. Track Statistics

II. Playlist Analysis

a. What is the most common word in the playlist name?

name	count
Country	35
Songs	34
Music	28
Chill	24
Summer	23
Party	21
Rock	19
My	18
New	17
Mix	17
Playlist	17
Good	17
Christmas	16
Rap	14
The	13
Old	12
2016	11
You	10
Workout	10
Up	9

only showing top 20 rows

Figure 8. Common Keywords in Playlist Names

b. What is the average playlist duration? Is there a maximum duration to a playlist? min?

Avg Playlist Duration	Max Playlist Duration	Min Playlist Duration
262.34 mins	963.62 mins	16.72 mins

Figure 9. Playlist Duration Statistics

c. What is the average playlist track number? Is there a maximum number of tracks to a playlist? Min?

Avg Number of Tracks	Max Number of Tracks	Min Number of Tracks
67.503	245	5

Figure 10. Playlist Track Statistics

III. Plotting the Data

We took a sample playlist from one of the million and broke it down even further.

User Artists

	artist_name	count
0	A$AP Rocky	15
1	G-Eazy	11
2	Kendrick Lamar	8
3	ScHoolboy Q	5
4	Ludacris	3

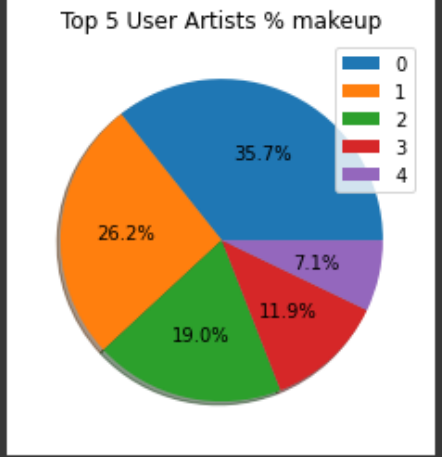

Figure 11. Artist Percentage Makeup in Playlist

Plotting the percent presence of individual artists out of their top 5 (*fig 11)*. We are able to determine the "hottest" artists in the specific playlist. The pie chart above shows the % makeup of each of the most popular artists. In this sample playlist, 35% of the music by their favorite artists are songs are by A$AP Rocky.

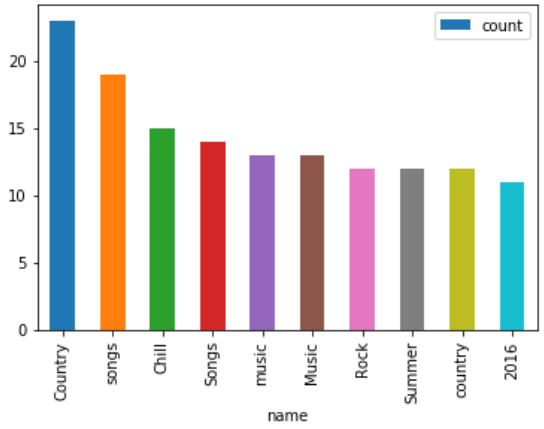

Figure 12. Playlist Keyword Frequency in Names

In this smaller slice of data, we gathered the most common words used to name each playlist. These are not entire playlist names. "Country" happened to be the most used word out of our sample of data.

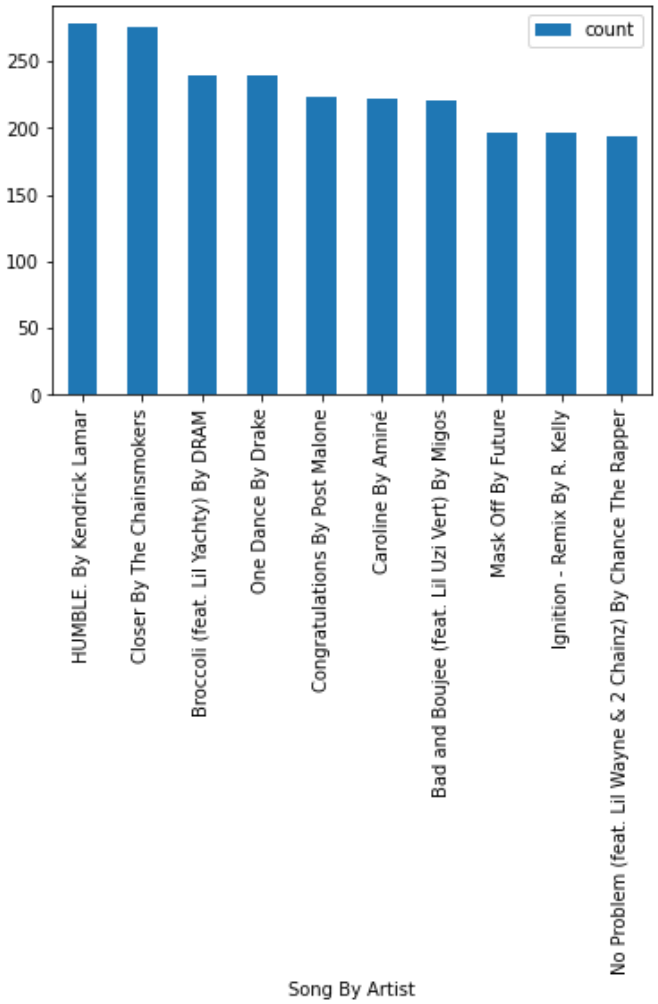

Figure 13. Song Frequency Across Playlists

The most common songs by artist across all playlists in our data sample. Despite Drake being the most popular artist across all playlists, this song by Kendrick Lamar made it into playlists more than the most popular song by Drake.

Results & Evaluation

Our project's implementation did an excellent job at answering the analytical questions we sought answers for at the beginning of the development period. The results were very much so in-line with what was expected from our own experience. The playlists in our data were all from 2010-2017 and the most popular music was what we were surrounded by the most at the time. Big rap artists like Drake, Kanye, and Eminem were in the top of our analyzing charts and even artists that were starting to gain traction during these years that are large now like Ed Sheeran are present on the top artists section of our tables. Other sources of this kind of music analysis like the Billboard 100 aren't as in-depth as the system we created. Our system was capable of tracking the amount of music an artist had in playlists globally, in an individual playlist, and the counts of each song through these two mediums as well.

Conclusion & Future Work

With extra time it would have been possible to add more functionality and answer deeper questions. One of the biggest issues our group sought to answer was how to recommend users music they were likely to enjoy. This ended up being a much larger problem to tackle and could've been finished with some more working hours being put into it. The biggest problem we ran into regarding this issue was properly filtering the song recommendations according to our proposed criteria. A little more time devoted to research and implementation could've resulted in a much more marketable deliverable, and an extremely useful and comprehensive music recommendation algorithm.

In addition to creating this algorithm, we would have enjoyed having more questions answered. The data included Spotify URIs to each of the songs and artists with valuable data able to be accessed through their API. Questions such as “What was the most popular year for music”, or “How many unpopular songs by popular artists made it into playlists” could have been answered. There is an entire section of data which was accessible to us but too large to cover

Investigating the data and Spark as a whole was fruitful not only for the questions we were able to answer, but also for the experience we accrued. Working with Spark and Pandas together to sort, filter, and efficiently work with data is a very valuable skill to have in the modern workplace.

in the limited scope of our project.

During the course of creating this project and data analysis, we had to overcome many roadblocks that were halting our progress. One of these roadblocks faced was the issue of formatting the data into workable chunks according to what each question required. Aggregating and filtering all of the user data in the specific way the questions of the project needed in order to be solved was an interesting challenge that was met with ample research and many hours of debugging and problem-solving.

If we were to start from scratch and begin the project anew, our group likely would have focused on the micro-applications of this larger set of data. While valuable conclusions can be drawn from the larger data analysis, the most marketable feature would have been the individual user deliverable – the recommendation algorithm. Much time was put into learning how to generate and plot data – mostly by cleaning the cluttering columns and rows from interfering with the more valuable pieces of data, Although the plots are beneficial to the project as a whole, more could have been done with them and the data they hold to give the user more music options to listen to.

REFERENCES

[1] https://developer.spotify.com/console/

[2] https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge