Structure of the Project:

Section 1: Introduction

There is no denying that the way we consume music has changed over the past few decades. Earlier, albums were sold in cassettes and compact disks(CDs) which everyone would buy from designated sellers. Now, with the rise of social media, things have definitely changed. Fewer people bother buying CDs from their favourite artists and instead turn to platforms like SoundCloud, Spotify, and Apple Music to listen to their favourite music. You wonder how do these record companies manage to hire music artists for record amounts and the answer lies in an increased dependency on Data Science. Data Science helps music companies to closely analyze trends and predict what their next big hit would be. They can easily take advantage of the vast amounts of data available to see the trajectory of the kind of music that appeals to a large audience and nudge their artists to produce such music.

What's the first thing that comes to your mind when your friend says that he has been hooked to a new song? Chances are, you think about a particular artist or band, maybe the chorus or the background music which really makes a song stand out. The reality is that big music companies have directed your attention towards a certain type of music- through years and years of data analysis- so that you are used to the kind of music they produce and more likely to listen. It's not a long stretch to say that music industries have designed their business model around making you accustomed to a certain type of music. The type of music is determined by music analytics and its potential to rise and compete with music produced by other music companies.

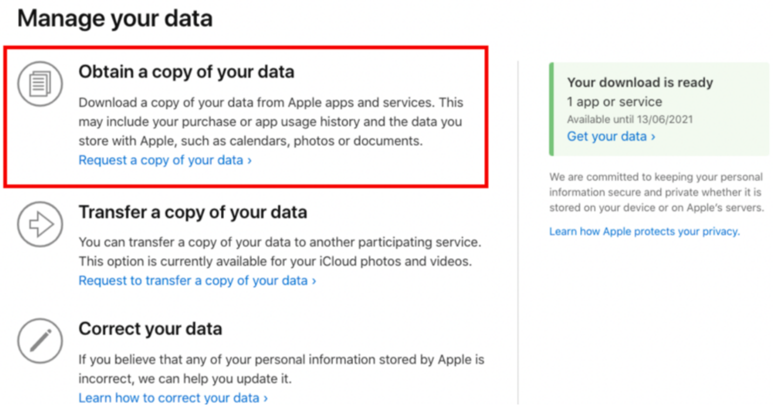
In conclusion, producing the next big hit isn’t about raw talent anymore; it’s about taking years of data into consideration and then choosing a song whose genre and lyrics have relevance to the time of release, which will cause it to go well with listeners. Music companies don’t have to depend on one artist either; in recent years, we’ve seen songs by previously unknown singers to become instant hits.

Section 2: Data Extraction:

Thanks to the European Union's General Data Protection Regulation (GDPR) requirements which were established in 2018, users have the right to access their personal data which includes information like basic identity information, webdata like location, IP addresses, cookie data. You can now request an electronic copy of your personal data, free of charge, upon request and can even inquire about how your data is used, stored, processed or transferred to other organizations. When I found out that I could request an archive with all my usage data since 2021, I requested a copy of my data following the required steps:

1. Head to Apple’s Data and Privacy log in page
2. Log in with the Apple ID for which you’d like to download data

3. Under Get a copy of your data and click Get Started



Manage your data

Obtain a copy of your data

Download a copy of your data from Apple apps and services. This may include your purchase or app usage history and the data you store with Apple, such as calendars, photos or documents.

Request a copy of your data ›

Transfer a copy of your data

You can transfer a copy of your data to another participating service. This option is currently available for your iCloud photos and videos.

Request to transfer a copy of your data ›

Correct your data

If you believe that any of your personal information stored by Apple is incorrect, we can help you update it.

Learn how to correct your data ›

Your download is ready

1 app or service

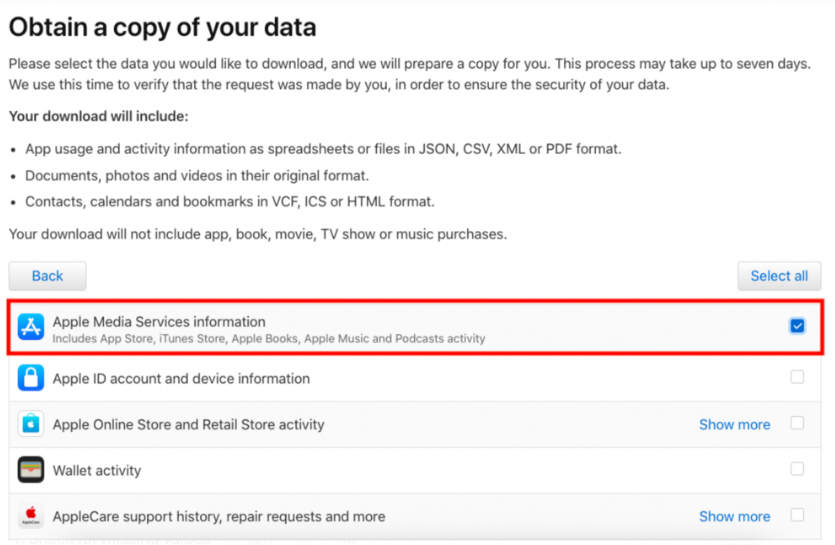
Available until 13/06/2021

Get your data ›

We are committed to keeping your personal information secure and private whether it is stored on your device or on Apple's servers.

Learn how Apple protects your privacy.

4. Select the data you’d like, 'App Store, Itunes Store, Apple Books and Apple Music'



Obtain a copy of your data

Please select the data you would like to download, and we will prepare a copy for you. This process may take up to seven days. We use this time to verify that the request was made by you, in order to ensure the security of your data.

Your download will include:

- App usage and activity information as spreadsheets or files in JSON, CSV, XML or PDF format.
- Documents, photos and videos in their original format.
- Contacts, calendars and bookmarks in VCF, ICS or HTML format.

Your download will not include app, book, movie, TV show or music purchases.

Back   Select all

Apple Media Services information ☑
Includes App Store, iTunes Store, Apple Books, Apple Music and Podcasts activity

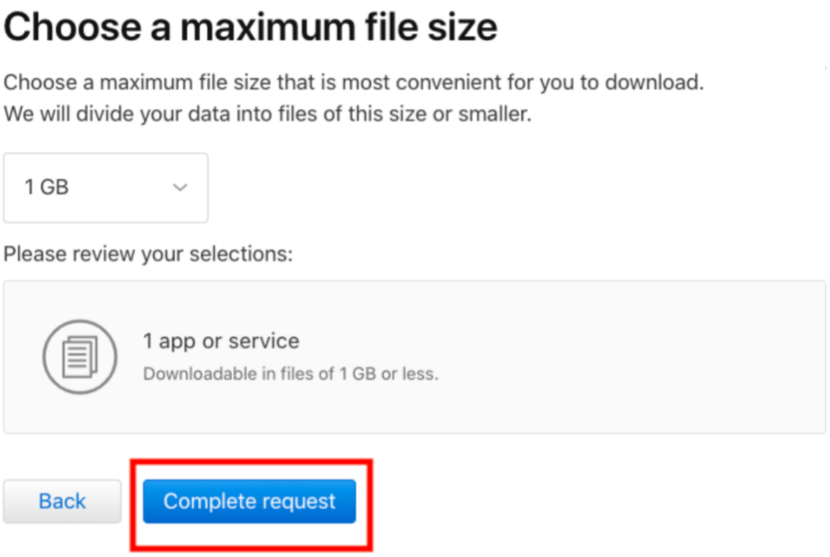
Apple ID account and device information ☐

Apple Online Store and Retail Store activity   Show more ☐

Wallet activity ☐

AppleCare support history, repair requests and more   Show more ☐

5. Choose the maximum default file size and click on Complete Request



Choose a maximum file size

Choose a maximum file size that is most convenient for you to download. We will divide your data into files of this size or smaller.

1 GB

Please review your selections:

1 app or service

Downloadable in files of 1 GB or less.

Back   Complete request

Section 3: Importing Data

Upon requesting the data to Apple, I received an archived zip file named Apple_Media_Services which contained a lot of folders. I investigated the Apple Music Activity folder which had a lot of files. Let’s look at what each of them contains:

- **Apple Music — Recently Played Containers** : albums, playlists recently played→ this is not relevant to figure out patterns or understand the usage of the service overtime
- **Apple Music — Recently Played Tracks :** tracks recently played→ this is not relevant to figure out patterns or understand the usage of the service
- **Apple Music Library Activity** : records all the actions performed with the library (either user actions, or automated software actions) → **relevant for our analysis!**
- **Apple Music Library Playlists** : describes the playlist created in the library (including its name, identifier, and the identifiers of each track it contains) → this is not relevant to figure out patterns or understand the usage of the service
- **Apple Music Library Tracks** : describes each track of the library (including its title, artist, genre, release year, album, when it was added to the library…) → **relevant for our analysis!**
- **Apple Music Likes and Dislikes**: lists the rating associated to a track, and when it was rated → **relevant for our analysis!**
- **Apple Music Play Activity**: lists the play activity history, with associated track info such as genre, provider, duration, activity timestamp and type, timezone → **extremely relevant for our analysis!**
- **Identifier Information**: matches a track title with an identifier → **this can be useful for our analysis!**
- **Music — Favorite Stations**: I believe this list the stations that you create → this is not relevant to figure out patterns or understand the usage of the service overtime

- **Music — Onboarding Artists**: the artists you select when first launching the service for Apple to start recommending you content → this is not relevant to figure out patterns or understand the usage of the service overtime

- **Music — Onboarding Genres**: the genres you select when first launching the service for Apple to start recommending you content → this is not relevant to figure out patterns or understand the usage of the service overtime

Out of all the content of the archive provided by Apple, I used the five files which are extremely relevant for our analysis. We clearly see two types of files: those that have information about tracks and listening activity, and the Apple Music Library Activity. This last file is going to be analysed independently of the others.

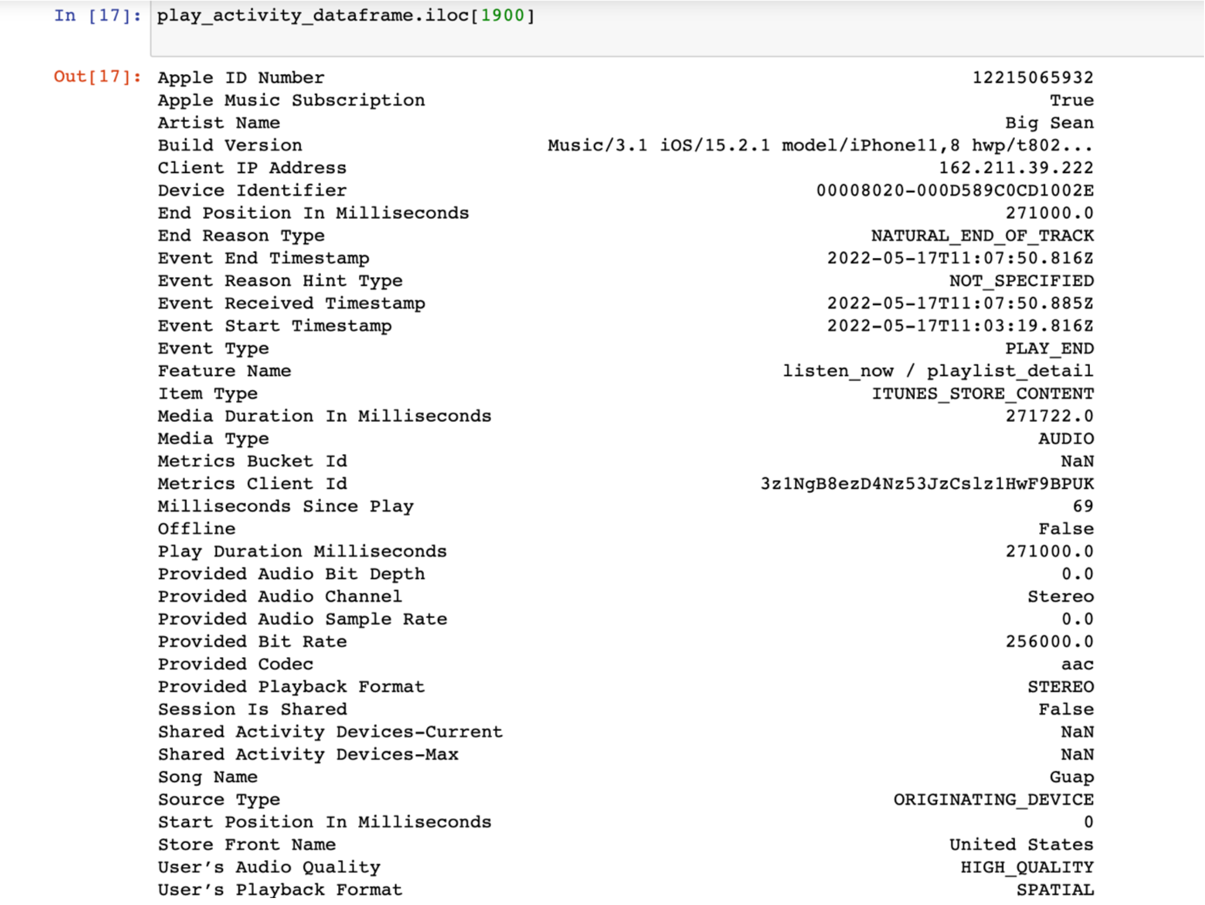
Section 4: Analyzing Play Activity:

Can I build a single dataframe that would allow me to build statistics and identify trends on the type of music I listen to, at what moment in time, if the trends change from month to month, how I usually find new tracks ?

The dataframe containing the most information about playing activity is Apple Music Play Activity. Hence, we use this dataframe as our base and enrich this dataframe with information obtained from other dataframes.

Step 1: Cleaning and Restructuring the Apple Music Play Activity Dataframe

Here, we have a look at how our dataframe looks like



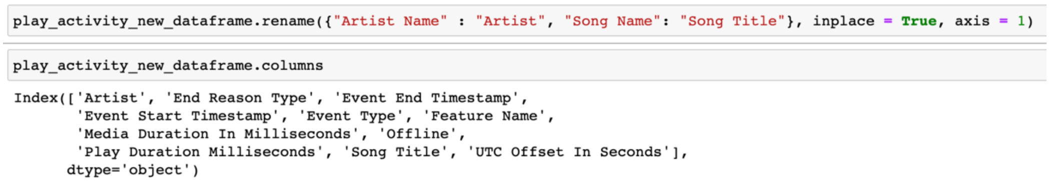
```
In [17]: play_activity_dataframe.iloc[1900]

Out[17]: Apple ID Number                                                          12215065932
         Apple Music Subscription                                                        True
         Artist Name                                                                 Big Sean
         Build Version                      Music/3.1 iOS/15.2.1 model/iPhone11,8 hwp/t802...
         Client IP Address                                                     162.211.39.222
         Device Identifier                                          00008020-000D589C0CD1002E
         End Position In Milliseconds                                                271000.0
         End Reason Type                                                 NATURAL_END_OF_TRACK
         Event End Timestamp                                         2022-05-17T11:07:50.816Z
         Event Reason Hint Type                                                 NOT_SPECIFIED
         Event Received Timestamp                                    2022-05-17T11:07:50.885Z
         Event Start Timestamp                                       2022-05-17T11:03:19.816Z
         Event Type                                                                  PLAY_END
         Feature Name                                            listen_now / playlist_detail
         Item Type                                                       ITUNES_STORE_CONTENT
         Media Duration In Milliseconds                                              271722.0
         Media Type                                                                     AUDIO
         Metrics Bucket Id                                                                NaN
         Metrics Client Id                                     3z1NgB8ezD4Nz53JzCslz1HwF9BPUK
         Milliseconds Since Play                                                           69
         Offline                                                                        False
         Play Duration Milliseconds                                                  271000.0
         Provided Audio Bit Depth                                                         0.0
         Provided Audio Channel                                                        Stereo
         Provided Audio Sample Rate                                                       0.0
         Provided Bit Rate                                                           256000.0
         Provided Codec                                                                   aac
         Provided Playback Format                                                      STEREO
         Session Is Shared                                                              False
         Shared Activity Devices-Current                                                  NaN
         Shared Activity Devices-Max                                                      NaN
         Song Name                                                                       Guap
         Source Type                                                       ORIGINATING_DEVICE
         Start Position In Milliseconds                                                     0
         Store Front Name                                                       United States
         User’s Audio Quality                                                    HIGH_QUALITY
         User’s Playback Format                                                       SPATIAL
```

At first glance, the following columns look interesting:

1. End Reason Type: To spot whether a track was skipped or played till the end of the track
2. Feature Name: To spot how the track was found
3. Artist Name and Song Name: To fetch information about a particular song
4. Event Start Timestamp: To identify when the track was listened to

The cleaning up of this dataframe will consist of the following steps:

1. **Rename the columns containing song title and artist**: We notice that this dataframe does not contain any ID number which can be used to match each row of this dataframe with information from other dataframes. Hence, we will rename the columns: Artist Name and Song Name and use these two columns for merging information from other dataframes.

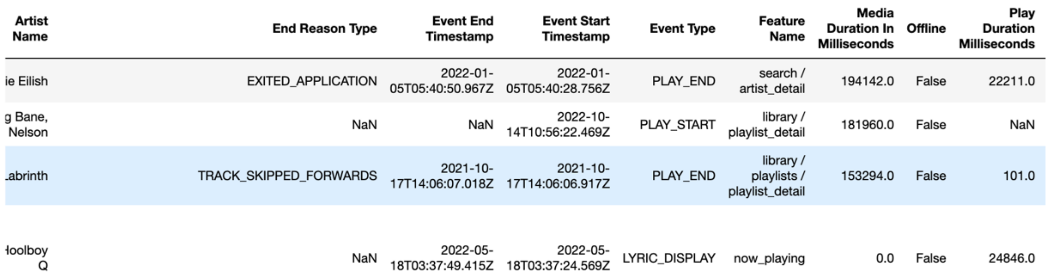


```
play_activity_new_dataframe.rename({"Artist Name" : "Artist", "Song Name": "Song Title"}, inplace = True, axis = 1)
```

```
play_activity_new_dataframe.columns
```

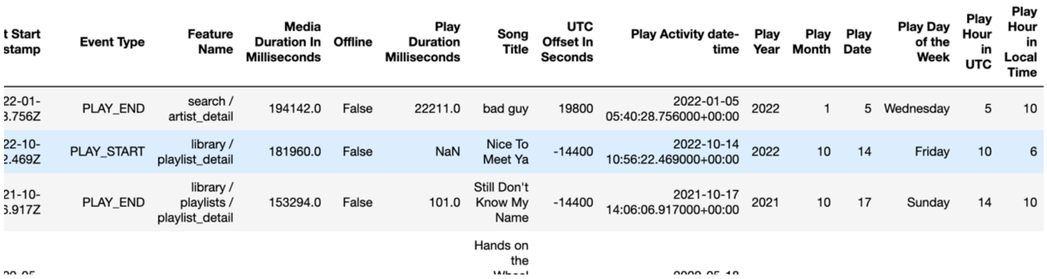
```
Index(['Artist', 'End Reason Type', 'Event End Timestamp',
       'Event Start Timestamp', 'Event Type', 'Feature Name',
       'Media Duration In Milliseconds', 'Offline',
       'Play Duration Milliseconds', 'Song Title', 'UTC Offset In Seconds'],
      dtype='object')
```

2. Then, we use the column "Event Start Timestamp" as a reference and when it is not available, we use the column "Event End Timestamp" as our reference point and obtain a timestamp column. Hence, this timestamp column is without any missing values.



Artist Name	End Reason Type	Event End Timestamp	Event Start Timestamp	Event Type	Feature Name	Media Duration In Milliseconds	Offline	Play Duration Milliseconds
ie Eilish	EXITED_APPLICATION	2022-01-05T05:40:50.967Z	2022-01-05T05:40:28.756Z	PLAY_END	search / artist_detail	194142.0	False	22211.0
g Bane, Nelson	NaN	NaN	2022-10-14T10:56:22.469Z	PLAY_START	library / playlist_detail	181960.0	False	NaN
.abrinth	TRACK_SKIPPED_FORWARDS	2021-10-17T14:06:07.018Z	2021-10-17T14:06:06.917Z	PLAY_END	library / playlists / playlist_detail	153294.0	False	101.0
ꟻoolboy Q	NaN	2022-05-18T03:37:49.415Z	2022-05-18T03:37:24.569Z	LYRIC_DISPLAY	now_playing	0.0	False	24846.0

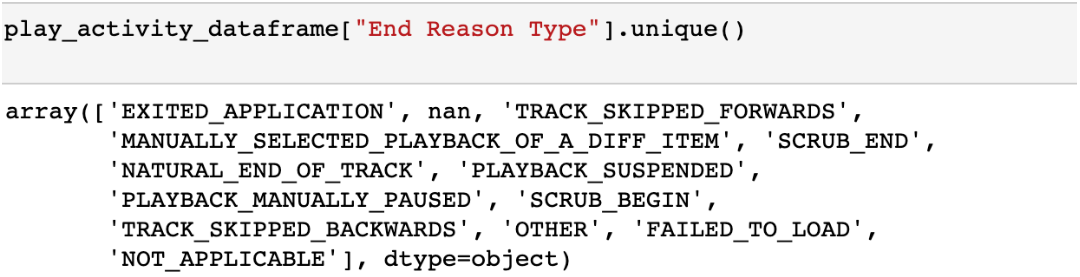
Using this time stamp column(Play Activity date-time column), I extract year, month, day of the month, and hour of the day for each track in UTC and local time and store them in different time columns.



t Start stamp	Event Type	Feature Name	Media Duration In Milliseconds	Offline	Play Duration Milliseconds	Song Title	UTC Offset In Seconds	Play Activity date-time	Play Year	Play Month	Play Date	Play Day of the Week	Play Hour in UTC	Play Hour in Local Time
22-01- 3.756Z	PLAY_END	search / artist_detail	194142.0	False	22211.0	bad guy	19800	2022-01-05 05:40:28.756000+00:00	2022	1	5	Wednesday	5	10
22-10- 2.469Z	PLAY_START	library / playlist_detail	181960.0	False	NaN	Nice To Meet Ya	-14400	2022-10-14 10:56:22.469000+00:00	2022	10	14	Friday	10	6
21-10- 5.917Z	PLAY_END	library / playlists / playlist_detail	153294.0	False	101.0	Still Don't Know My Name	-14400	2021-10-17 14:06:06.917000+00:00	2021	10	17	Sunday	14	10
						Hands on the								

3. Add a column with a flag for partial vs complete listening of a given track:

Now, we will add a column that would indicate partial vs complete listening of the song.



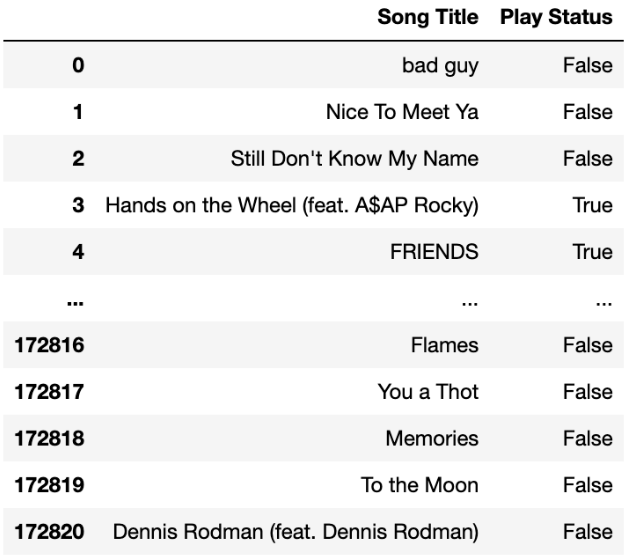
```
play_activity_dataframe["End Reason Type"].unique()
```

```
array(['EXITED_APPLICATION', nan, 'TRACK_SKIPPED_FORWARDS',
       'MANUALLY_SELECTED_PLAYBACK_OF_A_DIFF_ITEM', 'SCRUB_END',
       'NATURAL_END_OF_TRACK', 'PLAYBACK_SUSPENDED',
       'PLAYBACK_MANUALLY_PAUSED', 'SCRUB_BEGIN',
       'TRACK_SKIPPED_BACKWARDS', 'OTHER', 'FAILED_TO_LOAD',
       'NOT_APPLICABLE'], dtype=object)
```

So, for any given song, we can use "End reason Type" to identify:

1. Whether a song was skipped or listened partially(TRACK_SKIPPED_FORWARDS, TRACK_SKIPPED_BACKWARDS, SCRUB_BEGIN)
2. Listened to entirely(NATURAL_END_OF_TRACK)

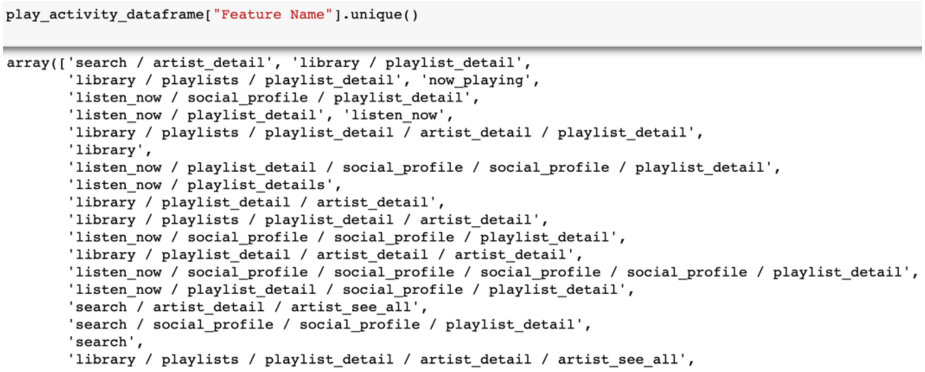
If the "End Reason Type" for a particular song is Natural_End_Of_Track and if the play duration is above the media duration in milliseconds, we consider the track to be listened to completely.



	Song Title	Play Status
0	bad guy	False
1	Nice To Meet Ya	False
2	Still Don't Know My Name	False
3	Hands on the Wheel (feat. A$AP Rocky)	True
4	FRIENDS	True
...	...	...
172816	Flames	False
172817	You a Thot	False
172818	Memories	False
172819	To the Moon	False
172820	Dennis Rodman (feat. Dennis Rodman)	False

This is how we use the Play Status column as an indicator for every song. If it’s True, the song was listened entirely and if it’s False, the song was skipped.

4. **Add a column with a simplified ‘origin’ of the song**: The image below demonstrates how Apple stores the information regarding the origin of a song.

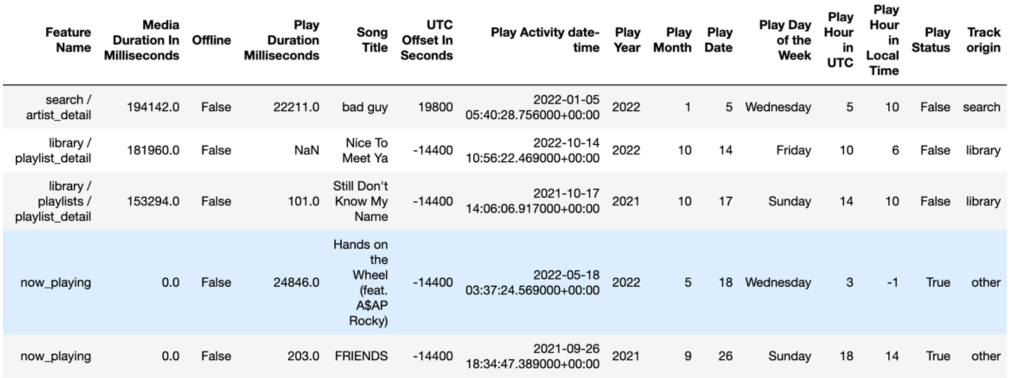


```
play_activity_dataframe["Feature Name"].unique()
```

```
array(['search / artist_detail', 'library / playlist_detail',
       'library / playlists / playlist_detail', 'now_playing',
       'listen_now / social_profile / playlist_detail',
       'listen_now / playlist_detail', 'listen_now',
       'library / playlists / playlist_detail / artist_detail / playlist_detail',
       'library',
       'listen_now / playlist_detail / social_profile / social_profile / playlist_detail',
       'listen_now / playlist_details',
       'library / playlist_detail / artist_detail',
       'library / playlists / playlist_detail / artist_detail',
       'listen_now / social_profile / social_profile / playlist_detail',
       'library / playlist_detail / artist_detail / artist_detail',
       'listen_now / social_profile / social_profile / social_profile / social_profile / playlist_detail',
       'listen_now / playlist_detail / social_profile / playlist_detail',
       'search / artist_detail / artist_see_all',
       'search / social_profile / social_profile / playlist_detail',
       'search',
       'library / playlists / playlist_detail / artist_detail / artist_see_all',
```

We can use this column to filter and find the origin of the song. I categorised them into four categories:

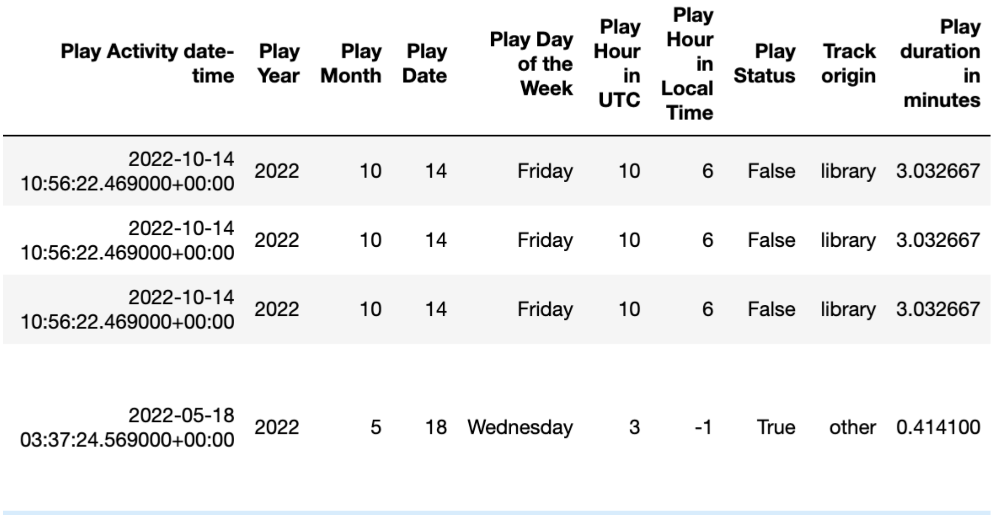
1. Search (this category includes songs that I have browsed manually on the app)
2. Library (this category includes songs that I have listened through my own playlists)
3. Radio (this category includes songs that I have listened through the Listen Now feature on Apple Music which usually plays my favourite songs and provides me with personalized recommendations)
4. Other(this category includes songs played using Siri, Alexa)



Feature Name	Media Duration In Milliseconds	Offline	Play Duration Milliseconds	Song Title	UTC Offset In Seconds	Play Activity date-time	Play Year	Play Month	Play Date	Play Day of the Week	Play Hour in UTC	Play Hour in Local Time	Play Status	Track origin
search / artist_detail	194142.0	False	22211.0	bad guy	19800	2022-01-05 05:40:28.756000+00:00	2022	1	5	Wednesday	5	10	False	search
library / playlist_detail	181960.0	False	NaN	Nice To Meet Ya	-14400	2022-10-14 10:56:22.469000+00:00	2022	10	14	Friday	10	6	False	library
library / playlists / playlist_detail	153294.0	False	101.0	Still Don't Know My Name	-14400	2021-10-17 14:06:06.917000+00:00	2021	10	17	Sunday	14	10	False	library
now_playing	0.0	False	24846.0	Hands on the Wheel (feat. A$AP Rocky)	-14400	2022-05-18 03:37:24.569000+00:00	2022	5	18	Wednesday	3	-1	True	other
now_playing	0.0	False	203.0	FRIENDS	-14400	2021-09-26 18:34:47.389000+00:00	2021	9	26	Sunday	18	14	True	other

The image above demonstrates how the TrackOrigin column stores the information regarding the origin of a song.

5. Add a column with a calculation of the listening duration in minutes: We use appropriate nesting to handle two specific types of cases: songs with no NA values for both Start and End Timestamps and songs with missing values in one of these two columns. To handle the latter, we make use of the Play Status column that we just added to the data frame.



Play Activity date-time	Play Year	Play Month	Play Date	Play Day of the Week	Play Hour in UTC	Play Hour in Local Time	Play Status	Track origin	Play duration in minutes
2022-10-14 10:56:22.469000+00:00	2022	10	14	Friday	10	6	False	library	3.032667
2022-10-14 10:56:22.469000+00:00	2022	10	14	Friday	10	6	False	library	3.032667
2022-10-14 10:56:22.469000+00:00	2022	10	14	Friday	10	6	False	library	3.032667
2022-05-18 03:37:24.569000+00:00	2022	5	18	Wednesday	3	-1	True	other	0.414100

The image above demonstrates that the column PlayDurationinMinutes which is derived using the existing information in our dataframe.

6. **Remove outliers of listening duration:** We remove the outliers if it has a value for listening duration above the 99th percentile and replace this value by the duration of the media in Milliseconds.

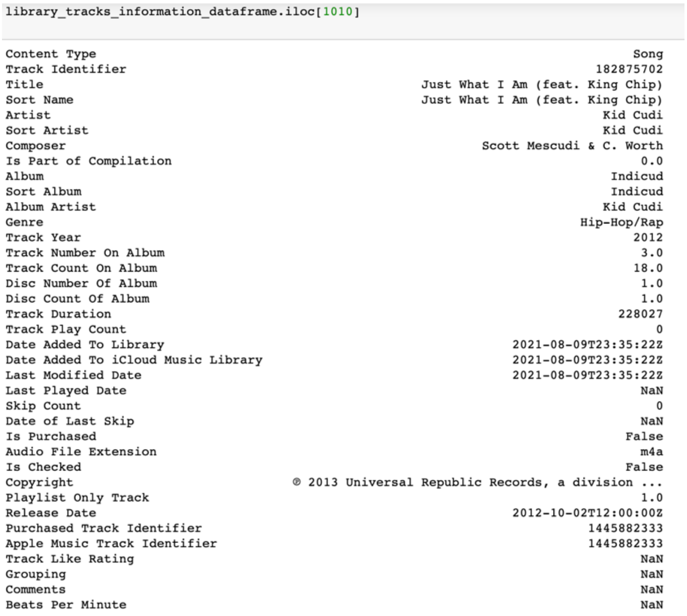
7. **Drop unused columns:** I spent some time cleaning up this dataframe to get a simplified dataframe which is easy to work with. Not all columns in this dataframe are useful for data analysis; hence, the drop() function is used to remove any unwanted columns. We get rid of the following columns:

1. Apple ID Number
2. Apple Music Subscription
3. Build Version
4. Client IP Address
5. Device Identifier
6. End Position in Milliseconds
7. Event Reason Hint Type
8. Event Received Timestamp
9. Item Type
10. Media Type
11. Metrics Bucket Id
12. Metrics Client Id
13. Milliseconds Since Play
14. Provided Audio Bit Depth
15. Provided Audio Channel
16. Provided Audio Sample Rate
17. Provided Bit Rate
18. Provided Codec
19. Provided Playback Format

20. Session Is Shared
21. Shared Activity Devices-Current
22. Shared Activity Devices-Max
23. Source Type
24. Start Position in Milliseconds
25. Store Country Name
26. User’s Audio Quality
27. User’s Playback Format

Section 5: Restructuring Library Tracks Related Information Dataframe

Here, we look at the structure of this data frame:



```
library_tracks_information_dataframe.iloc[1010]
```

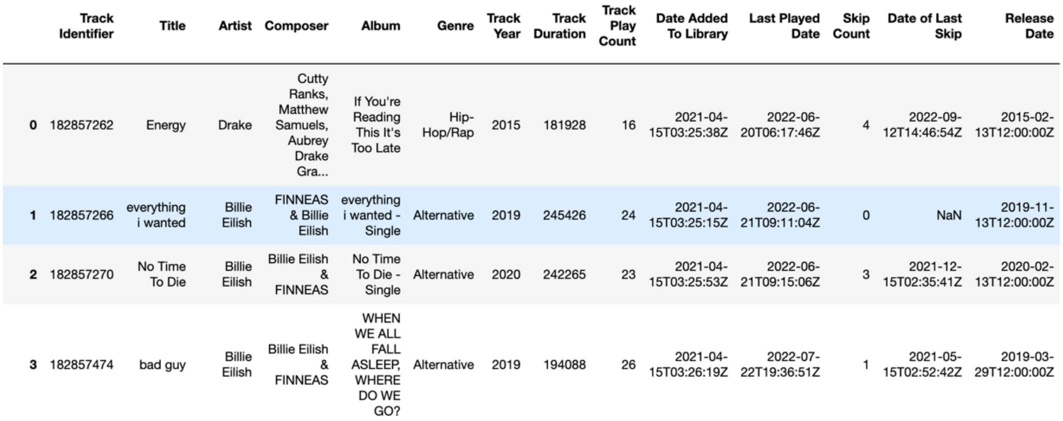
```
Content Type                                                                      Song
Track Identifier                                                             182875702
Title                                                 Just What I Am (feat. King Chip)
Sort Name                                             Just What I Am (feat. King Chip)
Artist                                                                        Kid Cudi
Sort Artist                                                                   Kid Cudi
Composer                                                      Scott Mescudi & C. Worth
Is Part of Compilation                                                             0.0
Album                                                                          Indicud
Sort Album                                                                     Indicud
Album Artist                                                                  Kid Cudi
Genre                                                                      Hip-Hop/Rap
Track Year                                                                        2012
Track Number On Album                                                              3.0
Track Count On Album                                                              18.0
Disc Number Of Album                                                               1.0
Disc Count Of Album                                                                1.0
Track Duration                                                                  228027
Track Play Count                                                                     0
Date Added To Library                                             2021-08-09T23:35:22Z
Date Added To iCloud Music Library                                2021-08-09T23:35:22Z
Last Modified Date                                                2021-08-09T23:35:22Z
Last Played Date                                                                   NaN
Skip Count                                                                           0
Date of Last Skip                                                                  NaN
Is Purchased                                                                     False
Audio File Extension                                                               m4a
Is Checked                                                                       False
Copyright                            ® 2013 Universal Republic Records, a division ...
Playlist Only Track                                                                1.0
Release Date                                                      2012-10-02T12:00:00Z
Purchased Track Identifier                                                  1445882333
Apple Music Track Identifier                                                1445882333
Track Like Rating                                                                  NaN
Grouping                                                                           NaN
Comments                                                                           NaN
Beats Per Minute                                                                   NaN
```

With this dataframe, there is not much to do really, besides dropping some columns that are not used later on. So, I dropped the following columns from the dataframe:

1. Content Type
2. Sort Name
3. Sort Artist
4. Is Part of Compilation
5. Sort Album
6. Album Artist
7. Track Number on Album
8. Track Count on Album
9. Disc Number of Album
10. Disc Count of Album
11. Date Added To iCloud Music Library
12. Last Modified Date
13. Is Purchased
14. Audio File Extension
15. Is Checked
16. Copyright
17. Playlist Only Track

18. Grouping
19. Comments
20. Beats Per Minute
21. Rating
22. Album Rating
23. Remember Playback Position
24. Album Like Rating
25. Album Rating Method
26. Work Name
27. Movement Name
28. Movement Number
29. Movement Count
30. Display Work Name

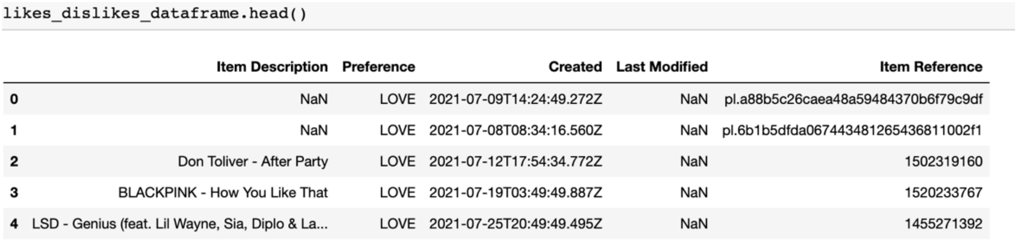
This is how the dataframe looks like after removing the unnecessary columns:



	Track Identifier	Title	Artist	Composer	Album	Genre	Track Year	Track Duration	Track Play Count	Date Added To Library	Last Played Date	Skip Count	Date of Last Skip	Release Date
0	182857262	Energy	Drake	Cutty Ranks, Matthew Samuels, Aubrey Drake Gra...	If You're Reading This It's Too Late	Hip-Hop/Rap	2015	181928	16	2021-04-15T03:25:38Z	2022-06-20T06:17:46Z	4	2022-09-12T14:46:54Z	2015-02-13T12:00:00Z
1	182857266	everything i wanted	Billie Eilish	FINNEAS & Billie Eilish	everything i wanted - Single	Alternative	2019	245426	24	2021-04-15T03:25:15Z	2022-06-21T09:11:04Z	0	NaN	2019-11-13T12:00:00Z
2	182857270	No Time To Die	Billie Eilish	Billie Eilish & FINNEAS	No Time To Die - Single	Alternative	2020	242265	23	2021-04-15T03:25:53Z	2022-06-21T09:15:06Z	3	2021-12-15T02:35:41Z	2020-02-13T12:00:00Z
3	182857474	bad guy	Billie Eilish	Billie Eilish & FINNEAS	WHEN WE ALL FALL ASLEEP, WHERE DO WE GO?	Alternative	2019	194088	26	2021-04-15T03:26:19Z	2022-07-22T19:36:51Z	1	2021-05-15T02:52:42Z	2019-03-29T12:00:00Z

Section 6: Restructuring Likes and Dislikes Dataframe

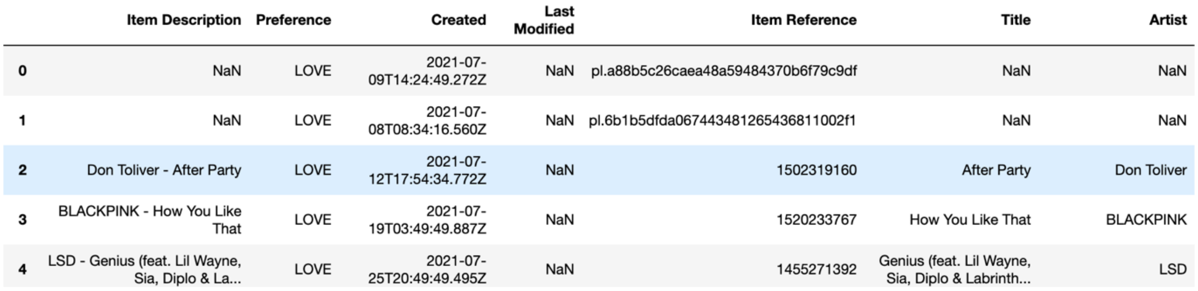
Here, we look at the original structure of this dataframe:



```
likes_dislikes_dataframe.head()
```

	Item Description	Preference	Created	Last Modified	Item Reference
0	NaN	LOVE	2021-07-09T14:24:49.272Z	NaN	pl.a88b5c26caea48a59484370b6f79c9df
1	NaN	LOVE	2021-07-08T08:34:16.560Z	NaN	pl.6b1b5dfda067443481265436811002f1
2	Don Toliver - After Party	LOVE	2021-07-12T17:54:34.772Z	NaN	1502319160
3	BLACKPINK - How You Like That	LOVE	2021-07-19T03:49:49.887Z	NaN	1520233767
4	LSD - Genius (feat. Lil Wayne, Sia, Diplo & La...	LOVE	2021-07-25T20:49:49.495Z	NaN	1455271392

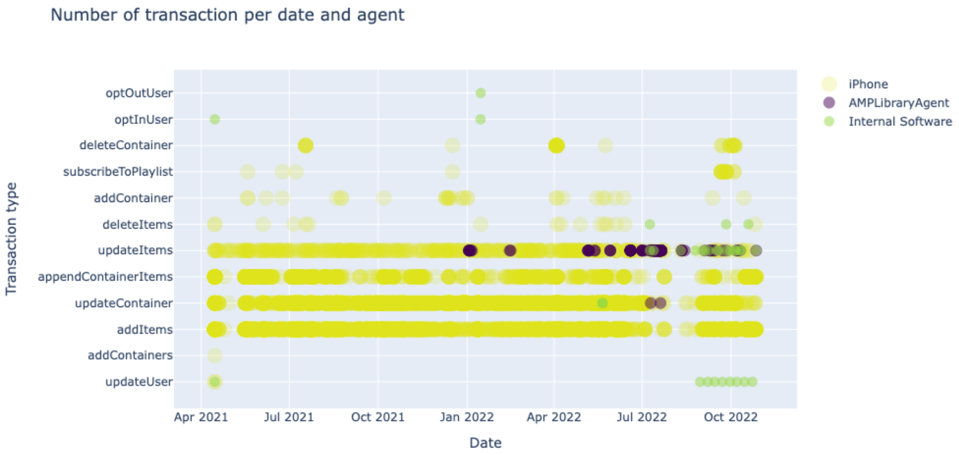
As we see that the name of the Artist and the Song are stored in the same column ‘Item Description’, I parse this column to create two new columns, each storing the Song Title and Artist Name



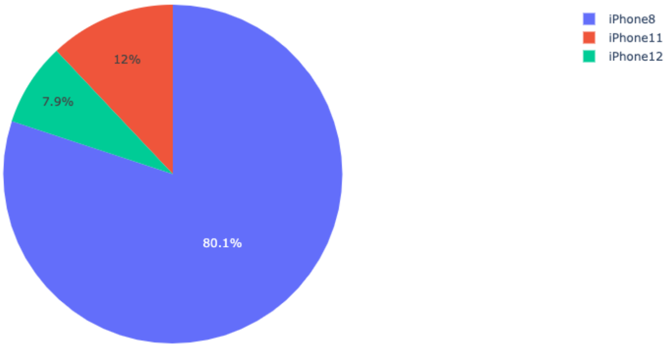
	Item Description	Preference	Created	Last Modified	Item Reference	Title	Artist
0	NaN	LOVE	2021-07-09T14:24:49.272Z	NaN	pl.a88b5c26caea48a59484370b6f79c9df	NaN	NaN
1	NaN	LOVE	2021-07-08T08:34:16.560Z	NaN	pl.6b1b5dfda067443481265436811002f1	NaN	NaN
2	Don Toliver - After Party	LOVE	2021-07-12T17:54:34.772Z	NaN	1502319160	After Party	Don Toliver
3	BLACKPINK - How You Like That	LOVE	2021-07-19T03:49:49.887Z	NaN	1520233767	How You Like That	BLACKPINK
4	LSD - Genius (feat. Lil Wayne, Sia, Diplo & La...	LOVE	2021-07-25T20:49:49.495Z	NaN	1455271392	Genius (feat. Lil Wayne, Sia, Diplo & Labrinth...	LSD

Section 7: Analyzing Library Activity Dataframe

I wanted to visualize a timeline of the activity inside my library. The Y-Axis in the plot below demonstrates all the actions performed inside my library from three different sources: iPhone, AMPLibrary Agent, and Internal Software. The X-Axis demonstrates the period since I have started using the application.



- The platform is mostly used through my iPhone
- The Internal software was most active during August- October 2022.



This pie-chart demonstrates the distribution of my library activity across three models of the iPhone I have used in the past one and a half year. I have been most active using this application through my iPhone 8

Section 8: Building a data structure for Tracks

After cleaning and restructuring all the data frames, I needed a way to match items from one data frame to another. As we don't have a column with an unique identifier which could be used to match songs from one dataframe to another dataframe, I created a Track class instance which stores all the information for each song from all the dataframes that we have restructured. The idea is to create a new data structure named Track and for each instance, we use this Track class to update information from the various dataframes.

- For each input dataframe, we try to identify the rows that represents a song we already saw and for which we already have an instance. We use a similarity score between the 'Title && Artist' string combinations to know whether we have seen that song before (i.e we already have a track instance for a given item). For example, comparing 'Bad Guy && Billie Eilish' and 'Bad Guy (Radio Edit) && Billie Eilish' will return a high similarity score. We create or update track instances as needed. Additionally, for each track instance, we record in which dataframe we gathered information from (using the row index)

- For each artist, we track all the songs listened to with the help of a dictionary.

- While processing our data, we exclude songs that do not contain a Title ('NaN'), or those we could not find a close match using 'Title && Artist' string combination.

The logic is as follows:

1. Step 1: We loop through the Apple Music Library dataframe and we create a track instance whenever we encounter a new song. We update an existing track instance when we have seen this song before.

2. Step 2: We loop through Identifier Information data frame. As this dataframe contains only title and id, we are not going to be able to create new instances of Tracks (too little information about a track), so we simply update existing instances when we find a match with the ids

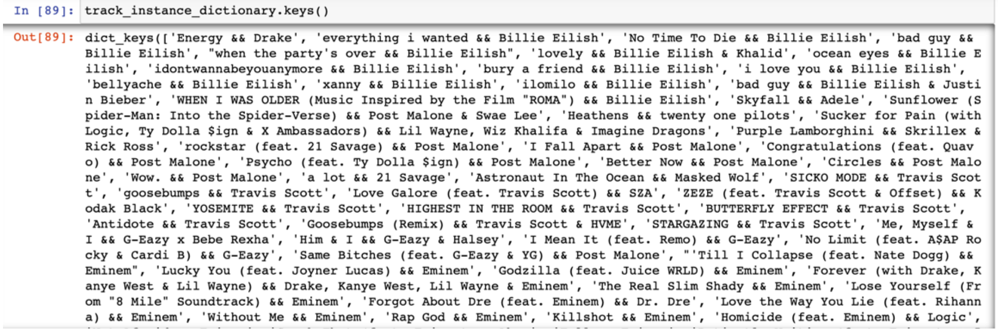
3. Step 3: We loop through the Apple Music Play Activity dataframe and we create a track instance when we encounter a new song. We update an existing instance when we already saw a similar song before.

4. Step 4: We loop through the Apple Music Likes and Dislikes dataframe, and again as this dataframe contains very little information about each track, we update existing instances when we already saw a similar song (similar here meaning with a similar combination of Title and Artist)

Results:

- We have one dictionary named track_instance_dictionary which keeps track of the title/artist combination with the reference of the associated track instance. The keys for this dictionary indicate the title&&artist combinations and the values assigned are the track instances.

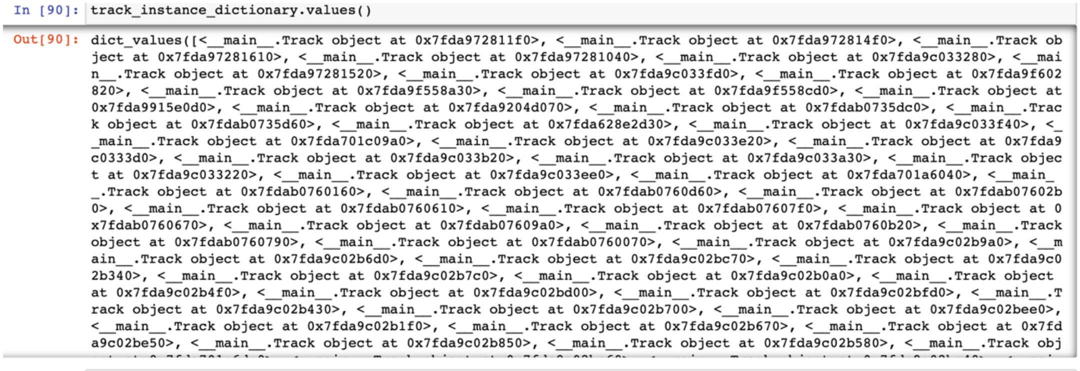
Keys for the dictionary:



```
In [89]: track_instance_dictionary.keys()
```

```
Out[89]: dict_keys(['Energy && Drake', 'everything i wanted && Billie Eilish', 'No Time To Die && Billie Eilish', 'bad guy &&
         Billie Eilish', "when the party's over && Billie Eilish", 'lovely && Billie Eilish & Khalid', 'ocean eyes && Billie E
         ilish', 'idontwannabeyouanymore && Billie Eilish', 'bury a friend && Billie Eilish', 'i love you && Billie Eilish',
         'bellyache && Billie Eilish', 'xanny && Billie Eilish', 'ilomilo && Billie Eilish', 'bad guy && Billie Eilish & Justi
         n Bieber', 'WHEN I WAS OLDER (Music Inspired by the Film "ROMA") && Billie Eilish', 'Skyfall && Adele', 'Sunflower (S
         pider-Man: Into the Spider-Verse) && Post Malone & Swae Lee', 'Heathens && twenty one pilots', 'Sucker for Pain (with
         Logic, Ty Dolla $ign & X Ambassadors) && Lil Wayne, Wiz Khalifa & Imagine Dragons', 'Purple Lamborghini && Skrillex &
         Rick Ross', 'rockstar (feat. 21 Savage) && Post Malone', 'I Fall Apart && Post Malone', 'Congratulations (feat. Quav
         o) && Post Malone', 'Psycho (feat. Ty Dolla $ign) && Post Malone', 'Better Now && Post Malone', 'Circles && Post Malo
         ne', 'Wow. && Post Malone', 'a lot && 21 Savage', 'Astronaut In The Ocean && Masked Wolf', 'SICKO MODE && Travis Scot
         t', 'goosebumps && Travis Scott', 'Love Galore (feat. Travis Scott) && SZA', 'ZEZE (feat. Travis Scott & Offset) && K
         odak Black', 'YOSEMITE && Travis Scott', 'HIGHEST IN THE ROOM && Travis Scott', 'BUTTERFLY EFFECT && Travis Scott',
         'Antidote && Travis Scott', 'Goosebumps (Remix) && Travis Scott & HVME', 'STARGAZING && Travis Scott', 'Me, Myself &
         I && G-Eazy x Bebe Rexha', 'Him & I && G-Eazy & Halsey', 'I Mean It (feat. Remo) && G-Eazy', 'No Limit (feat. A$AP Ro
         cky & Cardi B) && G-Eazy', 'Same Bitches (feat. G-Eazy & YG) && Post Malone', "'Till I Collapse (feat. Nate Dogg) &&
         Eminem", 'Lucky You (feat. Joyner Lucas) && Eminem', 'Godzilla (feat. Juice WRLD) && Eminem', 'Forever (with Drake, K
         anye West & Lil Wayne) && Drake, Kanye West, Lil Wayne & Eminem', 'The Real Slim Shady && Eminem', 'Lose Yourself (Fr
         om "8 Mile" Soundtrack) && Eminem', 'Forgot About Dre (feat. Eminem) && Dr. Dre', 'Love the Way You Lie (feat. Rihann
         a) && Eminem', 'Without Me && Eminem', 'Rap God && Eminem', 'Killshot && Eminem', 'Homicide (feat. Eminem) && Logic',
```

Values assigned to the keys for this dictionary:

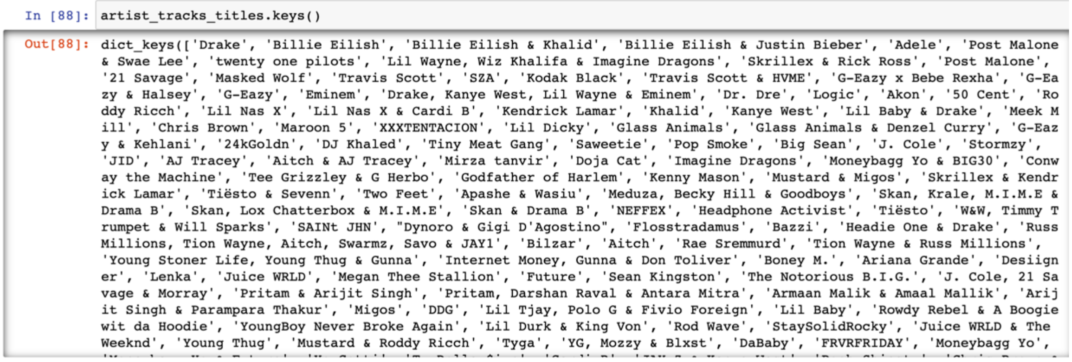


```
In [90]: track_instance_dictionary.values()
```

```
Out[90]: dict_values([<__main__.Track object at 0x7fda972811f0>, <__main__.Track object at 0x7fda972814f0>, <__main__.Track ob
         ject at 0x7fda97281610>, <__main__.Track object at 0x7fda97281040>, <__main__.Track object at 0x7fda9c033280>, <__mai
         n__.Track object at 0x7fda97281520>, <__main__.Track object at 0x7fda9c033fd0>, <__main__.Track object at 0x7fda9f602
         820>, <__main__.Track object at 0x7fda9f558a30>, <__main__.Track object at 0x7fda9f558cd0>, <__main__.Track object at
         0x7fda9915e0d0>, <__main__.Track object at 0x7fda9204d070>, <__main__.Track object at 0x7fdab0735dc0>, <__main__.Trac
         k object at 0x7fdab0735d60>, <__main__.Track object at 0x7fda628e2d30>, <__main__.Track object at 0x7fda9c033f40>, <_
         _main__.Track object at 0x7fda701c09a0>, <__main__.Track object at 0x7fda9c033e20>, <__main__.Track object at 0x7fda9
         c0333d0>, <__main__.Track object at 0x7fda9c033b20>, <__main__.Track object at 0x7fda9c033a30>, <__main__.Track objec
         t at 0x7fda9c033220>, <__main__.Track object at 0x7fda9c033ee0>, <__main__.Track object at 0x7fda701a6040>, <__main_
         _.Track object at 0x7fdab0760160>, <__main__.Track object at 0x7fdab0760d60>, <__main__.Track object at 0x7fdab07602b
         0>, <__main__.Track object at 0x7fdab0760610>, <__main__.Track object at 0x7fdab07607f0>, <__main__.Track object at 0
         x7fdab0760670>, <__main__.Track object at 0x7fdab07609a0>, <__main__.Track object at 0x7fdab0760b20>, <__main__.Track
         object at 0x7fdab0760790>, <__main__.Track object at 0x7fdab0760070>, <__main__.Track object at 0x7fda9c02b9a0>, <__m
         ain__.Track object at 0x7fda9c02b6d0>, <__main__.Track object at 0x7fda9c02bc70>, <__main__.Track object at 0x7fda9c0
         2b340>, <__main__.Track object at 0x7fda9c02b7c0>, <__main__.Track object at 0x7fda9c02b0a0>, <__main__.Track object
         at 0x7fda9c02b4f0>, <__main__.Track object at 0x7fda9c02bd00>, <__main__.Track object at 0x7fda9c02bfd0>, <__main__.T
         rack object at 0x7fda9c02b430>, <__main__.Track object at 0x7fda9c02b700>, <__main__.Track object at 0x7fda9c02bee0>,
         <__main__.Track object at 0x7fda9c02b1f0>, <__main__.Track object at 0x7fda9c02b670>, <__main__.Track object at 0x7fd
         a9c02be50>, <__main__.Track object at 0x7fda9c02b850>, <__main__.Track object at 0x7fda9c02b580>, <__main__.Track obj
```

- We have another dictionary named artist_tracks_titles which keeps track of all the titles for an artist, including different spellings of the same title. The keys for this dictionary indicate the various artists and the values assigned are the song titles.

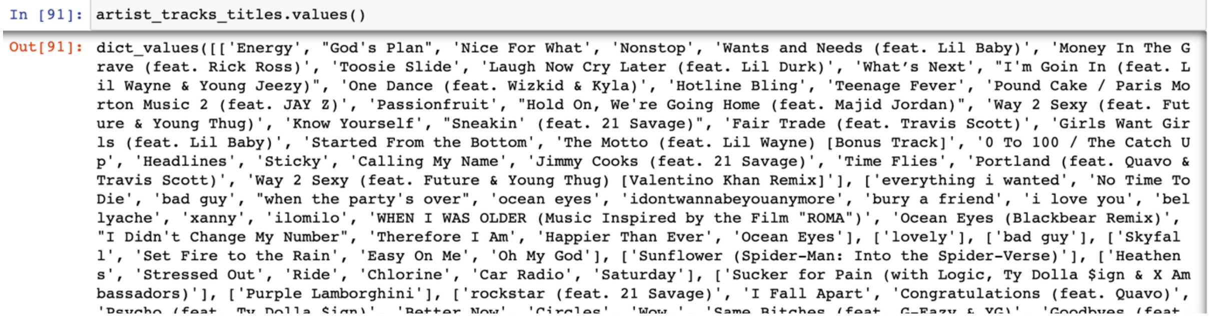
Keys for the dictionary:



```
In [88]: artist_tracks_titles.keys()
```

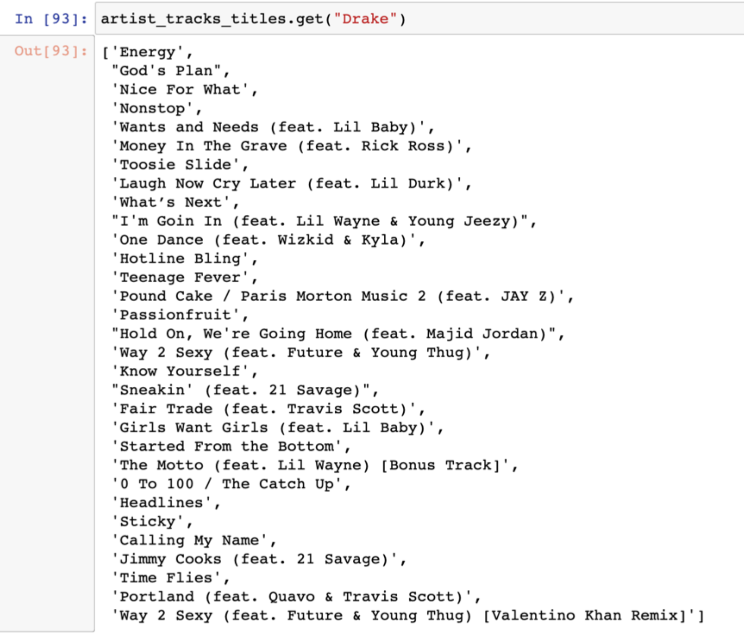
```
Out[88]: dict_keys(['Drake', 'Billie Eilish', 'Billie Eilish & Khalid', 'Billie Eilish & Justin Bieber', 'Adele', 'Post Malone
         & Swae Lee', 'twenty one pilots', 'Lil Wayne, Wiz Khalifa & Imagine Dragons', 'Skrillex & Rick Ross', 'Post Malone',
         '21 Savage', 'Masked Wolf', 'Travis Scott', 'SZA', 'Kodak Black', 'Travis Scott & HVME', 'G-Eazy x Bebe Rexha', 'G-Ea
         zy & Halsey', 'G-Eazy', 'Eminem', 'Drake, Kanye West, Lil Wayne & Eminem', 'Dr. Dre', 'Logic', 'Akon', '50 Cent', 'Ro
         ddy Ricch', 'Lil Nas X', 'Lil Nas X & Cardi B', 'Kendrick Lamar', 'Khalid', 'Kanye West', 'Lil Baby & Drake', 'Meek M
         ill', 'Chris Brown', 'Maroon 5', 'XXXTENTACION', 'Lil Dicky', 'Glass Animals', 'Glass Animals & Denzel Curry', 'G-Eaz
         y & Kehlani', '24kGoldn', 'DJ Khaled', 'Tiny Meat Gang', 'Saweetie', 'Pop Smoke', 'Big Sean', 'J. Cole', 'Stormzy',
         'JID', 'AJ Tracey', 'Aitch & AJ Tracey', 'Mirza tanvir', 'Doja Cat', 'Imagine Dragons', 'Moneybagg Yo & BIG30', 'Conw
         ay the Machine', 'Tee Grizzley & G Herbo', 'Godfather of Harlem', 'Kenny Mason', 'Mustard & Migos', 'Skrillex & Kendr
         ick Lamar', 'Tiësto & Sevenn', 'Two Feet', 'Apashe & Wasiu', 'Meduza, Becky Hill & Goodboys', 'Skan, Krale, M.I.M.E &
         Drama B', 'Skan, Lox Chatterbox & M.I.M.E', 'Skan & Drama B', 'NEFFEX', 'Headphone Activist', 'Tiësto', 'W&W, Timmy T
         rumpet & Will Sparks', 'SAINt JHN', "Dynoro & Gigi D'Agostino", 'Flosstradamus', 'Bazzi', 'Headie One & Drake', 'Russ
         Millions, Tion Wayne, Aitch, Swarmz, Savo & JAY1', 'Bilzar', 'Aitch', 'Rae Sremmurd', 'Tion Wayne & Russ Millions',
         'Young Stoner Life, Young Thug & Gunna', 'Internet Money, Gunna & Don Toliver', 'Boney M.', 'Ariana Grande', 'Desiign
         er', 'Lenka', 'Juice WRLD', 'Megan Thee Stallion', 'Future', 'Sean Kingston', 'The Notorious B.I.G.', 'J. Cole, 21 Sa
         vage & Morray', 'Pritam & Arijit Singh', 'Pritam, Darshan Raval & Antara Mitra', 'Armaan Malik & Amaal Mallik', 'Arij
         it Singh & Parampara Thakur', 'Migos', 'DDG', 'Lil Tjay, Polo G & Fivio Foreign', 'Lil Baby', 'Rowdy Rebel & A Boogie
         wit da Hoodie', 'YoungBoy Never Broke Again', 'Lil Durk & King Von', 'Rod Wave', 'StaySolidRocky', 'Juice WRLD & The
         Weeknd', 'Young Thug', 'Mustard & Roddy Ricch', 'Tyga', 'YG, Mozzy & Blxst', 'DaBaby', 'FRVRFRIDAY', 'Moneybagg Yo',
```

Values assigned to the keys for the dictionary:



```
In [91]: artist_tracks_titles.values()

Out[91]: dict_values([['Energy', "God's Plan", 'Nice For What', 'Nonstop', 'Wants and Needs (feat. Lil Baby)', 'Money In The G
         rave (feat. Rick Ross)', 'Toosie Slide', 'Laugh Now Cry Later (feat. Lil Durk)', 'What’s Next', "I'm Goin In (feat. L
         il Wayne & Young Jeezy)", 'One Dance (feat. Wizkid & Kyla)', 'Hotline Bling', 'Teenage Fever', 'Pound Cake / Paris Mo
         rton Music 2 (feat. JAY Z)', 'Passionfruit', "Hold On, We're Going Home (feat. Majid Jordan)", 'Way 2 Sexy (feat. Fut
         ure & Young Thug)', 'Know Yourself', "Sneakin' (feat. 21 Savage)", 'Fair Trade (feat. Travis Scott)', 'Girls Want Gir
         ls (feat. Lil Baby)', 'Started From the Bottom', 'The Motto (feat. Lil Wayne) [Bonus Track]', '0 To 100 / The Catch U
         p', 'Headlines', 'Sticky', 'Calling My Name', 'Jimmy Cooks (feat. 21 Savage)', 'Time Flies', 'Portland (feat. Quavo &
         Travis Scott)', 'Way 2 Sexy (feat. Future & Young Thug) [Valentino Khan Remix]'], ['everything i wanted', 'No Time To
         Die', 'bad guy', "when the party's over", 'ocean eyes', 'idontwannabeyouanymore', 'bury a friend', 'i love you', 'bel
         lyache', 'xanny', 'ilomilo', 'WHEN I WAS OLDER (Music Inspired by the Film "ROMA")', 'Ocean Eyes (Blackbear Remix)',
         "I Didn't Change My Number", 'Therefore I Am', 'Happier Than Ever', 'Ocean Eyes'], ['lovely'], ['bad guy'], ['Skyfal
         l', 'Set Fire to the Rain', 'Easy On Me', 'Oh My God'], ['Sunflower (Spider-Man: Into the Spider-Verse)'], ['Heathen
         s', 'Stressed Out', 'Ride', 'Chlorine', 'Car Radio', 'Saturday'], ['Sucker for Pain (with Logic, Ty Dolla $ign & X Am
         bassadors)'], ['Purple Lamborghini'], ['rockstar (feat. 21 Savage)', 'I Fall Apart', 'Congratulations (feat. Quavo)',
         'Psycho (feat. Ty Dolla $ign)', 'Better Now', 'Circles', 'Wow.', 'Same Bitches (feat. G-Eazy & YG)', 'Goodbyes (feat.
```

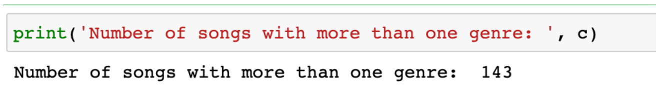
For Example: I want to look at all the song titles associated with Drake, I can easily get that information:



```
In [93]: artist_tracks_titles.get("Drake")

Out[93]: ['Energy',
          "God's Plan",
          'Nice For What',
          'Nonstop',
          'Wants and Needs (feat. Lil Baby)',
          'Money In The Grave (feat. Rick Ross)',
          'Toosie Slide',
          'Laugh Now Cry Later (feat. Lil Durk)',
          'What’s Next',
          "I'm Goin In (feat. Lil Wayne & Young Jeezy)",
          'One Dance (feat. Wizkid & Kyla)',
          'Hotline Bling',
          'Teenage Fever',
          'Pound Cake / Paris Morton Music 2 (feat. JAY Z)',
          'Passionfruit',
          "Hold On, We're Going Home (feat. Majid Jordan)",
          'Way 2 Sexy (feat. Future & Young Thug)',
          'Know Yourself',
          "Sneakin' (feat. 21 Savage)",
          'Fair Trade (feat. Travis Scott)',
          'Girls Want Girls (feat. Lil Baby)',
          'Started From the Bottom',
          'The Motto (feat. Lil Wayne) [Bonus Track]',
          '0 To 100 / The Catch Up',
          'Headlines',
          'Sticky',
          'Calling My Name',
          'Jimmy Cooks (feat. 21 Savage)',
          'Time Flies',
          'Portland (feat. Quavo & Travis Scott)',
          'Way 2 Sexy (feat. Future & Young Thug) [Valentino Khan Remix]']
```

Section 9: Checking for duplicates

Here. we look at the number of songs with duplicates or discrepancies as we have collected information from multiple dataframes.

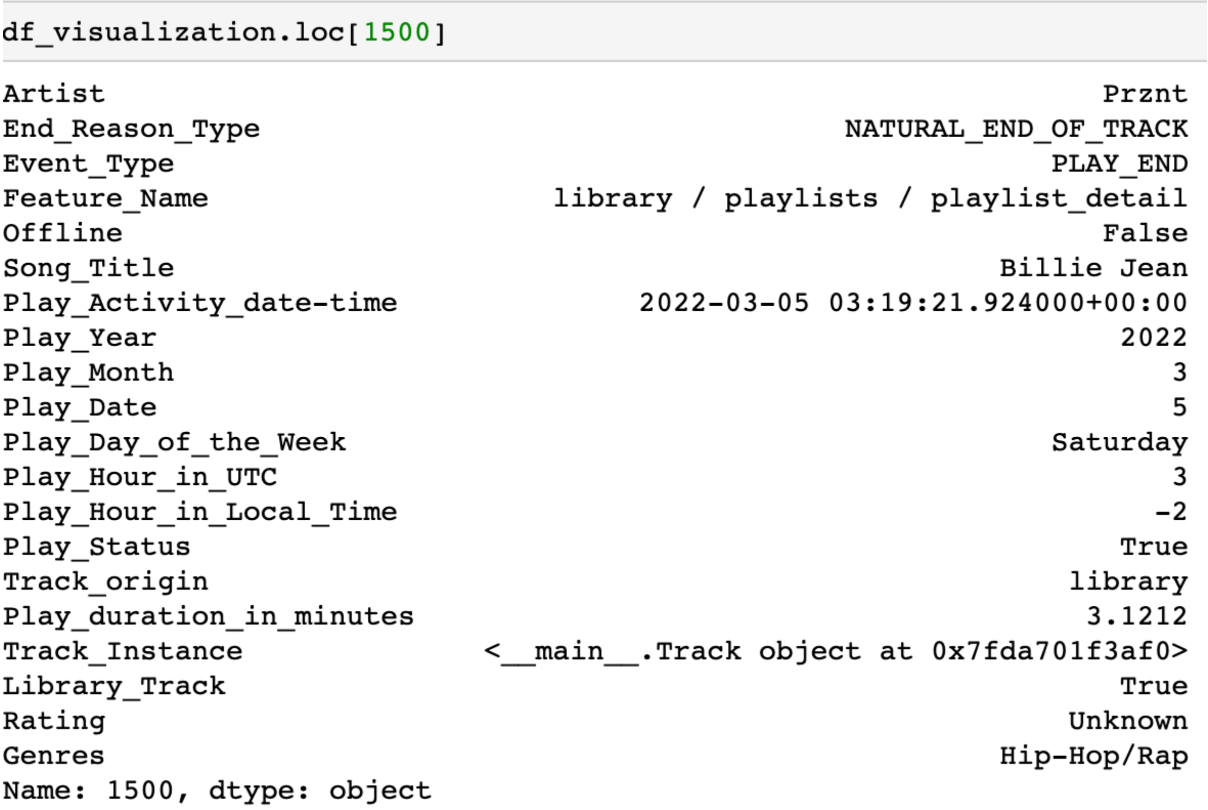


```
print('Number of songs with more than one genre: ', c)

Number of songs with more than one genre:  143
```

This is actually a great thing as it could allow building up recommendations using more than one genre to match songs!

Section 10: Merging this Tracks Information with Play Activity dataframe

As, I had decided to the play activity dataframe as my base for visualization, I enrich this dataframe with all the information I gathered using the Tracks data structure. I can easily add information on each row about each track such as the genres, whether the track is in my library or not, or if it has been rated as ‘Love’ or not.



```
df_visualization.loc[1500]
```

```
Artist                                                          Prznt
End_Reason_Type                                  NATURAL_END_OF_TRACK
Event_Type                                                   PLAY_END
Feature_Name                    library / playlists / playlist_detail
Offline                                                         False
Song_Title                                                Billie Jean
Play_Activity_date-time              2022-03-05 03:19:21.924000+00:00
Play_Year                                                        2022
Play_Month                                                          3
Play_Date                                                           5
Play_Day_of_the_Week                                         Saturday
Play_Hour_in_UTC                                                    3
Play_Hour_in_Local_Time                                            -2
Play_Status                                                      True
Track_origin                                                  library
Play_duration_in_minutes                                       3.1212
Track_Instance              <__main__.Track object at 0x7fda701f3af0>
Library_Track                                                    True
Rating                                                        Unknown
Genres                                                    Hip-Hop/Rap
Name: 1500, dtype: object
```

Section 11: Conclusion of Feature Construction

The image above clearly demonstrates all the information that I have derived for each song that I have listened to. This information includes:

1. When was the track listened? This information is displayed through the column 'Play_Activity_date-time' which is further broken down into the year, month, day of the month, day of the week, hour of the day in UTC and Local time.

2. A Reference to the Track Instance

3. Was the song skipped or listened entirely? This information is demonstrated through the 'Play_Status' column which has two binary values: True indicating that the song was completed and False indicating that the song was skipped

4. How was the song found? This information is displayed through 'Track_Origin' column which contains the simplified origin of the song in four categories: Library,, Search, Radio and Other.

5. What's the genre of the song? This information is displayed through the 'Genres' column

6. Is the track in my library? This information is displayed through the 'Library_Track' column which has two binary values: True indicating that the track is in my library and False indicating that the track is not my library.

7. What's the rating of the song? Liked/Disliked/Unknown? This information is displayed through the 'Rating' column in the data frame.

8. How long was the song played for? This information is displayed through the 'Play_duration_in_minutes' column.

Now, I am ready to use this data frame and answer some questions using some powerful visualization tools.